HAL open science

Learning to generate human videos

Yaohui Wang

▶ To cite this version:

Yaohui Wang. Learning to generate human videos. Artificial Intelligence [cs.AI]. Université Côte d'Azur, 2021. English. NNT : 2021COAZ4116 . tel-03662376v3

HAL Id: tel-03662376

https://theses.hal.science/tel-03662376v3

Submitted on 9 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Apprendre à Générer des Vidéos de Personnes

Yaohui WANG

Inria Sophia Antipolis, STARS

Soutenue le : **30 Septembre 2021**

Présentée en vue de l'obtention du grade de docteur en
INFORMATIQUE d'**Université Côte d'Azur**

Devant le jury composé de:

George Drettakis	Inria Sophia Antipolis, France	Président
Ivan Laptev	Inria Paris, École normale supérieure, France	Rapporteur
Elisa Ricci	University of Trento, Italie	Rapporteur
Shiguang Shan	Chinese Academy of Sciences, Chine	Examinateur
Sergey Tulyakov	Snap Research, États-Unis	Examinateur
Panayiotis Georgiou	University of Southern California, États-Unis	Examinateur
Antitza Dantcheva	Inria Sophia Antipolis, France	Directeur de thèse
François Brémond	Inria Sophia Antipolis, France	Directeur de thèse

Inria

Learning to Generate Human Videos

Yaohui WANG

Inria Sophia Antipolis, STARS

Université Côte d'Azur

This dissertation is submitted for the degree of

Doctor of Philosophy

December 2021

Abstract

Generative Adversarial Networks (GANs) have witnessed increasing attention due to their abilities to model complex visual data distributions, which allow them to generate and translate realistic *images*. 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 and retaining of human appearance, (b) modeling spatio-temporal consistency, as well as (c) understanding of latent representation.

In this thesis, we propose three novel approaches towards generating high-visual quality videos and interpreting latent space in video generative models. We firstly introduce a method, which learns to conditionally generate videos based on single input images. Our proposed model allows for controllable video generation by providing various motion categories. Secondly, we present a model, which is able to produce videos from noise vectors by disentangling the latent space into appearance and motion. We demonstrate that both factors can be manipulated in both, conditional and unconditional manners. Thirdly, we introduce an unconditional video generative model that allows for interpretation of the latent space. We place emphasis on the interpretation and manipulation of motion. We show that our proposed method is able to discover semantically meaningful motion representations, which in turn allow for control in generated results. Finally, we describe a novel approach to combine generative modeling with contrastive learning for unsupervised person re-identification. Specifically, we leverage generated data as data augmentation and show that such data can boost re-identification accuracy.

Keywords : Generative Adversarial Networks, Video Generation, Deep Learning

Résumé

Les réseaux antagonistes génératifs (GAN) ont suscité une attention croissante en raison de leurs capacités à modéliser des distributions de données visuelles complexes, ce qui leur permet de générer et de traduire des images réalistes. Bien que la génération de vidéos réalistes soit la suite naturelle, elle est nettement plus difficile en ce qui concerne leur complexité et leur calcul, associés à la modélisation simultanée de l'apparence, ainsi que du mouvement de la personne dans la vidéo. Plus précisément, en inférant et en modélisant la distribution de vidéos, les modèles génératifs sont confrontés à trois défis principaux : (a) générer un nouveau mouvement et conserver l'apparence de la personne, (b) modéliser la cohérence spatio-temporelle, ainsi que (c) comprendre la représentation latente de la vidéo.

Dans cette thèse, nous proposons un certain nombre d'approches novatrices pour générer des vidéos de haute qualité visuelle et interpréter l'espace latent de la représentation de la vidéo dans ces modèles génératifs. Nous introduisons tout d'abord une méthode, qui apprend à générer conditionnellement des vidéos basées sur une seule image en entrée. Notre modèle proposé permet une génération de vidéo contrôlable en fournissant diverses catégories de mouvement. Deuxièmement, nous présentons un modèle, qui est capable de produire des vidéos à partir de vecteurs de bruit en dissociant l'apparence et le mouvement dans l'espace latent. Nous démontrons que les deux facteurs peuvent être manipulés de manière conditionnelle et inconditionnelle. Troisièmement, nous introduisons un modèle génératif inconditionnel de vidéos qui permet l'interprétation de l'espace latent. Nous mettons l'accent sur l'interprétation et la manipulation du mouvement. Nous montrons que la méthode proposée est capable de découvrir des représentations du mouvement sémantiquement significatives, qui à leur tour permettent le contrôle des vidéos générées. Enfin, nous décrivons une

nouvelle approche pour combiner la modélisation générative avec l'apprentissage contrastif pour la réidentification de personnes en mode non supervisé. Nous exploitons les données générées en tant qu'augmentation de données et montrons que ces données peuvent améliorer la précision de la ré-identification.

Mots-clés : Réseaux Antagonistes Génératifs, Génération Vidéo, Apprentissage Profond

Acknowledgements

Antitza and Francois, I would like to thank you for your greatest support during my PhD. Without the guidance from both of you, this thesis would not have been possible. Having both of you as my advisors was extremely lucky to me. Thanks Antitza for providing me freedom to explore topics which really interest me, and always encouraging me to pursue my ambitious and crazy goals. Thanks Francois for giving great advice on both my research and personal life. Thank you for always being available when I need your help. You use your own behaviour to teach me that the most important thing during PhD is not just to be a good researcher but to be a good man. It has been my great honor to have the opportunity to work with both of you.

I am grateful to ANR for generously funding my research under the grant ENVISION. I would like to thank Inria Sophia Antipolis for providing an excellent research environment and cluster NEF. I would also like to thank IDRIS for the GPU grant, which supported my research.

I am also grateful to Elisa Ricci, George Drettakis, Ivan Laptev, Panayiotis Georgiou, Shiguang Shan and Sergey Tulyakov. It was an honor to have you at my defense. I would like to especially thank my thesis reviewers Elisa Ricci and Ivan Laptev. Thank you for devoting your time and effort to review my manuscript.

Prof. Michèle Sebag, I would like to thank you for teaching me machine learning in Master AIC. Your great lessons started my research career. Also, many thanks to my dear partner Herilalaina, we have finished so many projects together and you provided me so much help to my life in France. I really enjoy the time we wrote code together.

I would like to thank my collaborator Piotr Bilinski. Thanks to the inspirational discussions we had. You provided me many interesting ideas.

I would also like to thank my advisor at Dassault Systèmes, Mohamed Amine Ayari, for offering me a great internship to work on interesting computer vision project.

Many thanks to Ujjwal. You gave me so much help on engineering when I first came to the team. You have taken care of me like a big brother. Thanks to all the adventures, scientific discussions and food we had together.

I truly enjoyed my time at Inria with all the fantastic members in STARS team, Di, David, Rui, Hao, Srijan, Juan-diego, Jen-Cheng, Thibaud, Valeriya, Tanay, Michal, Laura, Sébastien, Rachid, Happy, Abhijit, Vikas, Sabine, Jean-Paul, Monique, Laurence and Sandrine, and all my Chinese friends here, Zihao, Shuman, Wen, Qiao, Chuan, Tingting, Kun, Yang and Hui. We shared so much unforgettable lunch and coffee time in our lovely campus.

I am also grateful to the sea, the forest and mountains around Inria, they gave me so much great memories of Sophia Antipolis.

Last but not least, I am deeply grateful to my parents for their love and support, and I would like to give special thanks to my girlfriend, Yingqi for always being with me and encouraging me.

Contents

List of Figures xiii

List of Tables xxiii

1 Introduction 1

1.1 Goals 2

1.2 Motivation 3

1.3 Challenges 5

1.4 Thesis Outline 8

1.5 Contributions 9

1.5.1 Publications 10

1.5.2 Software contributions 11

2 Literature Review 13

2.1 Generative Adversarial Networks (GANs) 13

2.2 Evaluation metrics of GANs 15

2.2.1 Inception Score (IS) 15

2.2.2 Fréchet Inception Distance (FID) 15

2.3 Image generation 16

2.3.1 Unconditional image generation 17

2.3.2 Conditional image generation 19

2.4 Video generation 23

2.4.1 Unconditional video generation 23

2.4.2 Conditional video generation 24
2.5 Interpretability of GANs 25

3 Conditional Spatio-Temporal GAN for Video Generation 27
3.1 Introduction 28
3.2 Background 30
3.3 ImaGINator 31
3.3.1 Generator 32
3.3.2 Two-stream Discriminator 34
3.3.3 Learning 35
3.3.4 Architecture details 36
3.3.5 Implementation details and training strategy 39
3.4 Experiments 42
3.4.1 Datasets 42
3.4.2 Evaluation Metrics 43
3.4.3 Video quality evaluation 43
3.4.4 Controllable Video Generation. 46
3.4.5 Ablation Study 46

4 Disentangling Appearance and Motion for Video Generation 57
4.1 Introduction 58
4.2 Background 59
4.3 G^3AN 60
4.3.1 Generator 61
4.3.2 Discriminator 66
4.3.3 Training 66
4.4 Experiments 67
4.4.1 Implementation details 67
4.4.2 Datasets 68
4.4.3 Evaluation metrics 68

4.4.4 Quantitative Evaluation 69
4.4.5 Qualitative Evaluation 70
4.4.6 Ablation Study 73

5 Motion Interpretation in Video Generation 89
5.1 Introduction 90
5.2 Background 91
5.2.1 Linear Motion Decomposition (LMD) 94
5.2.2 Generator 95
5.2.3 Discriminator 96
5.2.4 Learning 97
5.2.5 Implementation details. 97
5.3 Experiments and Analysis 98
5.3.1 Datasets 98
5.3.2 Evaluation metric 99
5.3.3 Video quality evaluation 99
5.3.4 Interpretability evaluation 101
5.3.5 User study 107
5.3.6 Further analysis 108

6 Joint Generative and Contrastive Learning for Unsupervised Person ReID 111
6.1 Introduction 112
6.2 Background 114
6.3 Approach 116
6.3.1 View Generator (Generative Module) 117
6.3.2 View Contrast (Contrastive Module) 119
6.3.3 Joint Training 121
6.4 Experiments 122
6.4.1 Datasets and Evaluation Protocols 122
6.4.2 Implementation Details 122

6.4.3 Unsupervised ReID Evaluation 125
6.4.4 Generation Quality Evaluation 127

7 Discussion and future work **135**
7.1 Summary of contributions 135
7.2 Future work 137

Bibliography **139**

List of Figures

1.1 **Examples of machine creativity.** (Left) A humanoid draws a painting. (Right) Robot arms construct a paper crane. 1

1.2 **Examples of data.** (Left) VoxCeleb. (Middle) BAIR. (Right) UCF101. . . 2

1.3 **Video manipulation.** (a) Moving robot arm and (b) talking head. 3

1.4 **Use cases for social media.** (Left) Inside functions of Snapchat for various contents creation. (Right) A user is spreading common knowledge on Tiktok. 5

1.5 **Generator design.** (Left) 3D ConvNets based architecture. (Right) 2D ConvNets+RNN based architecture. 6

2.1 **Generative Adversarial Network.** 13

2.2 **Generator of DCGAN.** 17

2.3 **Generator architecture of Progressive GAN.** 18

2.4 **Architecture and generated samples of StyleGAN(2).** (a) StyleGAN generator architecture. (b) Generated images from StyleGAN2. 19

2.5 **Discriminators for class-conditioned image generation.** (a) cGANs (input concat). (b) cGANs (hidden concat). (c) ACGANs. (d) SNGAN. 20

2.6 **BigGAN architecture.** (a) Generator. (b) Residual block in generator. (c) Residual block in discriminator. 21

2.7 **Generated samples from BigGAN.** 21

2.8 **Framework of Pix2pix.** 22

2.9 **Framework of CycleGAN.** 22

2.10 **VGAN Architecture.** 23

2.11 **TGAN Architecture.** 24
2.12 **MoCoGAN Architecture.** 25
2.13 **Linear and non-linear latent walks.** 26

3.1 **ImaGINator architecture.** The proposed ImaGINator architecture incorporates *Generator G*, *image Discriminator* D_I, as well as *video Discriminator* D_V. G accepts c_a, c_m and noise as input, and seeks to generate realistic video sequences. While D_I discriminates whether the generated images contain an authentic appearance, D_V additionally determines whether the generated videos contain an authentic motion. 29
3.2 **Overview of the proposed ImaGINator**. In the *Generator G*, the *Encoder* firstly encodes an input image c_a into a single vector p. Then, the *Decoder* produces a video based on a motion c_m and a random vector z. By using spatio-temporal fusion, low level spatial feature maps from the *Encoder* are directly concatenated into the *Decoder*. While D_I discriminates whether the generated images contain an authentic appearance, D_V additionally determines whether the generated videos contain an authentic motion. 32
3.3 **Spatio-temporal fusion.** Blue and orange cuboids represent the intermediate feature maps in the Decoder and Encoder respectively. Our proposed fusion scheme enforces the Decoder reutilizing spatial information through skip connections. Based on such operations, temporal consistency can be modeled in multi-levels. 33
3.4 **Transposed 3D convolution** (on the left) v.s. **proposed Transposed (1+2)D convolution** (on the right). The transposed 3D convolutional filter of size $t \times w \times h$ has been decomposed into M transposed 1D temporal convolution filters $t \times 1 \times 1$ and a transposed 2D spatial convolution $1 \times w \times h$. The operation M denotes the number of 1D filters, t indicates the temporal size, and w and h indicate the spatial size. 34

3.5 **Network architecture of the Generator.** Our Generator G accepts an image of size $64 \times 64 \times 3$ as input and generates a 32-frame long video. G incorporates an image Encoder (*Conv1 - Conv5*) and a video Decoder (*Deconv6-1 - Deconv10-2*). Skip connections link Encoder and Decoder, with the goal of enforcing the Decoder to reuse appearance features directly. A motion category vector is replicated into feature maps and concatenated with each feature map in the Decoder (for different dataset, length of motion category vector is different, here we use 6 to represent MUG dataset). 38

3.6 **Network architecture of the image Discriminator,** containing five 2D convolutional layers of kernel size 4×4. 40

3.7 **Network architecture of the video Discriminator,** including five 3D convolutional layers, a motion category vector is firstly replicated and then concatenated with the feature map of the first layer (for different dataset, length of motion category vector is different, here we use 6 to represent MUG dataset). 41

3.8 **Example generated video frames** pertained to algorithms (a) VGAN, (b) MoCoGAN, as well as the (c) proposed ImaGINator. For each method, we present generated video frames for the four datasets: **Weizmann** (top-left), label *"Wave"*; **NATOPS** (top-right), label *"Hot Brakes"*; **MUG** (bottom-left), label *"Happiness"*; **UvA-NEMO** (Down-right), no label. All frames are sampled with a time step of 3. 44

3.9 **Controllable video generation in ImaGINator.** Starting from the same image (top left for both datasets), we generate videos associated to different labels (remaining frames). In (a) MUG, from top to bottom the labels are set as *"fear"*, *"anger"* and *"happiness"*. In (b) NATOPS, from top to bottom the labels are set as *"all clear"*, *"fold winds"* and *"brakes on"*. 46

3.10 **Comparison of use of merely (a) Adversarial loss and (b) Reconstruction loss.** We illustrate generated frames for (a) and (b) on four datasets. We observe that frames in (a) are sharper than (b), but (b) retains overall structures better than (a). Frames are sampled with time step 4. 48
3.11 **Sample generated frames** of ImaGINator with (a) transposed 3D and (b) transposed (1+2)D convolutions. 50
3.12 **Generated examples from UvA-NEMO.** 51
3.13 **Generated examples from MUG.** Labels are *happiness* (01,02,03,04), *anger* (05,06,07,08), *fear* (09,10,11,12), *sadness* (13,14,15) and *disgust* (16,17,18). 52
3.14 **Generated examples from NATOPS.** Labels are *Fold Wings* (01,02,03,04,05), *All Clear* (06,07,08,09), *Nosegear Steering* (10,11), *Turn Right* (12,13) and *Move Ahead* (14,15). 53
3.15 **Generated examples from Weizmann.** Labels are *One hand wave* (01,02,05,06), *Two hands wave* (03,04,11,12), *Bend* (07,08,13,14) and *Jack* (09,10). 54
3.16 **Generated samples from BAIR robot push.** 55

4.1 **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. 61
4.2 **G^3 module architecture.** 62
4.3 **Generator architecture.** 63
4.4 **Spatio-temporal fusion.** 64
4.5 **Factorized spatio-temporal Self-Attention (F-SA) module.** 65
4.6 **Comparison with the state-of-the-art** on MUG (top-left), Weizmann (top-right), UvA-NEMO (bottom-left) and UCF101 (bottom-right). 69
4.7 **Unconditional video generation** of G^{3}AN 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. 71

4.8 **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. 71

4.9 **Comparison between G^{3}AN and MoCoGAN.** Given fixed z_a and z_m, as well as two condition-labels *smile* and *surprise*, G^{3}AN and MoCoGAN generate correct facial expressions. However, while G^{3}AN preserves the appearance between rows, MoCoGAN alters the subject's appearance. . . . 73

4.10 **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. 74

4.11 **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. 75

4.12 **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). 75

4.13 **Ablation study.** Generated videos obtained by removing G_T (*top row*), removing G_S (*middle*), and both (*bottom row*). 76

4.14 **Unconditionally generated samples from G^{3}AN on UvA-NEMO.** We combine each z_a with three different z_m, obtaining three different videos for the same appearance. Each row represents a video sequence. 78

4.15 **Conditionally generated samples from G^{3}AN on MUG dataset.** Each row represents the result generated by combining a one-hot category label with the same z_a and randomly sampled z_m as input. 79

4.16 **Conditionally generated samples from G^{3}AN on MUG dataset.** Each row represents the result generated by combining a one-hot category label with the same z_a and randomly sampled z_m as input. 80
4.17 **Conditionally generated samples from G^{3}AN on Weizmann dataset.** Each row represents the result generated by combining a one-hot category label with the same z_a and randomly sampled z_m as input. 81
4.18 **Conditionally generated samples from G^{3}AN on Weizmann dataset.** Each row represents the result generated by combining a one-hot category label with the same z_a and randomly sampled z_m as input. 81
4.19 **Results of manipulating *first dimension* in appearance representation on MUG dataset.** a and b are results from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased. . . 82
4.20 **Results of manipulating *second dimension* in appearance representation on MUG dataset.** a and b are results from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased. . . 83
4.21 **Results of manipulating *first dimension* in appearance representation on Weizmann dataset.** a and b are from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased. . . 84
4.22 **Results of manipulating *second dimension* in appearance representation on Weizmann dataset.** a and b are from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased. . . 85
4.23 **Results of manipulating *first dimension* in appearance representation on UvA-NEMO dataset.** a and b are from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased. . . 86
4.24 **Results of manipulating motion representation on UvA-NEMO dataset.** a and b are results of manipulating *first* and *sixth* dimensions. From top to bottom in each sub-figure, values are increased. 87

4.25 **Results of manipulating motion representation on Weizmann dataset.** *a* and *b* are results of manipulating *first* and *second* dimensions. From top to bottom in each sub-figure, values are increased. 88

5.1 **Controllable video generation.** MintGAN learns to decompose motion into semantic motion-components. This allows for manipulations in the latent code to invoke motion in generated videos that is human interpretable. Top (a) robot arm moves backwards, bottom (a) robot arm moves to the right. Similarly, in (b) we are animating the face to 'talk' (top) and 'move head' (bottom). 90

5.2 **MintGAN-architecture.** MintGAN comprises of a Generator and a two-stream Discriminator. We design the architecture of the Generator based on proposed Linear Motion Decomposition. Specifically, a motion bank is incorporated in the Generator to learn and store a motion dictionary D, which contains motion-directions $[d_0, d_1, .., d_{N-1}]$. We use an appearance net G_A to map appearance noise z_a into a latent code w_0, which serves as the initial latent code of a generated video. A motion net G_M maps a sequence of motion noises $\{z_{m_t}\}_{t=1}^{T-1}$ into a sequence $\{A_t\}_{t=1}^{T-1}$, which represent motion magnitudes. Each latent code w_t is computed based on Linear Motion Decomposition using w_0, D and A_t. Generated video V is obtained by a synthesis net G_S that maps the sequence of latent codes $\{w_t\}_{t=0}^{T-1}$ into an image sequence $\{x_t\}_{t=0}^{T-1}$. Our discriminator comprises an image discriminator D_I and a Temporal Pyramid Discriminator (TPD) that contains several video discriminators D_{V_i}, leveraging different temporal speeds υ_i to improve generated video quality. While D_I accepts as input a randomly sampled image per video, each D_{V_i} is accountable for one temporal resolution. 93

5.3 **Analysis of α.** Mean and variance bar charts, indicating top 10 motion-directions with the highest values in $A_{\bar{t}}$. 101

5.4 **Time v.s. α.** Each figure represents a video sample. We illustrate one sample from BAIR-robot (left) and one from VoxCeleb2-mini (right), respectively. Top 5 dimensions in α are plotted in different color. 101

5.5 **Directions analysis on BAIR-robot.** A generated video sample, related optical flow images (top), activation of *only* d_1 (middle), and activation of *only* d_{511} (bottom). Optical flow images indicate that d_1 is accountable for moving the robot arm backward, whereas d_{511} for moving it left and right. . 102

5.6 **Optical flow quantization.** (a) Middlebury colorwheel, (b) $\lambda(x_{t,j})$ and H on the colorwheel, (c) one frame from BAIR-robot and (d) related optical flow. 103

5.7 **Direction analysis in VoxCeleb2-mini.** A generated video sample and associated optical flow images (top), by *only* activating d_0 (middle), and by *only* activating d_{511} (bottom). While d_0 controls the mouth region, d_{511} controls the head region. 103

5.8 **Global and local motion extraction.** (a) Generated image, (b) related optical flow, (c) semantic map, (d) mouth-flow image, and (e) face-flow image based on training with VoxCeleb2-mini. 104

5.9 **Two pre-defined trajectories.** (a) We provide a *linear* trajectory for d_1 and a *sinusoidal* trajectory for d_{511}. (b) We provide a *sinusoidal* trajectory for d_1 and a *linear* trajectory for d_{511}. 105

5.10 Eight-region color wheel 107

6.1 **Traditional v.s. our proposed method.** (Left) Traditional self-supervised contrastive learning maximizes agreement between representations (f_1 and f_2) of augmented views from Data Augmentation (DA). (Right) Joint generative and contrastive learning maximizes agreement between original and generated views. 113

6.2 **A schematic overview of GCL.** (a) General architecture of GCL: Generative and contrastive modules are coupled by the shared identity encoder E_{id}. (b) Generative module: The decoder G combines the identity features encoded by E_{id} and structure features E_{str} to generate a novel view x'_{new} with a cycle consistency. (c) Contrastive module: View-invariance is enhanced by maximizing the agreement between original $E_{id}(x)$, synthesized $E_{id}(x'_{new})$ and memory f_{pos} representations. 117

6.3 **Generated multi-view images.** Example images as generated by the View Generator via 3D mesh rotation based on left input image. 118

6.4 **Qualitative ablation study on the view-invariant losses.** For simplicity, $\mathscr{L}_{vi}$ denotes three view-invariant losses $\mathscr{L}_{vi} + \mathscr{L}'_{vi} + \mathscr{L}''_{vi}$, which helps E_{id} to extract view-invariant features (red shirt). 127

6.5 **Comparison of the generated images on Market-1501 dataset.** $\star$ refers to methods without sharing source code, whose examples are cropped from their papers. Examples of FD-GAN, IS-GAN, DG-Net and GCL are generated from six real images shown in the figure. 128

6.6 **Generated novel views on the three datasets.** 129

6.7 **Linear interpolation on identity features.** Identity features are swapped between left and right persons. 129

6.8 **Examples of generated novel views on Market-1501 training and test sets.** 131

6.9 **Examples of generated novel views on DukeMTMC-reID training and test sets.** 132

6.10 **Examples of generated novel views on MSMT17 training and test sets.** . 133

List of Tables

2.1 **Generator and discriminator loss functions in different GANs.** 14

3.1 **Network architecture of the Generator.** Our Generator incorporates an image Encoder (*Conv1 - Conv5*), as well as a video Decoder (*Deconv6-1 - Deconv10-2*). KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size. 37

3.2 **Network architecture of the image Discriminator.** KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size. 39

3.3 **Network architecture of the video Discriminator.** KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size. 39

3.4 **Evaluation of video quality.** We compare VGAN, MoCoGAN and proposed ImaGINator w.r.t. image quality (SSIM/PSNR) and video quality (FID). 45

3.5 **Evaluation of content consistency** of VGAN, MoCoGAN and proposed ImaGINator on the MUG dataset, represented by ACD-I and ACD-C scores. 45

3.6 **Subjective analysis.** Mean user preference of human raters comparing videos generated by the respective algorithms, as well as originated from all the datasets. 46

3.7 **Effectiveness of the proposed architecture**. We compare different architectures in both G and D to showcase the effectiveness of the proposed ImaGINator. 47

3.8 **Contribution of main components in** *G*. We evaluate the ablation of spatio-temporal fusion, transposed (1+2)D convolution, as well as noise vector. . . 47
3.9 **Evaluation results for models using different losses** on four datasets represented by video FID. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.) 49
3.10 **Evaluation of frame quality** between generated frames and ground truth on four datasets using SSIM. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.) 49
3.11 **Evaluation of frame quality** between generated frames and ground truth on four datasets using PSNR. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.) 49
3.12 **FID score and training time per epoch** of our approach with transposed 3D and transposed (1+2)D convolutions. 50

4.1 **Comparison with the state-of-the-art** on four datasets w.r.t. FID and IS. . 70
4.2 **Mean user preference** of human raters comparing videos generated by the respective algorithms, originated from all datasets. 70
4.3 **Contribution of main components in** *G*. 76
4.4 **Comparison of various convolution types in** *G*. 77
4.5 **Comparison of inserting F-SA at different hierarchical levels of G^{3}AN.** 77

5.1 **Comparison of MintGAN with four state-of-the-art models.** MintGAN systematically and significantly outperforms other methods on both datasets w.r.t. FID. The lower FID, the better video quality. 99
5.2 **Evaluation of TPD.** When replacing the initial 3D discriminator with TPD, the latter significantly and consistently improves the FID of all 4 state-of-art models for the VoxCeleb2-mini and BAIR-robot datasets. 100

5.3 **Ablation study on video discriminators in TPD.** Number of video discriminators associated to temporal resolutions. FID is reported for comparison. Lower FID indicates a superior quality of generated videos. 100

5.4 $\Delta\phi_i$ **on BAIR-robot.** Motion difference in four regions (R_0, R_1, R_2, R_3) caused by deactivating motion-directions. 105

5.5 $\Delta\Phi_{head}$ **and** $\Delta\Phi_{mouth}$ **on VoxCeleb2-mini.** Motion difference in head and mouth regions induced by deactivation of motion-directions. 106

5.6 **User study**: Mean opinion score for the question 'Which video clip is more realistic?' 107

5.7 **User study**: Mean opinion score for the question 'which direction is the robot arm moving?'. 108

5.8 **User study**: Mean opinion score for the question 'what moves the most?'. . 108

5.9 **Comparison of MintGAN with four state-of-the-art models.** MintGAN systematically outperforms the other models on **VoxCeleb2-mini** w.r.t. FID. 109

5.10 **Comparison of MintGAN with five state-of-the-art models.** MintGAN systematically outperforms the other models on **UCF101** w.r.t. IS and FID. Numbers are adopted from [131], except MintGAN. 109

6.1 **Comparison of unsupervised ReID methods (%) with a ResNet50 backbone on Market and Duke datasets.** We test our proposed method on several baselines, whose names are in brackets. * refers to our implementation based on authors' code. 123

6.2 **Comparison of unsupervised ReID methods (%) with a ResNet50 backbone on MSMT17.** * refers to our implementation based on authors' code. 124

6.3 **Ablation study on loss functions used in two modules.** (1). $\mathscr{L}_{gan}$ corresponds to generation w/o contrast. (2). $\mathscr{L}_{vi}^{woGAN}$ corresponds to contrast w/o generation. TDA denotes traditional data augmentation. (3). $\mathscr{L}_{gan} + \mathscr{L}_{vi}$ ($\mathscr{L}_{vi}'$ and $\mathscr{L}_{vi}''$) correspond to joint generative and contrastive learning. . . . 127

6.4 **Comparison of FID and SSIM on Market-1501 dataset.** U denotes the fully unsupervised setting. UDA denotes Duke→Market setting. 128

Chapter 1

Introduction

Creativity constitutes the forming of something new and valuable, which is considered an essential ability of human intelligence. Owing to such an ability, human beings are capable of finding novel solutions to unexpected problems, as well as producing appealing works of art. Building machines, which are able to mimic human intelligence w.r.t. thinking and creating has always been long-term crusade of computer scientists [12, 149] (see Fig. 1.1). Recently, deep generative models and in particular Generative Adversarial Networks (GANs) have witnessed remarkable success in various visual content creation tasks such as image generation and translation. Given that such tasks are considered as an early stage of higher level machine intelligence, discovering the potential of GANs has become a highly compelling and exciting research topic in Artificial Intelligence (AI).

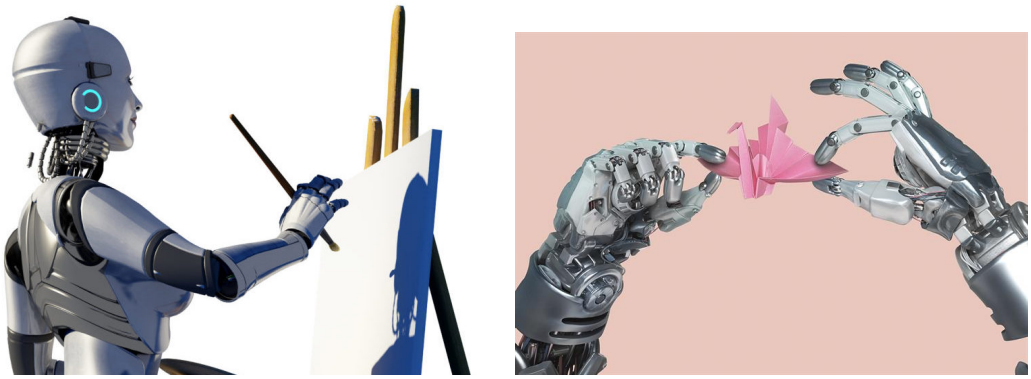


Figure 1.1 **Examples of machine creativity.** (Left) A humanoid draws a painting. (Right) Robot arms construct a paper crane.

Figure 1.2 **Examples of data.** (Left) VoxCeleb. (Middle) BAIR. (Right) UCF101.

1.1 Goals

Our main goal in this thesis is to develop GANs that create dynamic contents. In particular, we are interested in realistic videos generation, as well as model interpretability. Fig. 1.2 illustrates images of datasets which we mainly work with, i.e. VoxCeleb [112], BAIR [37] and UCF101 [141]. Below, we proceed to discuss our three main objectives.

1. Firstly, we focus on *high-quality video generation*. While the quality of generated samples in GANs is impacted by a number of factors such as objective function, model architecture and regularization, our major interest is *model architecture*, which is a novel topic in video generation. In particular, we aim at enabling generators to produce samples containing (a) temporal consistency, as well as (b) sharp and clear appearance. As we will discuss in Section 1.3, designing effective architectures that support this objective is challenging.

2. We proceed to *control generated videos* by disentangling the latent space. Specifically, we learn disentangled representations of the generative factors *appearance* and *motion*, which allow for individual and disjoint manipulation.

3. In addition, we aspire to discover semantics in the GAN-latent space, aiming to provide *interpretability of the generator*. Here, we place emphasis on finding interpretable motion representations that allow for control of subject-movement in generated videos. Fig. 1.3 illustrates two generated samples from our model, which we present in Section 6, i.e., talking heads and robot arms. In this objective we also develop evaluation

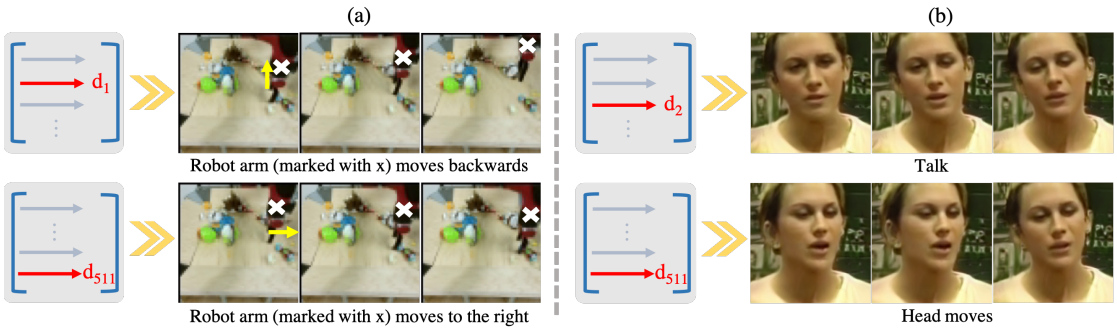


Figure 1.3 **Video manipulation.** (a) Moving robot arm and (b) talking head.

metrics, which quantify the correlation between representations and semantics. Such metrics are instrumental in comparing performance of each discovered representation.

1.2 Motivation

Recent development of social media platforms (e.g., TikTok and Snapchat) has brought to the fore remarkable interest in video content creation. In particular, users aim at designing short videos related to topics including fashion, sports, e-commerce and education. Due to the great potential in many domains, developing GANs for automated video generation has attracted increasing commercial and scientific attention associated to real-world applications and data augmentation, respectively, which we proceed to discuss.

Real-world applications. A key application for video generation has to do with *entertainment*. Creating short music videos, as well as Hollywood blockbusters entails considerable investments in time and resources. Video generation can greatly facilitate such a process. Trained by a large amount of movie data, machines can be beneficial in efficiently creating dynamic scenes. Current methods [17, 164, 163] have already been able to transfer motion from target video to input avatars. For example, enabling people to dance like Michael Jackson only requires few lines of code. In near future, we envision that video generation will allow audience to select their favorite actors, scenes and even storylines to produce custom made movies.

Further, video generation can greatly support video game development [77, 106]. By learning from massive gaming data, models are able to easily render interactions between agents and environments without additional requirement of game engines. Such portability will significantly decrease the time of development cycle, as games can be transferred and deployed directly across platforms and operating systems.

Data augmentation. Video generation is highly instrumental in data augmentation. Given growing model complexity, large-scale datasets are necessitated. Towards improving the model performance, enlarging existing datasets has become a direct and effective approach. For example, the enlargement from UCF101 [141] (10k videos) to Kinetics-400 [74] (300k videos) has substantially increased the accuracy of big visual models such as I3D [15] and 3D ResNet [51] in action recognition. However, collecting such large datasets is challenging. It requires months and even years of teamwork to download and preprocess datasets. In addition, the usage of downloaded videos leads to ethical problems related to privacy and transparency of personal information. Therefore, developing more efficient approaches for obtaining additional data is imperative. Several works [153, 152] explored the usage of generated (synthetic) data in training and highlighted related advantages in accessibility, controllability and privacy. GANs have inherently strong capacity in visual generation, incorporating two major benefits. Firstly, GANs allow for generation of unlimited amount of data. Therefore cumbersome data collection can be fully avoided. Secondly, GANs enable manipulation of generated samples. Attributes such as pose, illumination and scale, can be modified by walks in the latent space. Each manipulated sample is considered as an augmentation (or another 'view') of the original data. Such property is beneficial for a set of visual recognition tasks (e.g., view-invariant action recognition, self-supervised contrastive video learning), as invariance is learned from different 'views'. Predominantly, such 'multi-view' data is generally not available.

We proceed to discuss challenges related to open research questions.

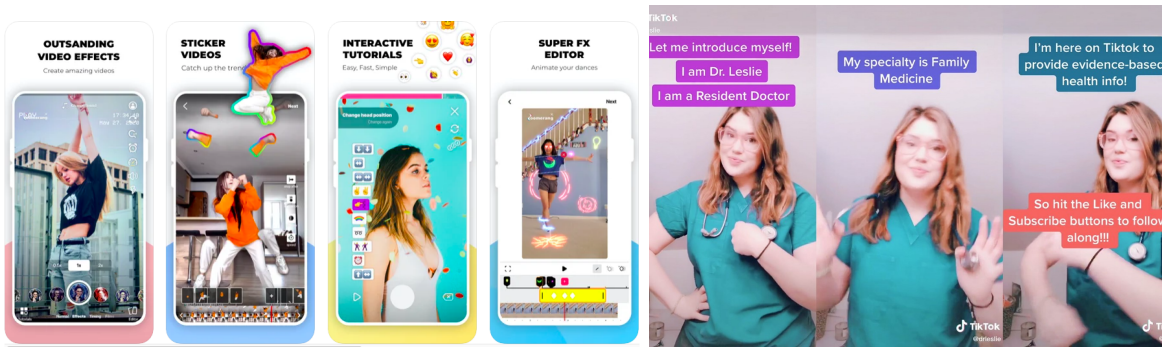


Figure 1.4 **Use cases for social media.** (Left) Inside functions of Snapchat for various contents creation. (Right) A user is spreading common knowledge on Tiktok.

1.3 Challenges

We have identified a set of challenges related to video generation that we proceed to enlist. This thesis is addressing some key challenges in both *visual generation* and *interpretability*. With respect to visual generation, the main challenges that we have addressed have to do with video representation and model design. For example, we have addressed following questions. How to represent a video? What should be an appropriate design of generator and discriminator? How to evaluate generated samples? With respect to interpretability, difficulties lie in the highly entangled latent space and model opacification. Developing methods to interpret the inner working of generative models remains a major research problem. Below, we discuss some challenges in detail.

Generator design. Deviating from an image only containing spatial information, a video incorporates a set of frames interconnected in the spatio-temporal domain. Towards modeling such inter-connections, appropriate generator architectures, which are able to endow random input noises with spatio-temporal consistency, are required. Generally, the objective of the generator is to upsample low-level representations to high-level semantics. It can be considered as an inverse problem of video understanding. Yet, generation remains a more challenging problem due to its extra requirements such as stable training, high visual-quality and interpretability. While two widely-used architectures in video understanding, namely 3D ConvNets and 2D ConvNet+RNN, have been explored reversibly for generation (Figure 1.5

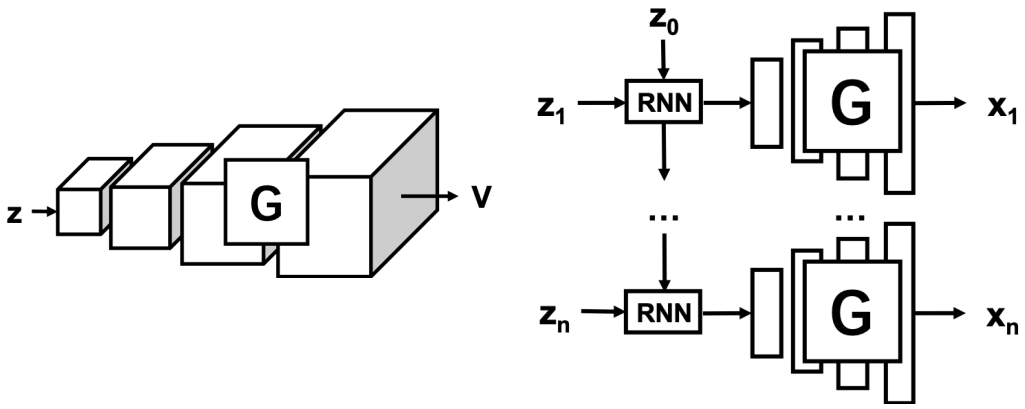


Figure 1.5 **Generator design.** (Left) 3D ConvNets based architecture. (Right) 2D ConvNets+RNN based architecture.

illustrates these two architectures), neither of the architectures outperforms the other due to respective limitations. 3D ConvNets suffer from large complexity with more training parameters, which may render models difficult to be optimized, impacting the visual quality negatively. On the other hand, incorporating an RNN in the generator, modeling long-term sequence, may result in unstable training owing to gradient vanishing and gradient explosion. Hence, designing good generator architecture is still challenging.

Discriminator design. Discriminators in GANs are beneficial in minimizing the distance between real and generated distributions. The associated capacity will strongly affect the quality of generated samples. Discriminators are accountable for ensuring that generated videos encompass *visual-quality*, as well as *temporal consistency*. For the latter, the transition between consecutive frames should be smooth. Having this in mind, design of discriminator-architectures remains an open question. In Chapter 4, we present a two-stream discriminator, which combines 3D ConvNets and 2D ConvNets to learn spatio-temporal distribution. In Chapter 6, we introduce a novel temporal pyramid discriminator equipped with only 2D ConvNets.

Entanglement of appearance and motion in videos. Appearance and motion are two major factors in videos. Without using additional information such as human keypoints or optical flow, learning to disentangle such factors is challenging. It requires building specific

model components to represent both factors, respectively. Due to lack of explicit formulation, designing model components to disentangle these two factors remains challenging. In Chapter 5 and Chapter 6, we introduce two different disentangling approaches, as well as their comparisons.

Interpretability. Deep neural networks have been widely used as black-boxes, and GANs are no exception. Due to the large amount of parameters, it is difficult to identify the types of knowledge GANs have learned. In addition, given that features such as textures, concepts, semantics and objects are represented in a hierarchical manner in GANs [9], discovering and locating information of interest becomes difficult. To interpret different features, specific methods are usually required. For example, to interpret attributes (e.g., gender, age) in StyleGANs [72, 73], pretrained classifiers were used to provide scores for generated samples [135]. At the same time for interpretation of pose, landmark detectors are required. In this thesis, we aim at interpreting motion in video GANs. However lack of prior knowledge is the most difficult part in this challenge. In Chapter 6, we discuss this in detail.

Evaluation. Lack of effective evaluation metrics is a major challenge in current GAN research. Since 'realism' mostly depends on human perception, subjective analysis has become a standard metric, which is inefficient and time-consuming. Towards evaluating GANs in an objective manner, two quantitative evaluation metrics, namely Inception Score (IS) [132] and Fréchet Inception Distance (FID) [54], have been proposed. They both use statistical methods, that rely on features extracted from pretrained models on large-scale datasets, in order to measure the distance between real and generated distributions. Due to large variability in space and time, evaluation in video generation remains challenging.

Data variation. In contrast to datasets available for image generation, which incorporate well-aligned faces (e.g., CelebA [98]) and objects (e.g., ImageNet [30]), datasets available for video generation entail richer variations in both, appearance and motion. In particular, complex scenes may contain multiple objects with different textures, poses and shapes. Further, each object has a moving trajectory and can encompass a set of variations related

to illumination and occlusion within one video. In addition, videos incorporate different movements in foreground and background. Learning a general model, which captures such spatio-temporal variations is challenging.

1.4 Thesis Outline

In this thesis we firstly design three GANs streamlined to generate human videos, and proceed with an innovative video GAN approach that allows for motion-interpretation. We then change gears and leverage GAN-generated data on a real-world problem, namely unsupervised person re-identification. These contributions are organized in following chapters.

Chapter 2 revisits literature with particular focus on (i) image and video generation, and (ii) GAN-based person re-identification.

Chapter 3 presents ImaGINator [169], a conditional GAN, which learns to generate video from a single input image. We introduce a spatio-temporal skip-connection architecture, in order to transfer multi-scale feature maps directly from encoder to decoder. We also incorporate motion class-labels (e.g., facial expressions, human actions) into the latent space for controllable video generation. Related results show that our proposed model is able to preserve well the appearance information, as well as generate videos corresponding to the class-labels.

Chapter 4 introduces G^3AN [166], a spatio-temporal generative model, which seeks to capture the distribution of high dimensional video data and to model appearance and motion in disentangled manner. As opposed to ImaGINator, G^3AN takes noise vectors from prior distribution, rather than images as inputs. It is able to generate videos in both, conditional and unconditional manner. We propose a three-stream Generator to decompose appearance and motion, 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. Experimental results show that such design can be instrumental in disentangling appearance and motion, as well as in manipulating generated results from both spaces.

Chapter 5 presents MintGAN [172], an unconditional video generative model, with a twofold objective, namely to generate high quality videos, as well as to allow for interpretation of the latent space. In this context, we place emphasis on motion interpretation. Towards this, we design the architecture of MintGAN-generator in accordance to proposed Linear Motion Decomposition (LMD) to decompose motion into semantic sub-spaces. LMD carries the assumption that motion can be represented by a dictionary, where the vectors forming an orthogonal basis in the latent space and each vector in the basis represent a semantic sub-space. To quantify motion in these sub-spaces, we propose a new evaluation metric by leveraging optical flow of generated samples. By doing this, we discover the semantic meanings inside those spaces by computing motion differences between activation and deactivation of related vectors. We also find that motion of generated videos can be controlled by manipulating those sub-spaces.

Chapter 6 describes GCL [20], a method to combine generative modeling and contrastive learning to boost the performance of unsupervised Person ReID. We design the generator to synthesize novel-view images to simulate multi-camera settings in the real-world. We leverage generated data as data augmentation for contrastive learning. Specifically, we propose a mesh-based view generator, which produces results that serve as references towards generating novel views of a person. In addition, we propose a view-invariant loss to facilitate contrastive learning between original and generated views. Deviating from previous GAN-based unsupervised Person ReID methods involving domain adaptation, we do not rely on a labeled source dataset, which renders our method more flexible.

Chapter 7 discusses future work and concludes this thesis.

1.5 Contributions

We proceed to list all publication contributions, as well as software that we developed in the course of this thesis. We will detail the contributions of four publications in Chapters 3-6.

1.5.1 Publications

— Y. Wang, A. Dantcheva, J. Broutart, P. Robert, F. Bremond, P. Bilinski. Comparing methods for assessment of facial dynamics in patients with major neurocognitive disorders. In *ECCV Workshop*, 2018. [177]

— Y. Wang, A. Dantcheva, F. Bremond. From attribute-labels to faces: face generation using a conditional generative adversarial network. In *ECCV Workshop*, 2018. [174]

— Y. Wang, P. Bilinski, F. Bremond, and A. Dantcheva. ImaGINator: Conditional Spatio-Temporal GAN for Video Generation. In *WACV*, 2020. [169] (Chapter 3)

— Y. Wang, P. Bilinski, F. Bremond, and A. Dantcheva. G3AN: Disentangling Appearance and Motion for Video Generation. In *CVPR*, 2020. [167] (Chapter 4)

— Y. Wang, A. Dantcheva. A video is worth more than 1000 lies. Comparing 3DCNN approaches for detecting deepfakes. In *FG*, 2020. [173]

— D. Yang, R. Dai, Y. Wang, R. Mallick, L. Minciullo, G. Francesca, and F. Bremond. Selective Spatio-Temporal Aggregation Based Pose Refinement System. In *WACV*, 2021. [190].

— Y. Wang, F. Bremond, and A. Dantcheva. MintGAN: Motion Interpretation in Video Generation. *arXiv*, 2021 [172] (Chapter 5)

— H. Chen*, Y. Wang*, B. Lagadec, A. Dantcheva, and F. Bremond. Joint Generative and Contrastive Learning for Unsupervised Person Re-identification. In *CVPR*, 2021 [20] (Chapter 6)

— D. Yang*; Y. Wang*; A. Dantcheva; L. Garattoni; G. Francesca; and F. Bremond. UNIK: A Unified Framework for Real-world Skeleton-based Action Recognition. In *BMVC* 2021 [192].

— D. Yang*; Y. Wang*; A. Dantcheva; L. Garattoni; G. Francesca; and F. Bremond. Self-Supervised Video Pose Representation Learning for Occlusion-Robust Action Recognition. In *FG* 2021 [191]

1.5.2 Software contributions

The code for four chapters of this thesis has been publicly released.

— ImaGINator: The code and pretrained models presented in [169] (Chapter 3) are available in https://github.com/wyhsirius/ImaGINator

— G^3AN: The code and pretrained models presented in [167] (Chapter 4) are available in https://wyhsirius.github.io/G3AN/

— MintGAN: The code and pretrained models presented in [172] (Chapter 5) are available in https://wyhsirius.github.io/InMoDeGAN/

— GCL: The code and pretrained models presented in [20] (Chapter 6) are available in https://github.com/chenhao2345/GCL

Chapter 2

Literature Review

We here revisit literature related to the topics covered in this thesis.

2.1 Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs), as introduced by Goodfellow *et al.* [48], incorporate two networks, a *Generator*, which generates new data instances and a *Discriminator*, which evaluates them for authenticity. The generator accepts noise as input and generates new samples of data in line with the observed training data. GANs have succeeded in applications such as image [13, 72, 73, 215, 64] and video generation [148, 157, 167, 172, 169], robotics [139, 58, 123] and medical image analysis [194, 181, 75, 179].

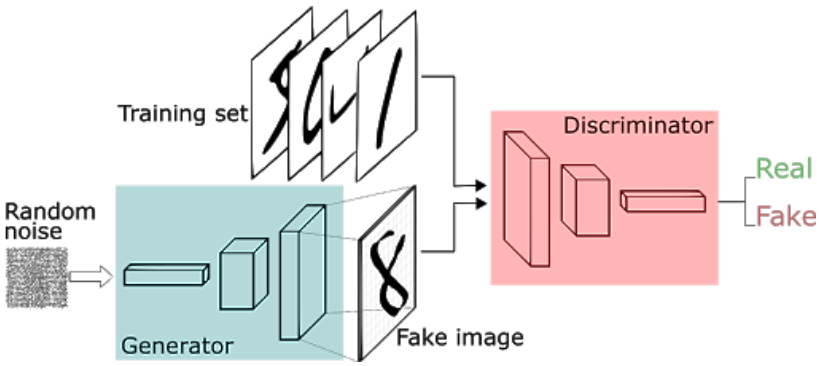


Figure 2.1 **Generative Adversarial Network.**

The objective of GANs is to learn a data distribution p_{data} from real samples $x \sim p_x$ via adversarial learning. G takes a noise vector z from a prior distribution p_z as input and produces a sample $G(z)$. D learns to infer p_{data}, where input data is drawn. D is trained to maximize the probability of assigning the correct label to both, real and generated samples, while G is simultaneously trained to minimize $log(1-D(G(z))$. Training is achieved via solving a two-player minimax game with value function:

$$\min_G \max_D V(G,D), \tag{2.1}$$

where

$$V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z))]. \tag{2.2}$$

However, in practice, a so called vanilla GAN suffers from several problems such as unstable training, mode collapse and low-quality results. Towards solving these issues, techniques such as novel loss functions [5, 11, 80, 50, 104] (see Tab. 2.1), regularization [80, 107, 127], normalization layers [110, 50], training strategies [55] and architectures [13, 72, 73, 111, 122] have been proposed. In this thesis, we have integrated some of the techniques in proposed video GANs towards achieving better visual quality and more stable training.

GAN	Discriminator Loss	Generator Loss
MM GAN [48]	$\mathscr{L}_D^{GAN} = -\mathbb{E}_{x\sim p_d}[log(D(x))] - \mathbb{E}_{\hat{x}\sim p_g}[log(1-D(\hat{x}))]$	$\mathscr{L}_G^{GAN} = \mathbb{E}_{\hat{x}\sim p_g}[log(1-D(\hat{x}))]$
NS GAN [48]	$\mathscr{L}_D^{NSGAN} = -\mathbb{E}_{x\sim p_d}[log(D(x))] - \mathbb{E}_{\hat{x}\sim p_g}[log(1-D(\hat{x}))]$	$\mathscr{L}_G^{NSGAN} = -\mathbb{E}_{\hat{x}\sim p_g}[log(D(\hat{x}))]$
WGAN [5]	$\mathscr{L}_D^{WGAN} = -\mathbb{E}_{x\sim p_d}[D(x)] - \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$	$\mathscr{L}_G^{WGAN} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
WGAN-GP [50]	$\mathscr{L}_D^{WGANGP} = \mathscr{L}_D^{WGAN} + \lambda\mathbb{E}_{\hat{x}\sim p_g}[(\left\|\nabla D(\alpha x + (1-\alpha\hat{x})^2\right\|_2 - 1)^2]$	$\mathscr{L}_G^{WGANGP} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
LSGAN [104]	$\mathscr{L}_D^{LSGAN} = -\mathbb{E}_{x\sim p_d}[(D(x)-1)^2] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})^2]$	$\mathscr{L}_G^{LSGAN} = -\mathbb{E}_{\hat{x}\sim p_g}[(D(\hat{x}-1))^2]$
DRAGAN [80]	$\mathscr{L}_D^{DRAGAN} = \mathscr{L}_D^{GAN} + \lambda\mathbb{E}_{\hat{x}\sim p_g+\mathscr{N}(0,c)}[(\left\|\nabla D(\hat{x})\right\|_2 - 1)^2]$	$\mathscr{L}_G^{DRAGAN} = \mathbb{E}_{\hat{x}\sim p_g}[log(1-D(\hat{x}))]$
BEGAN [11]	$\mathscr{L}_D^{BEGAN} = -\mathbb{E}_{x\sim p_d}[\left\|x - AE(x)\right\|_1] - k_t\mathbb{E}_{\hat{x}\sim p_g}[\left\|\hat{x} - AE(\hat{x})\right\|_1]$	$\mathscr{L}_G^{BEGAN} = \mathbb{E}_{\hat{x}\sim p_g}[\left\|\hat{x} - AE(\hat{x})\right\|_1]$

Table 2.1 **Generator and discriminator loss functions in different GANs.**

2.2 Evaluation metrics of GANs

Evaluation of GANs is challenging as it requires measuring both, visual-quality and diversity of generated samples. Two main evaluation metrics, i.e., Inception Score (IS) [133] and Fréchet Inception Distance (FID) [54] were proposed, in order to quantify the performance of GANs.

2.2.1 Inception Score (IS)

IS uses an Inception V3 [144] model pre-trained on ImageNet as feature extractor, it can be computed by

$$IS(G) = exp(\mathbb{E}_{x \sim p_g} D_{KL}(p(y|x) \mid\mid p(y))), \tag{2.3}$$

where $x \sim p_g$ indicates that x is an image sampled from p_g, $p(y|x)$ is the conditional class distribution, and $p(y) = \int_x p(y|x) p_g(x)$ is the marginal class distribution.

To compute IS, firstly an empirical marginal class distribution from sampled $x^{(i)}$ is required,

$$\hat{p}(y) = \frac{1}{N} \sum_{i=1}^{N} p(y|x^{(i)}), \tag{2.4}$$

where N is the number of images generated from GAN. Then an approximation to the expected KL-divergence is obtained using

$$IS(G) \approx exp(\mathbb{E}_{x \sim p_g} D_{KL}(p(y|x^{(i)}) \mid\mid \hat{p}(y))). \tag{2.5}$$

In practice, it is recommended to conduct 10 times computation with $N = 5,000$ and report the mean and standard deviation of the final score. Higher IS indicates better image quality.

2.2.2 Fréchet Inception Distance (FID)

While IS correlates well with human judgement of image quality, it does not consider the statistics of training dataset. Towards overcoming this drawback, FID was proposed to compare statistics of generated samples with the real world training samples. Similar

to IS, FID also uses an Inception V3 model [144] pre-trained on ImageNet to extract features and consider polynomials of coding unit functions. In practice, it only considers the first two polynomials, mean and covariance. It assumes that the coding units to follow a multidimensional Gaussian and the difference of two Gaussian is measured by Fréchet Distance [42]. The Fréchet Distance $d(.,.)$ between the Gaussian with mean and covariance (m,C) obtained from model sample distribution $p(.)$ and the Gaussian (m_w,C_w) obtained from real world distribution $p_w(.)$ is called the "Fréchet Inception Distance" (FID), which is denoted by [36]:

$$d^2((m,C),(m_w,C_w)) = \|m-m_w\|_2^2 + Tr(C+C_w-2(CC_w)^{1/2}). \tag{2.6}$$

In practice, it is recommended to generate more than 10,000 samples to compute FID, in order to prevent underestimated results. The lower FID values indicate better performance of GANs. Since FID considers the statistics of both, generated and real-world samples, it can provide more reliable result than IS in comparing performance of different GANs.

In this thesis, as we focus on video generation, we need to compare the distance of spatio-temporal distribution between generated and real video data. In Chapter 4- 6, we compute FID and IS by replacing Inception V3 with a spatio-temporal model ResNeXt101 [51] pre-trained on Kinetics [15], a large scale video understanding dataset. Previous work [130, 150] explored using other spatio-temporal models, e.g., C3D [146] and I3D [15] as feature extractors. We choose ResNeXt101 due to its state-of-the-art performance on Kinetics.

2.3 Image generation

In this section, we review previous research on the most prominent application of GANs, namely image generation. Specifically, we will mainly discuss unconditional (Section 2.3.1) and conditional image generation (Section 2.3.2). As introduced in Section 2.1, training of unconditional image generation is achieved via solving a two-player minimax game with the

objective

$$\mathscr{L}_{GAN}(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z))]. \tag{2.7}$$

In conditional GANs, G takes an additional input y as the control signal. The discriminator distinguishes real from fake by leveraging the information in y. In this case, the objective is

$$\mathscr{L}_{cGAN}(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log(D(x,y))] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z,y))]. \tag{2.8}$$

2.3.1 Unconditional image generation

Unconditional image generation aims at learning to map from a prior distribution (e.g., Gaussian) to a real-world data distribution. In this setting, models are usually trained to generate category-specific datasets (e.g., faces [98, 72], kitchens [196], cars [196], etc.).

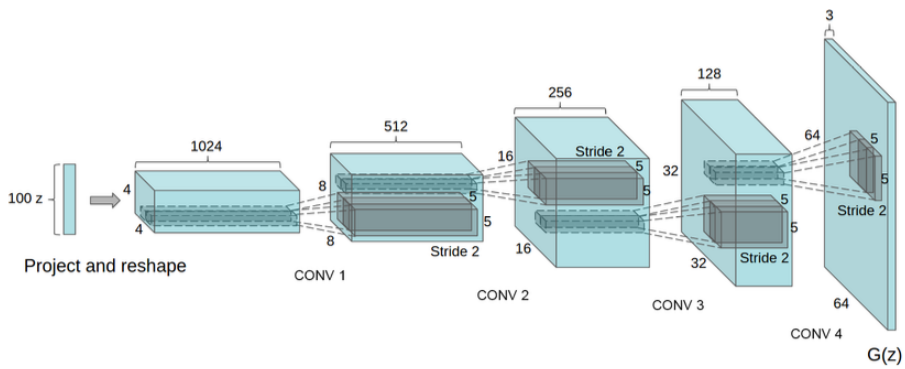


Figure 2.2 **Generator of DCGAN.**

DCGAN [122] firstly incorporated a fully convolutional architecture for unconditional image generation and unsupervised representation learning. We illustrate associated generator in Fig. 2.2. DCGAN contains five convolutional layers in both generator and discriminator respectively and can upsample an input vector to a 64×64 image. Although the results are preliminary, the design of DCGAN has strongly inspired most following GANs.

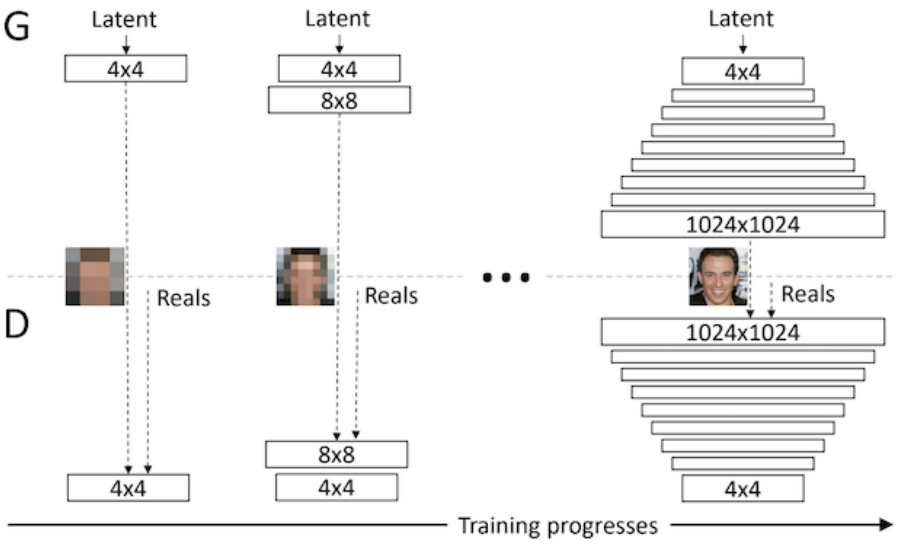
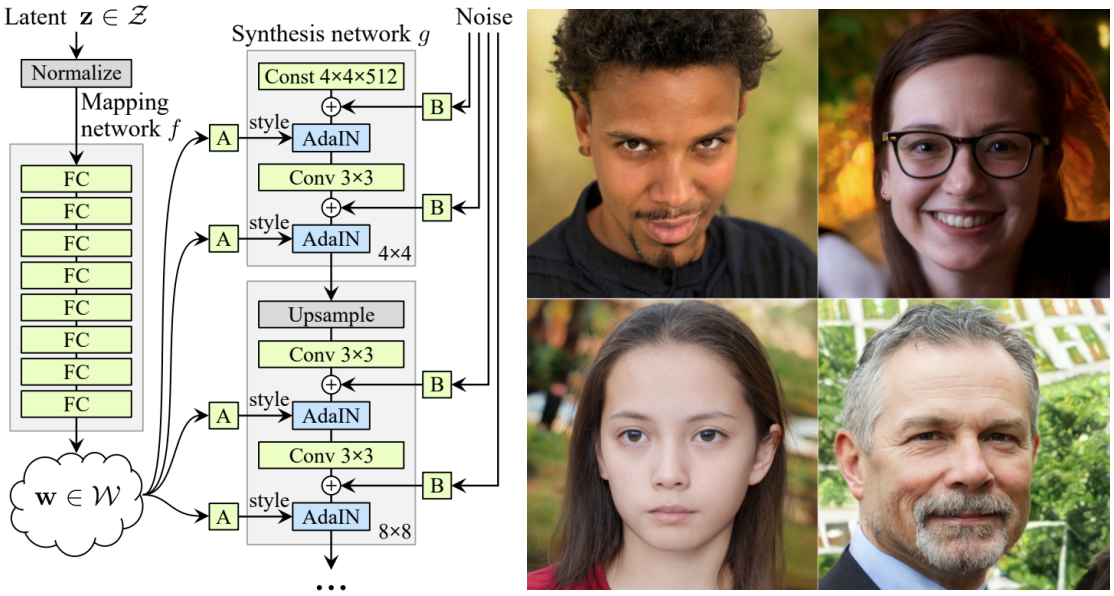


Figure 2.3 **Generator architecture of Progressive GAN.**

Recently, unconditional image generation has witnessed considerable progress. Progressive GAN [71] proposed a progressive training strategy (see Fig. 2.3) for high-resolution image generation. It started from generating low-resolution images (4×4) and reached high resolution, stage by stage. In each stage, it fused the RGB output from previous stages into current feature maps towards training a new model for higher resolution. Due to the progressive training, it was the first method, which can produce images of 1024 resolution. However, since its architecture still followed the design of DCGAN, there are still obvious artifacts in the generated images and some details are missing. In order to improve learning of the distribution pertained to training data, StyleGAN [72] (see Fig. 2.4a) introduced a novel style generator, which incorporates the Adaptive Instance Normalization (AdaIN) [60] in each convolutional layer. Deviating from previous methods, where the input vector can only be seen by the first layer, in StyleGAN, input noise vectors are mapped by a 8-layer MLP into intermediate latent codes, which are fed into each AdaIN layer as style information. Such design provides a more explicit way to control the style of generated images. Since the input vectors control the style information in all convolutional layers, it can generate high-quality images containing more details. Towards removing further artifacts, StyleGAN2 [73] introduced a demodulation layer to replace the original AdaIN. In addition, residual modules



(a) StyleGAN generator

(b) Images generated by StyleGAN2

Figure 2.4 **Architecture and generated samples of StyleGAN(2).** (a) StyleGAN generator architecture. (b) Generated images from StyleGAN2.

were used in both generator and discriminator to reduce the training time and further improve the image quality. We show generated samples from StyleGAN2 in Fig. 2.4b.

2.3.2 Conditional image generation

Conditional image generation aims at controlling generated images using various additional input information such as class labels, audio and images. Below, we proceed to discuss models using category labels and images.

Conditioned on category labels. In class-conditioned image generation, one crucial challenge has to do with integration of category labels in generator and discriminator. This profoundly influences the diversity and quality of generated images, as investigated in previous work [109, 113, 110] (see Fig. 2.5). CGAN [109] firstly introduced the concatenation of category labels and inputs in both, generator and discriminator. ACGAN [113] developed this idea by adding an auxiliary branch in discriminator towards classifying the real and generated images. Both methods have achieved good results in low-resolution datasets with

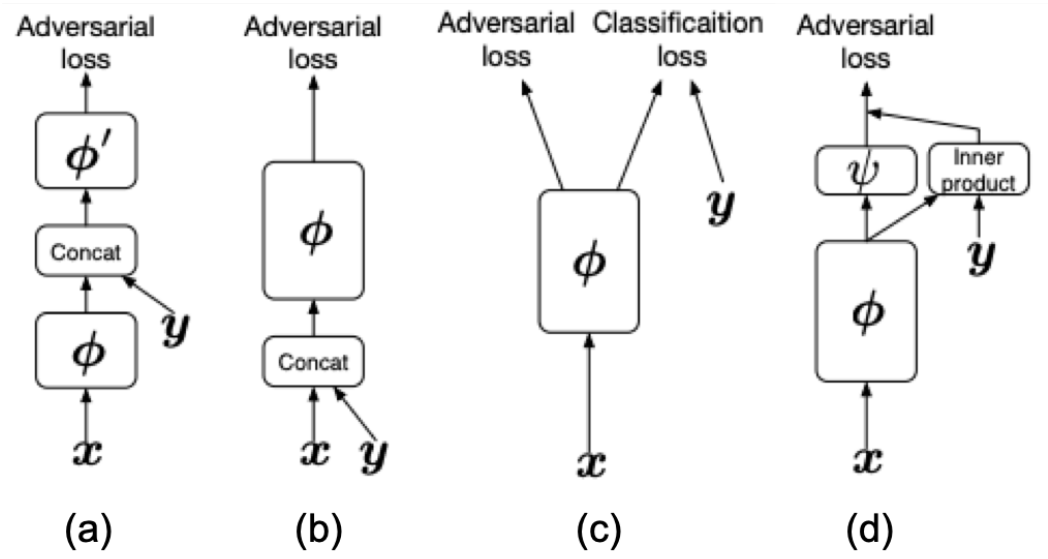


Figure 2.5 **Discriminators for class-conditioned image generation.** (a) cGANs (input concat). (b) cGANs (hidden concat). (c) ACGANs. (d) SNGAN.

few categories (e.g., CIFAR10). However when trained with large-scale datasets such as ImageNet [31], their performance significantly decreases. The reason is twofolds. Firstly, their architectures are based on DCGAN, which is not able to support high-resolution image generation. Secondly, the integration of category labels fails in discriminating different categories, in particular when the number of categories are large. Towards solving these two problems, SNGAN [110, 111] employed to use residual blocks in both generator and discriminator, achieving higher resolution images. To better learn the distributions of various categories in ImageNet,, Spectral Normalization (SN) and class-label projection in discriminator were employed. Associated experimental results showed that such two techniques significantly improve the performance of the model with much lower FID on ImageNet comparing to all the previous methods. BigGAN [13] followed this idea and designed a larger model (see Fig. 2.6). Towards finding the best strategy to train a generator for high-quality generation, BigGAN conducted a large number of experiments to explore combinations of different advanced techniques such as spectral normalization [110], category label projection [111], self-attention [200], conditional batch normalization [29] and TTUR [55]. When trained

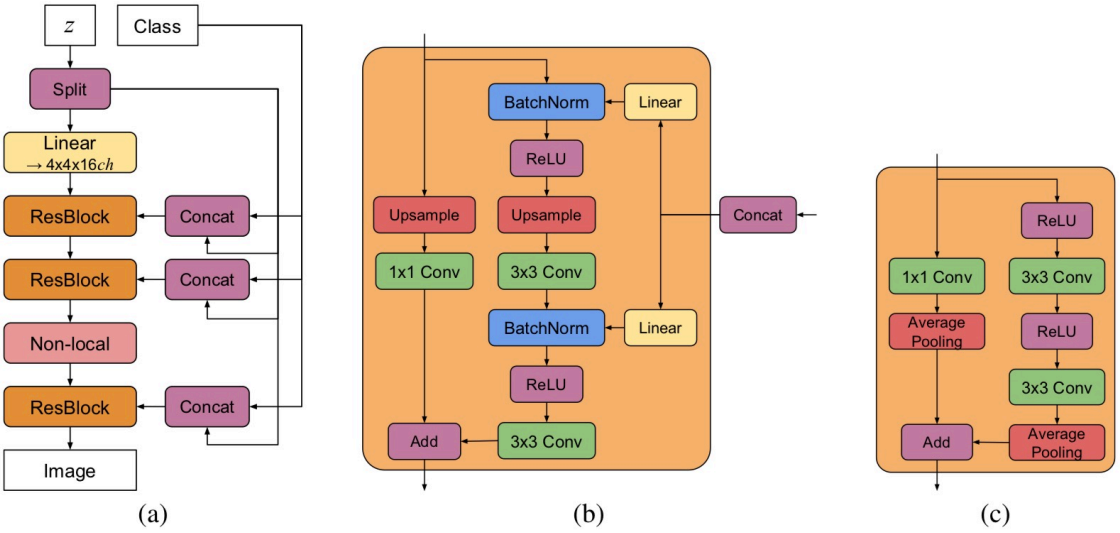


Figure 2.6 **BigGAN architecture.** (a) Generator. (b) Residual block in generator. (c) Residual block in discriminator.

under optimal settings, BigGAN achieved remarkable results on conditional image generation on ImageNet w.r.t. both FID and visual quality. We illustrate generated samples in Fig. 2.7.

Figure 2.7 **Generated samples from BigGAN.**

Conditioned on images. Images as condition can be considered as building a translation from a source image to a target one, and it is referred to as image-to-image translation. Pix2pix [64] was a first general framework in this context that combined perceptual and adversarial losses. We show the framework in Fig. 2.8. Pix2pix can translate semantic maps, depth maps, grey-scale images and even sketch images into new RGB images. However, the requirement of paired data (source-target pairs) during training time constrains its application

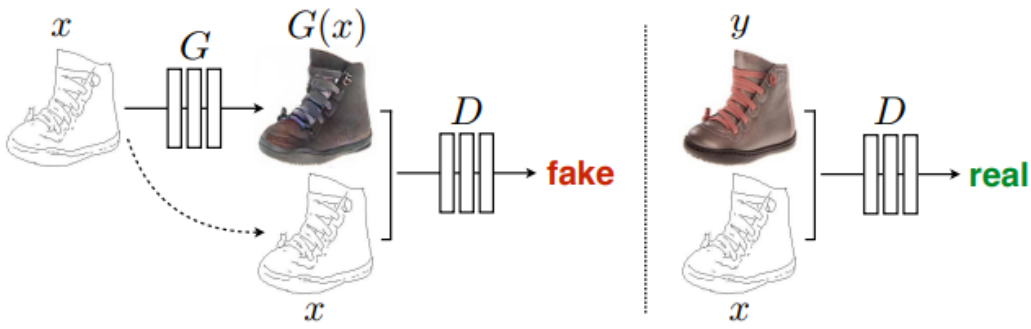


Figure 2.8 **Framework of Pix2pix.**

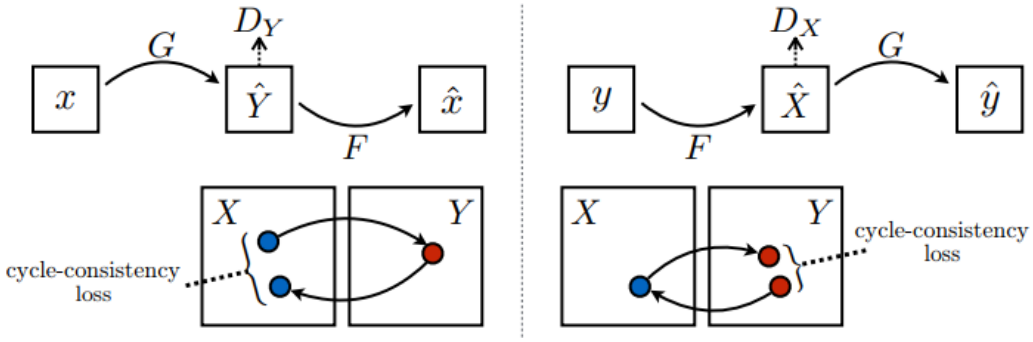


Figure 2.9 **Framework of CycleGAN.**

in some real-world cases (e.g., style transfer). To tackle this issue, CycleGAN [215] introduced a cycle consistency loss (see Fig. 2.9) aiming to translate images between different domains for unpaired data. Recently, towards reducing training time of cycle consistency loss, CUT [117] proposed a patchwise contrastive loss (PatchNCE) for unpaired image-to-image translation, which learns distribution of patches between input and output images. Another line in image-to-image translation is to learn disentangled representations while translating images. Several work [61, 96, 85, 47] proposed to use two different encoders to represent structure and content respectively from source and target images. By involving noises into the latent space, these methods can also translate images in a multi-modal manner. In Chapter 6, we will introduce our data augmentation method for unsupervised person ReID. We follow the idea of Pix2pix and MUNIT [61] to generate images of a person with different viewpoints using 3D mesh images. Our idea is to construct positive-negative samples for contrastive learning.

2.4 Video generation

In this section, we review works on unconditional (Section 2.4.1) and conditional video generation (Section 2.4.2).

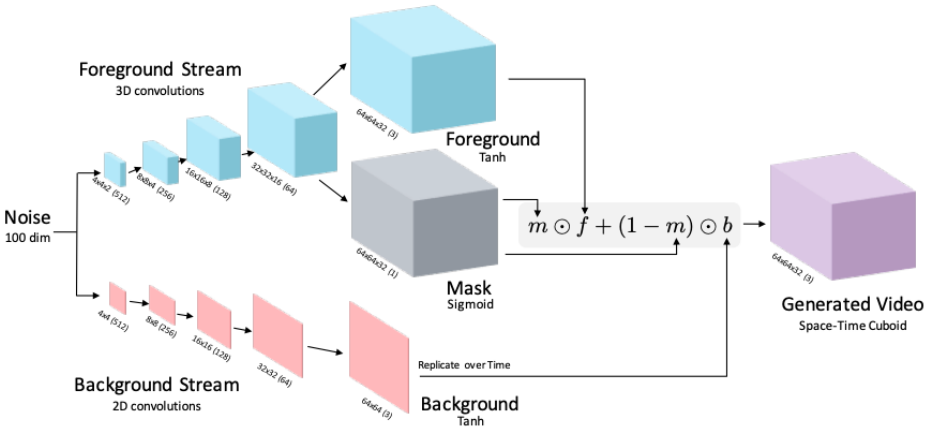


Figure 2.10 **VGAN Architecture.**

2.4.1 Unconditional video generation

Unconditional video generation aims at generating videos *without* additional information. Examples of methods include VGAN [157], TGAN [130] and MoCoGAN [148]. VGAN is equipped with a two-stream generator to generate foreground and background, respectively (see Fig. 2.10). In particular, foreground is generated using a spatio-temporal generator (3D generator), whereas background is produced using a 2D generator. In addition, spatio-temporal convolutions are used in the discriminator to learn the video distribution. Similarly, TGAN firstly generated a set of latent vectors corresponding to each frame using temporal convolutional layers (1D transposed convoltutions) and then transformed them into actual images using normal 2D generator (see Fig. 2.11). Further, TGAN proposed a singular value clipping method to improve the capacity of the discriminator. MoCoGAN aimed at decomposing latent representation into motion and content, in order to control both factors in the generated results. It used a GRU to model temporal information in the latent space. Each input of GRU is a noise vector sampled from Gaussian distribution. Content information

is represented by another noise vector and concatenated with each motion representation. Videos are produced using a 2D generator from such concatenated latent representation (see Fig. 2.13). MoCoGAN combined 3D and 2D discriminators to further improve the quality of generated videos. In Chapter 4 and Chapter 5, we introduce our approaches, which follow the line of research in learning disentangled and interpretable latent spaces in unconditional video generation.

2.4.2 Conditional video generation

In contrast to unconditional video generation methods, conditional video generation methods aim at controlling generated results using additional input information. Specifically, we discuss two types of conditions, class-labels and videos.

Conditioned on category labels. Category labels in this context are related to motion in the training datasets (e.g., facial expressions and human actions). Methods for class-conditioned image generation can also be applied to video generation. Current methods [157, 148, 130] show that using category labels in both generator and discriminator can control the generated videos. However, since appearance and motion are entangled, one major challenge has to do with independently controlling motion, without affecting appearance. In Chapter 3 and Chapter 4, we introduce two different approaches to tackle this challenge.

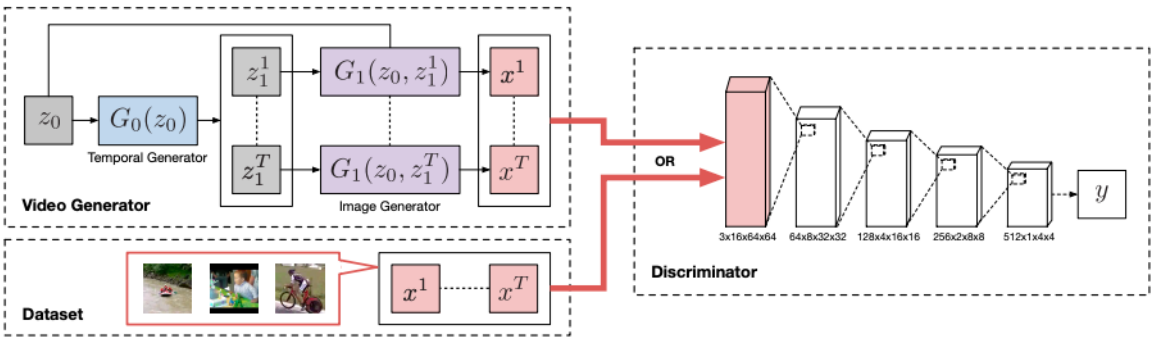


Figure 2.11 **TGAN Architecture.**

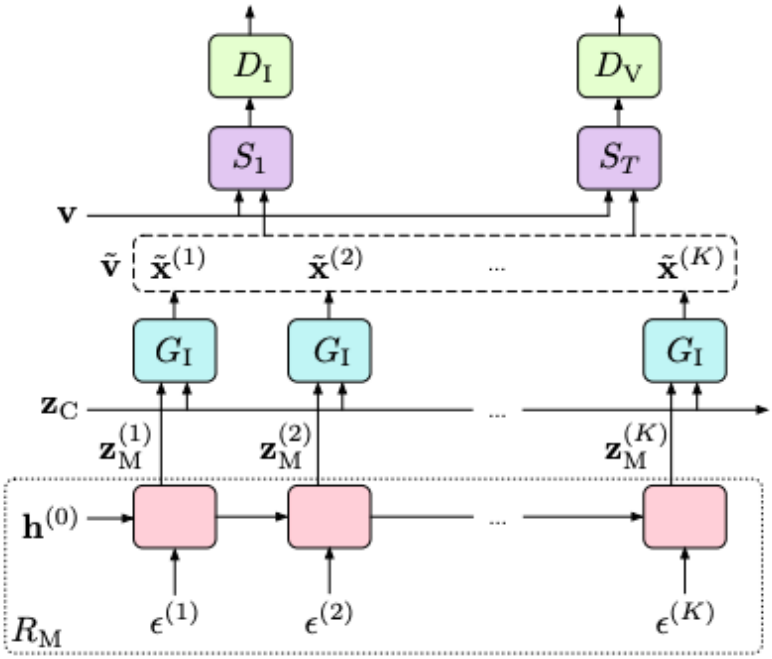


Figure 2.12 **MoCoGAN Architecture.**

Conditioned on videos. Due to challenges in modeling high dimensional video data, additional information such as semantic maps [116, 164, 163], human keypoints [67, 188, 161, 17, 199, 163], 3D mesh [204] as well as optical flow [89, 114] have been exploited to guide appearance and motion generation. Different from these methods, in this thesis, we aim to learn the full distributions of training datasets and generate videos without relying on additional information.

2.5 Interpretability of GANs

In an effort to open the black box representing GANs, very recent work has proposed different methods to understand both latent representations and inner representations of generator. Bau *et al.* [8, 9] sought to associate neurons in the generator with encoding of pre-defined visual concepts such as colors, textures and objects. Subsequent work [135, 46, 66, 158] proceeded to explore the interpretability of the latent space, seeking for latent representations corresponding to semantics in generated images. Supervised methods attempted to involve

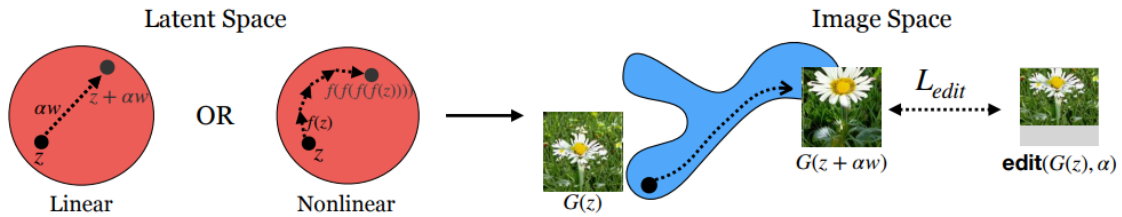


Figure 2.13 **Linear and non-linear latent walks.**

classifiers to discover the latent representation related to pre-defined concepts such as facial attributes [135], object structures [135, 66] and geometric transformations [66]. To learn these concepts in a supervised manner, a large amount of generated results are required, which renders these methods inefficient. Towards mitigating cumbersome and costly annotation, unsupervised learning methods were proposed, most notably PCA [62], SVD [136], Hessian Matrix [120] and pre-defined learnable matrix [158] in latent space. Linear [135, 66] and non-linear [66] *walks* in found interpretable latent representations enabled semantic concepts in the generated images to be modified.

In Chapter 5, we introduce our proposed approach to design an interpretable video GAN. Deviating from previous methods, our evolved architecture allows for high-quality video generation. We prioritize to interpret and manipulate *motion* in generated videos. We do so by instilling a-priori the generator with a motion representation module, which learns interpretable motion-components during training, rather than interpreting a-posteriori a pre-trained generator.

Chapter 3

Conditional Spatio-Temporal GAN for Video Generation

In this chapter, we study the problem of generating human videos from single images. It entails the challenging simultaneous generation of realistic and visually appealing appearance and motion. In this context, we propose a novel conditional GAN architecture, namely ImaGINator, which given a single image, a condition (e.g., motion-label of a facial expression or action) and noise, decomposes appearance and motion in both latent and high level feature spaces, generating realistic videos. This is achieved by (i) a novel spatio-temporal fusion scheme, which generates dynamic motion, while retaining appearance throughout the full video sequence by transmitting appearance (originating from the single image) through all layers of the network. In addition, we propose (ii) a novel transposed (1+2)D convolution, factorizing the transposed 3D convolutional filters into separate transposed temporal and spatial components, which yields significant gains in video quality and speed. We extensively evaluate our approach on the facial expression datasets MUG and UvA-NEMO, as well as on the action datasets NATOPS and Weizmann. We show that our approach achieves significantly better quantitative and qualitative results than the state-of-the-art.

3.1 Introduction

Generating realistic human videos based on single images brings to the fore following three challenges: (a) retaining of human identity appearance throughout the video, (b) generating (uncertain) motion, as well as (c) modeling of spatio-temporal consistency. Finding suitable representation learning methods, which are able to address these challenges, is critical to the final visual quality and plausibility of the rendered novel video sequences.

Existing methods predominantly treat generation of high dimensional video as a separate *two step* modeling of low-dimensional temporal and spatial generation. Such methods (e.g. MoCoGAN) [148], are grounded on the *seq2seq* [143] architecture. In particular associated video generation in such methods includes two steps: (1) motion generation in a latent space, proceeded by (2) motion and appearance-generation, where frames are generated individually, combining the single-input-image-appearance information with each motion vector generated in (1). These two steps aim at decomposing video generation into the generation of individual frames, which imparts the benefit of straightforward optimization. Two step methods fail to address the above named challenges (a) and (c), i.e. appearance is not sufficiently retained and spatio-temporal consistency is not modeled, as temporal consistency is not modeled in higher level spatial spaces.

In contrast to two step methods, VGAN [157] utilized a *single step* to generate future frames by leveraging on 3D convolution to model spatio-temporal features in high and low levels. We here note that utilizing 3D convolution directly challenges optimization. In addition, the generated video was decomposed into foreground and background, in two streams, which required an additional branch to model background information, increasing the complexity of the model.

Motivated by the above, in this chapter we present a novel conditional GAN model, referred to as ImaGINator, generating video sequences given a single image, a motion-category label (i.e., facial expression or human action), as well as noise. ImaGINator incorporates a *Generator* G, a *video Discriminator* D_V, as well as an *image Discriminator* D_I, as depicted in Fig. 3.1. It is streamlined to exploit the joint benefits of single and two-step methods by incorporating several new properties. First, we propose a novel *spatio-temporal*

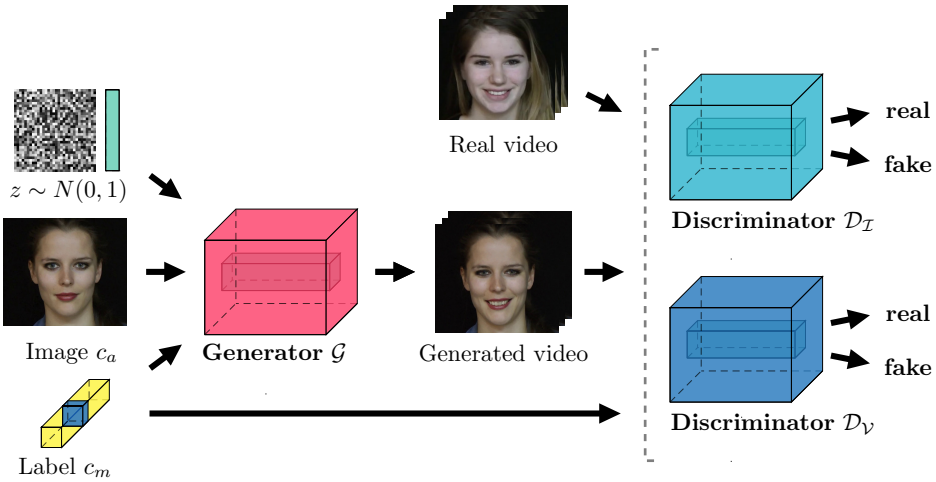


Figure 3.1 **ImaGINator architecture.** The proposed ImaGINator architecture incorporates *Generator G*, *image Discriminator* D_I, as well as *video Discriminator* D_V. G accepts c_a, c_m and noise as input, and seeks to generate realistic video sequences. While D_I discriminates whether the generated images contain an authentic appearance, D_V additionally determines whether the generated videos contain an authentic motion.

fusion mechanism, aiming at *retaining the appearance* by enforcing G to employ the spatial information in both, low and high feature levels. By injecting motion-category label into the *Decoder*, we enable G to place emphasis on generating solely motion. This is based on the hypothesis that a video can be disentangled into appearance and motion in the latent space, as well as in multi-level spatio-temporal feature spaces. While at each level appearance is retained, only the motion is being altered. Second, we introduce a novel *transposed (1+2)D convolution*, factorizing the transposed 3D convolutional filters into separate temporal and spatial components. This brings several benefits: (i) an additional nonlinear rectification allows the model to represent more complex functions, (ii) it facilitates optimization, as transposed (1+2)D convolution blocks are easier to optimize than the full transposed 3D convolutional filters, and (iii) it yields significant gains in both video quality and speed.

Towards comparing our algorithm with other video generation algorithms, we augment two state-of-the-art video generation algorithms, namely VGAN and MoCoGAN, in order to adhere to our problem setting. We proceed to provide a comparison, showing that our method outperforms these methods *qualitatively* (based on a human study of 30 subjects) and *quantitatively* on both, facial expression (MUG and UvA-NEMO), as well as human action

datasets (Weizmann and NATOPS) by presenting results pertaining to five evaluation metrics. In addition, we conduct an ablation study, which validates the effectiveness of components in ImaGINator.

3.2 Background

Conditional Generation accepts as inputs both, latent variables, as well as known auxiliary information, such as class labels. The majority of works have expanded either Generative Adversarial Networks (GANs) [48] or Variational Auto-Encoders (VAEs) [79] in this context, by augmenting GANs and VAEs with the capability of generating data samples based on class labels. Conditional generation has been beneficial in domain transfer, super-resolution imaging, video to video translation, as well as image and face editing [65, 214, 109, 69, 87, 164, 26, 72, 175, 176, 197, 166]. Most recently, a number of new techniques has been proposed to stabilize the training process of conditional GANs (cGANs) and improve the visual quality of generated images [110, 13]. Our proposed ImaGINator is a cGAN architecture, aiming at generating facial expressions / human actions from single images, where a category label is provided in both G and D.

Unsupervised video prediction based on multiple frames involves the use of multiple frames as input and the prediction of future frames by learning to extrapolate. Video prediction has been predominantly focused on predicting high-level semantics in video, such as action [129, 81, 41, 105, 155, 182, 34, 33], event [198, 59, 124], semantic segmentation [101], as well as motion [121, 160, 159, 91]. In contrast to such works, our model is targeted to generate a video sequence based on a *single frame*. Since future motion is very uncertain under this setting, we leverage action label as a guidance.

Video generation based on a single image is challenging and hence current methods have proposed to decompose it into sub-tasks. One line of scientific works has utilized in this additional context-information, e.g., human key points [67, 188, 161], 3D face mesh [204] and optical flow [89], as future motion guidance. This additional information is either pre-computed throughout the generated video [67, 204] or predicted based on an initial input

[188, 161]. The additional information guides a conditional image translation, which though results in lack of modeling spatio-temporal correlations.

Deviating from the above, several previous work [148, 157, 187] attempted to hallucinate future frames directly in the pixel space. The latter proposed a probabilistic model, predicting dynamic filters on the input image to render next frame, leading to prediction of only one future frame. MoCoGAN is based on a *seq2seq* [143] architecture, aiming at separating spatio-temporal generation into two steps (disentangling each video frame into motion and appearance in different latent spaces). However, such two-step generation omits the modeling of temporal consistency in higher spatial levels, which generally fails to retain original appearance. VGAN employs a single step method towards modeling multi-level spatio-temporal consistency through 3D convolution by decomposing videos into foreground and background. Although it models both, low and high level features, due to lack of frame quality constrains, generated results are of inherently lower visual quality, i.e., are blurry.

In this chapter, we present a single step architecture, which decomposes motion and appearance in multi-level feature spaces for image to video generation.

3.3 ImaGINator

Our goal is to generate a video sequence, given an appearance information (as a single image frame) and a motion-category label (e.g. determining the facial expression and humna action). We here assume that a video y can be decomposed into appearance c_a (originating from the input image) and motion c_m (originating from the label), based on which we proceed to generate videos. Hence, we formulate our task as learning a conditional mapping $G\colon \{z, c_a, c_m\} \rightarrow y$, where $z \sim \mathcal{N}(0, 1)$ denotes the noise vectors.

Towards achieving our goal, we present a framework that consists of the following 3 main components: (i) *Generator* G, that accepts c_a, c_m and noise as inputs, and seeks to generate realistic video sequences, (ii) *image Discriminator* D_I that determines the frame-level based appearance quality, and (iii) *video Discriminator* D_V, which additionally discriminates, whether the generated video sequences contain authentic motion. In the following we proceed

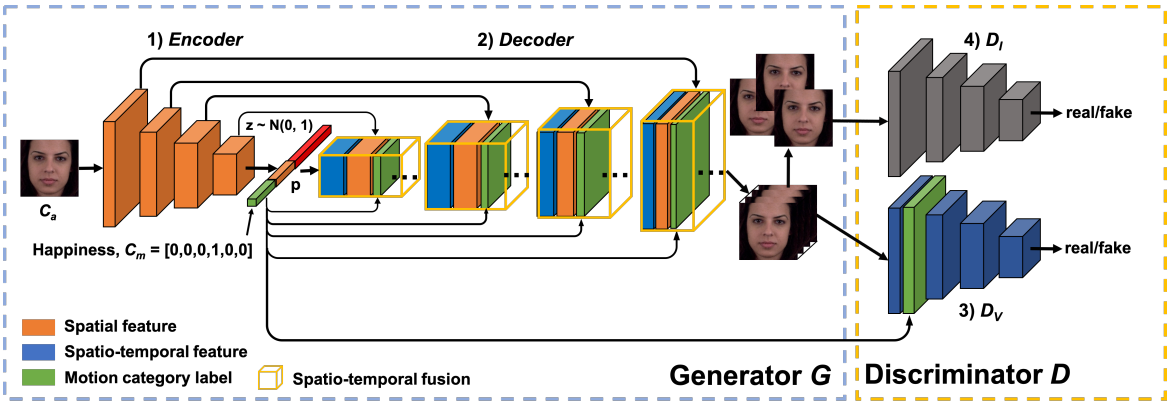


Figure 3.2 **Overview of the proposed ImaGINator**. In the *Generator G*, the *Encoder* firstly encodes an input image c_a into a single vector p. Then, the *Decoder* produces a video based on a motion c_m and a random vector z. By using spatio-temporal fusion, low level spatial feature maps from the *Encoder* are directly concatenated into the *Decoder*. While D_I discriminates whether the generated images contain an authentic appearance, D_V additionally determines whether the generated videos contain an authentic motion.

to describe the architecture of our video prediction network, providing details on G, D_I and D_V, as illustrated in Fig. 3.2. In addition, we elaborate on the proposed spatio-temporal fusion scheme, as well as the transposed (1+2)D convolution.

3.3.1 Generator

Our Generator G consists of an image *Encoder* and a video *Decoder* (see Fig. 3.2). The *Encoder* extracts appearance information in various layers, from shallow, fine layers to deep, coarse layers. It encodes the input image c_a into a latent vector p, and then by concatenating p, c_m as well as the noise vector z, the decoder generates a video sequence.

In our Generator G, we extend the idea of using 2 skip connections from the FCN-8 [99] to 4 skip connections, but with the difference that the original skip connections are applied to fuse predictions, whereas ours are applied to fuse appearance and motion spatio-temporal features. Our skip connections allow the Decoder to access low-level features directly from the Encoder, enabling the Decoder to reuse the appearance features at each time slice and to focus on generating motion.

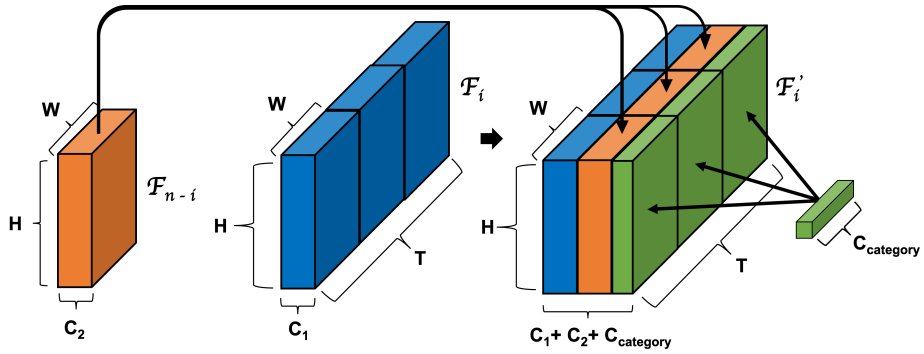


Figure 3.3 **Spatio-temporal fusion.** Blue and orange cuboids represent the intermediate feature maps in the Decoder and Encoder respectively. Our proposed fusion scheme enforces the Decoder reutilizing spatial information through skip connections. Based on such operations, temporal consistency can be modeled in multi-levels.

Spatio-temporal fusion. Let G have n layers and let $F_i \in \mathbb{R}^{H \times W \times C_1 \times T}$ be the feature map from the ith layer with C_1 number of channels in G, $f_{i,t} \in \mathbb{R}^{H \times W \times C_1}$, $t \in \{1, ..., T\}$ be the tth feature map in F_i and $F_{n-i} \in \mathbb{R}^{H \times W \times C_2}$ represent the feature map from $(n-i)^{th}$ layer, see Fig. 3.3. We design the outputs of each respective layer from our Decoder and Encoder to have the same spatial dimensions $H \times W$. We propose a fusion mechanism, concatenating each $f_{i,t}$ with F_{n-i} in a *channel-wise dimension* with a result of a new feature map $F_i^{'} \in \mathbb{R}^{H \times W \times (C_1 + C_2) \times T}$, named **spatio-temporal fusion**. Here we note that each initial feature map F_i represents spatio-temporal features of several consecutive frames in the generated video. By spatio-temporally fusing F_i and F_{n-i} directly in *different feature levels*, the input information can be well preserved in the generated video.

Further, we fuse the label (constituting a one-hot vector) directly into the Decoder, in order to provide each layer an access to the label. To do so, we firstly project the label onto one-hot feature map. Then, we spatio-temporally fuse the category label information into different layers in the Decoder. Our final feature map F_i is of size $H \times W \times (C_1 + C_2 + C_{category}) \times T$.

We note that 3D convolution, utilized in one step methods often brings to the fore generation of blurry videos, due to hard optimization. Nevertheless, benefiting from spatial and temporal decomposition, frames can be generated individually in a two step method. Hence, towards incorporating such decomposition in a one step method, we design a new convolution layer, integrating transposed (1+2)D convolution.

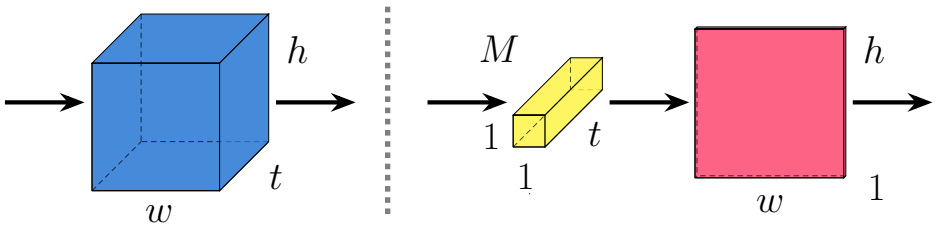


Figure 3.4 **Transposed 3D convolution** (on the left) v.s. **proposed Transposed (1+2)D convolution** (on the right). The transposed 3D convolutional filter of size $t \times w \times h$ has been decomposed into M transposed 1D temporal convolution filters $t \times 1 \times 1$ and a transposed 2D spatial convolution $1 \times w \times h$. The operation M denotes the number of 1D filters, t indicates the temporal size, and w and h indicate the spatial size.

Transposed (1+2)D Convolution. We propose to explicitly factorize transposed 3D convolutional filters into two separate and successive operations, M transposed 1D temporal convolutional filters followed by a 2D separate spatial components, which we refer to as transposed (1+2)D convolution, shown in Fig. 3.4. Such decomposition brings to the fore several benefits. The first benefit relates to an additional nonlinear rectification between these two operations, thus allowing the model to represent more complex functions. The second potential benefit is that the decomposition facilitates optimization, as transposed (1+2)D convolution blocks, with factorized temporal and spatial components, are easier to optimize than the full transposed 3D convolutional filters. Moreover, we show that factorizing the transposed 3D convolutional filters yields significant gains in both, video quality and speed, see Section 5.3. We note that proposed transposed (1+2)D convolution is inspired by decomposition of 3D convolutional filters [147].

3.3.2 Two-stream Discriminator

Towards improving image quality in video generation, we here design a two-stream *Discriminator* architecture, containing D_V, as well as D_I. While D_V has five 3D convolution layers, D_I contains only 2D convolutions with the same layer numbers of D_V. D_V accepts the full generated video as input, using proposed spatio-temporal fusion to fuse the 'one-hot feature map' of the category label and the output of the first layer, similarly like in G. D_V seeks to measure the KL divergence between the joint distributions $p(x_{real}, c_m)$ and

$p(x_{fake}, c_m)$. We randomly sample N frames out of real and generated video respectively as input.

3.3.3 Learning

We define our full objective function as

$$\begin{aligned} G^* &= \arg\min_{G} \max_{D_I, D_V} \mathscr{L}(G, D_I, D_V) \\ \mathscr{L}_{\mathscr{F}}(G, D_I, D_V) &= \mathscr{L}_{GAN}(G, D_I, D_V) + \lambda \mathscr{L}_{rec}(G), \end{aligned} \tag{3.1}$$

which contains two types of terms: an adversarial loss $\mathscr{L}_{GAN}$ for matching the distribution of generated images to the data distribution in the target domain, and a reconstruction loss $\mathscr{L}_{rec}$ for capturing the overall structure and coherence of a video. Due to the high dimensional video space, we introduce the λ parameter, which controls the relative importance of the objectives and stabilizes the training and balancing between losses.

Adversarial Losses. We apply adversarial losses to our mapping function G and its image Discriminator D_I and video Discriminator D_V. We express the objective as

$$\mathscr{L}_{GAN}(G, D_I, D_V) = \mathscr{L}_I(G, D_I) + \mathscr{L}_V(G, D_V), \tag{3.2}$$

where G attempts to generate videos $G(z, c_a, c_m)$, while D_I and D_V aim to distinguish between translated samples and real samples. G seeks to minimize this objective against adversaries D_I and D_V, which attempt to maximize it, i.e. $\min_G \max_{D_I, D_V} \mathscr{L}_{GAN}(G, D_I, D_V)$. The loss $\mathscr{L}_I$ and the loss $\mathscr{L}_V$ are defined as follows

$$\begin{aligned} \mathscr{L}_I &= \mathbb{E}_{x' \sim p_{data}}[log(D_I(x'))] + \mathbb{E}_{z \sim p_z(z), c_a, c_m}[1 - log(D_I(G(z, c_a, c_m)'))], \\ \mathscr{L}_V &= \mathbb{E}_{x \sim p_{data}, c_m}[log(D_V(x, c_m))] + \mathbb{E}_{z \sim p_z(z), c_a, c_m}[1 - log(D_V(G(z, c_a, c_m), c_m))]. \end{aligned} \tag{3.3}$$

$\mathscr{L}_I$ denotes the loss function related to D_I, $\mathscr{L}_V$ represents the loss function related to D_V, and $(\cdot)'$ characterizes N frames sampled from real and generated videos. Both losses,

encompassed in D_I and D_V, are based on the Cross-Entropy loss.

Reconstruction Loss. We define our video-level reconstruction loss as

$$\mathscr{L}_{rec} = \mathbb{E}[||x_{real} - G(z, c_a, c_m)||_1], \tag{3.4}$$

the reconstruction loss is aimed at capturing the overall structure and coherence of a video. It uses $\mathscr{L}_1$ loss in order to generate sharp videos. By combining it with $\mathscr{L}_{GAN}$, it fosters G to create more realistic videos and to reconstruct the original real ones at the same time.

3.3.4 Architecture details

Generator. We illustrate the architecture of generator in Fig. 3.5. It consists of two parts, (a) an image Encoder, containing five 2D convolutional layers (*Conv1 - Conv5*) and (b) a video Decoder with five transposed (1+2)D convolutions (*Deconv 6-1 - Deconv10-2*). Each transposed (1+2)D convolution has two separate and successive operations, M transposed 1D temporal convolutional filters followed by a transposed 2D spatial convolution. In all layers of the Generator, we use the Batch Normalization [63], followed by the *LeakyReLU* after each convolution and transposed convolution, except for the last layer, where we directly use the *Tanh* activation function after the transposed convolution.

Towards generating a video, the Encoder firstly encodes an input image of size $64 \times 64 \times 3$ into a latent vector of size 100, proceeds to combine it with a noise vector of size 512, as well as with a one-hot category vector towards formulating a representation of video in a latent space. Then, the Decoder generates a video based on this representation. Each transposed 1D convolutional layer (except *Deconv6-1*) in the Decoder merges three different types of feature maps as input through *spatio-temporal fusion*, (i) a motion map from its last 2D layer, (ii) an appearance map from the corresponding layer in the Encoder through skip connections, as well as (iii) a one-hot category map replicated from the one-hot category vector. All feature maps share the same spatial size. In particular, we capture the feature maps from

layers *Conv1*, *Conv2*, *Conv3*, *Conv4*, in order to fuse with the outputs from layers *Deconv9-2*, *Deconv8-2*, *Deconv7-2* and *Deconv6-2*, respectively. Details of the Generator are exhibited in Table 3.1.

Layers	**Type**	**KN**	**KS**	**S**	**P**
Conv1	Conv2D	64	4x4	2x2	1x1
Conv2	Conv2D	128	4x4	2x2	1x1
Conv3	Conv2D	256	4x4	2x2	1x1
Conv4	Conv2D	512	4x4	2x2	1x1
Conv5	Conv2D	100	4x4	1x1	No
Deconv6-1	TransConv1D	4096	2x1x1	1x1x1	No
Deconv6-2	TransConv2D	512	1x4x4	1x1x1	No
Deconv7-1	TransConv1D	3072	4x1x1	2x1x1	1x0x0
Deconv7-2	TransConv2D	256	1x4x4	1x2x2	0x1x1
Deconv8-1	TransConv1D	1536	4x1x1	2x1x1	1x0x0
Deconv8-2	TransConv2D	128	1x4x4	1x2x2	0x1x1
Deconv9-1	TransConv1D	768	4x1x1	2x1x1	1x0x0
Deconv9-2	TransConv2D	64	1x4x4	1x2x2	0x1x1
Deconv10-1	TransConv1D	36	4x1x1	2x1x1	1x0x0
Deconv10-2	TransConv2D	3	1x4x4	1x2x2	0x1x1

Table 3.1 **Network architecture of the Generator.** Our Generator incorporates an image Encoder (*Conv1* - *Conv5*), as well as a video Decoder (*Deconv6-1* - *Deconv10-2*). KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size.

Discriminator. Our ImaGINator includes two Discriminators, an *image* Discriminator D_I, as well as a *video* Discriminator D_V. The input of D_I entails N randomly sampled frames, either from real or generated videos. In our experiments, we set $N = 16$. D_I provides as output a scalar value, indicating whether the frames are real or fake. D_I is represented by a network of five 2D convolutional layers. The kernel size in all layers is 4×4, see Fig. 3.6.

D_V discriminates videos based on the related realistic appearance and motion. It is represented by a network containing five 3D convolutional layers, see Fig. 3.7. While $4 \times 4 \times 4$ kernels have been applied in the first four layers, one $2 \times 4 \times 4$ kernel is featured in the last layer ($T \times H \times W$ denotes time step, height and width of a kernel respectively). A one-hot category vector is replicated into a one-hot category map of the same spatial size of

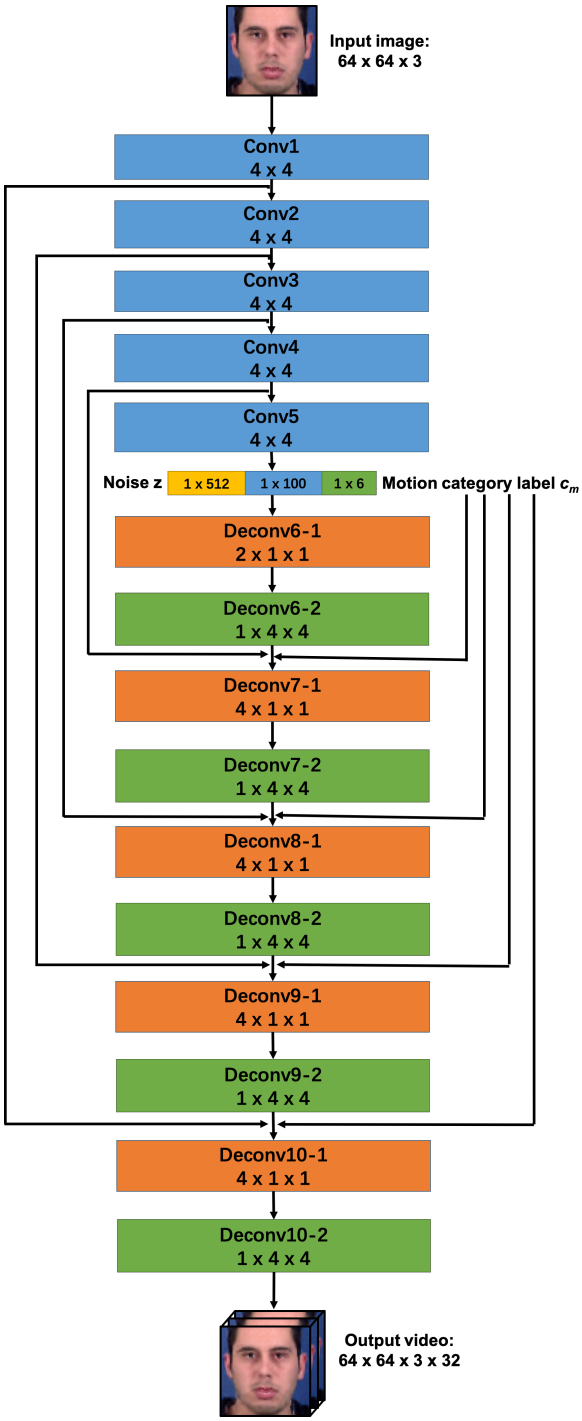


Figure 3.5 **Network architecture of the Generator.** Our Generator G accepts an image of size $64 \times 64 \times 3$ as input and generates a 32-frame long video. G incorporates an image Encoder (*Conv1* - *Conv5*) and a video Decoder (*Deconv6-1* - *Deconv10-2*). Skip connections link Encoder and Decoder, with the goal of enforcing the Decoder to reuse appearance features directly. A motion category vector is replicated into feature maps and concatenated with each feature map in the Decoder (for different dataset, length of motion category vector is different, here we use 6 to represent MUG dataset).

the output feature map of *Conv1*. Then, *Conv2* takes the concatenation of both feature maps as input.

In all layers in both Discriminators, we use the Spectral Normalization (SN) [110], followed by the *LeakyReLU* after each convolution, except for the the last layer, where we use *Sigmoid* activation function after the normalization. Details pertained to the network architecture of the Discriminators are presented in Table 3.2 (image Discriminator) and Table 3.3 (video Discriminator), respectively.

Layers	**Type**	**KN**	**KS**	**S**	**P**
Conv1	Conv2D	64	4x4	2x2	1x1
Conv2	Conv2D	128	4x4	2x2	1x1
Conv3	Conv2D	256	4x4	2x2	1x1
Conv4	Conv2D	512	4x4	2x2	1x1
Conv5	Conv2D	1	4x4	1x1	No

Table 3.2 **Network architecture of the image Discriminator.** KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size.

Layers	**Type**	**KN**	**KS**	**S**	**P**
Conv1	Conv3D	64	4x4x4	2x2x2	1x1x1
Conv2	Conv3D	128	4x4x4	2x2x2	1x1x1
Conv3	Conv3D	256	4x4x4	2x2x2	1x1x1
Conv4	Conv3D	512	4x4x4	2x2x2	1x1x1
Conv5	Conv3D	1	2x4x4	1x1x1	No

Table 3.3 **Network architecture of the video Discriminator.** KN = Kernel Numbers, KS = Kernel Size, S = Stride, P = Padding size.

3.3.5 Implementation details and training strategy

Our method is implemented using PyTorch. We train the entire network end-to-end with the standard back-propagation algorithm on a single GTX 1080Ti GPU. We employ ADAM optimizer [78] with $\beta = 0.5$. Moreover, we apply spectral normalization on both D_I and D_V to stabilize training, as proposed by Miyoto *et al.* [110]. We observe that given the same learning rate for D_I, D_V and G during training, D_I and D_V typically learn faster than G. The reason for this might be that the spatio-temporal convolution is more efficient at

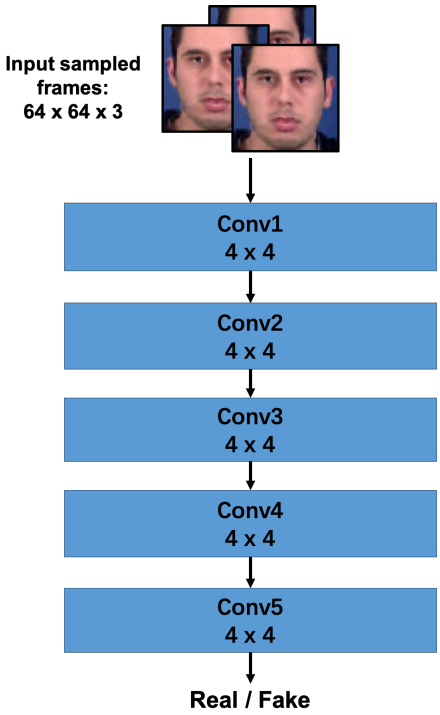


Figure 3.6 **Network architecture of the image Discriminator,** containing five 2D convolutional layers of kernel size 4×4.

differentiating than at generating, as pointed out by Goodfellow *et al.* [48] and Radford *et al.* [122]. In order to circumvent this disparity, we set the learning rate to $2e^{-4}$ for G, and $5e^{-5}$ for both D_I and D_V. λ is set $1e^{-3}$ to balance two types of losses.

To train the network, we firstly provide an input frame, as well as corresponding category label to G to generate possible videos. Then D_V and D_I distinguish between real and fake videos and frames based on the respective quality. Specifically, when training D_V, we provide two types of negative samples, generated videos with correct labels $(x_{real}, c_{correct})$ and real videos with wrong labels (x_{real}, c_{wrong}). We observe that such training enforces D_V to learn from diverse samples and at the same time enables the generation of realistic samples. We provide details in Algorithm 1.

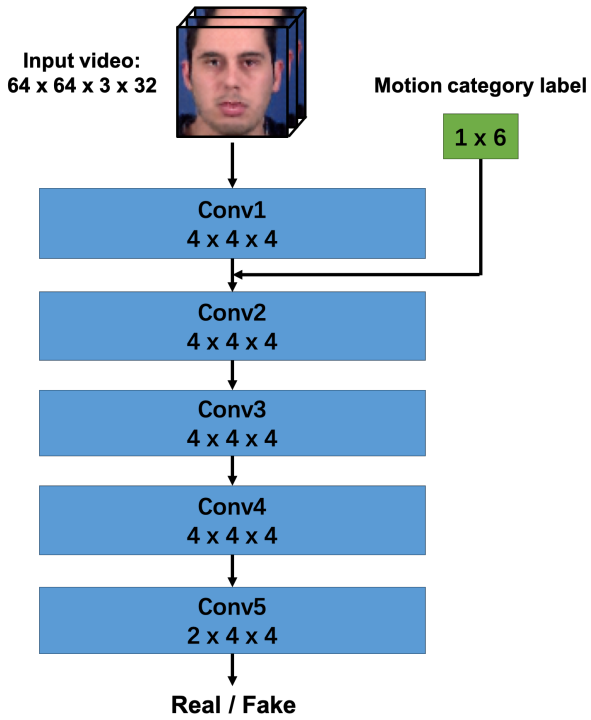


Figure 3.7 **Network architecture of the video Discriminator,** including five 3D convolutional layers, a motion category vector is firstly replicated and then concatenated with the feature map of the first layer (for different dataset, length of motion category vector is different, here we use 6 to represent MUG dataset).

Algorithm 1 ImaGINator Training Algorithm

Input: minibatch x, $x^{'}$, input image c_a, correct c_m, wrong $\hat{c_m}$

for each step **do**

$z \sim \mathcal{N}(0, I)$

$x_{recon} \leftarrow G(z, c_a, c_m)$

$s_{real} \leftarrow D_V(x, c_m) + D_I(x^{'})$

$s_{recon} \leftarrow D_V(x_{recon}, c_m) + D_I(x^{'}_{recon})$

$s_w \leftarrow D_V(x, \hat{c_m}) + D_I(x^{'})$

$\mathcal{L}_D \leftarrow \log(s_r) + 0.5[\log(1 - s_w) + \log(1 - s_{recon})]$

$D_V \leftarrow D_V - \alpha \partial \mathcal{L}_D / \partial D_V$

$D_I \leftarrow D_I - \alpha \partial \mathcal{L}_D / \partial D_I$

$\mathcal{L}_{recon} \leftarrow ||x - x_{recon}||_1$

$\mathcal{L}_G \leftarrow \log(s_{recon}) + \lambda \mathcal{L}_{recon}$

$G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$

end for

3.4 Experiments

This section presents the evaluation of ImaGINator. We first describe datasets and evaluation metrics used in this work. We then present quantitative and qualitative comparison with other methods w.r.t. video quality. Finally, we showcase an ablation study to demonstrate the effectiveness of proposed architecture and loss function.

3.4.1 Datasets

MUG Facial Expression dataset [2] contains 931 videos of 52 subjects (data of 42 subjects is employed for training and 10 for testing), performing 7 facial expressions, namely "happy", "sad", "surprise", "anger", "disgust", "fear" and "neutral".

NATOPS Aircraft Handling Signals dataset [140] contains video sequences of 20 subjects (data of 15 subjects is employed for training and 5 for testing), performing 24 gestures including "all clear" and "move ahead". Each subject repeats each gesture 20 times.

Weizmann Action dataset [49] contains 90 videos of 9 subjects (data of 6 subjects is employed for training and 3 for testing), performing 10 actions, e.g.,"wave" and "bend". We augment this dataset by doubling the number of videos using horizontal flipping transformation.

UvA-NEMO Smile dataset [35] contains 597 video sequences of smiling individuals. It contains 400 subjects (data of 320 subjects is employed training and 80 for testing) with 1 or 2 videos per subject. In the context of UvA-NEMO we do not provide any category to our model, since the dataset features only one facial expression.

In all our experiments, frames are scaled to 64×64 pixels. We use a time step 2 to sample frames from facial expression datasets and a time step of 1 from human action datasets. MUG and UvA-NEMO are pre-processed by detecting faces in OpenFace [4] and cropping them in each frame.

3.4.2 Evaluation Metrics

The Video Fréchet Inception Distance (**FID**) [164] is a video generation metric. It measures both visual quality and temporal consistency of generated videos. We use 3D ResNeXt-101 as a feature extractor and calculate Video FID as: $\|\mu - \widetilde{\mu}\|^2 + Tr(\Sigma + \widetilde{\Sigma} - 2\sqrt{\Sigma\widetilde{\Sigma}})$, where μ and Σ are mean and covariance matrix computed from real feature vectors, and $\widetilde{\mu}$, and $\widetilde{\Sigma}$ are computed from generated data. Lower Video FID scores represent a superior quality of generated videos.

The Structural Similarity Index Measure (**SSIM**) indicates the structure similarity between real and reconstruction images, Peak Signal-to-Noise Ratio (**PSNR**) quantifies the image quality. High SSIM and PSNR scores indicate higher quality of generated images.

The Average Content Distance (**ACD-C**) [148] measures content consistency of a generated video. For facial expression videos, we first use OpenFace [4], which outperforms human performance in face recognition, to extract a feature vector pertaining to the detected face. Then, we compute the ACD-C as an average $\mathscr{L}_2$ pairwise distance for a per-frame vector in a video. Smaller values indicate similar faces in consecutive frames of a generated video. However, the original ACD-C only signifies the face-identity-consistency between each pair of frames, lacking the information on general identity preservation. Therefore, we also use the **ACD-I** measure [205], the extension corresponding to the average of all $\mathscr{L}_2$ pairwise distances between each generated frame and the respective input frame.

3.4.3 Video quality evaluation

We proceed to compare our proposed ImaGINator to state-of-the-art video generation methods MoCoGAN and VGAN, both quantitatively and qualitatively. For the latter we report results pertained to a subjective analysis comparing the three methods.

Quantitative Analysis. For all methods, we sample 10 initial frames from each video sequence in each testing set. Both benchmark methods have been tuned with the best parameters on all training sets. All methods are trained to generate video sequences of 32

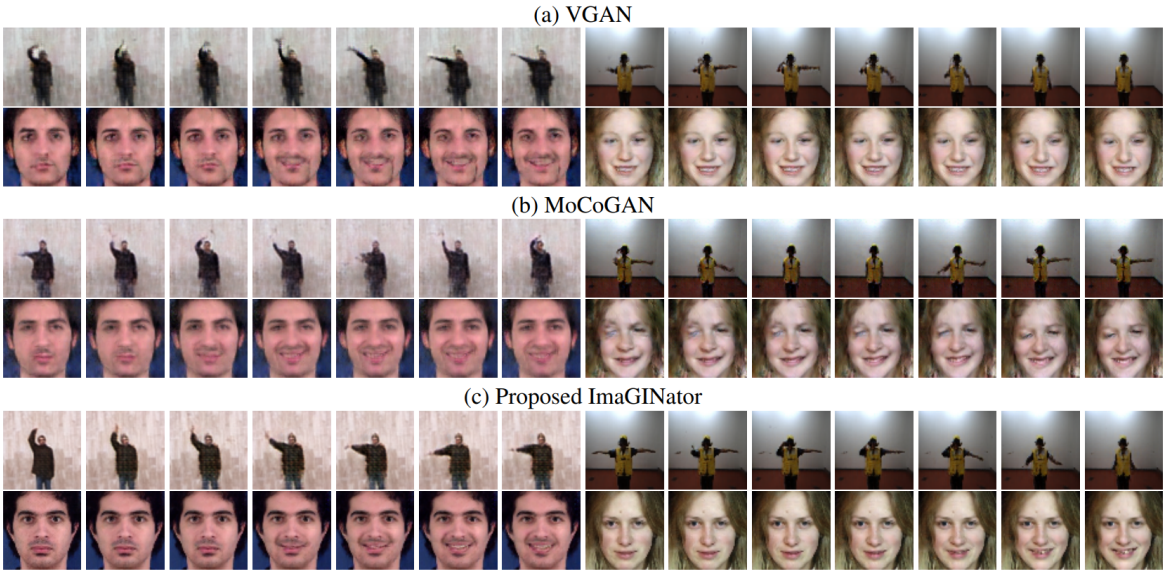


Figure 3.8 **Example generated video frames** pertained to algorithms (a) VGAN, (b) MoCoGAN, as well as the (c) proposed ImaGINator. For each method, we present generated video frames for the four datasets: **Weizmann** (top-left), label *"Wave"*; **NATOPS** (top-right), label *"Hot Brakes"*; **MUG** (bottom-left), label *"Happiness"*; **UvA-NEMO** (Down-right), no label. All frames are sampled with a time step of 3.

frames with an image size 64×64 pixels. Example generated frames of different methods are shown in Fig. 3.8.

We firstly report reconstruction capabilities of our approach using SSIM and PSNR scores in Table 3.4. Our results show that the ImaGINator outperforms MoCoGAN and VGAN, w.r.t. SSIM and PSNR metrics, indicating that our proposed spatio-temporal fusion mechanism can well preserve the structure information of input image in the full generated video.

Then, we report FID scores for the three methods in Table 3.4. Our proposed ImaGINator achieves the lowest numbers on all four datasets, suggesting that videos generated by our method have the best temporal consistency and visual quality. We also show generates samples in Fig. 3.12 (Uva-NEMO), Fig. 3.13 (MUG), Fig. 3.14 (NATOPS) and Fig. 3.15 (Weizmann). This proves that modeling temporal consistency in higher spatial level can generate more realistic videos.

	MUG		NATOPS		Weizmann		UvA-NEMO	
	SSIM/PSNR	FID	SSIM/PSNR	FID	SSIM/PSNR	FID	SSIM/PSNR	FID
VGAN [157]	0.28/14.54	74.72	0.72/20.09	167.71	0.29/15.78	127.31	0.21/13.43	30.01
MoCoGAN [148]	0.58/18.16	45.46	0.74/21.82	49.46	0.42/17.58	116.08	0.45/16.58	29.81
ImaGINator	**0.75/22.63**	**29.02**	**0.88/27.39**	**26.86**	**0.73/19.67**	**99.80**	**0.66/20.04**	**16.16**

Table 3.4 **Evaluation of video quality.** We compare VGAN, MoCoGAN and proposed ImaGINator w.r.t. image quality (SSIM/PSNR) and video quality (FID).

Then, we evaluate the content consistency for facial expression generation using ACD-C and ACD-I scores. Our results on the MUG dataset are presented in Table 3.5. The proposed ImaGINator outperforms both MoCoGAN and VGAN, on both ACD-C and ACD-I scores. The results confirm the ability of the proposed spatio-temporal fusion scheme to effectively preserve the appearance information in the generated videos.

Methods	ACD-C	ACD-I
VGAN [157]	0.272	0.932
MoCoGAN [148]	0.158	0.904
ImaGINator	**0.131**	**0.431**
Reference	0.102	0.206

Table 3.5 **Evaluation of content consistency** of VGAN, MoCoGAN and proposed ImaGINator on the MUG dataset, represented by ACD-I and ACD-C scores.

Subjective Analysis. In addition, we conduct a subjective analysis, where we ask 30 human raters to pairwise compare videos generated by our approach with those generated by the state-of-the-art. We report the mean user preference in Table 4.2. We observe that human raters express a strong preference for the proposed framework over MoCoGAN (83.32% v.s. 16.68%) and VGAN (85.43% v.s. 14.57%), which is consistent with the above listed quantitative results. Further, we compare real videos from all the datasets with generated video sequences from our method. The human raters ranked 20.82% of videos from our ImaGINator as more realistic than real videos, which we find highly encouraging.

Methods	Rater preference (%)
ImaGINator / MoCoGAN [148]	**83.32** / 16.68
ImaGINator / VGAN [157]	**85.43** / 14.57
MoCoGAN [148] / VGAN [157]	**70.85** / 29.15
ImaGINator / Real videos	20.82 / **79.18**

Table 3.6 **Subjective analysis.** Mean user preference of human raters comparing videos generated by the respective algorithms, as well as originated from all the datasets.

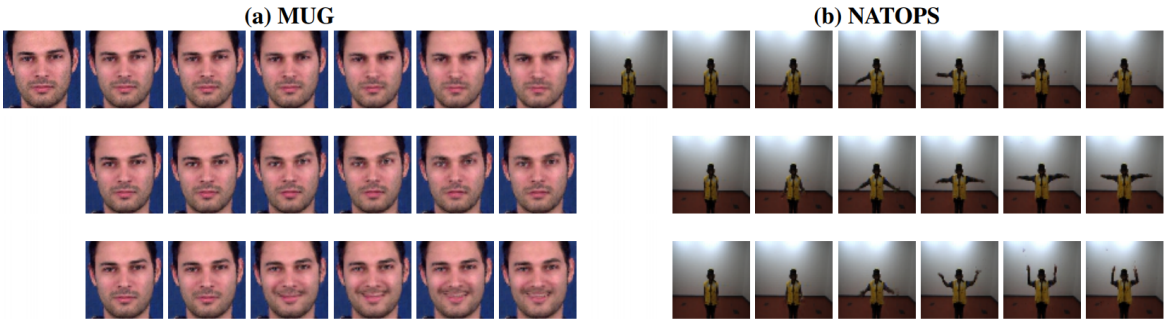


Figure 3.9 **Controllable video generation in ImaGINator.** Starting from the same image (top left for both datasets), we generate videos associated to different labels (remaining frames). In (a) MUG, from top to bottom the labels are set as *"fear"*, *"anger"* and *"happiness"*. In (b) NATOPS, from top to bottom the labels are set as *"all clear"*, *"fold winds"* and *"brakes on"*.

3.4.4 Controllable Video Generation.

We further conduct an experiment on the MUG and NATOPS datasets, where starting from the same image, we generate various videos associated to different labels (facial expressions / actions). Our results are presented in Fig. 3.9. These results confirm the ability of our approach to generate new videos based on single images and category-labels.

3.4.5 Ablation Study

We here focus on showcasing the general effectiveness of our architecture, as well as the effectiveness related to each component of the proposed Generator.

Firstly, in the Generator G, we compare the performance of fully transposed 3D convolution with the proposed transposed (1+2)D convolution, and in the Discriminator D, we mainly focus on analyzing the usage of D_I. In addition, we compare each architecture with

the model of the same architecture, but using an auxiliary classifier in D, similar to ACGAN loss [113], which we refer as $D_V(ac)$. Our results are presented in Table 3.7.

Generator	Discriminator	MUG	NATOPS
3D	$D_V(ac)$	37.71	65.28
(1+2)D	$D_V(ac)$	32.57	52.43
3D	$D_V(ac), D_I$	33.08	57.65
(1+2)D	$D_V(ac), D_I$	29.91	48.41
3D	D_V	36.93	50.08
(1+2)D	D_V	29.80	40.57
3D	D_V, D_I	27.94	42.10
(1+2)D	D_V, D_I	**24.36**	**26.86**

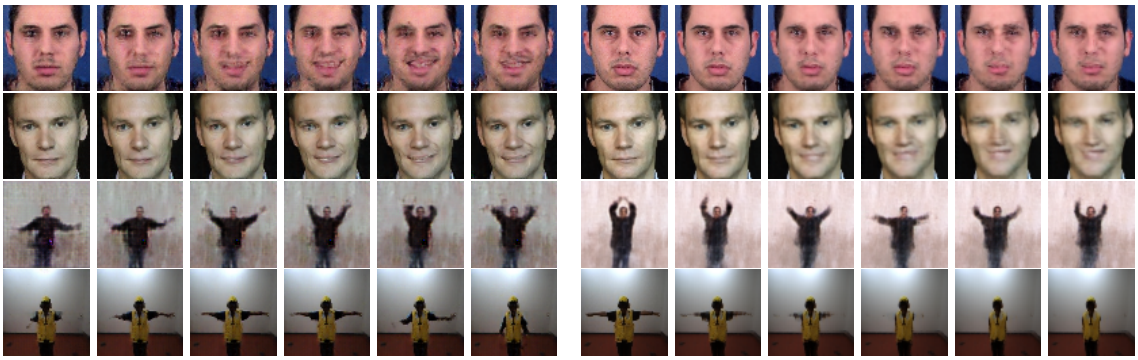
Table 3.7 **Effectiveness of the proposed architecture**. We compare different architectures in both G and D to showcase the effectiveness of the proposed ImaGINator.

Our results show that given the same Discriminator, models using transposed (1+2)D convolution provide consistently lower video FID scores than models using transposed 3D convolution. The results confirm that our proposed transposed (1+2)D layer systematically improves video quality. Moreover, we show that adding D_I is beneficial, as well as that concatenating label vectors directly into spatio-temporal feature maps exceeds the performances of using auxiliary classifier in conditional video generation, see Table 3.7. This is especially true if the number of categories is large. A similar observation has been reported by Miyato and Koyama [111] in the context of conditional image generation.

Furthermore, we showcase that the spatio-temporal fusion contributes predominantly to video quality, see Table 3.8, and hence re-injecting spatial features and modeling temporal consistency in higher spatial level is an effective way to generate realistic videos. Finally, our results confirm that adding noise in the latent space is beneficial, as depicted in Table 3.8.

Architecture	MUG	NATOPS
ImaGINator, w/o ST fusion	46.02	62.89
ImaGINator, w/o (1+2)D	27.94	42.10
ImaGINator, w/o noise	32.38	32.05
ImaGINator	**24.36**	**26.86**

Table 3.8 **Contribution of main components in** G. We evaluate the ablation of spatio-temporal fusion, transposed (1+2)D convolution, as well as noise vector.



(a) Adversarial loss (b) Reconstruction loss

Figure 3.10 **Comparison of use of merely (a) Adversarial loss and (b) Reconstruction loss.** We illustrate generated frames for (a) and (b) on four datasets. We observe that frames in (a) are sharper than (b), but (b) retains overall structures better than (a). Frames are sampled with time step 4.

Towards evaluating the pertinence of reconstruction and adversarial losses in our loss function, we conduct two experiments. While the first experiment integrates merely the *adversarial loss* in ImaGINator, omitting the reconstruction loss; the second experiment merely integrates *reconstruction loss*, omitting the adversarial loss.

As shown in Table 3.9, models only using adversarial loss achieve lower video FID than those only using reconstruction loss. However, results in Table 3.10 and Table 3.11 indicate that the use of reconstruction loss manifests in significantly higher SSIM and PSNR than models only using adversarial loss. We conclude that adversarial loss is instrumental in improving the perceptual quality of videos, as it enforces the Generator to create videos, matching the distribution of the training data. At the same time, the reconstruction loss encourages the Generator to produce frames, resembling the ground truth by reducing the pixel-wise distance, see Fig. 3.10.

In contrast to both single loss experiments, the ImaGINator (using both losses), w.r.t. both evaluation metrics achieves the best results. Hence, both types of losses are complementary and pertinent for the performance of the ImaGINator.

Finally, we compare video quality and training speed of our approach when using (i) transposed 3D convolutional filters only, and (ii) our transposed (1+2)D convolutional filters

	Adv. Loss	Recon. Loss	Two losses
MUG	35.62	45.43	**29.02**
NATOPS	33.97	61.32	**26.86**
Weizmann	150.48	217.58	**99.80**
UvA-NEMO	19.29	30.72	**16.16**

Table 3.9 **Evaluation results for models using different losses** on four datasets represented by video FID. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.)

	Adv. Loss	Recon. Loss	Two Losses
MUG	0.54	0.74	**0.75**
NATOPS	0.87	0.88	**0.88**
Weizmann	0.50	0.54	**0.73**
UvA-NEMO	0.64	0.66	**0.66**

Table 3.10 **Evaluation of frame quality** between generated frames and ground truth on four datasets using SSIM. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.)

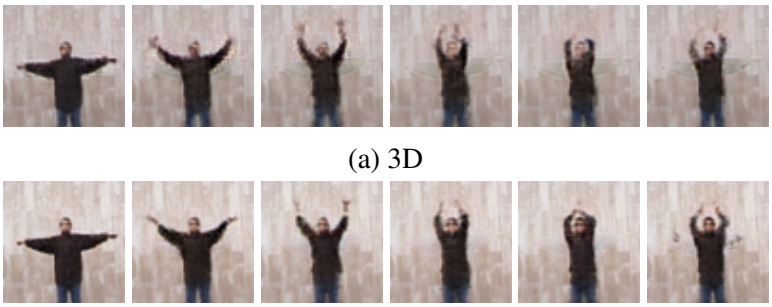
	Adv. Loss	Recon. Loss	Two Losses
MUG	19.24	22.60	**22.63**
NATOPS	26.72	27.10	**27.39**
Weizmann	17.01	18.03	**19.67**
UvA-NEMO	19.87	20.02	**20.04**

Table 3.11 **Evaluation of frame quality** between generated frames and ground truth on four datasets using PSNR. (**Adv. Loss** indicates *adversarial loss*, **Recon. Loss** indicates *Reconstruction Loss* and **Two losses** represents our proposed ImaGINator loss function.)

only, both having the same number of parameters for a fair comparison. The quantitative and qualitative results on the Weizmann dataset are presented in Table 3.12 and Fig. 3.11.

Architecture	FID	Training time
Transposed 3D convolution	110.5	16.7s
Transposed (1+2)D convolution	**99.8**	**11.9s**

Table 3.12 **FID score and training time per epoch** of our approach with transposed 3D and transposed (1+2)D convolutions.



(a) 3D

(b) (1+2)D

Figure 3.11 **Sample generated frames** of ImaGINator with (a) transposed 3D and (b) transposed (1+2)D convolutions.

The results confirm that factorizing the transposed 3D convolutional filters into separate temporal and spatial components brings benefits: (i) an additional nonlinear rectification allows the model to represent more complex functions, (ii) optimization is facilitated, as transposed (1+2)D convolution blocks are easier to optimize than the full transposed 3D convolutional filters, and (iii) significant gains are yielded in both video quality and speed. Therefore, in the following evaluations we use our approach with the transposed (1+2)D convolution filters only.

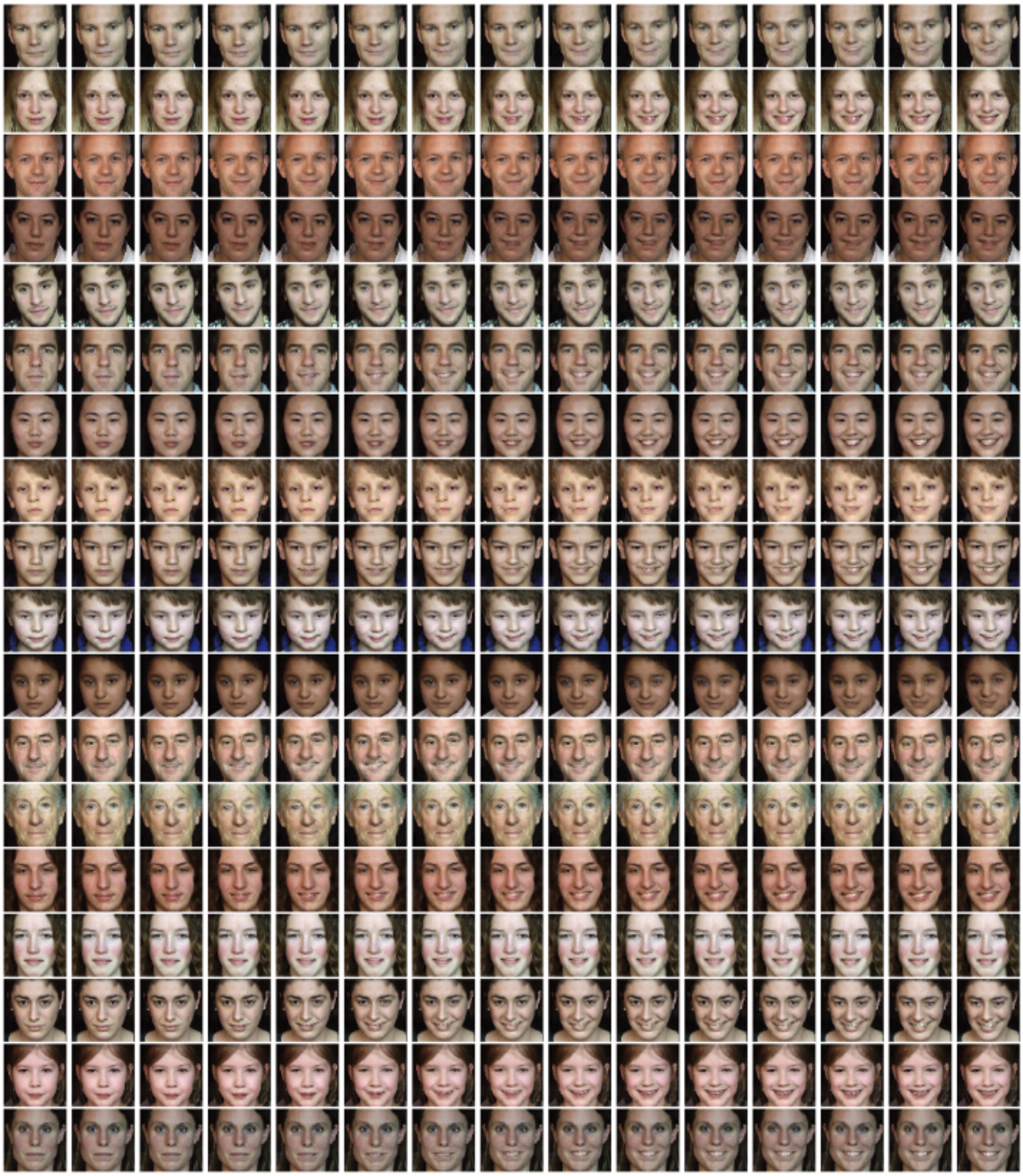


Figure 3.12 **Generated examples from UvA-NEMO.**

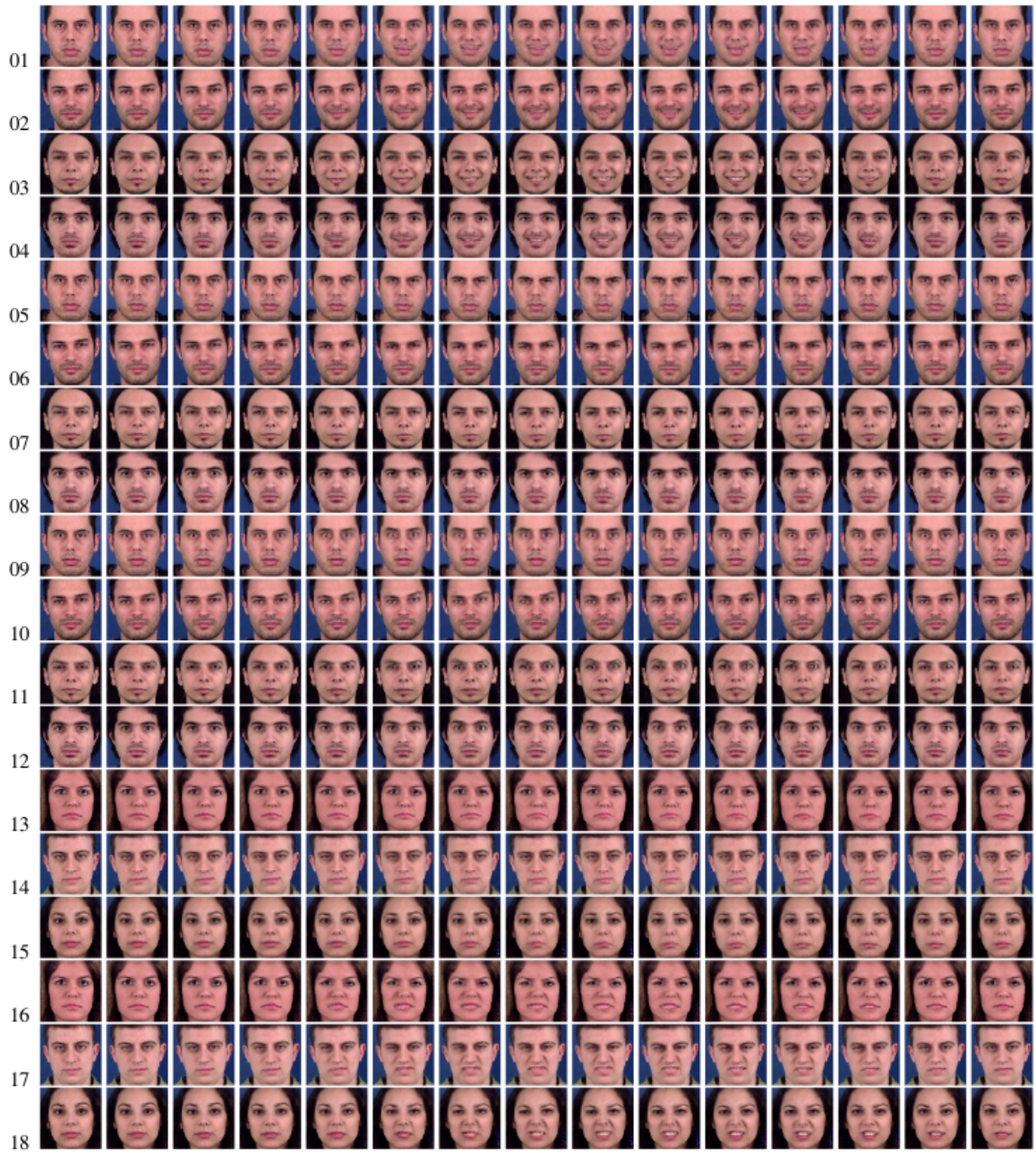


Figure 3.13 **Generated examples from MUG.** Labels are *happiness* (01,02,03,04), *anger* (05,06,07,08), *fear* (09,10,11,12), *sadness* (13,14,15) and *disgust* (16,17,18).

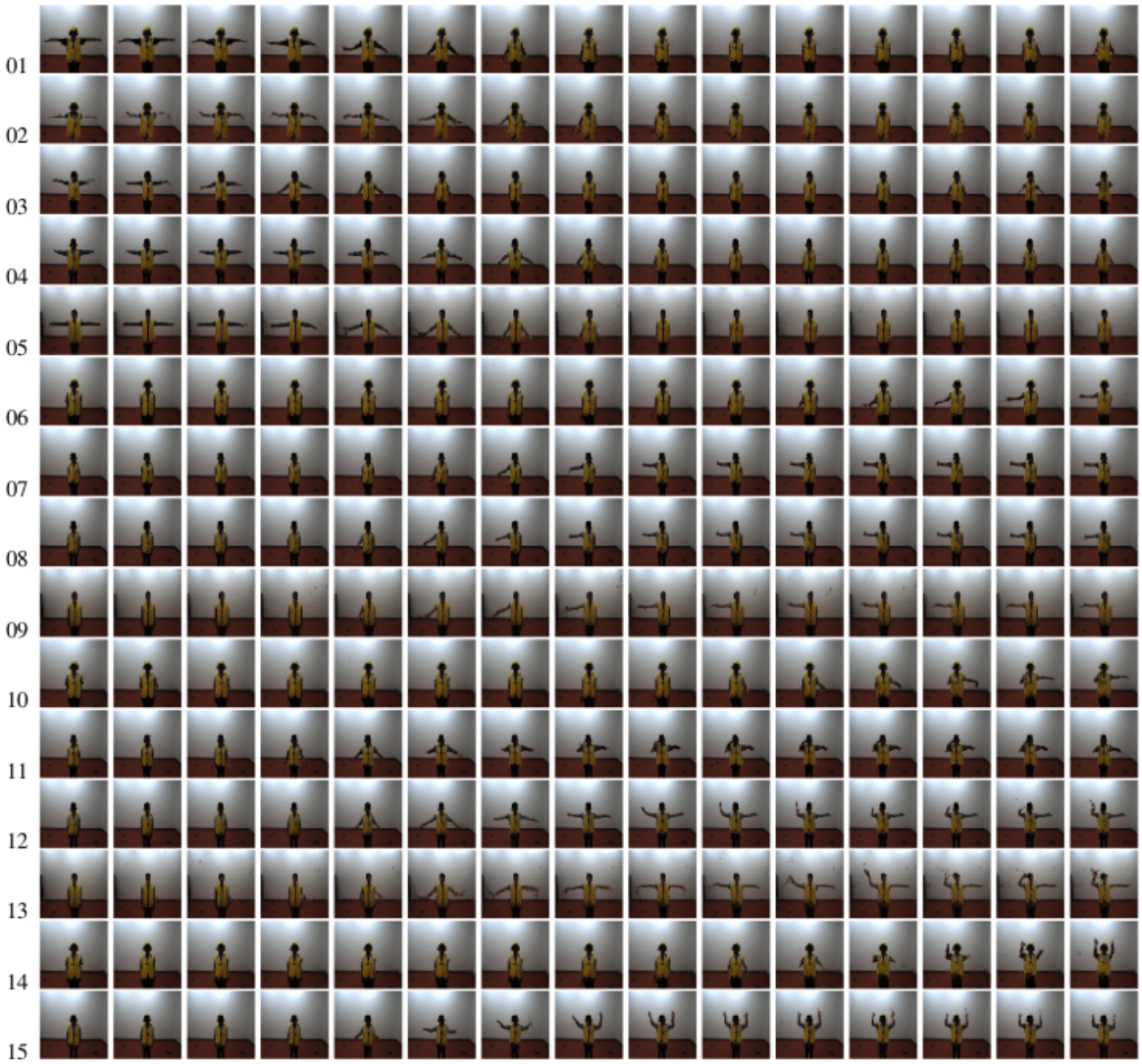


Figure 3.14 **Generated examples from NATOPS.** Labels are *Fold Wings* (01,02,03,04,05), *All Clear* (06,07,08,09), *Nosegear Steering* (10,11), *Turn Right* (12,13) and *Move Ahead* (14,15).

Figure 3.15 **Generated examples from Weizmann.** Labels are *One hand wave* (01,02,05,06), *Two hands wave* (03,04,11,12), *Bend* (07,08,13,14) and *Jack* (09,10).

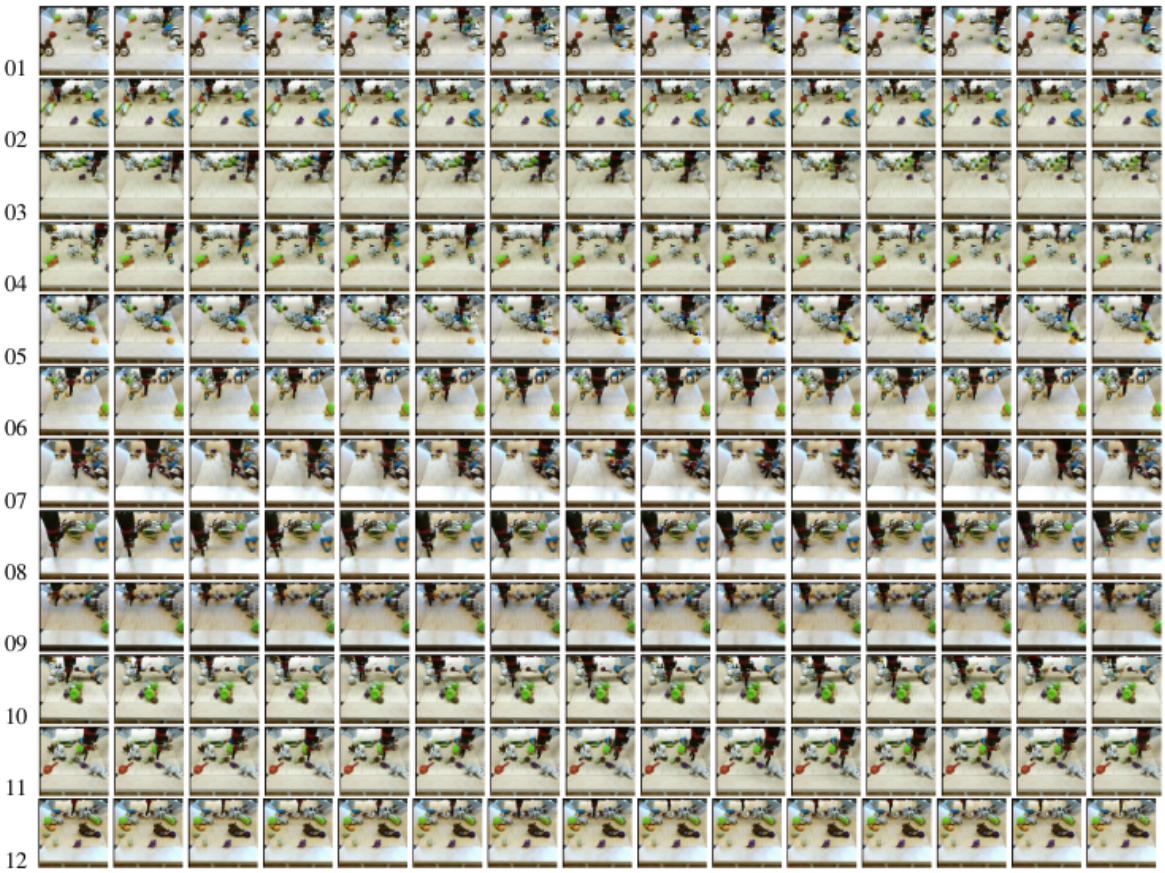


Figure 3.16 **Generated samples from BAIR robot push.**

Chapter 4

Disentangling Appearance and Motion for Video Generation

In the previous chapter, we have introduced a method to generate video from one single image and motion-category label. Deviating from that, in this chapter we focus on the highly intricate problem of video generation without additional input. Specifically, based on noise vectors, we generate an appearance, e.g. human face and body, which we concurrently animate, by a facial expression or human action. We introduce G^3AN, 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.

4.1 Introduction

Generative Adversarial Networks (GANs) [48] have witnessed increasing attention due to their ability to model complex data distributions, which allows them to *generate* realistic *images* [13, 70, 72, 82, 103, 110, 186, 203], as well as to translate images [3, 65, 126, 137]. 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.

G^3AN, 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 [10]. In this context, our G^3AN 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.

In summary, this work makes several contributions. First, we introduce a novel generative model, G^3AN, 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. Second, we propose a novel three-stream generator, which takes into account the learning of individual appearance features (*spatial stream*), motion features (*temporal stream*) and smoothing of generated videos (*main stream*) at the same time. Third, we propose 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. In addition, we conduct extensive qualitative and quantitative evaluation, which demonstrates that G^3AN systematically and significantly outperforms state-of-the-art baselines on a set of datasets.

4.2 Background

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 [159]. Finding suitable representation learning methods, which are able to address these challenges is critical. Existing methods include approaches based on Variational Autoencoders (VAEs) [79], auto-regressive models, as well as most prominently Generative Adversarial Networks (GANs) [48].

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 [116, 164], human keypoints [67, 188, 161, 17], 3D face mesh [204] and optical flow [89] can be instrumental as guidance for appearance and motion generation. This additional information is either pre-computed throughout the generated video [67, 204, 17] or predicted based on an initial input image [188]. 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 [157] or latent representation [130, 148]. VGAN [157] was equipped with a two-stream spatio-temporal Generator, generating foreground and background separately. TGAN [130] 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 Fig. 4.6. The closest work to ours is MoCoGAN [148], 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 G^{3}AN. 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 Fig. 4.9. 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 [10]. Disentangling a number of factors in *still images* has been widely explored in recent works [22, 103, 138, 84]. 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 Fig. 4.7), suggest that the results are not satisfactory.

4.3 G^{3}AN

G^{3}AN aims 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. G^{3}AN consists of a three-stream Generator G, as well as a two-stream Discriminator D, as illustrated in Fig. 5.2. While G aims at generating videos with the ability to modulate appearance and motion

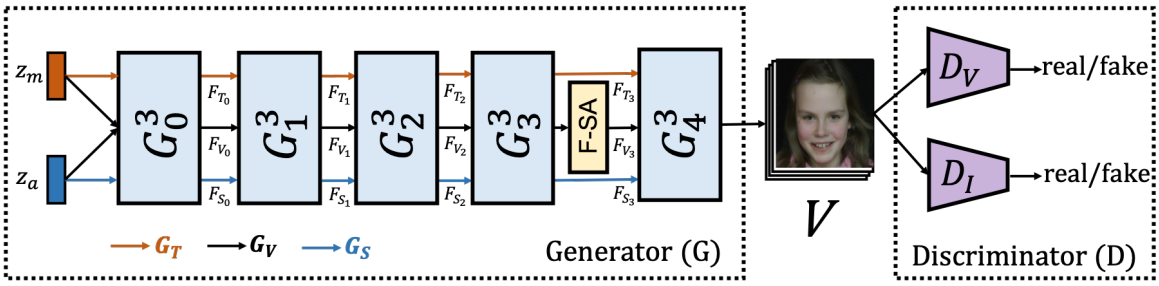


Figure 4.1 **Overview of our G^{3}AN architecture.** G^{3}AN consists of a three-stream Generator and a two-stream Discriminator. The Generator contains five stacked G^3 modules, a factorized self-attention (F-SA) module, and takes as input two random noise vectors, z_a and z_m, aiming at representing appearance and motion, respectively.

disjointly, D accounts for distinguishing generated samples from real data, in both, videos and frames, respectively.

4.3.1 Generator

Hierarchical Generator with G^3-modules. We design G in a hierarchical structure of G^3 modules. Specifically, we have N levels of hierarchy, denoted as $\mathrm{G}^3_{n=0\ldots N-1}$. The first G^3 module, G^3_0 accepts as input the two noise vectors z_a and z_m. The remaining modules $\mathrm{G}^3_{n=1\ldots N-1}$, 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^3_{n-1} module. We illustrate the detailed architecture in Fig. 4.3.

Each G^3_n 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} (see Fig. 4.2). They are designed to generate three different types of features. The spatial stream G_{S_n}, denoted by a blue line in Fig. 5.2 and Fig. 4.2, 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 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}.

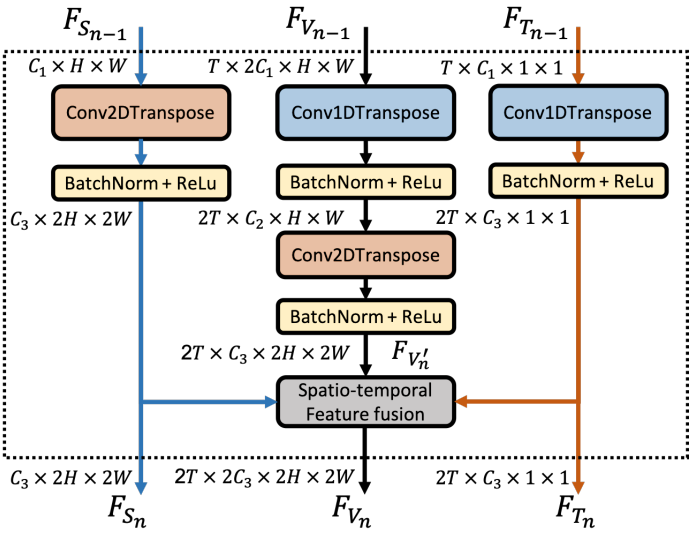


Figure 4.2 **G^3 module architecture.**

Factorized transposed spatio-temporal convolution. 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 5.3.

Spatio-temporal fusion is the key-element to learn well disentangled features, taking 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 Fig. 4.4). Firstly, spatial and temporal replications are applied on F_{T_n} and F_{S_n} respectively, in order to obtain two new feature maps $F^R_{T_n}$ and $F^R_{S_n}$. Both new feature maps have the same spatio-temporal size as $F_{V_n'}$. Next, $F^R_{T_n}$ and $F_{V_n'}$ are combined through a position-wise addition, creating a new spatio-temporal embedding $F_{V_n''}$. Finally, $F^R_{S_n}$ 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.

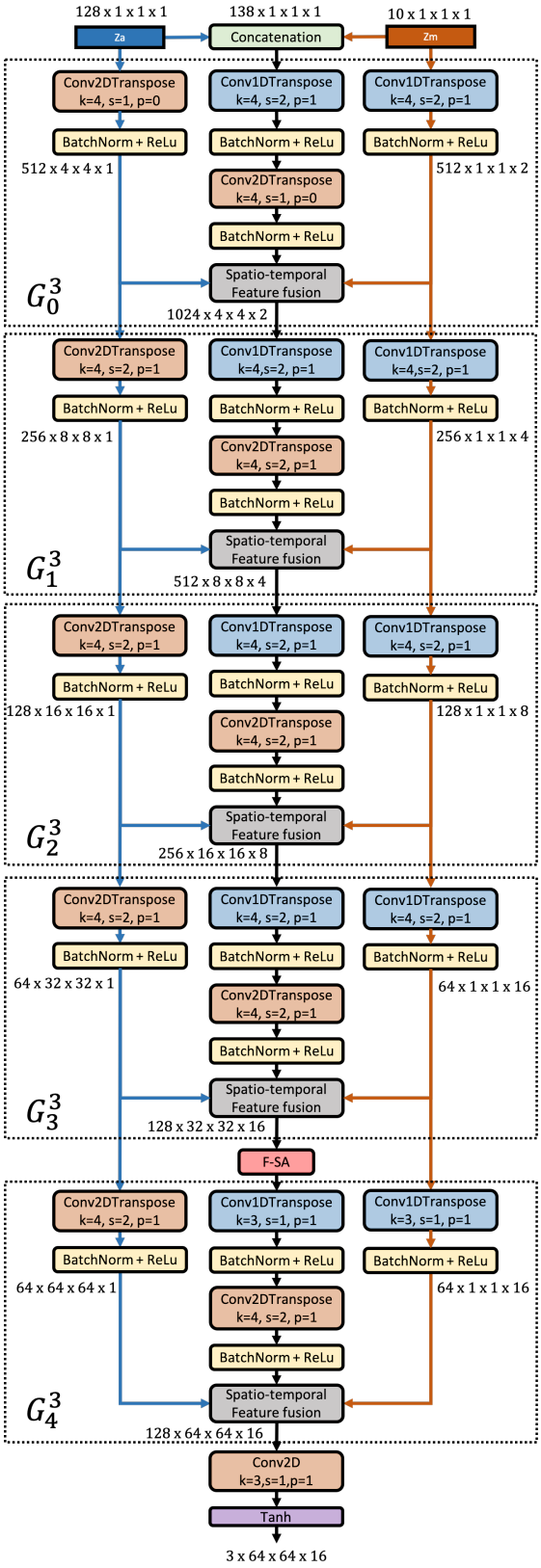


Figure 4.3 **Generator architecture.**

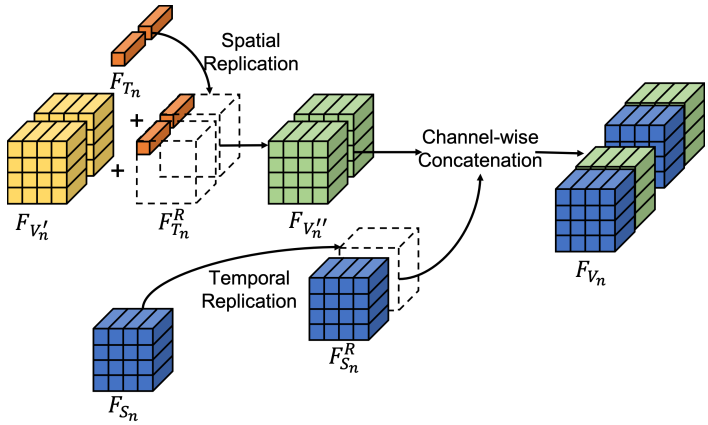
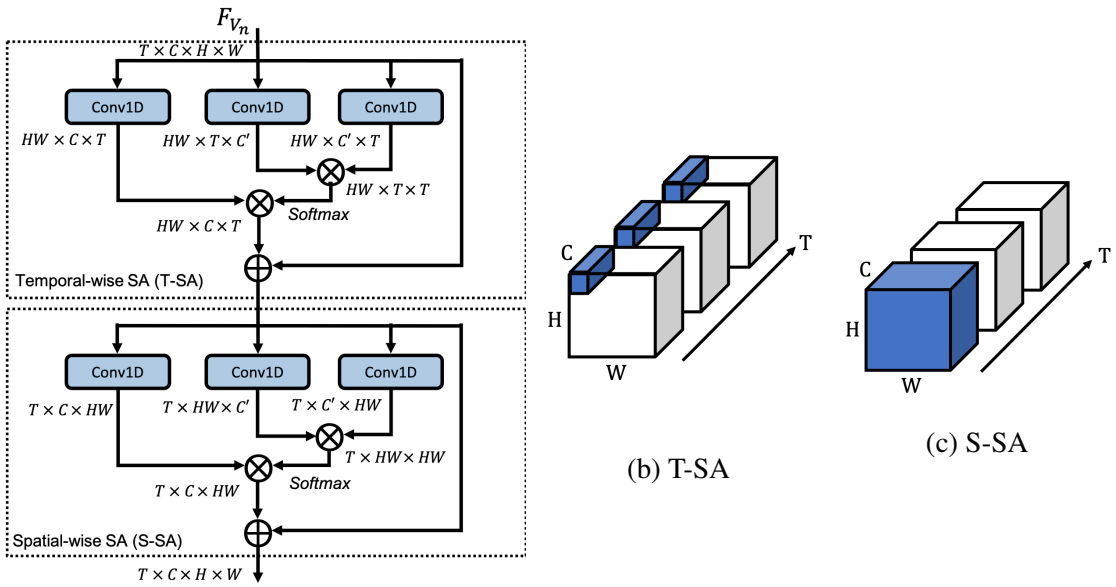


Figure 4.4 **Spatio-temporal fusion.**

Factorized spatio-temporal Self-Attention (F-SA). While Self-Attention (SA) has been successfully applied in image generation within SAGAN [201], it has not been explored in the context of spatio-temporal video generation. Here, we incorporate a spatio-temporal SA module, enabling G to utilize cues from all spatio-temporal feature positions and model relationships between widely separated regions. However, computing correlation between each position with all others in a 3D spatio-temporal feature map is very computationally expensive, particularly, if applied on higher feature maps in G. Therefore, we propose a novel *factorized spatio-temporal self-attention*, namely F-SA, as shown in Fig. 4.5. 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^{3}AN, we apply F-SA on the output of the G$^3_3$ in the G_V stream, which leads to the best video quality. We report related evaluation results of applying F-SA at various hierarchy-layers of the G^{3}AN in Section 5.3.

Our F-SA contains a Temporal-wise Self-Attention (T-SA) followed by a Spatial-wise Self-Attention (S-SA). Given spatio-temporal feature maps in the G_V stream, $F_{V_n} = x \in \mathbb{R}^{C \times T \times H \times W}$, where T and $H \times W$ denote temporal and spatial size, respectively. We firstly perform T-SA on $C \times T$ dimensions of x, where attention is only calculated along T for each position in x (see Fig. 4.5b). Then, S-SA is performed on $C \times H \times W$ dimensions and attention maps are obtained for all spatial position at each time step (see Fig. 4.5c).



(a) F-SA (b) T-SA (c) S-SA

Figure 4.5 **Factorized spatio-temporal Self-Attention (F-SA) module.**

While we apply T-SA, x is firstly transformed into two feature spaces f_t and g_t, in order to compute temporal self-attention

$$a^t_{s,ji} = \frac{\exp(t_{s,ij})}{\Sigma^T_{i=1}\exp(t_{s,ij})}, \; where \; t_{s,ij} = f_t(x_{s,i})^T g_t(x_{s,j}) \tag{4.1}$$

where a^t_{ji} indicates the correlation between j^{th} and i^{th} time instances for each position s in x. Then we apply attention maps on $h_t(x)$, which is the transformed feature map of x in h_t feature space. Finally we multiply the output of the attention layer by a scalar parameter γ_t and we add back the input feature map in order to obtain the final output of T-SA y^t.

$$y^t_{s,j} = \gamma_t \sum_{i=0}^{T} a^t_{s,ji} h_t(x_{s,i}) + x_{s,j}, \; h_t(x_{s,i}) = W_{h_t} x_{s,i} \tag{4.2}$$

Similar to T-SA, S-SA uses f_s, g_s and h_s to project y^t into three different feature spaces. γ_s is a learnable scalar parameter multiplied with the output after attention layer. S-TA is computed as following for each time step t.

$$a^s_{t,ji} = \frac{\exp(s_{t,ij})}{\Sigma_{i=1}^{N}\exp(s_{t,ij})},\ where\ s_{t,ij} = f_s(x_{t,i})^T g_s(x_{t,j}) \tag{4.3}$$

$$y^s_{t,j} = \gamma_s \sum_{i=0}^{N} a^s_{t,ji} h_s(x_{t,i}) + x_{t,j},\ h_s(x_{t,i}) = W_{h_s} x_{t,i} \tag{4.4}$$

In the above formulation, f_t, g_t, h_t, f_s, g_s and h_s are implemented as $1 \times 1 \times 1$ convolutions. For memory efficiency, we reduce channel numbers to $C^{'} = C/k$, where $k = 8$ for f_t, g_t, f_s and g_s in all our experiments.

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.

4.3.3 Training

Given our two-stream Discriminator architecture, G^{3}AN simultaneously optimizes D_V and D_I. Both losses use the GAN loss function proposed in DCGAN [122]. The objective functions of G^{3}AN can be expressed as

$$\begin{aligned} G^* &= arg \min_{G} \max_{D_I, D_V} \mathscr{L}(G, D_I, D_V), \\ \mathscr{L}(G, D_I, D_V) &= \mathscr{L}_I(G, D_I) + \mathscr{L}_V(G, D_V), \end{aligned} \tag{4.5}$$

where $\mathscr{L}_I$ denotes the loss function related to D_I, L_V represents the loss function related to D_V.

$$\begin{aligned}\mathscr{L}_I &= \mathbb{E}_{x^{'}\sim p_{data}}[log(D_I(x^{'}))] + \mathbb{E}_{z_a\sim p_{z_a}, z_m\sim p_{z_m}}[log(1-D_I(G(z_a,z_m)^{'}))],\\ \mathscr{L}_V &= \mathbb{E}_{x\sim p_{data}}[log(D_V(x))] + \mathbb{E}_{z_a\sim p_{z_a}, z_m\sim p_{z_m}}[log(1-D_V(G(z_a,z_m)))],\end{aligned} \tag{4.6}$$

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.

4.4 Experiments

This section presents the evaluation of G^{3}AN. We firstly describe implementation details, datasets and evaluation metrics used in this work. We secondly present quantitative and qualitative comparison with other methods w.r.t. video quality. Specifically, we evaluate and compare videos generated from G^{3}AN, VGAN, TGAN and MoCoGAN, quantitatively and qualitatively on all four datasets. Then, we test *conditional* and *unconditional* video generation, where we aim to demonstrate the effectiveness of the proposed decomposition method. Next, 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.

4.4.1 Implementation details

The entire network is implemented using PyTorch. We employ Adam optimizer [78] with β_1=0.5 and β_2=0.999. Learning rate is set to be $2e^{-4}$ 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.

4.4.2 Datasets

MUG Facial Expression dataset [2] contains 1254 videos of 86 subjects, performing 6 facial expressions, namely *happy*, *sad*, *surprise*, *anger*, *disgust* and *fear*.

UvA-NEMO Smile dataset [35] 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 [14][1].

Weizmann Action dataset [49] consists of videos of 9 subjects, performing 10 actions such as *wave* and *bend*. We augment it by horizontally flipping the existing videos.

UCF101 dataset [141] contains 13,320 videos of 101 human action classes. Similarly to TGAN [130], 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.

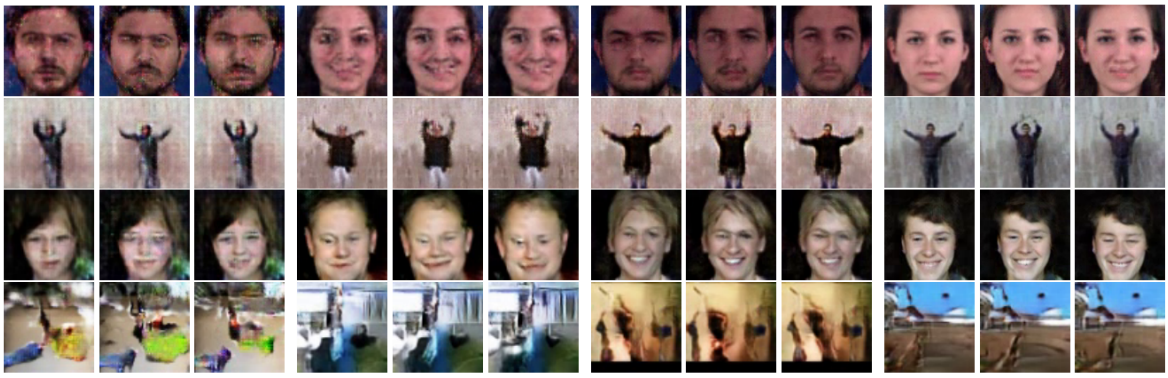
4.4.3 Evaluation metrics

We use the extension of two most commonly used metrics in image generation, the Inception Score (IS) [134] and Fréchet Inception Distance (FID) [54], in video level by using a pre-trained 3D CNN [51] as feature extractor, similar to [164].

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

1. https://github.com/1adrianb/face-alignment



(a) **VGAN** (b) **TGAN** (c) **MoCoGAN** (d) **G^{3}AN**

Figure 4.6 **Comparison with the state-of-the-art** on MUG (top-left), Weizmann (top-right), UvA-NEMO (bottom-left) and UCF101 (bottom-right).

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.

4.4.4 Quantitative Evaluation

We compare G^{3}AN with three state-of-the-art methods, namely VGAN, TGAN, as well as MoCoGAN. We report two evaluation metrics on the above four datasets. Comparison results among different methods are presented in Table 4.1. Our method consistently achieves the lowest FID on all datasets, suggesting that videos generated by G^{3}AN 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 Fig. 4.6.

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. G^{3}AN / Real videos). The posed question was ”Which video clip is more **realistic**?“. We report the mean user preference in Table 4.2.

	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^3AN	**67.12**	**119.22**	**86.01**	**91.21**	**3.62**

Table 4.1 **Comparison with the state-of-the-art** on four datasets w.r.t. FID and IS.

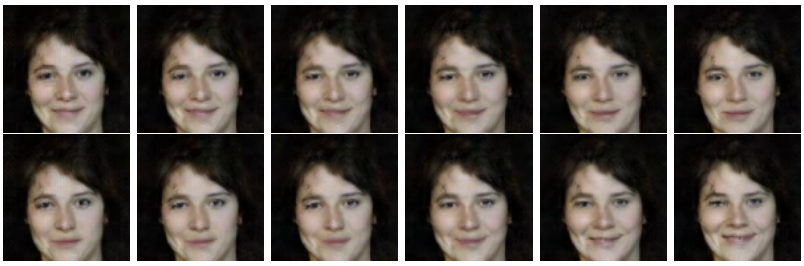
We observe that human raters express a strong preference for the proposed framework G^3AN over MoCoGAN (84.26% v.s. 15.74%), TGAN (87.31% v.s. 12.69%) and VGAN (90.24% v.s. 9.76%), which is consistent with the above listed quantitative results. Further, we compare real videos from all datasets with the generated video sequences from our method. The human raters ranked 25.71% of videos from our G^3AN as more realistic than real videos, which we find highly encouraging.

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

Table 4.2 **Mean user preference** of human raters comparing videos generated by the respective algorithms, originated from all datasets.

4.4.5 Qualitative Evaluation

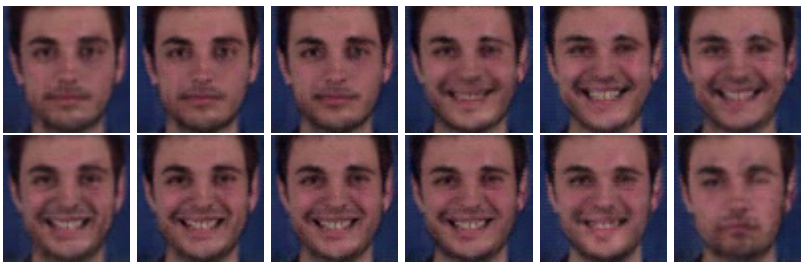
We conduct an **unconditional generation** experiment utilizing the Uva-NEMO dataset, where we fix z_a and proceed to randomly vary motion, z_m. Associated generated samples from G^3AN and MoCoGAN are shown in Fig. 4.7. While we observe the varying motion in the video sequences generated by G^3AN, the 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.



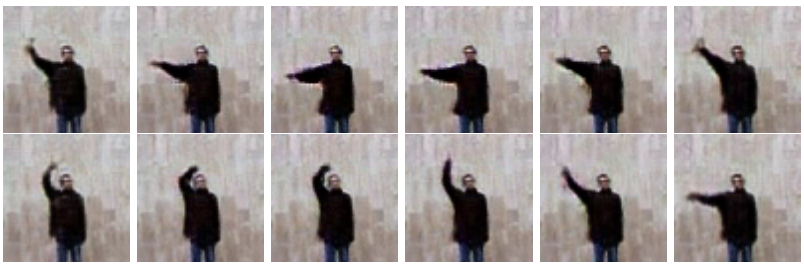
(a) **G^{3}AN**

(b) **MoCoGAN**

Figure 4.7 **Unconditional video generation** of G^{3}AN 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.



(a) **MUG**: *Happiness*



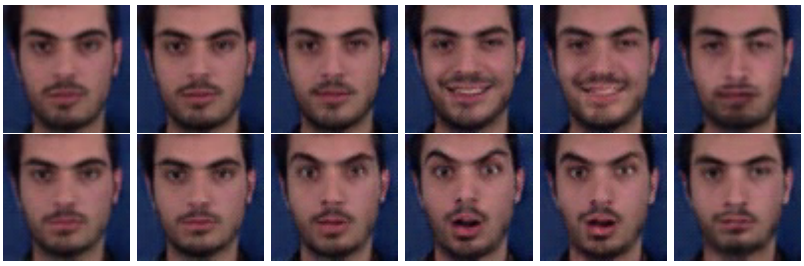
(b) **Weizmann**: *One hand waving*

Figure 4.8 **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.

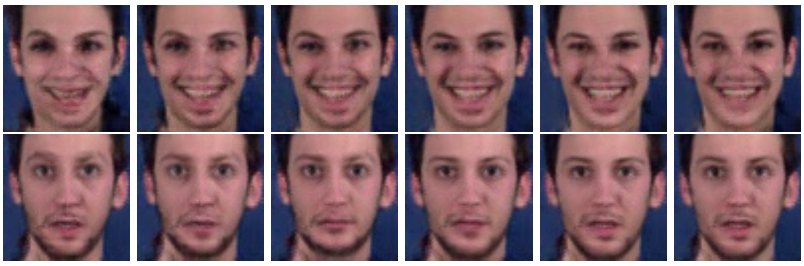
Conditional video generation. Further, we leverage on labels of the MUG and Weizmann datasets, in order to analyze conditional video generation. Towards this, we here concatenate a one-hot category vector and motion noise z_m, feeding it into G_T. We note that the inputs of G_S and G_V remain the same as in the setting of unconditional generation. Related results show that when varying motion-categories, while having a fixed appearance, G^{3}AN correctly generates an identical facial appearance, with appropriate category-based motion (facial expressions and body actions), see Fig. 4.8. 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, G^{3}AN is able to generate *the same 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 Fig. 4.9. 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 G^{3}AN 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 [56], we proceed to illustrate the ability of our model to learn latent representations by manipulating 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 Fig. 4.10. Interestingly such factors can be related to semantics, e.g., facial view point in Fig. 4.10a, mustache in Fig. 4.10b, and color of pants in Fig. 4.10c. 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 Fig. 4.11. Such



(a) **G^{3}AN**



(b) **MoCoGAN**

Figure 4.9 **Comparison between G^{3}AN and MoCoGAN.** Given fixed z_a and z_m, as well as two condition-labels *smile* and *surprise*, G^{3}AN and MoCoGAN generate correct facial expressions. However, while G^{3}AN preserves the appearance between rows, MoCoGAN alters the subject's appearance.

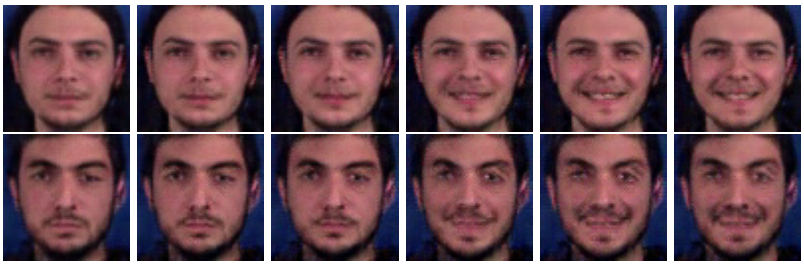
observations show that our method learns to interpolate between different data points in motion- and appearance-latent spaces, respectively.

Addition of appearance representations. We here *add* appearance vectors, aiming to analyze the resulting latent 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 *summary appearance* pertaining to z_{a_0} and z_{a_1}, with identical motion as z_{m_0} and z_{m_1}, see Fig. 4.12.

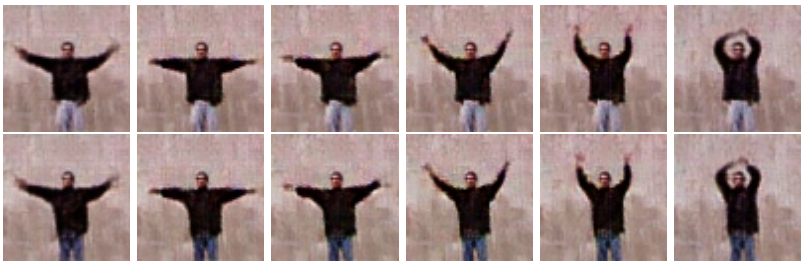
4.4.6 Ablation Study

We here seek to study the effectiveness of proposed G^{3}AN 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

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



(b) Manipulation of *third* dimension on MUG

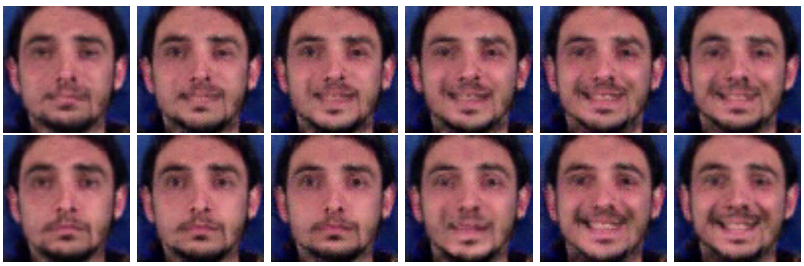


(c) Manipulation of *second* dimension on Weizmann

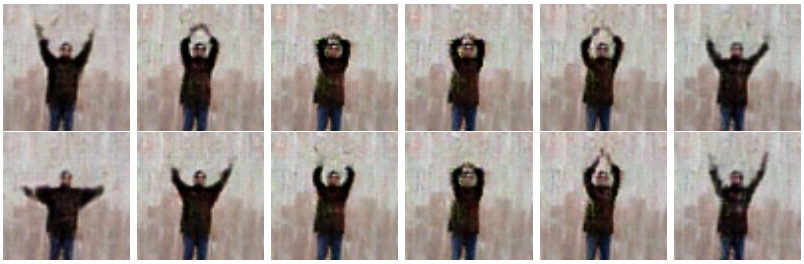
Figure 4.10 **Latent appearance representation manipulation.** For each dataset, each row shares the same motion representation, whereas from top to bottom values in one dimension of appearance representation are increased. See SM for more samples.

generate different subjects, however for each person the facial movement is constant, see Fig. 4.13 (*top*). Similarly, when G_S is removed, changing motion will affect subject's identity, whereas the appearance vector loses its efficacy, see Fig. 4.13 (*middle*). When removing both, G_T and G_S, appearance and motion are entangled and they affect each other, see Fig. 4.13 (*bottom*). This demonstrates the effective disentanglement brought to the fore by the streams G_S and G_T.

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 4.3. The results show that after removing all three components, video quality is the poorest, which proves that all of them contribute to the final

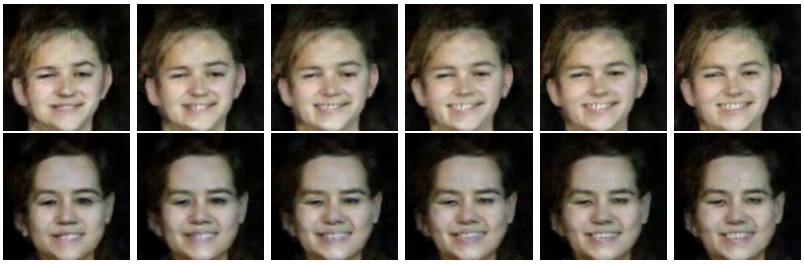


(a) Manipulation of *sixth* dimension of MUG



(b) Manipulation of *second* dimension on Weizmann

Figure 4.11 **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 4.12 **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).

(a) z_a, z_{m_0} (b) z_a, z_{m_1}

Figure 4.13 **Ablation study.** Generated videos obtained by removing G_T (*top row*), removing G_S (*middle*), and both (*bottom row*).

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**	**UvA**	**Weizmann**	**UCF101**	
	FID ↓	FID ↓	FID ↓	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 4.3 **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 4.4. 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 temporal information, which we believe to be important in video generation tasks.

	MUG	**UvA**	**Weizmann**	**UCF101**	
	FID ↓	FID ↓	FID ↓	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 4.4 **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 4.5. 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 [9], *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	**UvA**	**Weizmann**	**UCF101**	
	FID ↓	FID ↓	FID ↓	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 4.5 **Comparison of inserting F-SA at different hierarchical levels of G^3AN.**

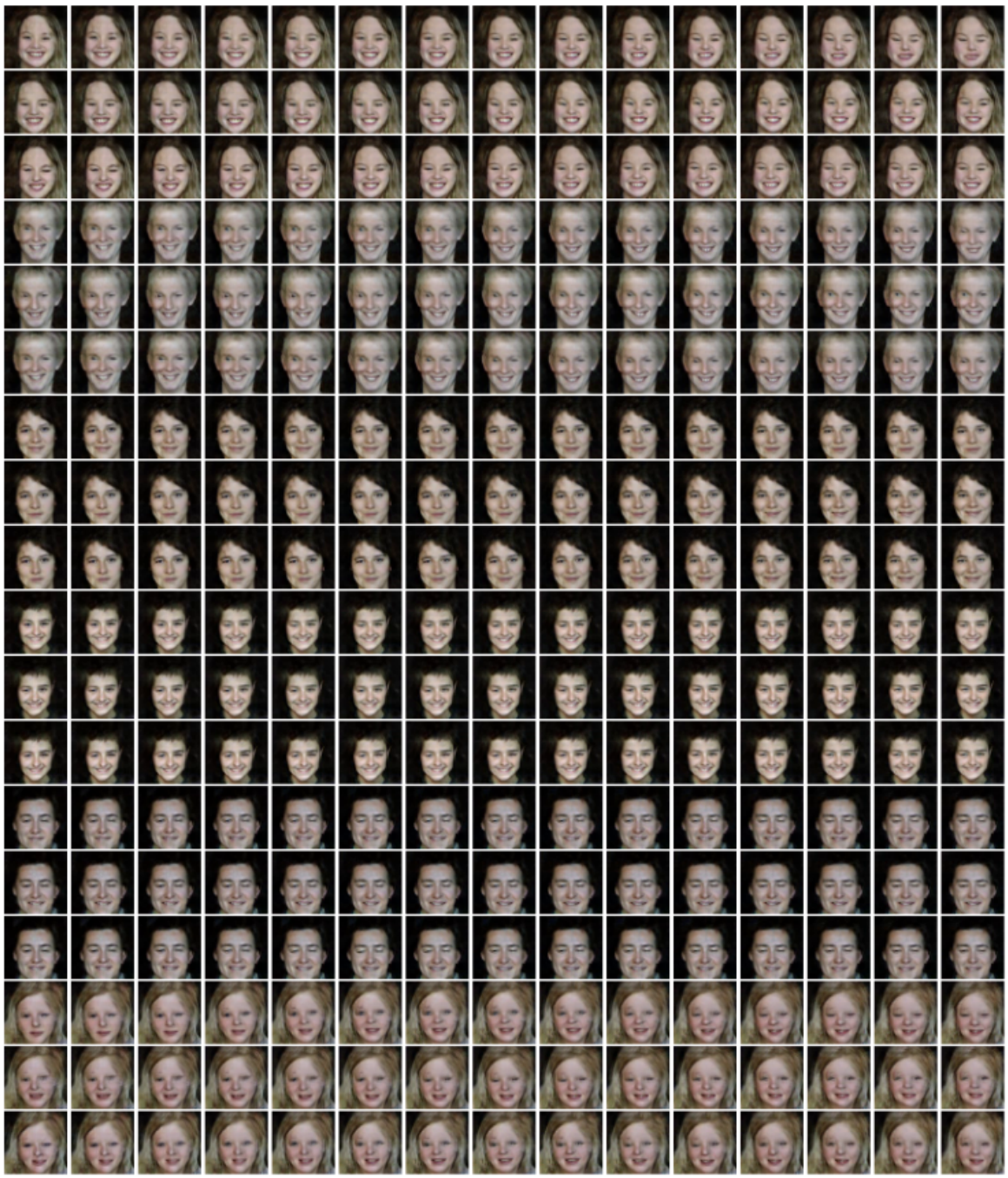


Figure 4.14 **Unconditionally generated samples from G^{3}AN on UvA-NEMO.** We combine each z_a with three different z_m, obtaining three different videos for the same appearance. Each row represents a video sequence.

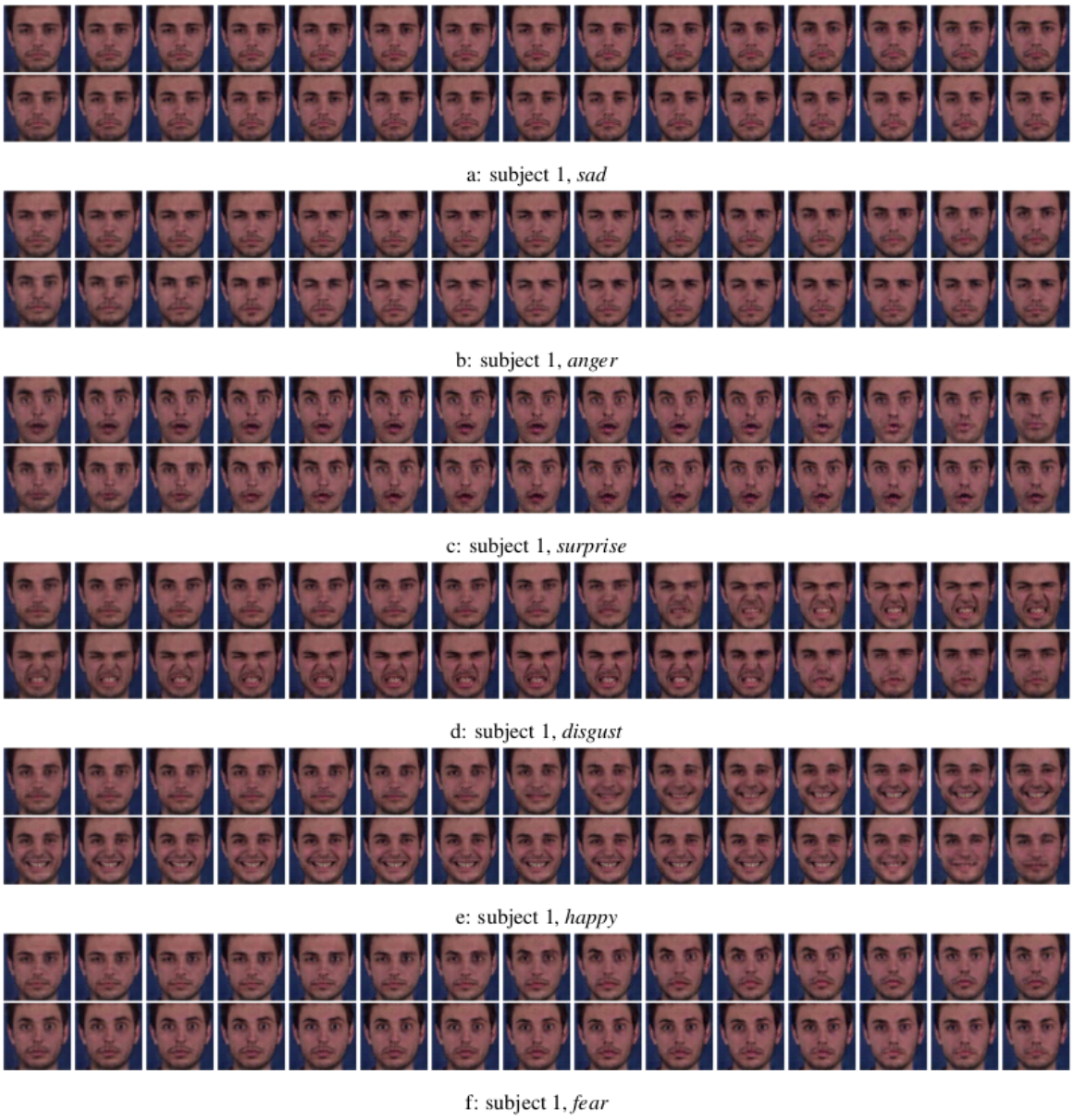


Figure 4.15 **Conditionally generated samples from G^{3}AN on MUG dataset.** Each row represents the result generated by combining a one-hot category label with the same z_a and randomly sampled z_m as input.

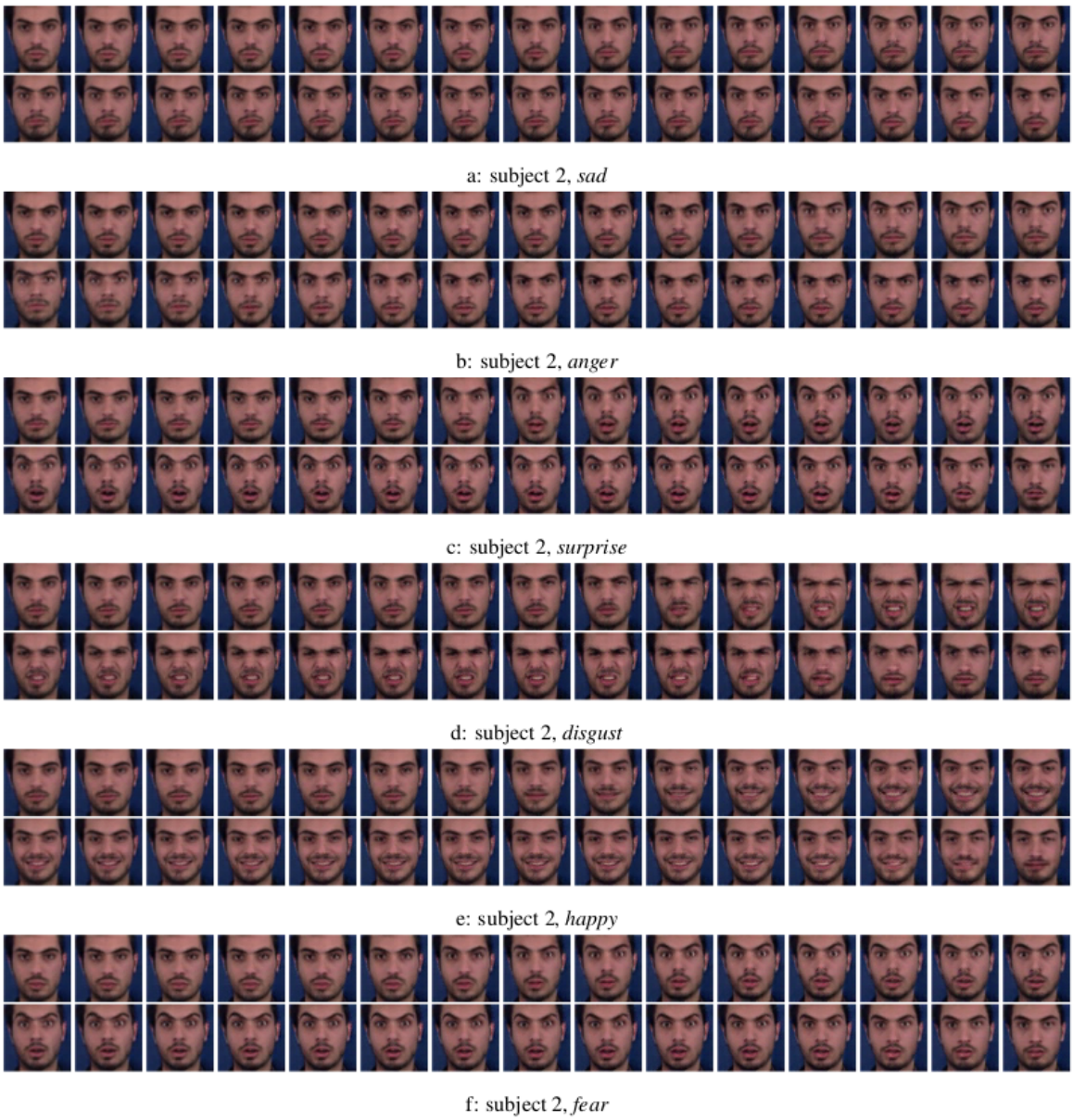


Figure 4.16 **Conditionally generated samples from G^{3}AN on MUG dataset.** Each row represents the result generated by combining a one-hot category label with the same z_a and randomly sampled z_m as input.

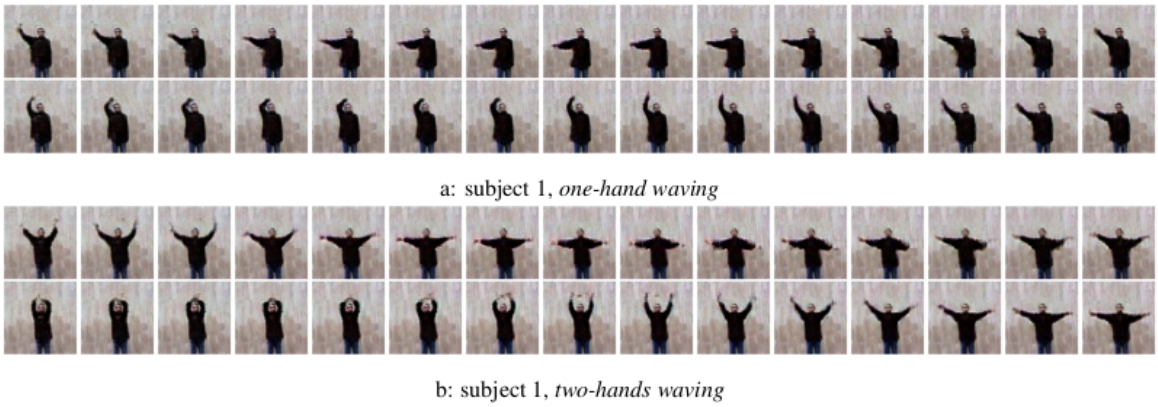


Figure 4.17 **Conditionally generated samples from G^{3}AN on Weizmann dataset.** Each row represents the result generated by combining a one-hot category label with the same z_a and randomly sampled z_m as input.

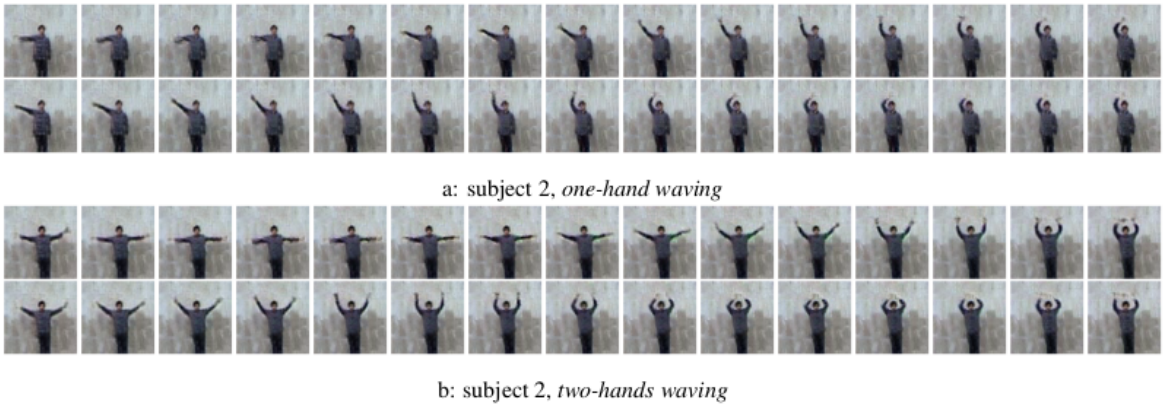


Figure 4.18 **Conditionally generated samples from G^{3}AN on Weizmann dataset.** Each row represents the result generated by combining a one-hot category label with the same z_a and randomly sampled z_m as input.

a: subject 1

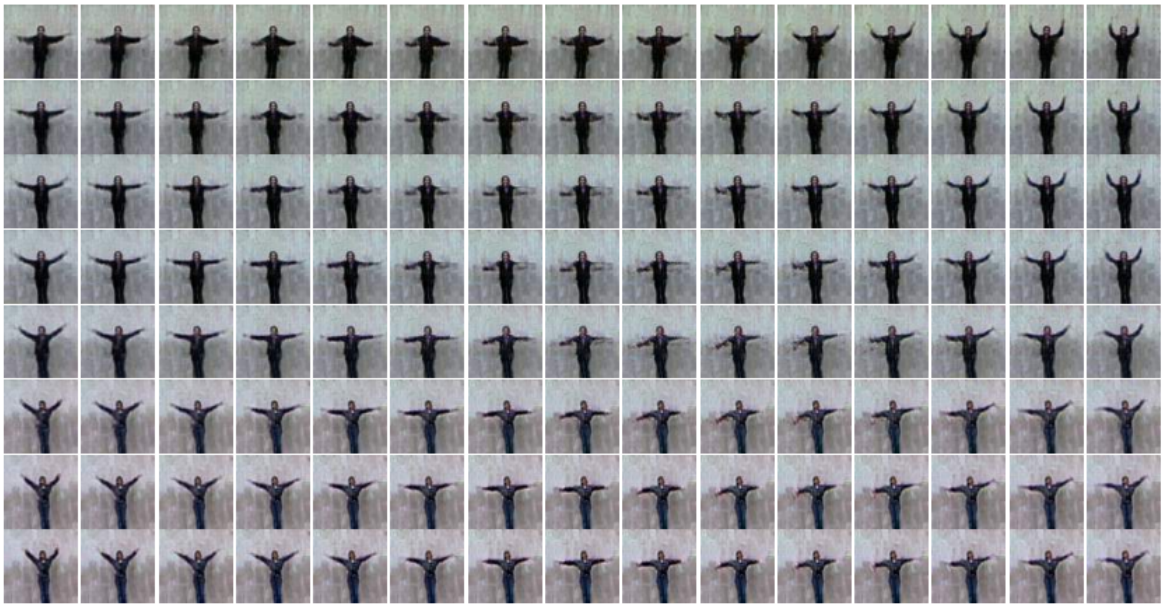
b: subject 2

Figure 4.19 **Results of manipulating *first dimension* in appearance representation on MUG dataset.** *a* and *b* are results from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased.

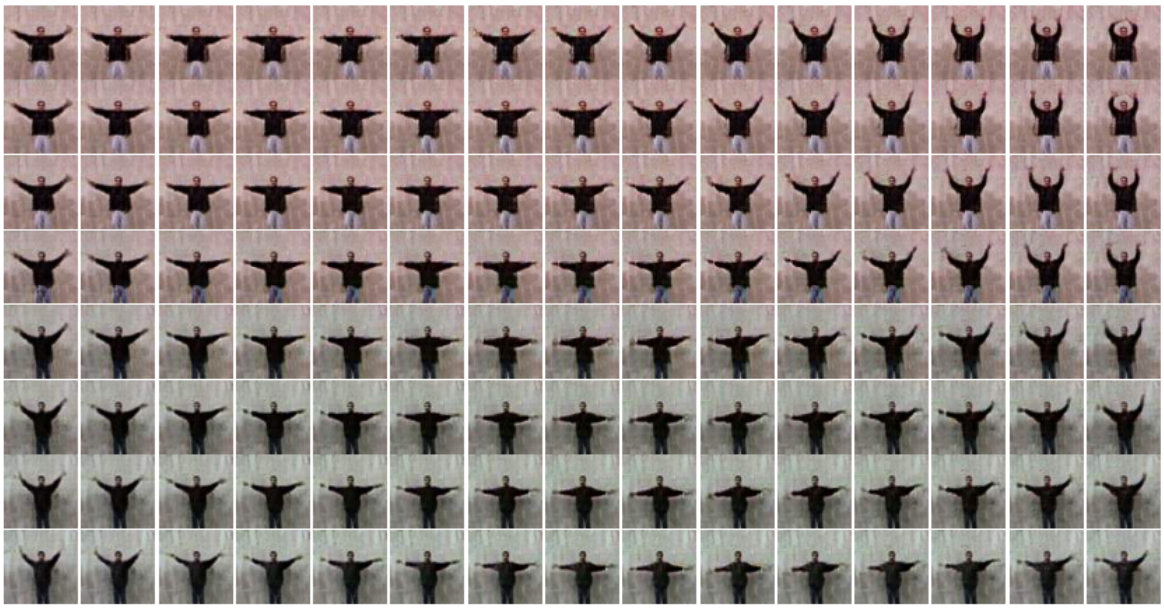
a: subject 1

b: subject 2

Figure 4.20 **Results of manipulating *second dimension* in appearance representation on MUG dataset.** a and b are results from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased.

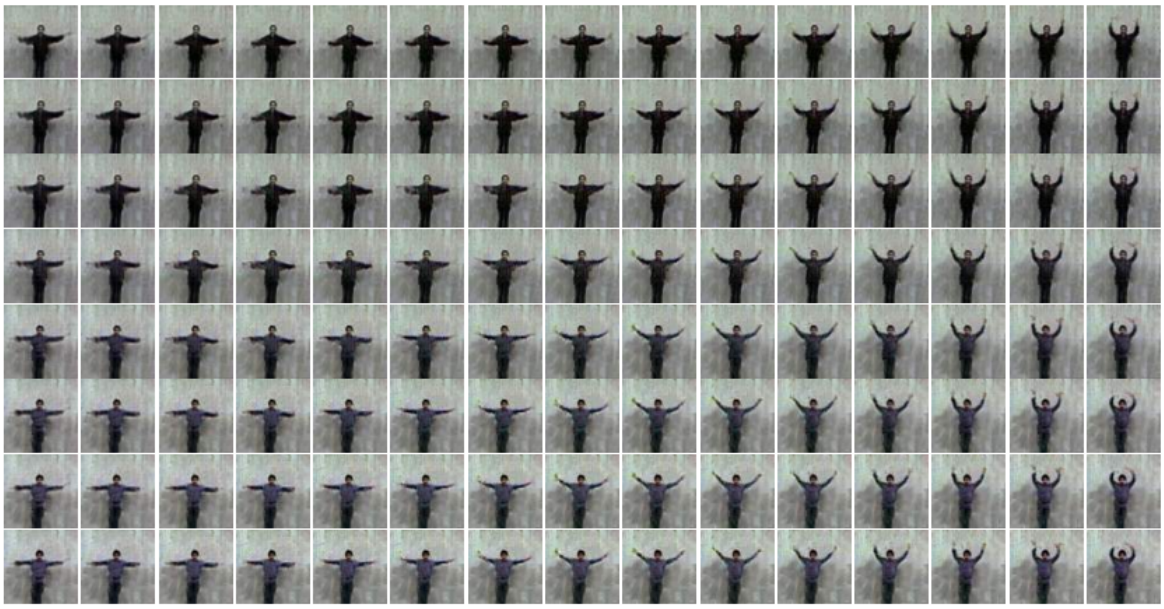


a: subject 1

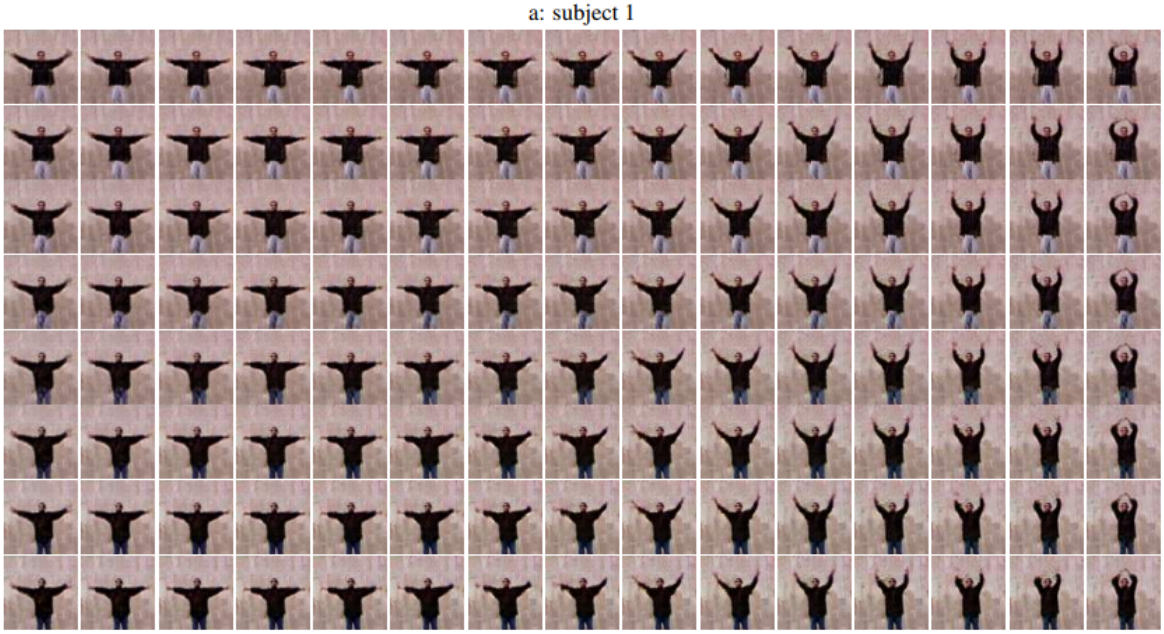


b: subject 2

Figure 4.21 **Results of manipulating *first dimension* in appearance representation on Weizmann dataset.** a and b are from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased.



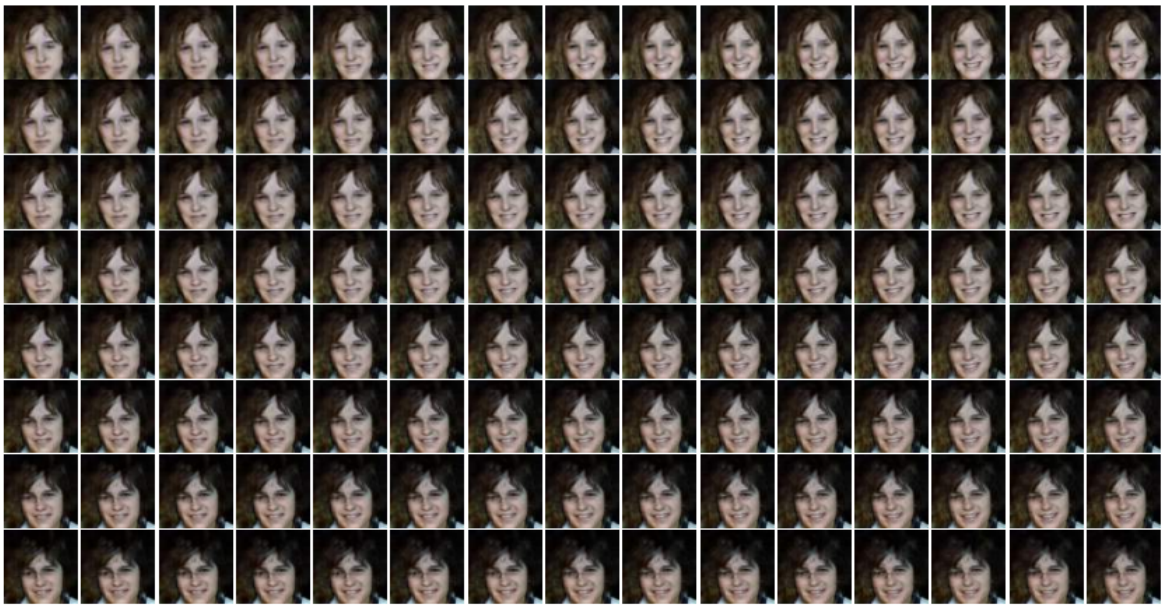
a: subject 1



b: subject 2

Figure 4.22 **Results of manipulating *second dimension* in appearance representation on Weizmann dataset.** a and b are from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased.

a: subject 1



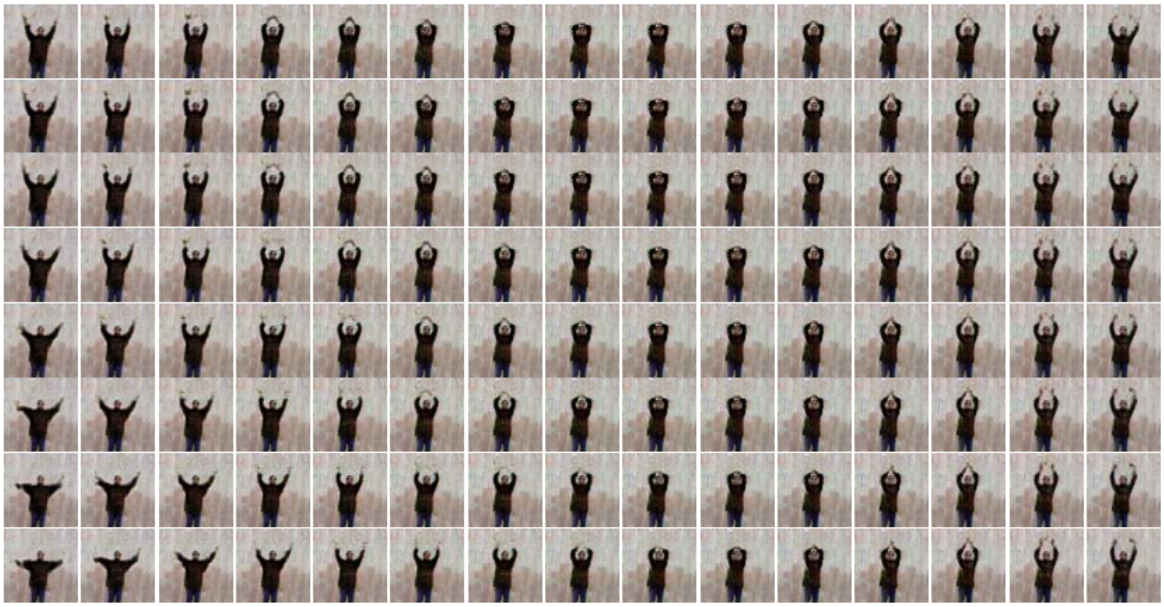
b: subject 2

Figure 4.23 **Results of manipulating *first dimension* in appearance representation on UvA-NEMO dataset.** *a* and *b* are from two randomly sampled z_a. From top to bottom in each sub-figure, values of *first dimension* are increased.

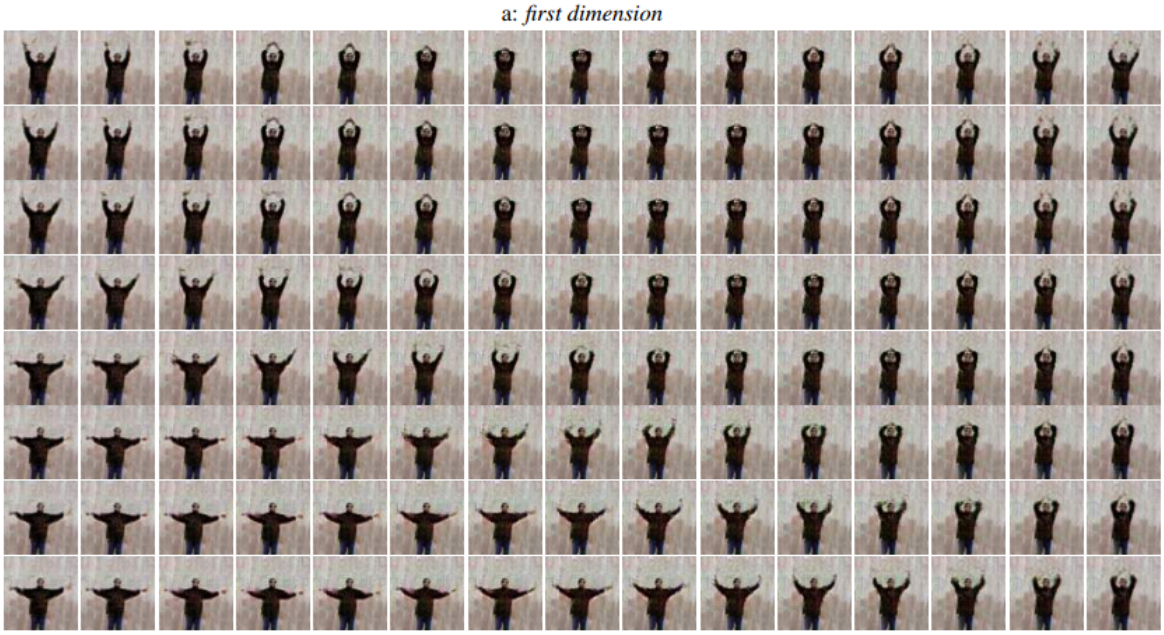
a: second dimension

b: *sixth dimension*

Figure 4.24 **Results of manipulating motion representation on UvA-NEMO dataset.** *a* and *b* are results of manipulating *first* and *sixth* dimensions. From top to bottom in each sub-figure, values are increased.



a: *first dimension*



b: *second dimension*

Figure 4.25 **Results of manipulating motion representation on Weizmann dataset.** *a* and *b* are results of manipulating *first* and *second* dimensions. From top to bottom in each sub-figure, values are increased.

Chapter 5

Motion Interpretation in Video Generation

In the previous chapters, we have described two methods to generate videos in conditional and unconditional manners. Both methods are designed to generate videos of good visual quality and spatio-temporal consistency. The first method takes an input image and motion label to generate videos, whereas the second method generates videos only from input noises. In this chapter, we introduce an unconditional video generative model, MintGAN, targeted to allow for interpretation of the latent space. Towards this, we design a model that generates high quality videos, placing emphasis on the interpretation and manipulation of *motion*. Specifically, we decompose motion into semantic sub-spaces, which allow for control of generated samples. We design the generator of MintGAN in accordance to proposed Linear Motion Decomposition, which carries the assumption that motion can be represented by a dictionary, whose atoms form an orthogonal basis in the latent space. Each vector in the basis represents a semantic sub-space. In addition, a Temporal Pyramid Discriminator analyzes videos at different temporal resolutions. Extensive quantitative and qualitative analysis shows that our model systematically and significantly outperforms state-of-the-art methods on the VoxCeleb2-mini, BAIR-robot and UCF101 datasets w.r.t. video quality, as well as confirms that decomposed sub-spaces are interpretable and moreover, generated motion is controllable.

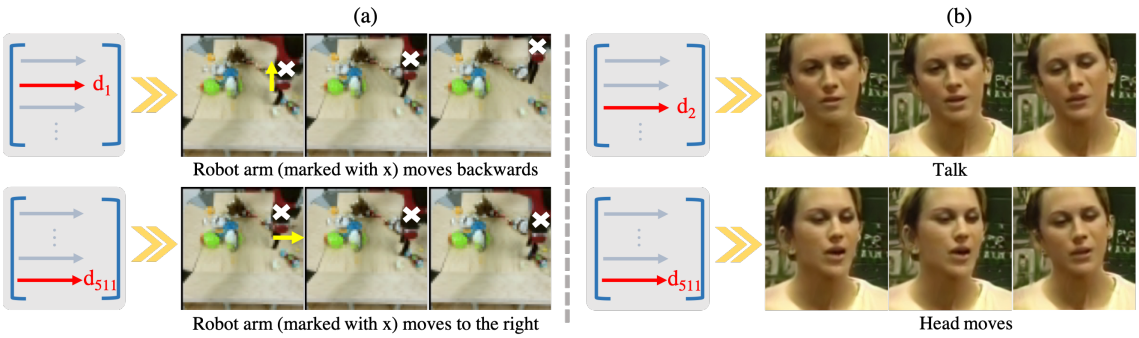


Figure 5.1 **Controllable video generation.** MintGAN learns to decompose motion into semantic motion-components. This allows for manipulations in the latent code to invoke motion in generated videos that is human interpretable. Top (a) robot arm moves backwards, bottom (a) robot arm moves to the right. Similarly, in (b) we are animating the face to 'talk' (top) and 'move head' (bottom).

5.1 Introduction

Videos signify more complex data than images, due to the additional temporal dimension. While some research works showed early results in video generation [157, 130, 148, 168], related interpretability is yet to be revealed. Such interpretability and hence steerability is of particular interest, as it would render video GANs highly instrumental in a number of down-stream applications such as *video editing* [164] and *data augmentation* [154, 151]. Motivated by the above, we here consider the following question: Can we control and manipulate the complex visual world created by video GANs?

In order to answer this new and intricate question, we propose a new video GAN that decomposes and enables interpretation of motion, which we refer to as MintGAN. In particular, we aim to interpret the latent space of MintGAN by finding sub-spaces, which are endowed with semantic meanings. Once such sub-spaces have been identified, walking along certain trajectories within them allows for targeted modification of generated videos. Specifically, we here place emphasis on interpreting and modifying *motion*. We note that the posed research question deviates from current efforts on interpreting *appearance* [66, 135, 158] in the latent space.

This new problem necessitates an original architecture, streamlined to allow for analysis of the latent motion representation. Hence, we propose a new interpretable architecture,

which we design based on the assumption that motion can be decomposed into independent *semantic* motion-components. Therefore, we define the motion space by a linear combination of *semantic* motion-components, which can reflect 'talking' and 'robot arm moving left and right'. We implement named decomposition via a motion bank in our generator. Once trained, MintGAN allows for the incorporation (elimination) of corresponding motion-components in the generated videos by activating (deactivating) associated latent directions, see Figure 5.1.

Meaningful interpretation is only justified in an architecture that is able to *generate high-quality videos*. To ensure that, we design a two-stream discriminator, which incorporates an image discriminator, as well as a novel Temporal Pyramid Discriminator (TPD) that contains a number of video discriminators. The latter leverages on a set of temporal resolutions that are related to temporal speed. We show that while our proposed discriminator incorporates 2D ConvNets, it is consistently superior to 3D-discriminators. We evaluate proposed MintGAN on three large datasets, namely VoxCeleb2-mini [112], BAIR-robot [37] and UCF101 [141]. In extensive qualitative and quantitative evaluation, we show that MintGAN systematically and significantly outperforms state-of-the-art baselines w.r.t. video quality. In addition, we propose an evaluation framework for motion interpretability and proceed to demonstrate that MintGAN is interpretable, as well as steerable. Finally, we provide experiments, showcasing generation of higher-resolution, as well as longer videos.

5.2 Background

Image Generation. Recently, both conditional [13, 165, 118, 215, 65, 117] and unconditional [70, 72, 73, 97, 92, 206] image generation methods have witnessed considerable progress. Related to our work, notably StyleGAN [72] and StyleGAN2 [73] have advanced the state-of-the-art in unconditional image generation. Related architecture incorporates modulation based convolutional layers, which re-introduce a latent code at different layers of the network. Alterations of the latent code correspond to particular manipulations in generated images. For example basic operations such as adding a vector, linear interpolation,

and crossover in the latent space cause expression transfer, morphing, and style transfer in generated images.

Video Generation. While realistic video generation is the natural sequel of image generation, it entails a number of challenges related to complexity and computation, associated to the simultaneous modeling of appearance, as well as motion. Current video generation can be categorized based on related input data into *unconditional* and *conditional* methods.

Unconditional video generation seeks to map noise to video, directly and in the absence of other constraints. Examples of unconditional methods include VGAN [157], TGAN [130], MoCoGAN [148] and G^3AN [168]. VGAN was equipped a two-stream generator to generate foreground and background separately. TGAN firstly generated a set of latent vectors corresponding to each frame and then aimed at transforming them into actual images. MoCoGAN and G^3AN decomposed the latent representation into motion and appearance, aiming at controlling both factors. We note that named methods have learned to capture spatio-temporal distribution based on shallow architectures. Such works predominantly focused on improving the quality of generated videos, rather than exploring interpretability of the latent space. While MoCoGAN and G^3AN disentangled appearance and motion, no further investigation on underlying semantics was provided. As opposed to that, our main goal in this paper is to gain insight into the latent space, seeking to dissect complex motion into semantic latent sub-spaces.

In contrast to unconditional video generation, *conditional* video generation aims at achieving videos of high visual quality, following image-to-image generation [27, 65, 61]. In this context and due to challenges in modeling of high dimensional video data, additional information such as lables [170], semantic maps [116, 164, 163], human key-points [67, 188, 161, 17, 199, 163], 3D face mesh [204] and optical flow [89, 114] have been exploited to guide motion generation. We note that given the provided motion-prior, in such methods, generative models do not learn to capture the full motion distribution.

We note that video generation is based on a noise vector as input, whereas *video prediction* aims at predicting future frames based on a set of *existing frames*. Towards latter, additional

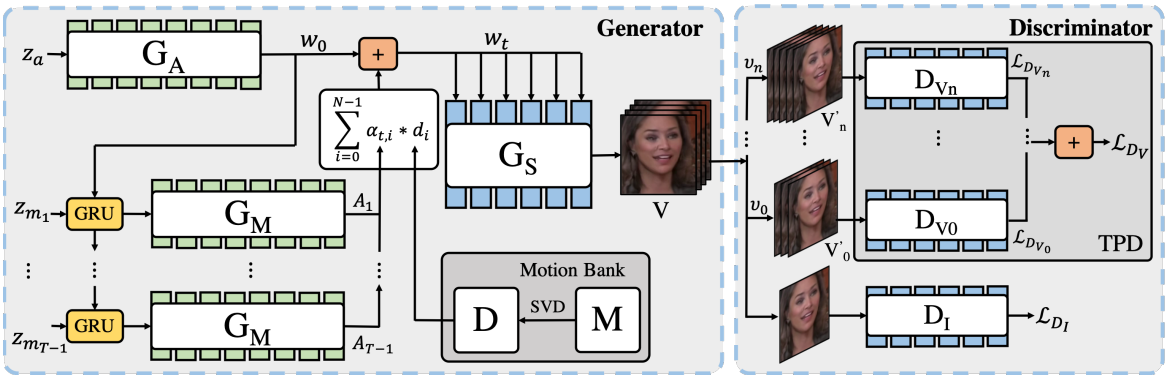


Figure 5.2 **MintGAN-architecture.** MintGAN comprises of a Generator and a two-stream Discriminator. We design the architecture of the Generator based on proposed Linear Motion Decomposition. Specifically, a motion bank is incorporated in the Generator to learn and store a motion dictionary D, which contains motion-directions $[d_0, d_1, .., d_{N-1}]$. We use an appearance net G_A to map appearance noise z_a into a latent code w_0, which serves as the initial latent code of a generated video. A motion net G_M maps a sequence of motion noises $\{z_{m_t}\}_{t=1}^{T-1}$ into a sequence $\{A_t\}_{t=1}^{T-1}$, which represent motion magnitudes. Each latent code w_t is computed based on Linear Motion Decomposition using w_0, D and A_t. Generated video V is obtained by a synthesis net G_S that maps the sequence of latent codes $\{w_t\}_{t=0}^{T-1}$ into an image sequence $\{x_t\}_{t=0}^{T-1}$. Our discriminator comprises an image discriminator D_I and a Temporal Pyramid Discriminator (TPD) that contains several video discriminators D_{V_i}, leveraging different temporal speeds υ_i to improve generated video quality. While D_I accepts as input a randomly sampled image per video, each D_{V_i} is accountable for one temporal resolution.

priors are instrumental, such as optical flow or [83], human pose [156, 202], provided by additional modules that cater such scenario. We here focus on unconditional video generation and associated interpretability, hence video prediction is out of scope for this paper.

GAN Interpretation. In an effort to open the black box representing GANs, Bau *et al.* [9, 8] sought to associate neurons in the generator with the encoding of pre-defined visual concepts such as colors, textures and objects. Subsequent works [135, 46, 66, 158] proceeded to explore the interpretability of the latent space, seeking for latent representations corresponding to different semantics in generated images. *Linear* [135, 66] and *non-linear* [66] *walks* in the latent space enabled for semantic concepts in the generated images to be modified. Deviating from previous work, in this chapter we focus on interpreting MintGAN, in order to discover semantics related to *motion*.

5.2.1 Linear Motion Decomposition (LMD)

We formulate unconditional video generation as learning a function G_S that maps a sequence of latent codes $S = \{w_t\}_{t=0}^{T-1}, w_t \sim \mathscr{W} \subset \mathbb{R}^N \ \forall t$ to a sequence of images $V = \{x_t\}_{t=0}^{T-1}, x_t \sim \mathscr{X} \subset \mathbb{R}^{H \times W \times 3}$, such that $G_S(w_t) = x_t, \forall t \in [0, T-1]$, where T denotes the length of the video. S is obtained by mapping a sequence of noises $Z = \{z_t\}_{t=0}^{T-1}, z_t \sim \mathscr{Z} \subset \mathbb{R}^N$ into the $\mathscr{W}$ space. However, such mapping jointly learns appearance and motion, rendering $\mathscr{W}$ challenging to be interpreted. *W.r.t.* interpretable $\mathscr{W}$, and in hindsight to our core objective, we propose to decompose motion into linear independent components.

Given a video of high visual quality and spatio-temporal consistency, we assume that motion between consecutive frames follows a *transformation* $\mathscr{T}_{t \to (t+1)}$, so that $G_S(w_{t+1}) = \mathscr{T}_{t \to t+1}(G_S(w_t))$. Based on the idea of equivariance [86, 28, 57], an alteration in the latent space causes a corresponding alteration in the output, consequently a transition $\tau_{t \to t+1}$ affecting the latent space results in $G_S(\tau_{t \to t+1}(w_t)) = \mathscr{T}_{t \to t+1}(G_S(w_t))$.

Recent works [66, 135] showed that for a given image-transformation $\mathscr{T}$ such as shifting and zooming, there exists a vector d in the latent space, which represents the direction of $\mathscr{T}$. By linearly navigating in this direction with a magnitude α, a corresponding transformation $\mathscr{T}(G(w)) = G(w + \alpha * d)$ is witnessed in generated images.

Therefore, we assume that any transition $\tau_{t \to t+1}$ associated to $\mathscr{T}_{t \to t+1}$ can be represented as a composition of motion-directions in a **motion dictionary** $D = [d_0, d_1, .., d_{N-1}], d_i \in \mathbb{R}^N$. We constrain these motion directions to form an orthogonal basis, so that

$$< d_i, d_j >= \begin{cases} 0 & i \neq j \\ 1 & i = j. \end{cases} \tag{5.1}$$

If these directions are interpretable, manipulating the magnitude of any of them should inflict a specific semantic change in the output, without affecting other directions. Therefore, in transformation $\mathscr{T}_{t \to t+1}$, the magnitude $A_t = [\alpha_{t,0}, \alpha_{t,1}, ..., \alpha_{t,N-1}], \alpha_{t,i} \in \mathbb{R}$ varies. Each $a_{t,i}$ denotes the magnitude pertained to the ith direction at time step t. Based on this, we define

the Linear Motion Decomposition (LMD) as following

$$\tau_{t\to t+1}(w_t) = w_t + \sum_{i=0}^{N-1} \alpha_{t,i}\, d_i, \tag{5.2}$$

where the transformation between consecutive frames is indicated as

$$\begin{aligned} G_S(w_{t+1}) &= \mathscr{T}_{t\to t+1}(G_S(w_t)) \\ &= G_S(\tau_{t\to t+1}(w_t)) \\ &= G_S\left(w_t + \sum_{i=0}^{N-1} \alpha_{t,i}\, d_i\right). \end{aligned} \tag{5.3}$$

The general term of w_t is hence

$$w_t = w_0 + \sum_{i=0}^{N-1}\sum_{j=0}^{t-1} \alpha_{j,i}\, d_i,\ t \in [1, T-1]. \tag{5.4}$$

So far, we have succeeded transferring learning w_t from an unknown motion space into learning three variables from three sub-spaces which contain clear meanings, namely initial appearance code w_0, magnitude sequence $\{A_t\}_{t=1}^{T-1}$, as well as associated motion-directions $[d_0, d_1 ... d_{N-1}]$. We proceed to elaborate on how we implement described linear motion decomposition in our architecture.

5.2.2 Generator

The initial latent code w_0 serves as a representation of *appearance* in the first and all following frames of an output video. At the same time, the vector A_t represents a set of magnitudes associated to motion directions in a transition and hence is accountable for *motion*. Taking that into account, we decompose $\mathscr{Z}$ into two separated spaces $\mathscr{Z}_{\mathscr{A}}$ and $\mathscr{Z}_{\mathscr{M}}$ representing appearance and motion, respectively. Hence w_0 is generated by mapping an appearance noise $z_a \sim \mathscr{Z}_{\mathscr{A}}$ using an appearance net G_A. A_t is mapped from the motion noise $z_{m_t} \sim \mathscr{Z}_{\mathscr{M}}$ by a motion net G_M. In order to ensure temporal consistency in the latent space,

we integrate a GRU [25] with its initial code set to be z_a prior to the mapping. We note that G_A and G_M are two independent 8-layer MLPs.

Based on proposed LMD, the motion dictionary D is entitled to an orthogonal basis. We propose to find a matrix, with eigenvectors representing d_i. More specifically, we pre-define a matrix $M \in \mathbb{R}^{N \times N}$ and devise it trainable, updating it along with the parameters in the generator. D is represented as the transpose of *right singular vectors* of M, $M = U\Sigma V^T$ and $D = V^T$. Each d_i is an eigenvector of matrix $M^T M$ and is learned based on adversarial learning. Once trained, M captures the motion distribution of the training dataset and decomposes it into N independent directions. We show that directions are interpretable and moreover can be manipulated, which results in related modifications of generated results, see Section 5.3.4. M is initialized randomly and updated with other parameters in G via back-propagation. We refer to M and D jointly as **motion bank**.

We adapt the architecture proposed by Karras *et al.* [73] in G_S. We note that G_S serves as a rendering network, which incorporates a sequence of convolutional blocks aiming to up-sample a learned constant into high resolution images. In each block, convolutional layers are modulated by the respective input w_t, in order to learn different appearances. Each w_t is computed according to Equation (5.8) and serves as input of G_S to generate related frame $x_t = G_S(w_t)$.

5.2.3 Discriminator

Temporal speed in videos has been a pertinent cue in action recognition [40, 189]. We note that videos sampled at temporal speeds $\{\upsilon_i | i \in \mathbb{R}, 0 \leq i < n\}$, which represent temporal resolutions, provide a set of motion features. For this reason, we propose a Temporal Pyramid Discriminator (TPD) that leverages videos of different temporal resolutions in order to ensure high video quality in generation.

Principally, our discriminator is inspired from the two-stream architectures of MoCoGAN [148] and G^{3}AN [168]. We have a stream comprising an image discriminator D_I, as well as a stream incorporating the proposed TPD. While the input of D_I is a randomly sampled

frame, TPD accepts as input a full video sequence. TPD includes a set of video discriminators $\{D_{V_i} | i \in \mathbb{R}, 0 \leq i < n\}$, where each D_{V_i} is accountable for one temporal resolution.

Deviating from previous work [148, 168], we here propose to leverage 2D ConvNets in D_V rather than 3D ConvNets. We apply time to channel (TtoC) to concatenate sampled frames in channel dimension, in order to construct a video sampled at speed υ_i into an image $V_i^{'} \in \mathbb{R}^{H \times W \times K}$, where $\frac{K}{3}$ denotes the number of sampled frames. We surprisingly find that such design can substantially improve the visual quality, while ensuring temporal consistency of generated videos. We report experimental results in Section 5.3.3.

5.2.4 Learning

We use non-saturating loss [48] with $\mathscr{R}_1$ regularization [108, 73] as our objective function, in accordance to the setting of StyleGAN2 [73]. The loss of TPD, $\sum_{i=0}^{n-1} \mathscr{L}_{D_{V_i}}$, combines the losses of each video discriminator D_{V_i} in the pyramid. The network is optimized based on the full objective

$$\min_{G} \left(\lambda \sum_{i=0}^{n-1} \max_{D_{V_i}} \mathscr{L}_{D_{V_i}} + \max_{D_I} \mathscr{L}_{D_I} \right), \tag{5.5}$$

where n is a hyperparameter denoting the number of video discriminators to be used during training. We empirically identify appropriate n values in our two datasets, see Section 5.3.3. λ aims to balance the loss between D_I and TPD.

5.2.5 Implementation details.

We implement MintGAN using PyTorch [119]. All experiments are conducted on 8 V100 GPUs (32GB) with total batch size 32 (4 videos per GPU). We use Adam optimizer [78] with a learning rate 0.002 and set $\beta_1 = 0.0$, $\beta_2 = 0.99$. Dimensions of z_a and z_m are set to be 512 and 256, respectively. We pre-define $N = 512$ learnable directions in the motion dictionary, the dimension of each direction is set to be 512. λ is set to be 0.5 for all experiments.

5.3 Experiments and Analysis

We present extensive experiments, which include the following. In *video quality evaluation*, we quantitatively evaluate the ability of MintGAN to generate realistic videos and compare related results with four state-of-the-art methods for unconditional video generation. We then analyze the effectiveness of the proposed TPD. In addition, we provide an ablation study, which indicates the appropriate number of temporal resolutions for different datasets.

In *interpretability evaluation*, we aim to discover interpretable directions in the motion dictionary. Towards this, we propose a new evaluation framework that quantifies motion in generated videos using optical flow. We show that directions in the motion dictionary, based on our proposed framework, are indeed semantically meaningful. Further, we demonstrate that generated videos can be easily modified by manipulating such directions. Notably, our model allows for controllable video generation based on pre-defined trajectories for different directions.

Finally, we conduct analysis of *linear interpolation*, *high-resolution video generation* and go beyond training data to explore *longer video generation*.

5.3.1 Datasets

VoxCeleb2-mini dataset. We construct a subset of VoxCeleb2, where we randomly select 2 diverse videos per each of the 6,000 subjects. We note that videos include large appearance diversity.

BAIR robot pushing dataset. The dataset incorporates stationary videos of a robot arm moving and pushing a set of objects. We use the training set of this dataset which contains 40,000 short videos.

UCF101 dataset. The dataset contains 13,320 videos from Youtube of 101 human action classes.

We note that UCF101 is only used to evaluate high-resolution video generation.

5.3.2 Evaluation metric

For VoxCeleb2-mini and BAIR-robot, We use video FID [54] to quantitatively evaluate visual quality and temporal consistency in generated videos.We appropriate ResNeXt-101 [51] pre-trained on Kinetics-400 [16] as feature extractor and use outputs from last fully connected layer to compute the FID. We randomly sample 10,000 videos to compute the values for each experiment. To evaluate results on UCF101, we follow the evaluation protocol introduced in TGANv2 [131].

5.3.3 Video quality evaluation

We firstly compare MintGAN with four state-of-the-art methods, namely VGAN, TGAN, MoCoGAN, as well as G^3AN. We generate videos pertained to named methods with spatial resolution of 64×64 and temporal length of 32 for VGAN and 16 for the other methods. Related FIDs are reported in Tab. 5.1. MintGAN systematically outperforms other methods w.r.t. video quality by obtaining the lowest FID on both datasets. This is a pertinent prerequisite for latent space interpretation, as only highly realistic videos would allow for a meaningful interpretation.

Method	**VoxCeleb2-mini**	**BAIR-robot**
VGAN [157]	38.13	147.23
TGAN [130]	23.05	120.22
MoCoGAN [148]	12.69	13.68
G^3AN [168]	3.32	1.58
MintGAN	**2.37**	**1.31**

Table 5.1 **Comparison of MintGAN with four state-of-the-art models.** MintGAN systematically and significantly outperforms other methods on both datasets w.r.t. FID. The lower FID, the better video quality.

Effectiveness of TPD. We replace the original 3D discriminators in VGAN, TGAN, MoCoGAN, as well as G^3AN with TPD, maintaining all training configurations as in the previous experiment. We report FIDs related to original and proposed discriminators in all algorithms and both datasets in Tab. 5.2. We observe that TPD improves the results of

all methods significantly and consistently. This confirms that videos sampled with a set of temporal resolutions contain different features, which are beneficial in the discriminator.

On a different but related note, we observe during training that models without image discriminator (VGAN and TGAN) tend to reach mode collapse rapidly on BAIR-robot (high FID in Tab. 5.2). This is rather surprising, as BAIR-robot constitutes the simpler of the two datasets, comprising videos of a robot arm moving, with a fixed background. The occurrence of very similar scenes might be the reason for the challenging distinguishing of real and fake spatial information in the absence of an image discriminator.

Method	**VoxCeleb2-mini**		**BAIR-robot**	
	3D	TPD	3D	TPD
VGAN [157]	38.13	16.33	147.23	93.71
TGAN [130]	23.05	21.24	120.22	120.04
MoCoGAN [148]	12.69	7.07	13.68	3.16
G^3AN [168]	3.32	2.98	1.58	1.50

Table 5.2 **Evaluation of TPD.** When replacing the initial 3D discriminator with TPD, the latter significantly and consistently improves the FID of all 4 state-of-art models for the VoxCeleb2-mini and BAIR-robot datasets.

In addition, we conduct an **ablation study**, seeking to determine the optimal number of temporal resolutions in TPD for both datasets. Associated results are reported in Tab. 5.3, which suggest that while for VoxCeleb2-mini, which contains complex motion, we achieve the lowest FID on four temporal resolutions (16, 8, 4, 2 frames), for BAIR-robot, which is simpler w.r.t. occurring motion, three resolutions (16, 8, 4 frames) suffice.

TPD type	**VoxCeleb2-mini**	**BAIR-robot**
$D_{V_0}, D_{V_1}, D_{V_2}, D_{V_3}$	**2.37**	1.56
$D_{V_0}, D_{V_1}, D_{V_2},$	2.65	**1.31**
D_{V_0}, D_{V_1}	2.76	1.33
D_{V_0}	2.84	1.58

Table 5.3 **Ablation study on video discriminators in TPD.** Number of video discriminators associated to temporal resolutions. FID is reported for comparison. Lower FID indicates a superior quality of generated videos.

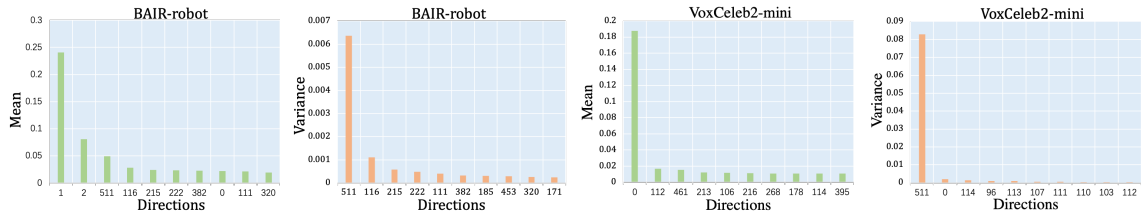


Figure 5.3 **Analysis of α.** Mean and variance bar charts, indicating top 10 motion-directions with the highest values in $A_{\bar{t}}$.

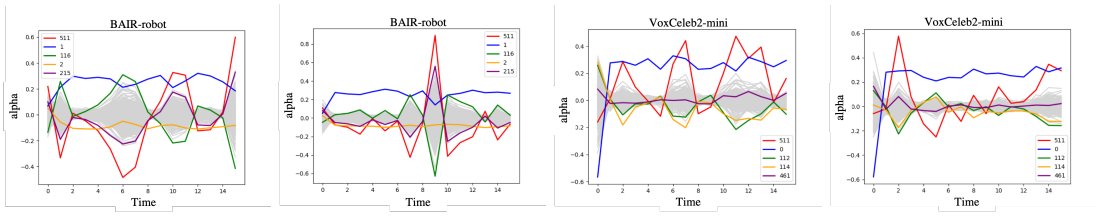


Figure 5.4 **Time v.s. α.** Each figure represents a video sample. We illustrate one sample from BAIR-robot (left) and one from VoxCeleb2-mini (right), respectively. Top 5 dimensions in α are plotted in different color.

5.3.4 Interpretability evaluation

Above, we have provided experimental proof that MintGAN is able to generate high quality videos. In this section, we focus on discussing, how to leverage those videos to find interpretable directions in the motion dictionary. Towards this, firstly we analyze α, seeking to find directions with the highest impact.

Then, we present our proposed evaluation framework to quantify motion, in order to find semantic meaning of such directions. Next, we show generated results based on manipulation of such directions. Finally, we demonstrate that our model allows for controllable generation by navigating in found interpretable directions in pre-defined trajectories.

Do all directions contribute equally? As per Equation (5.8), each $\alpha_{j,i}$ indicates the magnitude of d_i at time step j. We sample $10,000$ latent codes as evaluation set and compute mean and variance over time, for the full set, in order to obtain $A_{\bar{t}} = [\alpha_{\bar{t},0}, \alpha_{\bar{t},1}, \ldots, \alpha_{\bar{t},N-1}], \alpha_{\bar{t},i} \in \mathbb{R}$. Figure 5.3 shows mean and variance values of the 10 most pertinent dimensions in $A_{\bar{t}}$ for both datasets. We note that for both datasets, $\alpha_{\bar{t},511}$ has the largest variance, which indicates that d_{511} leads to the strongest motion variation in generated videos. At the same time,

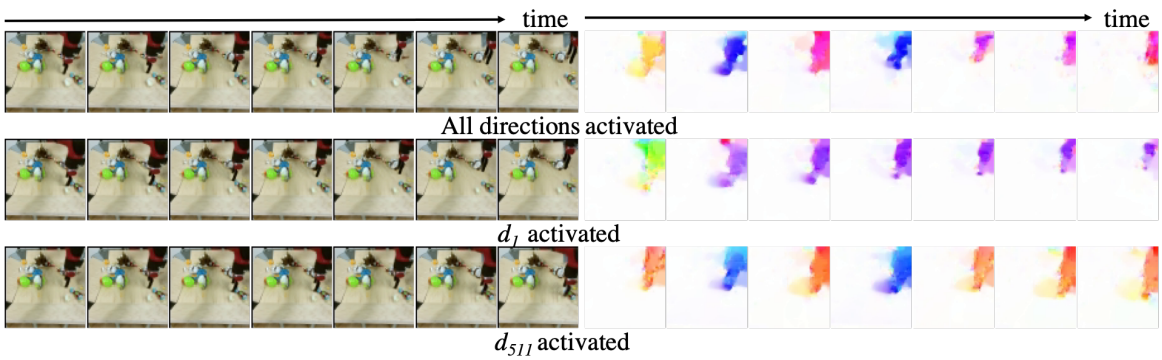


Figure 5.5 **Directions analysis on BAIR-robot.** A generated video sample, related optical flow images (top), activation of *only* d_1 (middle), and activation of *only* d_{511} (bottom). Optical flow images indicate that d_1 is accountable for moving the robot arm backward, whereas d_{511} for moving it left and right.

$\alpha_{\bar{t},1}$ (BAIR-robot) and $\alpha_{\bar{t},0}$ (VoxCeleb2-mini) encompass highest mean values, respectively. Therefore, we have that d_1 (BAIR-robot) and d_0 (VoxCeleb2-mini) show high and continuous magnitudes, respectively.

Moreover, we are interested in the course of each $\alpha_{j,i}$ over time, which we portray in Figure 5.4. Specifically, we randomly select one sample per dataset and highlight a set of $\alpha_{0:15,i}$ in different colors. We have that, while $\alpha_{0:15,511}$ (in red) has the largest amplitude in both datasets, $\alpha_{0:15,1}$ (BAIR-robot) and $\alpha_{0:15,0}$ (VoxCeleb2-mini) (in blue) maintain high but steady values over time, respectively. This supports our findings, as displayed in Figure 5.3. One explanation could be d_{511} corresponds to the largest singular value, so it should identify a dimension, which entails the largest variance, and hence its length impact mostly the space and is responsible to the strongest motion.

Based on the above, we conclude that directions in the motion dictionary *do not* contribute equally in composing motion. There exists directions which represent the major motions while others are complementary in the generated results.

Are motion components interpretable? We here aim to semantically quantify motion directions by a novel framework using optical flow. Firstly, we represent the optical flow according to the Middlebury evaluation [7]. Specifically, we partition the flow into four histogram bins, namely R_0, R_1, R_2 and R_3, to cover the 360° range of orientation and

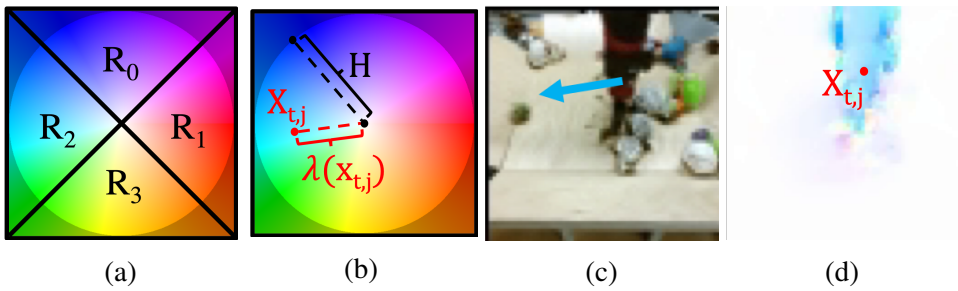


Figure 5.6 **Optical flow quantization.** (a) Middlebury colorwheel, (b) $\lambda(x_{t,j})$ and H on the colorwheel, (c) one frame from BAIR-robot and (d) related optical flow.

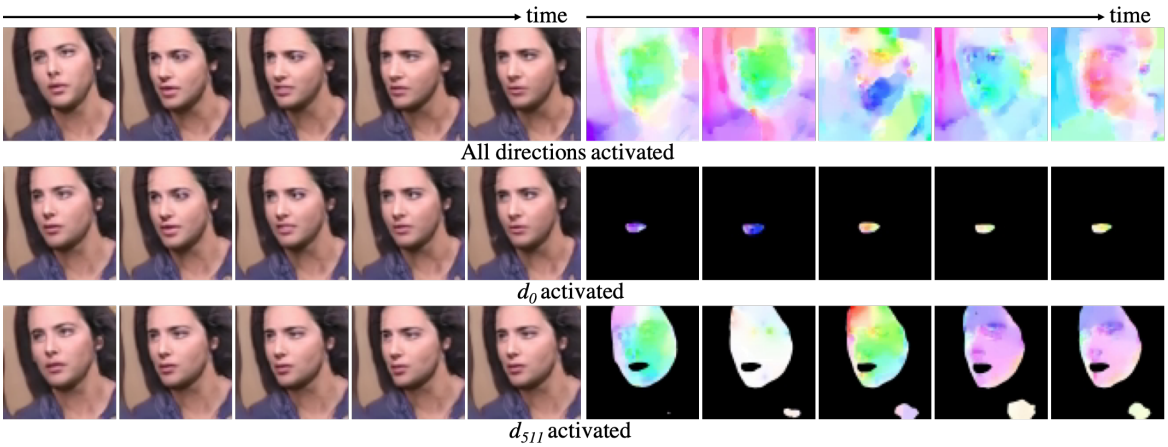


Figure 5.7 **Direction analysis in VoxCeleb2-mini.** A generated video sample and associated optical flow images (top), by *only* activating d_0 (middle), and by *only* activating d_{511} (bottom). While d_0 controls the mouth region, d_{511} controls the head region.

amplitude, see Figure 5.6. While different motion directions are represented by the hue values, motion magnitude is indicated by the brightness. Hence each R_i represents a motion range. Next, for any given optical flow video, we quantify motion in R_i as following.

$$\phi_i = \frac{1}{N_i} \sum_{t=0}^{T-1} \sum_{j=0}^{N-1} \frac{\lambda(x_{t,j})}{H} \mathbb{1}_{R_i}(x_{t,j}), i \in \{0,1,2,3\}, \tag{5.6}$$

with total motion in the video being computed as

$$\Phi = \frac{1}{N} \sum_{i=0}^{3} \sum_{t=0}^{T-1} \sum_{j=0}^{N-1} \frac{\lambda(x_{t,j})}{H} \mathbb{1}_{R_i}(x_{t,j}), \tag{5.7}$$

Figure 5.8 **Global and local motion extraction.** (a) Generated image, (b) related optical flow, (c) semantic map, (d) mouth-flow image, and (e) face-flow image based on training with VoxCeleb2-mini.

where $x_{t,j}$ denotes the value of the jth pixel at time step t in an optical flow video, which contains N color pixels in total. N_i denotes the total number of color pixels in R_i. $\lambda(x_{t,j})$ measures the distance from $x_{t,j}$ to the center of the colorwheel, whose radius is H (see Figure 5.6). A large ϕ_i indicates a frequent and strong motion appearing in R_i.

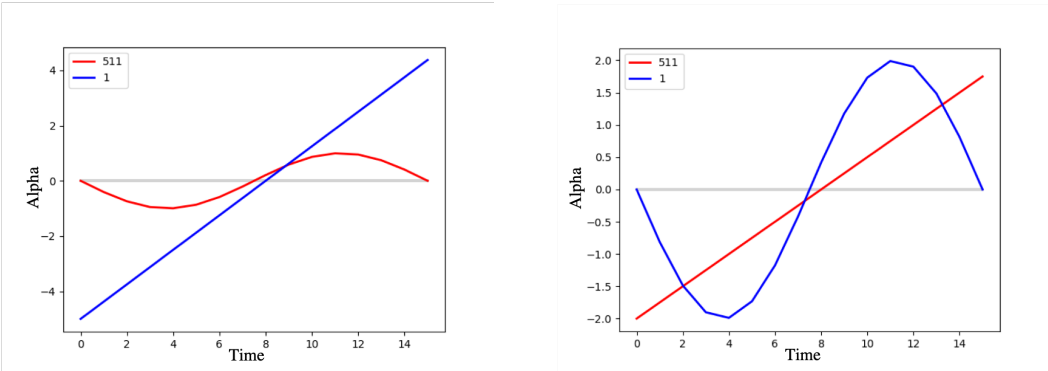
For BAIR-robot, we proceed to evaluate the set of directions d_1, d_2, d_{116} and d_{511}, as they exhibit the highest impact according to Figure 5.3. Our idea is to quantify the motion difference $\Delta\phi_i = \phi_i^{d_k} - \phi_i$ in each R_i, when d_k is deactivated (set $\alpha_k = 0$) in original videos.

We sample 1000 videos and deactivate each of the chosen directions, respectively, building an evaluation dataset containing 6000 samples (1000 original + 5000 deactivated). We report averaged ϕ_i over the full evaluation set for each region in Tab. 5.4. When d_1 is deactivated, motion in R_0 and R_3 are strongly reduced. Similarly for d_{511}, ϕ_1 and ϕ_2 obtain the largest decline. We note that for some directions motion changes are minor. As (R_0, R_3) and (R_1, R_2) are opposite regions, d_1 and d_{511} represent symmetric motions. To illustrate this, we generate samples by *only* activating d_1 and *only* activating d_{511}, respectively, while maintaining other directions deactivated. Figure 5.5 shows one sample and related optical flow, from which we deduce that the results match our quantitative evaluation, which suggested that d_1 represents 'robot arm moving back and forth', and d_{511} represents 'robot arm moving left and right'.

As we have already found interpretable directions, we show, by providing pre-defined trajectories to d_1 and d_{511}, that we are able to control generated videos. We show two types of α-trajectories over time for d_1 and d_{511} in Fig. 5.9a and Fig. 5.9b, respectively. While in Fig. 5.9a a *linear* trajectory is provided for d_1 and a *sinusoidal* trajectory for d_{511}, in Fig. 5.9b, d_1 and d_{511} are activated oppositely. We illustrate generated videos by activating d_1, d_{511}, as

	$\Delta\phi_0$	$\Delta\phi_1$	$\Delta\phi_2$	$\Delta\phi_3$
d_1	**-0.008**	0.017	0.002	**-0.033**
d_2	-0.001	0.002	0.002	-0.005
d_{116}	0.000	-0.001	0.001	0.000
d_{511}	0.007	**-0.087**	**-0.059**	0.019

Table 5.4 $\Delta\phi_i$ **on BAIR-robot.** Motion difference in four regions (R_0, R_1, R_2, R_3) caused by deactivating motion-directions.



(a) Trajectory type 1.

(b) Trajectory type 2.

Figure 5.9 **Two pre-defined trajectories.** (a) We provide a *linear* trajectory for d_1 and a *sinusoidal* trajectory for d_{511}. (b) We provide a *sinusoidal* trajectory for d_1 and a *linear* trajectory for d_{511}.

well as both directions, respectively, whereas all other directions maintain deactivated (set α to 0). Related results indicate that the robot arm can indeed be controlled directly with different trajectories. Generated results are illustrated on website [1].

VoxCeleb2-mini comprises a more complex dataset than BAIR-robot with videos containing concurrent global motion (e.g., head moving), as well as local motion (e.g., talking). For VoxCeleb2-mini we therefore analyze global and local motion by focusing on head and mouth regions, computing facial semantic maps, and further head-flow and mouth-flow videos for each sample (see Figure 5.8). We use the method of Yu *et al.* [195] to extract facial semantic maps.

For VoxCeleb2-mini we proceed to select the top 4 directions d_0, d_{112}, d_{114}, and d_{511} from Figure 5.3 and sample 1000 videos for evaluation. Deviating from above, we here quantify video motion changes in head $\Delta\Phi_{head}$ and mouth regions $\Delta\Phi_{mouth}$, respectively. Tab. 5.5

1. https://wyhsirius.github.io/InMoDeGAN/

shows that while deactivation of d_{511} triggers the largest motion decline in the head region, the deactivation of d_0 leads to the largest decline of mouth-motion. Considering that head movement contributes to mouth movement, we compute $\Delta\Phi_{mouth} - \Delta\Phi_{head}$, excluding global from local motion. However, d_0 still remains highest contribution to mouth motion. Similar to BAIR-robot, we illustrate samples by activating *only* d_0, and *only* d_{511}, respectively, in Figure 5.7. While d_0 reflects mouth motion, d_{511} represents head motion. This is conform to our quantitative evaluation.

	$\Delta\Phi_{head}$	$\Delta\Phi_{mouth}$	$\Delta\Phi_{mouth}$-$\Delta\Phi_{head}$
d_0	-0.012	**-0.052**	**-0.040**
d_{112}	-0.001	-0.005	-0.005
d_{114}	-0.000	-0.005	-0.005
d_{511}	**-0.036**	-0.027	0.008

Table 5.5 $\Delta\Phi_{head}$ **and** $\Delta\Phi_{mouth}$ **on VoxCeleb2-mini.** Motion difference in head and mouth regions induced by deactivation of motion-directions.

We also analyze the interpretability of other directions, despite the fact that related contribution is minor. In doing so, we find basic semantic motions such as zooming and rotation. Walks in such directions correspond to facial geometric transformations. We firstly conduct linear walks along each direction d_i

$$\begin{aligned} \alpha_t &= \alpha_{min} + t * \frac{(\alpha_{max} - \alpha_{min})}{T}, \; t \in [0, T-1], \\ w_{t,i} &= w_0 + \alpha_t \, d_i, \; i \in [0, 511], \\ x_{t,i} &= G_S(w_{t,i}), \end{aligned} \tag{5.8}$$

where we set α_{min} and α_{max} to be -5 and +5, respectively and we interpolate using $T = 16$ points. Then we construct videos $V_i = \{x_t\}_{t=0}^{15}$ for direction d_i. Towards analyzing the videos, we divide the color wheel into 8 regions (see Fig. 5.10) and use proposed method to compute motion differences in each region

$$\begin{aligned} \phi_i = \frac{1}{N_i} \sum_{t=0}^{T-1} \sum_{j=0}^{N-1} \frac{\lambda(x_{t,j})}{H} \mathbb{1}_{R_i}(x_{t,j}), \\ i \in \{0,1,2,3,4,5,6,7\}. \end{aligned} \tag{5.9}$$

Finally, we use *k-means* to group 512 directions into 16 different clusters based on their motion differences in each region. We observe that directions within one cluster contain similar semantic meanings. Generated results are illustrated on website [2].

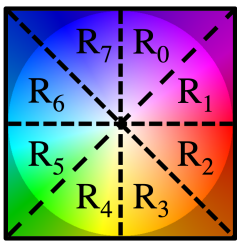


Figure 5.10 Eight-region color wheel

We conclude that directions in our motion dictionary are interpretable, whereas activation and manipulation of them enable control of motion in generated videos.

5.3.5 User study

We asked 30 human raters to evaluate generated video quality, as well as the interpretability. Towards evaluating quality, we show paired videos and ask the raters, to rate 'which clip is more realistic'. Each video-pair contains one generated video from our method, whereas the second video is either *real* or generated from other methods. As shown in Tab. 5.6, users rate that our method is more realistic than MoCoGAN and G^3AN across both datasets. It is encouraging that in the comparison of our method with real videos, users rated in 25% that our method was more realistic for the 64×64 setting.

Methods	**User Preference (%)**	
	VoxCeleb2-mini	BAIR-robot
ours / MoCoGAN	93.00 / 7.00	80.00 / 20.00
ours / G^3AN	85.33 / 14.67	61.33 / 38.67
ours / real (64×64)	25.00 / 75.00	48.67 / 51.33
ours / real (128×128)	16.00 / 84.00	-

Table 5.6 **User study**: Mean opinion score for the question 'Which video clip is more realistic?'

2. https://wyhsirius.github.io/InMoDeGAN/

Towards evaluating the interpretability of found motion-directions, we asked the question 'which direction is the robot arm moving?' for BAIR-robot dataset. Human raters were able to choose 'back and forth' and 'left and right'. For VoxCeleb-mini, the question posed was 'what moves the most?' and options were 'mouth' and 'head'. We compare the accordance of obtained results of user study and quantitative evaluation results. We report the accuracy of 'users/actions' in Tab. 5.7 and Tab. 5.8. Related results indicate that found directions correspond to the human evaluation.

Direction (BAIR-robot)	**Acc. (%)**
d_1 (back and forth)	100.00
d_{511} (left and right)	85.33

Table 5.7 **User study**: Mean opinion score for the question 'which direction is the robot arm moving?'.

Direction (VoxCeleb-mini)	**Acc. (%)**
d_0 (mouth)	92.22
d_{511} (head)	80.00

Table 5.8 **User study**: Mean opinion score for the question 'what moves the most?'.

5.3.6 Further analysis

We conduct more analysis of our model, including *linear interpolation*, *high-resolution video generation* and *longer video generation*. Generated results are illustrated on website [3].

High-resolution video generation. We evaluate our generated high-resolution (128×128) videos pertained to both, VoxCeleb2-mini and UCF101 [141]. We use the evaluation protocol introduced in the main paper for VoxCeleb2-mini. Results are reported in Tab. 5.9. Naturally, higher resolution corresponds to better (lower) FID.

Towards a fair comparison with state-of-the-art results on UCF101, we use the evaluation protocol introduced in TGANv2 [131]. It uses a C3D [146] that has been pre-trained on UCF101 as feature extractor. We report video results w.r.t. Inception Score (IS) and Fréchet

3. https://wyhsirius.github.io/InMoDeGAN/

Inception Distance (FID) in Tab. 5.10. Our method outperforms other methods using both evaluation metrics w.r.t. high-resolution video generation.

Method	FID (↓)
VGAN [157] (64×64)	38.13
TGAN [130] (64×64)	23.05
MoCoGAN [148] (64×64)	12.69
G^3AN [167] (64×64)	3.32
MintGAN (64×64)	2.37
MintGAN (128×128)	**0.25**

Table 5.9 **Comparison of MintGAN with four state-of-the-art models.** MintGAN systematically outperforms the other models on **VoxCeleb2-mini** w.r.t. FID.

Method	IS (↑)	FID (↓)
VGAN [157]	$8.31 \pm .09$	-
TGAN [130]	$11.85 \pm .07$	-
MoCoGAN [148]	$12.42 \pm .03$	-
ProgressiveVGAN [1]	$13.59 \pm .07$	-
TGANv2 [131]	$26.60 \pm .47$	3431 ± 19
MintGAN	$\mathbf{28.25 \pm .05}$	$\mathbf{3390 \pm 83}$

Table 5.10 **Comparison of MintGAN with five state-of-the-art models.** MintGAN systematically outperforms the other models on **UCF101** w.r.t. IS and FID. Numbers are adopted from [131], except MintGAN.

Chapter 6

Joint Generative and Contrastive Learning for Unsupervised Person ReID

This chapter presents our last contribution, while aims at learning from synthesis data to improve the performance of unsupervised person re-identification. Recent self-supervised contrastive learning provides an effective approach for unsupervised person re-identification (ReID) by learning invariance from different views (transformed versions) of an input. In this chapter, we incorporate a Generative Adversarial Network (GAN) and a contrastive learning module into one joint training framework. While the GAN provides online data augmentation for contrastive learning, the contrastive module learns view-invariant features for generation. In this context, we present a mesh-based view generator. Specifically, mesh projections serve as references towards generating novel views of a person. In addition, we present a view-invariant loss to facilitate contrastive learning between original and generated views. Deviating from previous GAN-based unsupervised ReID methods involving domain adaptation, we do not rely on a labeled source dataset, which makes our method more flexible. Extensive experimental results show that our method significantly outperforms state-of-the-art methods under both, fully unsupervised and unsupervised domain adaptive settings on several large scale ReID datsets.

6.1 Introduction

A person re-identification (ReID) system is targeted at identifying subjects across different camera views. In particular, given an image containing a person of interest (as query) and a large set of images (gallery set), a ReID system ranks gallery-images based on visual similarity with the query. Towards this, ReID systems are streamlined to bring to the fore discriminative representations, which allow for robust comparison of query and gallery images. In this context, *supervised* ReID methods [18, 142] learn representations guided by human-annotated labels, which is time-consuming and cumbersome. Towards omitting such human annotation, researchers increasingly place emphasis on *unsupervised* person ReID algorithms [167, 88, 94], which learn directly from unlabeled images and thus allow for scalability in real world deployments.

Recently, self-supervised contrastive methods [52, 21] have provided an effective retrieval-based approach for unsupervised representation learning. Given an image, such methods maximize agreement between two augmented views of one instance (see Fig. 6.1). *Views* refer to transformed versions of the same input. As shown in very recent works [21, 23], data augmentation enables a network to explore view-invariant features by providing augmented views of a person, which are instrumental in building robust representations. Such and similar methods considered traditional data augmentation techniques, e.g., 'random flipping', 'cropping', and 'color jittering'. Generative Adversarial Networks (GANs) [48] constitute a novel approach for data augmentation. As opposed to traditional data augmentation, GANs are able to modify id-unrelated features substantially, while preserving id-related features, which is highly beneficial in contrastive ReID.

Previous GAN-based methods [6, 32, 216, 90, 180, 210] considered unsupervised ReID as an unsupervised domain adaptation (UDA) problem. Under the UDA setting, researchers used both, a labeled source dataset, as well as an unlabeled target dataset to gradually adjust a model from a source domain into a target domain. GANs can be used in cross-domain style transfer, where labeled source domain images are generated in the style of a target domain. However, the UDA setting necessitates a large-scale labeled source dataset. Scale and quality of the source dataset strongly affect the performance of UDA methods. Recent research has

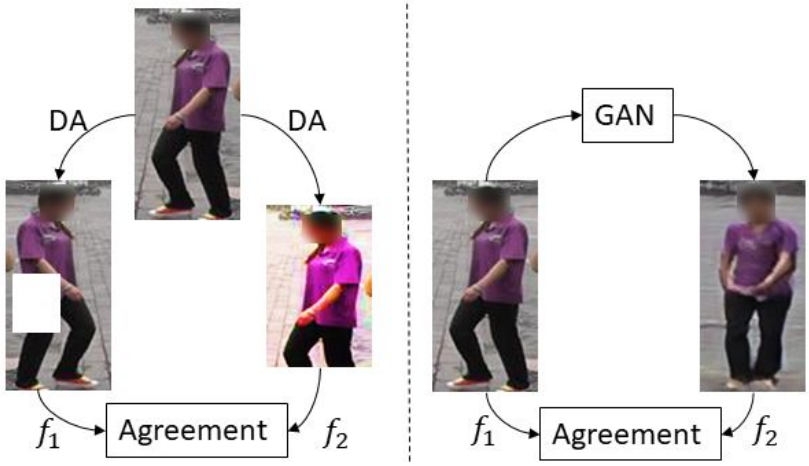


Figure 6.1 **Traditional v.s. our proposed method.** (Left) Traditional self-supervised contrastive learning maximizes agreement between representations (f_1 and f_2) of augmented views from Data Augmentation (DA). (Right) Joint generative and contrastive learning maximizes agreement between original and generated views.

considered fully unsupervised ReID [167, 88], where under the fully unsupervised setting, a model directly learns from unlabeled images without any identity labels. Self-supervised contrastive methods [52, 21] belong to this category. In this work, we use a GAN as a novel view generator for contrastive learning, which does not require a labeled source dataset.

Here, we aim at enhancing view diversity for contrastive learning via generation under the fully unsupervised setting. Towards this, we introduce a mesh-based novel view generator. We explore the possibility of disentangling a person image into identity features (color distribution and body shape) and structure features (pose and view-point) under the fully unsupervised ReID setting. We estimate 3D meshes from unlabeled training images, then rotate these 3D meshes to simulate new structures. Compared to skeleton-guided pose transfer [45, 90], which neglects body shape, mesh recovery [68] jointly estimates pose and body shape. Estimated meshes preserve body shape during the training, which facilitates the generation and provides more visual clues for fine-grained ReID. Novel views can be generated by combining identity features with new structures.

Once we obtain the novel views, we design a pseudo label based contrastive learning module. With the help of our proposed view-invariant loss, we maximize representation

similarity between original and generated views of a same person, whereas representation similarity of other persons is minimized.

Our proposed method incorporates generative and contrastive modules into one framework, which are trained jointly. Both modules share the same identity feature encoder. The generative module disentangles identity and structure features, then generates diversified novel views. The novel views are then used in the contrastive module to improve the capacity of the shared identity feature encoder, which in turn improves the generation quality. Both modules work in a mutual promotion way, which significantly enhances the performance of the shared identity feature encoder in unsupervised ReID. Moreover, our method is compatible with both UDA and fully unsupervised settings. With a labeled source dataset, we obtain better performance by alleviating the pseudo label noise.

In summary, this chapter makes several contributions. First, we propose a joint generative and contrastive learning framework for unsupervised person ReID. Generative and contrastive modules mutually promote each other's performance. Second, in the generative module, we introduce a 3D mesh based novel view generator, which is more effective in body shape preservation than skeleton-guided generators. Third, in the contrastive module, a view-invariant loss is proposed to reduce intra-class variation between original and generated images, which is beneficial in building view-invariant representations under a fully unsupervised ReID setting. In addition, we overcome the limitation of previous GAN-based unsupervised ReID methods that strongly rely on a labeled source dataset. Our method significantly surpasses the performance of state-of-the-art methods under both, fully unsupervised, as well as UDA settings.

6.2 Background

Unsupervised representation learning. Recent contrastive instance discrimination methods [185, 52, 21] have witnessed a significant progress in unsupervised representation learning. The basic idea of instance discrimination has to do with the assumption that each image is a single class. Contrastive predictive coding (CPC) [115] included an InfoNCE loss

to measure the ability of a model to classify positive representation amongst a set of unrelated negative samples, which has been commonly used in following works on contrastive learning. Recent contrastive methods treated unsupervised representation learning as a retrieval task. Representations can be learnt by matching augmented views of a same instance from a memory bank [185, 52] or a large mini-batch [21]. MoCoV2 [23] constitutes the improved version of the MoCo [52] method, incorporating larger data augmentation. We note that data augmentation is pertinent in allowing a model to learn robust representations in contrastive learning. However, only traditional data augmentation was used in aforementioned methods.

Data augmentation. MoCoV2 [23] used 'random crop', 'random color jittering', 'random horizontal flip', 'random grayscale' and 'gaussian blur'. However, 'random color jittering' and 'grayscale' were not suitable for fine-grained person ReID, because such methods for data augmentation tend to change the color distribution of original images. In addition, 'Random Erasing' [209] has been a commonly used technique in person ReID, which randomly erases a small patch from an original image. Cross-domain Mixup [102] interpolated source and target domain images, which alleviated the domain gap in UDA ReID. Recently, Generative Adversarial Networks (GANs) [48] have shown great success in image [73, 72, 13] and video synthesis [148, 168, 17, 172, 171]. GAN-based methods can serve as a method for evolved data augmentation by conditionally modifying id-unrelated features (style and structure) for supervised ReID. CamStyle [213] used the CycleGAN-architecture [215] in order to transfer images from one camera into the style of another camera. FD-GAN [45] was targeted to generate images in a pre-defined pose, so that images could be compared in the same pose. IS-GAN [38] was streamlined to disentangle id-related and id-unrelated features by switching both local and global level identity features. DG-Net [208] recolored grayscale images with a color distribution of other images, targeting to disentangle identity features. Deviating from such supervised GAN-based methods, our method generates novel views by rotating 3D meshes in an *unsupervised* manner.

Unsupervised person ReID. Recent unsupervised person ReID methods were predominantly based on UDA. Among UDA-based methods, several works [162, 93] used semantic

attributes to facilitate domain adaptation. Other works [184, 43, 19, 193, 44] assigned pseudo labels to unlabeled images and proceeded to learn representations with pseudo labels. Transferring source dataset images into the style of a target dataset represents another line of research. SPGAN [32] and PTGAN [180] used CycleGAN [215] as domain style transfer-backbone. HHL [210] aims at transferring cross-dataset camera styles. ECN [211, 212] exploited invariance from camera style transferred images for UDA ReID. CR-GAN [24] employed parsing-based masks to remove noisy backgrounds. PDA [90] included skeleton estimation to generate person images with different poses and cross-domain styles. DG-Net++ [216] jointly disentangled id-related/id-unrelated features and transferred domain styles. While the latter is related to our our method, we aim at training jointly a GAN-based online data augmentation, as well as a contrastive discrimination, which renders the labeled source dataset unnecessary, rather than transferring style.

Fully unsupervised methods do not require any identity labels. BUC [94] represented each image as a single class and gradually merged classes. In addition, TSSL [183] considered each tracklet as a single class to facilitate cluster merging. SoftSim [95] utilized similarity-based soft labels to alleviate label noise. MMCL [167] assigned multiple binary labels and trained a model in a multi-label classification way. JVTC and JVTC+ [88] added temporal information to refine visual similarity based pseudo labels. We note that all aforementioned fully unsupervised methods learn from pseudo labels. We show in this work that disentangling view-invariant identity features is possible in fully unsupervised ReID, which can be an add-on to boost the performance of previous pseudo label based methods.

6.3 Approach

We refer to our proposed method as joint *Generative and Contrastive Learning* as GCL. The general architecture of GCL comprises of two modules, namely a View Generator, as well as a View Contrast Module, see Fig. 6.2. Firstly, the View Generator uses cycle-consistency on both, image and feature reconstructions in order to disentangle identity and structure features. It combines identity features and mesh-guided structure features to generate one

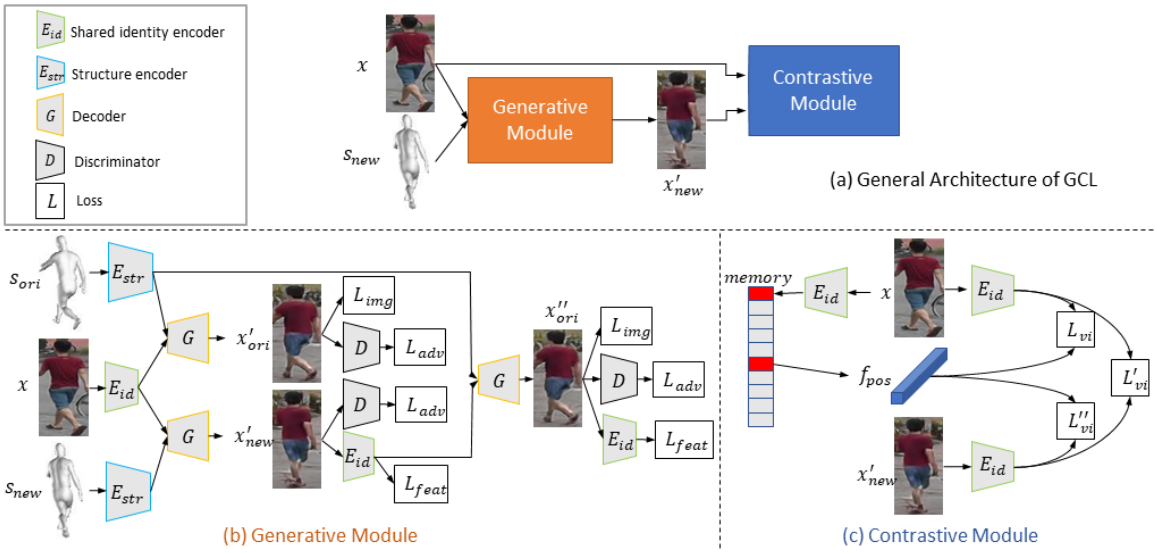


Figure 6.2 **A schematic overview of GCL.** (a) General architecture of GCL: Generative and contrastive modules are coupled by the shared identity encoder E_{id}. (b) Generative module: The decoder G combines the identity features encoded by E_{id} and structure features E_{str} to generate a novel view x'_{new} with a cycle consistency. (c) Contrastive module: View-invariance is enhanced by maximizing the agreement between original $E_{id}(x)$, synthesized $E_{id}(x'_{new})$ and memory f_{pos} representations.

person in new view-points. Then, original and generated views are exploited as positive pairs in the View Contrast Module, which enables our network to learn view-invariant identity features. We proceed to elaborate on both modules in the following.

6.3.1 View Generator (Generative Module)

As shown in Fig. 6.2, the proposed View Generator incorporates 4 networks: an identity encoder E_{id}, a structure encoder E_{str}, a decoder G and an image discriminator D. Given an unlabeled person ReID dataset $\mathscr{X} = \{x_1, x_2, ..., x_N\}$, we generate corresponding 3D meshes with a popular 3D mesh generator Human Mesh Recovery (HMR) [68], which simultaneously estimates body shape and pose from a single RGB image. Here, we denote the 2D projection of a 3D mesh as original structure s_{ori}. Then, as depicted in Fig. 6.3, we rotate each 3D mesh by $45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ$ and 315°, respectively and proceed to randomly pick one 2D projection of these rotated meshes as a new structure s_{new}. We use the 3D mesh rotation to mimic view-point variance from different cameras. Next, unlabeled

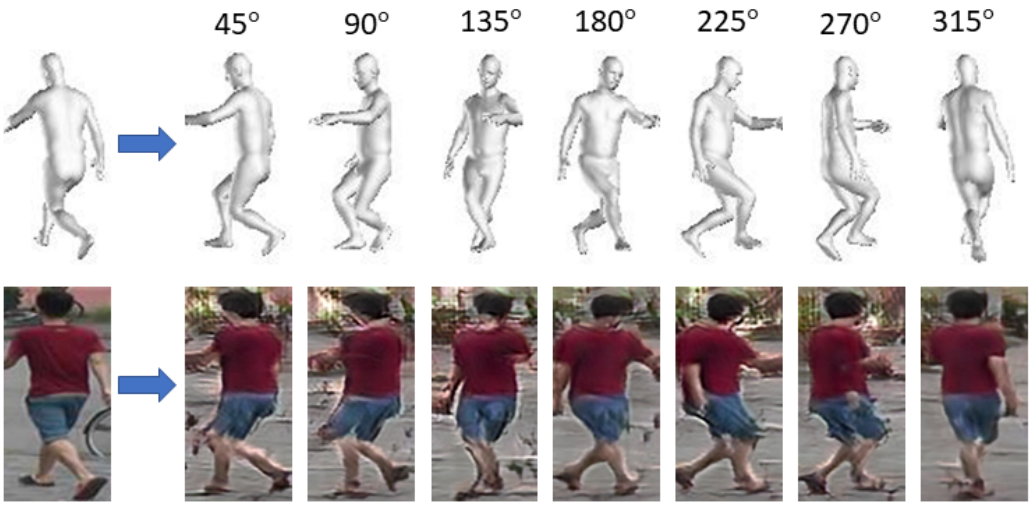


Figure 6.3 **Generated multi-view images.** Example images as generated by the View Generator via 3D mesh rotation based on left input image.

images are encoded to identity features by the identity encoder $E_{id} : x \rightarrow f_{id}$, whereas both original and new structures are encoded to structure features by the structure encoder $E_{str} : s_{ori} \rightarrow f_{str(ori)}, s_{new} \rightarrow f_{str(new)}$. Combining both, identity and structure features, the decoder generates synthesized images $G : (f_{id}, f_{str(ori)}) \rightarrow x'_{ori}, (f_{id}, f_{str(new)}) \rightarrow x'_{new}$, where a prime is used to represent generated images.

Given the lack of real images corresponding to the new structures, we consider a cycle consistency [215] to reconstruct the original image by swapping the structure features in the View Generator. We encode and decode once again to get synthesized images in original structures $G(E_{id}(x'_{new}), s_{ori}) \rightarrow x''_{ori}$. We calculate an image reconstruction loss as follows.

$$\mathscr{L}_{img} = \mathbb{E}[\|x - x'_{ori}\|_1] + \mathbb{E}[\|x - x''_{ori}\|_1] \tag{6.1}$$

In addition, we compute a feature reconstruction loss

$$\mathscr{L}_{feat} = \mathbb{E}[\|f_{id} - E_{id}(x'_{new})\|_1] + \mathbb{E}[\|f_{id} - E_{id}(x''_{ori})\|_1]. \tag{6.2}$$

The discriminator D attempts to distinguish between real and generated images with the adversarial loss

$$\begin{aligned}\mathscr{L}_{adv} = &\mathbb{E}[\log D(x) + \log(1 - D(x'_{ori}))] + \\ &\mathbb{E}[\log D(x) + \log(1 - D(x'_{new}))] + \\ &\mathbb{E}[\log D(x) + \log(1 - D(x''_{ori}))].\end{aligned} \tag{6.3}$$

Consequently, the overall GAN loss combines the above named losses with weighting coefficients λ_{img} and λ_{feat}

$$\mathscr{L}_{gan} = \lambda_{img}\mathscr{L}_{img} + \lambda_{feat}\mathscr{L}_{feat} + \mathscr{L}_{adv}. \tag{6.4}$$

6.3.2 View Contrast (Contrastive Module)

The previous reconstruction and adversarial losses work in an unconditional manner. They only explore identity features within the original view-point, which renders appearance representations view-variant. In rotating an original mesh to a different view-point, e.g., from front to side view-point, the generation is prone to fail due to lack of information pertained to the side view. This issue can be alleviated by enhancing the view-invariance of representations.

Given an anchor image x, the first step is to find positive images that belong to the same identity and negative images that belong to different identities. Here, we store all instance representations in a memory bank [185], which stabilizes pseudo labels and enlarges the number of negatives during the training with mini-batches. The memory bank $\mathscr{M}$ is updated with a momentum coefficient α.

$$\mathscr{M}[i]^t = \alpha \cdot \mathscr{M}[i]^{t-1} + (1 - \alpha) \cdot f^t \tag{6.5}$$

where $\mathscr{M}[i]^t$ and $\mathscr{M}[i]^{t-1}$ respectively refer to the identity feature vector in the t and $t-1$ epochs.

We use a clustering algorithm DBSCAN [39] on all memory bank feature vectors to generate pseudo identity labels $\mathscr{Y} = \{y_1, y_2, ..., y_J\}$, which are renewed at the beginning of every epoch.

Given the obtained pseudo labels, we have N_{pos} positive and N_{neg} negative instances for each training instance. N_{pos} and N_{neg} vary for different instances. For simplicity in a mini-batch training, we fix common positive and negative numbers for every training instance. Given an image x, we randomly sample K instances that have different pseudo identities and one instance representation f_{pos} that has the same pseudo identity with x from the memory bank. Note that f_{pos} is from a random positive image that usually has a pose and camera style different from x and x'_{new}. x and x'_{new} are encoded by E_{id} into identity feature vectors f and f'_{new}. Next, f, f'_{new} and f_{pos} are used in turn to form three positive pairs. The f'_{new} and K different identity instances in the memory bank are used as K negative pairs. Towards learning robust view-invariant representations, we extend the InfoNCE loss [115] into a view-invariant loss between original and generated views. We use $sim(u,v) = \frac{u}{\|u\|_2} \cdot \frac{v}{\|v\|_2}$ to denote the cosine similarity. We define the view-invariant loss as a softmax log loss of $K+1$ pairs as following.

$$\mathscr{L}_{vi} = \mathbb{E}[\log(1 + \frac{\sum_{i=1}^{K} \exp(sim(f'_{new}, k_i)/\tau)}{\exp(sim(f, f_{pos})/\tau)})] \tag{6.6}$$

$$\mathscr{L}'_{vi} = \mathbb{E}[\log(1 + \frac{\sum_{i=1}^{K} \exp(sim(f'_{new}, k_i)/\tau)}{\exp(sim(f'_{new}, f)/\tau)})] \tag{6.7}$$

$$\mathscr{L}''_{vi} = \mathbb{E}[\log(1 + \frac{\sum_{i=1}^{K} \exp(sim(f'_{new}, k_i)/\tau)}{\exp(sim(f'_{new}, f_{pos})/\tau)})], \tag{6.8}$$

where τ indicates a temperature coefficient that controls the scale of calculated similarities. $\mathscr{L}_{vi}$ maximizes the invariance between original and memory positive views. $\mathscr{L}'_{vi}$ maximizes the invariance between synthesized and original views. $\mathscr{L}''_{vi}$ maximizes the invariance between synthesized and memory positive views. Meanwhile, the synthesized view is pushed away from K negative views in the latent space. Replacing $sim(f'_{new}, k_i)$ in Equation (6.6), Equation 6.7 and Equation (6.8) with $sim(f, k_i)$ is another possibility, which pushes away the original view from negative instances. After testing, $sim(f'_{new}, k_i)$ works better, because

pushing away the synthesized view from negative instances aid the generation of more accurate synthesized views that look different from the K negative instances.

6.3.3 Joint Training

Our proposed GCL framework is trained in a joint training way. Both GAN and contrastive instance discrimination can be trained in a self-supervised manner. While the GAN learns a data distribution via adversarial learning on each instance, contrastive instance discrimination learns representations by retrieving each instance from candidates. In our designed joint training, the two modules work as two collaborators with the same objective: enhancing the quality of representations built by the shared identity encoder E_{id}. We formulate our GCL as an approach to augment contrast for unsupervised ReID. Firstly, the generative module generates online data augmentation, which enhances the positive view diversity for contrastive module. Secondly, the contrastive module, in turn, learns view-invariant representations by matching original and generated views, which refine the generation quality. The joint training boosts both modules simultaneously. Our joint training conducts forward propagation initially on the generative module and subsequently on the contrastive module. Back-propagation is then conducted with an overall loss that combines Equation (6.4), Equation (6.6), Equation (6.7) and Equation (6.8),

$$\mathscr{L}_{all} = \mathscr{L}_{gan} + \mathscr{L}_{vi} + \mathscr{L}'_{vi} + \mathscr{L}''_{vi} \tag{6.9}$$

To accelerate the training process and alleviate the noise from imperfect generation quality at beginning epochs, we need to warm up the four modules used in the View Generator E_{id}, E_{str}, G and D. We firstly use a state-of-the-art unsupervised ReID method to warm up E_{id}, which is then considered as a baseline in our ablation studies. Generally speaking, any unsupervised ReID method can be used to warm up E_{id}. Before conducting the View Contrast, we freeze E_{id} and warm up E_{str}, G, and D only with GAN loss in Equation (6.4) for 40 epochs. In the following, we bring in the memory bank and the pseudo labels to jointly train the whole

framework with $\mathscr{L}_{all}$ for another 20 epochs. During the joint training, pseudo labels are updated at the beginning of every epoch.

6.4 Experiments

6.4.1 Datasets and Evaluation Protocols

Three mainstream person ReID datasets are considered in our experiments, including Market-1501 [207], DukeMTMC-reID [125] and MSMT17 [180]. Market-1501 is composed of 12,936 images of 751 identities for training and 19,732 images of 750 identities for test captured from 6 cameras. DukeMTMC-reID contains 16,522 images of 702 persons for training, 2,228 query images and 17,661 gallery images of 702 persons for test from 8 cameras. MSMT17 is a larger dataset, which contains 32,621 training images of 1,041 identities and 93,820 testing images of 3,060 identities collected from 15 cameras.

Following state-of-the-art unsupervised ReID methods [167, 88], we evaluate our proposed method GCL under fully unsupervised setting on the three datasets and under four UDA benchmark protocols, including Market→Duke, Duke→Market, Market→MSMT and Duke→MSMT. We report both quantitative and qualitative results for unsupervised person ReID and view generation.

6.4.2 Implementation Details

We firstly present network design details of E_{id}, E_{str}, G and D. In the following descriptions, we write the size of feature maps in channel×height×width. Our model design is mainly inspired by [208, 216]. (1) E_{id} is a ImageNet [128] pre-trained ResNet50 [53] with slight modifications. The original fully connected layer is replaced by a fully connected embedding layer, which outputs identity representations f in $512\times1\times1$ for the View Contrast. In parallel, we add a part average pooling that outputs identity features f_{id} in $2048\times4\times1$ for the View Generator. (2) E_{str} is composed of four convolutional and four residual layers, which output structure features f_{str} in $128\times64\times32$. (3) G contains four residual and four

Method	Reference	Market1501					DukeMTMC-reID				
		Source	mAP	Rank1	Rank5	Rank10	Source	mAP	Rank1	Rank5	Rank10
BUC [94]	AAAI'19	None	29.6	61.9	73.5	78.2	None	22.1	40.4	52.5	58.2
SoftSim [95]	CVPR'20	None	37.8	71.7	83.8	87.4	None	28.6	52.5	63.5	68.9
TSSL [183]	AAAI'20	None	43.3	71.2	-	-	None	38.5	62.2	-	-
MMCL [167]	CVPR'20	None	45.5	80.3	89.4	92.3	None	40.2	65.2	75.9	80.0
JVTC [88]	ECCV'20	None	41.8	72.9	84.2	88.7	None	42.2	67.6	78.0	81.6
JVTC+ [88]	ECCV'20	None	47.5	79.5	89.2	91.9	None	50.7	74.6	82.9	85.3
MMCL*	This paper	None	45.1	79.5	89.0	91.9	None	40.9	64.8	75.2	79.8
JVTC*	This paper	None	47.2	75.4	86.7	90.5	None	43.9	66.8	77.6	81.0
JVTC+*	This paper	None	50.9	79.1	89.8	92.9	None	52.8	74.9	83.3	85.8
ours(MMCL*)	This paper	None	54.9	83.7	91.6	94.0	None	49.3	69.7	79.7	82.8
ours(JVTC*)	This paper	None	63.4	83.7	91.6	94.3	None	53.3	72.4	82.0	84.9
ours(JVTC+*)	This paper	None	**66.8**	**87.3**	**93.5**	**95.5**	None	**62.8**	**82.9**	**87.1**	**88.5**
ECN [211]	CVPR'19	Duke	43.0	75.1	87.6	91.6	Market	40.4	63.3	75.8	80.4
PDA [90]	ICCV'19	Duke	47.6	75.2	86.3	90.2	Market	45.1	63.2	77.0	82.5
CR-GAN [24]	ICCV'19	Duke	54.0	77.7	89.7	92.7	Market	48.6	68.9	80.2	84.7
SSG [43]	ICCV'19	Duke	58.3	80.0	90.0	92.4	Market	53.4	73.0	80.6	83.2
MMCL [167]	CVPR'20	Duke	60.4	84.4	92.8	95.0	Market	51.4	72.4	82.9	85.0
ACT [193]	AAAI'20	Duke	60.6	80.5	-	-	Market	54.5	72.4	-	-
DG-Net++ [216]	ECCV'20	Duke	61.7	82.1	90.2	92.7	Market	63.8	78.9	87.8	90.4
JVTC [88]	ECCV'20	Duke	61.1	83.8	93.0	95.2	Market	56.2	75.0	85.1	88.2
ECN+ [212]	PAMI'20	Duke	63.8	84.1	92.8	95.4	Market	54.4	74.0	83.7	87.4
JVTC+ [88]	ECCV'20	Duke	67.2	86.8	95.2	97.1	Market	66.5	80.4	**89.9**	92.2
MMT [44]	ICLR'20	Duke	71.2	87.7	94.9	96.9	Market	65.1	78.0	88.8	**92.5**
CAIL [102]	ECCV'20	Duke	71.5	88.1	94.4	96.2	Market	65.2	79.5	88.3	91.4
ACT*	This paper	Duke	59.1	78.8	88.9	91.7	Market	51.5	70.9	80.0	83.4
JVTC*	This paper	Duke	65.0	85.7	93.6	95.6	Market	56.5	73.9	84.5	87.7
JVTC+*	This paper	Duke	67.6	87.0	95.2	97.0	Market	66.7	81.0	**89.9**	91.5
ours(ACT*)	This paper	Duke	66.7	83.9	91.4	93.4	Market	55.4	71.9	81.6	84.6
ours(JVTC*)	This paper	Duke	73.4	89.1	95.0	96.6	Market	60.4	77.2	86.2	88.4
ours(JVTC+*)	This paper	Duke	**75.4**	**90.5**	**96.2**	**97.1**	Market	**67.6**	**81.9**	88.9	90.6

Table 6.1 **Comparison of unsupervised ReID methods (%) with a ResNet50 backbone on Market and Duke datasets.** We test our proposed method on several baselines, whose names are in brackets. * refers to our implementation based on authors' code.

Method	Reference	MSMT17				
		Source	mAP	R1	R5	R10
MMCL [167]	CVPR'20	None	11.2	35.4	44.8	49.8
JVTC [88]	ECCV'20	None	15.1	39.0	50.9	56.8
JVTC+ [88]	ECCV'20	None	17.3	43.1	53.8	59.4
JVTC*	This paper	None	13.4	36.0	48.8	54.9
JVTC+*	This paper	None	16.3	40.4	55.6	61.8
ours(JVTC*)	This paper	None	18.0	41.6	53.2	58.4
ours(JVTC+*)	This paper	None	**21.3**	**45.7**	**58.6**	**64.5**
ECN [211]	CVPR'19	Market	8.5	25.3	36.3	42.1
SSG [43]	ICCV'19	Market	13.2	31.6	49.6	-
MMCL [167]	CVPR'20	Market	15.1	40.8	51.8	56.7
ECN+ [212]	PAMI'20	Market	15.2	40.4	53.1	58.7
JVTC [88]	ECCV'20	Market	19.0	42.1	53.4	58.9
DG-Net++ [216]	ECCV'20	Market	22.1	48.4	60.9	66.1
CAIL [102]	ECCV'20	Market	20.4	43.7	56.1	61.9
MMT [44]	ICLR'20	Market	22.9	49.2	63.1	68.8
JVTC+ [88]	ECCV'20	Market	25.1	48.6	65.3	68.2
JVTC*	This paper	Market	17.1	39.6	53.3	59.3
JVTC+*	This paper	Market	20.5	44.0	59.5	71.1
ours(JVTC*)	This paper	Market	21.5	45.0	57.1	66.5
ours(JVTC+*)	This paper	Market	**27.0**	**51.1**	**63.9**	**69.9**
ECN [211]	CVPR'19	Duke	10.2	30.2	41.5	46.8
SSG [43]	ICCV'19	Duke	13.3	32.2	51.2	-
MMCL [167]	CVPR'20	Duke	16.2	43.6	54.3	58.9
ECN+ [212]	PAMI'20	Duke	16.0	42.5	55.9	61.5
JVTC [88]	ECCV'20	Duke	20.3	45.4	58.4	64.3
DG-Net++ [216]	ECCV'20	Duke	22.1	48.8	60.9	65.9
MMT [44]	ICLR'20	Duke	23.3	50.1	63.9	69.8
CAIL [102]	ECCV'20	Duke	24.3	51.7	64.0	68.9
JVTC+ [88]	ECCV'20	Duke	27.5	52.9	70.5	75.9
JVTC*	This paper	Duke	19.9	45.4	59.1	64.9
JVTC+*	This paper	Duke	23.6	49.4	65.2	71.1
ours(JVTC*)	This paper	Duke	24.9	50.8	63.4	68.9
ours(JVTC+*)	This paper	Duke	**29.7**	**54.4**	**68.2**	**74.2**

Table 6.2 **Comparison of unsupervised ReID methods (%) with a ResNet50 backbone on MSMT17.** * refers to our implementation based on authors' code.

convolutional layers. Every residual layer contains two adaptive instance normalization layers [60] that transform f_{id} into scale and bias parameters. (4) D is a multi-scale PatchGAN [65] discriminator at 64×32, 128×64 and 256×128.

Then, we present the training and testing configuration details. Our framework is implemented in Pytorch and trained with one Nvidia Titan RTX GPU. (1) For the E_{id} warm-up, we consider JVTC [88], because it is a state-of-the-art ReID method that is compatible with both fully unsupervised and UDA settings. We also test other baselines, e.g., MMCL [167] and ACT [193] to demonstrate the generalizability of our method. (2) For training, inputs are resized to 256×128. We empirically set a large weight $\lambda_{img} = \lambda_{feat} = 5$ for reconstruction in Equation (6.4). With a batch size of 16, we use SGD to train E_{id} and Adam optimizer to train E_{str}, G and D. Learning rate is set to 1×10^{-4} during the warm-up. In the joint-training, learning rate in Adam is set to 1×10^{-4} and 3.5×10^{-4} in SGD and are multiplied by 0.1 after 10 epochs. (3) In the View Contrast module, we set the momentum coefficient $\alpha = 0.2$ in Equation (6.5) and the temperature $\tau = 0.04$ in Equation (6.6). The number of negatives K is 8192. DBSCAN density radius is set to 2×10^{-3}. (4) For testing, only E_{id} is conserved and outputs representations f of dimension 512.

Important parameters are set by a grid search on the fully unsupervised Market-1501 benchmark. The temperature τ is searched from $\{0.03, 0.04, 0.05, 0.06, 0.07\}$ and finally is set to 0.04. A smaller τ increases the scale of similarity scores in the Equation (6.6), Equation (6.7) and Equation (6.8), which makes view-invariant losses more sensitive to inter-instance difference. However, when τ is set to 0.03, these losses become too sensitive and make the training unstable. The number of negatives K is searched from $\{2048, 4096, 8192\}$. A larger K pushes away more negatives in the view-invariant losses. Since the Market-1501 dataset has only 12936 training images, we set $K = 8192$.

6.4.3 Unsupervised ReID Evaluation

Comparison with state-of-the-art methods. Table 6.1 shows the quantitative results on the Market-1501 and DukeMTMC-reID datasets. Table 6.2 shows the quantitative results on the MSMT17 dataset. Our method is mainly designed for fully unsupervised ReID. Under

this setting, we test the performance of GCL with three different baselines, including MMCL, JVTC and JVTC+. Our implementation of the three baselines provides results that are slightly different from those mentioned in the corresponding papers. Thus, we firstly report results of our implementations and then add our GCL on these baselines. Our method improves the performance of the baselines by large margins. These improvements show that GANs are not limited to cross-domain style transfer for unsupervised ReID.

Under the UDA setting, we also evaluate the performance of GCL with three different baselines, including ACT, JVTC and JVTC+. The labeled source dataset is only used to warm up our identity encoder E_{id}, but not used in our joint generative and contrastive training. Compared to fully unsupervised methods, the UDA warmed E_{id} is stronger and extracts improved identity features. Thus, the performance of UDA methods is generally higher than fully unsupervised methods. With a strong baseline JVTC+, our GCL achieves state-of-the-art performance.

Ablation Study. To better understand the contribution of generative and contrastive modules, we conduct ablation experiments on the two fully unsupervised benchmarks: Market-1501 and DukeMTMC-reID. Quantitative results with a JVTC baseline are reported in Table 6.3. By gradually adding loss functions on the baseline, our ablation experiments correspond to three scenarios. (1) Only Generation: with only $\mathscr{L}_{gan}$, our generation module disentangles identity and structure features. Since there is no inter-view constraint, E_{id} tends to extract view-specific identity features, which decreases the ReID performance. (2) Only Contrast: we use $\mathscr{L}_{vi}^{woGAN} = \mathbb{E}[\log(1 + \frac{\sum_{i=1}^{K} \exp(sim(f,k_i)/\tau)}{\exp(sim(f,f_{pos})/\tau)})]$ to train our contrastive module without generation. We also add a set of traditional data augmentation, including random flipping, cropping, jittering, erasing, to train our contrastive module like a traditional memory bank based contrastive method. (3) Joint Generation and Contrast: $\mathscr{L}_{vi}$, $\mathscr{L}'_{vi}$ and $\mathscr{L}''_{vi}$ enhance the view-invariance of identity representations between original, synthesized and memory-stored positive views, while negative views are pushed away.

We also conduct a qualitative ablation study, where synthesized novel views without and with view-invariant losses are illustrated in Fig. 6.4. Results confirm that E_{id} extracts

Loss	Market-1501		DukeMTMC-reID	
	mAP	Rank1	mAP	Rank1
Baseline	47.2	75.4	43.9	66.8
$+\mathscr{L}_{gan}$	41.6	69.0	25.8	45.9
$+\mathscr{L}_{vi}^{woGAN}$	47.8	75.2	44.1	67.8
$+\mathscr{L}_{vi}^{woGAN}+TDA$	53.7	78.7	48.5	70.0
$+\mathscr{L}_{gan}+\mathscr{L}_{vi}$	54.1	79.4	47.4	68.4
$+\mathscr{L}_{gan}+\mathscr{L}_{vi}+\mathscr{L}_{vi}'$	59.2	82.2	50.5	71.0
$+\mathscr{L}_{gan}+\mathscr{L}_{vi}+\mathscr{L}_{vi}'+\mathscr{L}_{vi}''$	**63.4**	**83.7**	**53.3**	**72.4**

Table 6.3 **Ablation study on loss functions used in two modules.** (1). $\mathscr{L}_{gan}$ corresponds to generation w/o contrast. (2). $\mathscr{L}_{vi}^{woGAN}$ corresponds to contrast w/o generation. TDA denotes traditional data augmentation. (3). $\mathscr{L}_{gan}+\mathscr{L}_{vi}$ ($\mathscr{L}_{vi}'$ and $\mathscr{L}_{vi}''$) correspond to joint generative and contrastive learning.

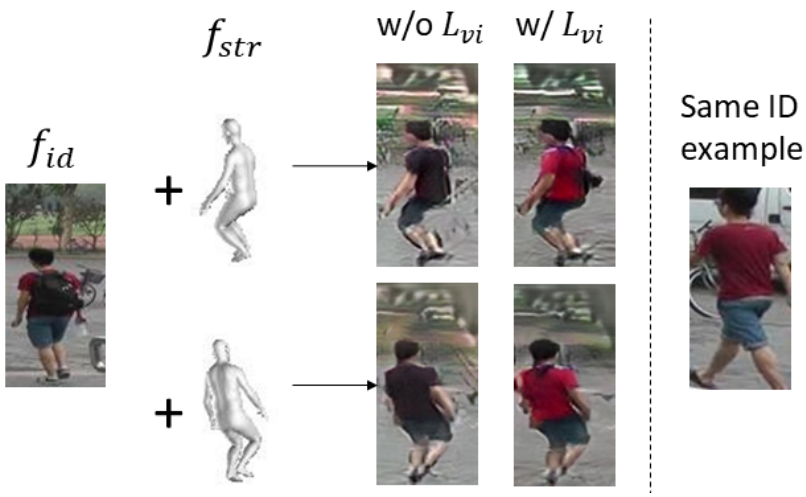


Figure 6.4 **Qualitative ablation study on the view-invariant losses.** For simplicity, $\mathscr{L}_{vi}$ denotes three view-invariant losses $\mathscr{L}_{vi}+\mathscr{L}_{vi}'+\mathscr{L}_{vi}''$, which helps E_{id} to extract view-invariant features (red shirt).

view-specific identity features (black bag), in the case that view-invariant losses are not used. Given view-invariant losses, E_{id} is able to extract view-invariant identity features (red shirt).

6.4.4 Generation Quality Evaluation

Comparison with state-of-the-art methods. We compare generated images between our proposed GCL under the JVTC [88] warmed fully unsupervised setting and state-of-the-art GAN-based ReID methods in Fig. 6.5. FD-GAN [45], IS-GAN [38] and DG-Net [208] are supervised Re-ID methods. Since the source code of these three methods is available, we

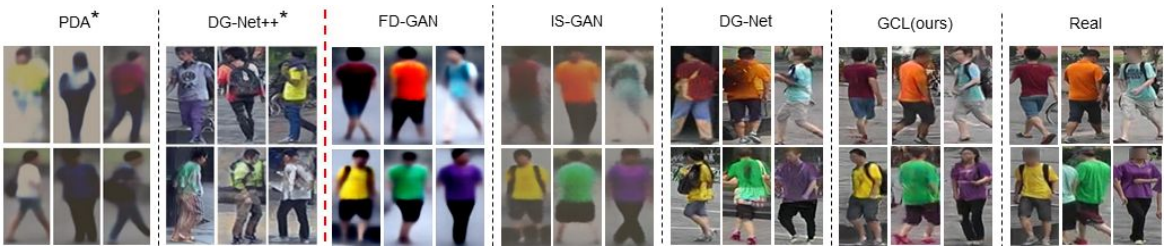


Figure 6.5 **Comparison of the generated images on Market-1501 dataset.** $\star$ refers to methods without sharing source code, whose examples are cropped from their papers. Examples of FD-GAN, IS-GAN, DG-Net and GCL are generated from six real images shown in the figure.

Method	FID(realism)	SSIM(diversity)
Real	**7.22**	0.350
FD-GAN [45]	216.88	0.271
IS-GAN [38]	281.63	0.165
DG-Net [208]	18.24	0.360
Ours(U)	59.86	0.367
Ours(UDA)	53.07	**0.369**

Table 6.4 **Comparison of FID and SSIM on Market-1501 dataset.** U denotes the fully unsupervised setting. UDA denotes Duke→Market setting.

compare generated images of same identities. We observe that there exists blur in images generated by FD-GAN and IS-GAN. DG-Net generates sharper images, but different body shapes and some incoherent objects (bags and clothes) are observed. PDA [90] and DG-Net++ [216] are UDA methods, whose source code is not yet released. We can only compare several generated images with unknown identities as illustrated in their papers. PDA generates blurred cross-domain images, whose quality is similar to FD-GAN and IS-GAN. DG-Net++ extends DG-Net into cross-domain generation, which has same problems of body shape and incoherent objects. Our GCL preserves better body shape information and does not generate incoherent objects. Moreover, our GCL is a fully unsupervised method.

We use Fréchet Inception Distance (FID) [55] to measure visual quality, as well as Structural SIMilarity (SSIM) [178] to capture structure diversity of generated images. In Table 6.4, we compare our method with FD-GAN [45], IS-GAN [38] and DG-Net [208], whose source code is available. FID measures the distribution distance between generated and real images, where a lower FID represents the case, where generated images are similar

Figure 6.6 **Generated novel views on the three datasets.**

Figure 6.7 **Linear interpolation on identity features.** Identity features are swapped between left and right persons.

to real ones. SSIM measures the intra-class structural similarity, where a larger SSIM represents a larger diversity. We note that DG-Net is outperforms our method w.r.t. FID, as the distribution is better maintained with ground truth identities in the supervised method DG-Net. However, our method is superior to DG-Net w.r.t. SSIM, as DG-Net swaps intra-dataset structures, whereas our rotated meshes build structures that do not exist in the original dataset.

More discussion. To validate, whether identity and structure features can be really disentangled under a fully unsupervised ReID setting, two experiments are conducted by changing firstly only structure features and then only identity features. Results in Fig. 6.6 show that

changing structure features only change structures and do not affect appearances. We also fix structure features and linearly interpolate two random identity feature vectors. Results in Fig. 6.7 show that identity features only change appearances and do not affect structures in generated images. More examples are shown in Fig. 6.8 (Market-1501), Fig. 6.9 (DukeMTMC) and Fig.6.10 (MSMT17).

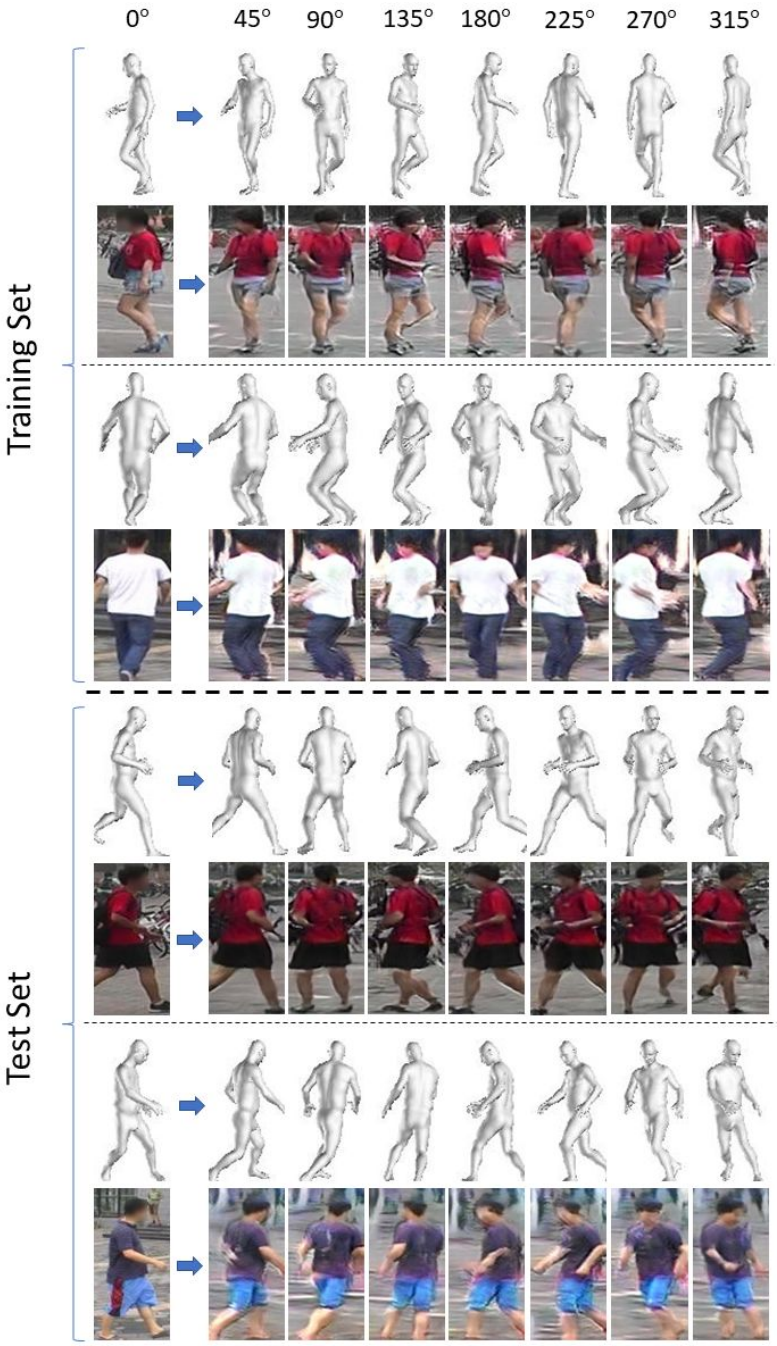


Figure 6.8 **Examples of generated novel views on Market-1501 training and test sets.**

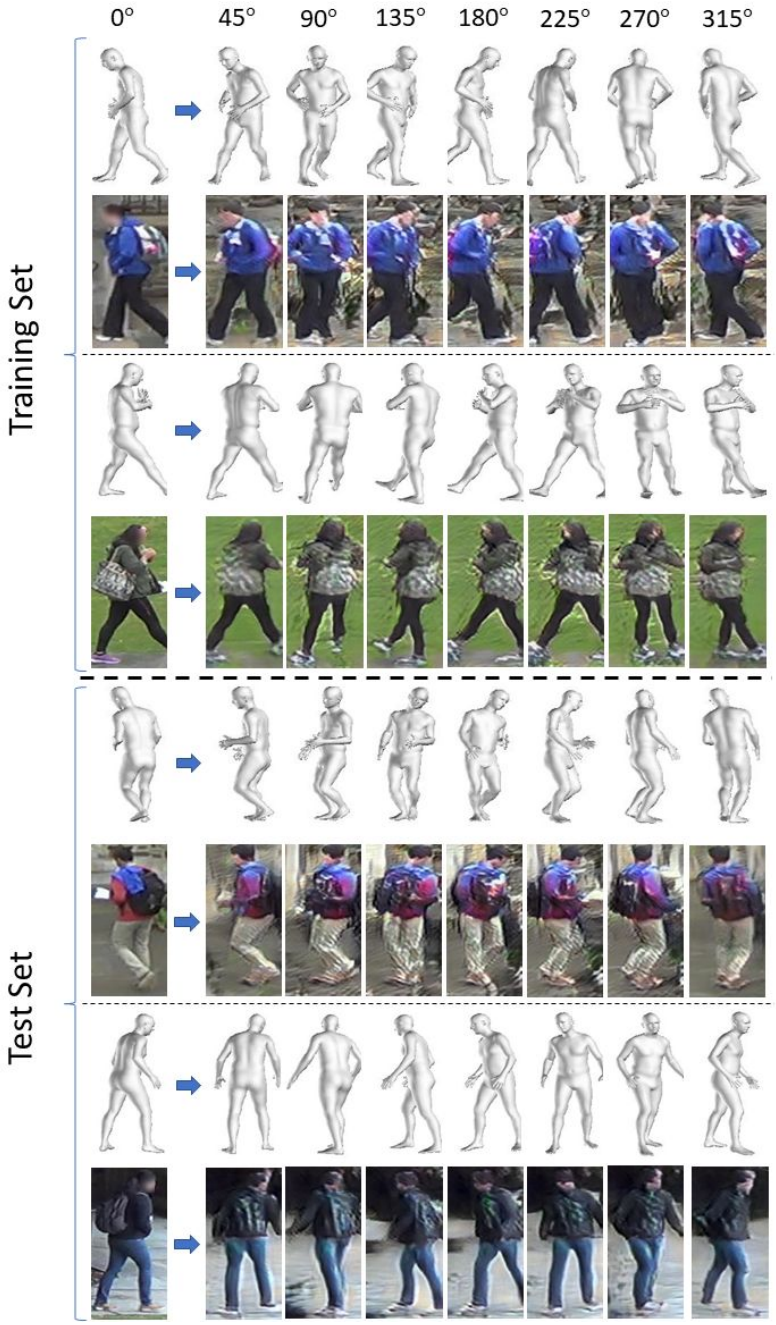


Figure 6.9 **Examples of generated novel views on DukeMTMC-reID training and test sets.**

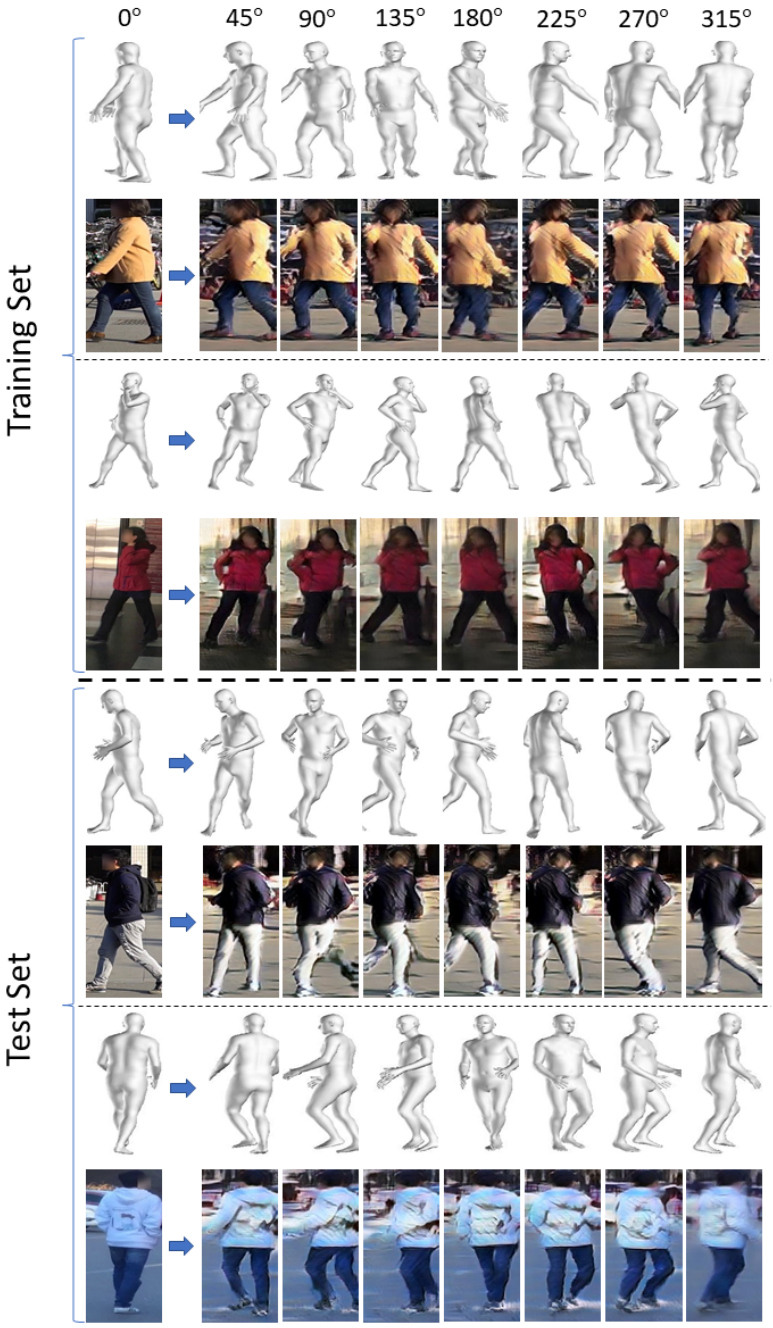


Figure 6.10 **Examples of generated novel views on MSMT17 training and test sets.**

Chapter 7

Discussion and future work

In conclusion, we have advanced the frontier of video generative modeling, as well as the interpretability of latent representations in the context of video generation. Our approaches enable new visual generation and manipulation in *machine creativity*. We are able to create unseen identities, as well as to control motion, which to the best of our knowledge, has been the first endeavor in this direction. In addition, we have demonstrated that combining generative modeling with contrastive learning can boost the performance of unsupervised person re-identification.

Below, we conclude this thesis by providing a summary of contributions and by outlining future research directions, that build on our current video generation algorithms.

7.1 Summary of contributions

— In Chapter 3, we have introduced our ImaGINator, endowed with the ability to effectively generate videos *based on a single image and a motion condition*. Specifically, we focused on the setting, where the human appearance is determined by a single input image, and motion is determined by a class-label. Our experiments showed that proposed framework generates high-quality videos, preserving the input appearance. We further showed that motion can be controlled by providing different class-labels.

— In Chapter 4, we have presented G^3AN, a novel video GAN seeking to *learn disentangled representations of generative factors (appearance and motion)* from human video data. We demonstrated that the proposed three-stream generator can effectively decompose the latent space, which allows for individual manipulation of both factors. Our experiments showed that spatio-temporal self-attention in the generator is able to model global spatio-temporal representations and improve the quality of generated videos. Finally, we showed that in both conditional and unconditional video generation settings, G^3AN systematically and significantly outperformed previous methods.

— In Chapter 5, we have introduced our MintGAN, which was designed to *learn an interpretable motion latent space* in video generation. We demonstrated that with the proposed new interpretable architecture, designed based on Linear Motion Decomposition (LMD) assumption, MintGAN was able to learn human-understandable motion directions, enabling direct manipulation of generated videos. Our experiments showcased that proposed 2D ConvNets based Temporal Pyramid Discriminator (TPD), streamlined to analyze videos at different temporal resolutions, outperformed the previous 3D ConvNets based discriminator in video quality. Finally, we further demonstrated that MintGAN was able to generate longer videos, as well as videos of higher resolution.

— In Chapter 6, we have presented a *joint generative and contrastive learning framework for unsupervised person ReID*. We demonstrated that generative and contrastive modules mutually promote each other's performance. In the generative module, we introduced a 3D mesh based novel view generator, which was more effective in body shape preservation than skeleton-guided generators. In the contrastive module, a view-invariant loss was proposed to reduce intra-class variation between original and generated images, which was beneficial in building view-invariant representations under a fully unsupervised ReID setting. Further, we overcame the limitation of previous GAN-based unsupervised ReID methods that strongly rely on a labeled source dataset. Our experiments showed that proposed method significantly surpassed the

performance of state-of-the-art methods under both, fully unsupervised, as well as unsupervised domain adaptation settings.

7.2 Future work

My future research will progress along the following paths.

High-resolution video generation. We have introduced three video generation methods in Chapters 4-6, respectively. However, associated results are limited in resolution and far from perfect w.r.t. video quality. The main reason for these limitations is that it is challenging to train GANs to produce high-resolution videos, due to large model-complexity, training instability and optimization issues. These challenges call for better network architectures, as well as for more robust loss functions and stable training procedures. We note that recent works in high-resolution image generation have achieved success in both conditional [13] and unconditional [72, 73] settings. One further solution has to do with building video generation models on top of such pretrained image generators (StyleGAN2), in order to take advantage of their powerful generation capacity. In this solution, models will focus on modeling temporal consistency, rather than building visual features from scratch. The next step in video generation constitutes developing simpler, lower complexity and memory-efficient architectures.

Self-supervised image animation. Image animation aims at transferring motion from a driving video to a target image. It is challenging, since motion representations are difficult to be directly extracted from RGB videos. Most current work uses existing extractors to obtain additional motion information such as human keypoints [17, 199, 163], semantic maps [17, 199, 163] and 3D meshes [145, 76]. However, these additional information is not always accessible in real-world videos due to pose variation, illumination and occlusion. To address this issue, we intend to build a model based on Linear Motion Decomposition presented in Chapter 6. As we have an explicit formulation of appearance and motion, models can dispense with the constraints of additional information and learn the disentanglement

of two factors directly from RGB videos. We have preliminary results based on this idea that are promising. In future, we will explore cross-domain image animation based on this approach. For example, we envision to transfer *human motion* in videos such as making coffee, changing tires or cooking, *to animate robot arms*.

Learning from synthetic video data. We have demonstrated in Chapter 6 that GAN-generated images can be used for data augmentation for novel 'views' in unsupervised person ReID. In future work, we aim to extend this idea to video-related tasks such as activity recognition and detection. Our idea is to firstly interpret the latent space to discover the representations related to a general 'view' concept (e.g., lighting, shifting, viewpoint). Then we will augment the training data by manipulating these factors. A similar idea has been explored by Varol *et al.* [151] to generate data based on Skinned Multi-Person Linear Model (SMPL) [100], validating it in a supervised learning setting. Our objective is more general, as we intend to learn discriminative spatio-temporal features from synthetic data in a totally self-supervised learning manner, without leveraging other information.

Bibliography

[1] Acharya, D., Huang, Z., Paudel, D. P., and Van Gool, L. (2018). Towards high resolution video generation with progressive growing of sliced wasserstein gans. *arXiv preprint arXiv:1810.02419*.

[2] Aifanti, N., Papachristou, C., and Delopoulos, A. (2010). The mug facial expression database. In *WIAMIS*.

[3] AlBahar, B. and Huang, J.-B. (2019). Guided image-to-image translation with bi-directional feature transformation. In *ICCV*.

[4] Amos, B., Ludwiczuk, B., and Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU School of Computer Science.

[5] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. *ICML*.

[6] Bak, S., Carr, P., and Lalonde, J.-F. (2018). Domain adaptation through synthesis for unsupervised person re-identification. In *ECCV*.

[7] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *IJCV*.

[8] Bau, D., Zhu, J.-Y., Strobelt, H., Lapedriza, A., Zhou, B., and Torralba, A. (2020). Understanding the role of individual units in a deep neural network. *PNAS*.

[9] Bau, D., Zhu, J.-Y., Strobelt, H., Zhou, B., Tenenbaum, J. B., Freeman, W. T., and Torralba, A. (2019). Gan dissection: Visualizing and understanding generative adversarial networks. In *ICLR*.

[10] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828.

[11] Berthelot, D., Schumm, T., and Metz, L. (2017). BEGAN: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*.

[12] Boden, M. A. et al. (2004). *The creative mind: Myths and mechanisms*. Psychology Press.

[13] Brock, A., Donahue, J., and Simonyan, K. (2019). Large scale GAN training for high fidelity natural image synthesis. In *ICLR*.

[14] Bulat, A. and Tzimiropoulos, G. (2017). How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *ICCV*.

[15] Carreira, J. and Zisserman, A. (2017a). Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

[16] Carreira, J. and Zisserman, A. (2017b). Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

[17] Chan, C., Ginosar, S., Zhou, T., and Efros, A. A. (2019). Everybody dance now. In *ICCV*.

[18] Chen, H., Lagadec, B., and Bremond, F. (2020a). Learning discriminative and generalizable representations by spatial-channel partition for person re-identification. In *WACV*.

[19] Chen, H., Lagadec, B., and Bremond, F. (2021a). Enhancing diversity in teacher-student networks via asymmetric branches for unsupervised person re-identification. In *WACV*.

[20] Chen, H., Wang, Y., Lagadec, B., Dantcheva, A., and Bremond, F. (2021b). Joint generative and contrastive learning for unsupervised person re-identification. In *CVPR*.

[21] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020b). A simple framework for contrastive learning of visual representations. In *ICML*.

[22] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*.

[23] Chen, X., Fan, H., Girshick, R., and He, K. (2020c). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

[24] Chen, Y., Zhu, X., and Gong, S. (2019). Instance-guided context rendering for cross-domain person re-identification. In *ICCV*.

[25] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP*.

[26] Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*.

[27] Chu, C., Zhmoginov, A., and Sandler, M. (2017). CycleGAN: a master of steganography. *NIPS Workshop*.

[28] Cohen, T., Weiler, M., Kicanaoglu, B., and Welling, M. (2019). Gauge equivariant convolutional networks and the icosahedral CNN. In *ICML*.

[29] de Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., and Courville, A. C. (2017). Modulating early visual processing by language. In *NIPS*.

[30] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009a). Imagenet: A large-scale hierarchical image database. In *CVPR*.

[31] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009b). Imagenet: A large-scale hierarchical image database. In *CVPR*.

[32] Deng, W., Zheng, L., Ye, Q., Kang, G., Yang, Y., and Jiao, J. (2018). Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *CVPR*.

[33] Denton, E. and Fergus, R. (2018). Stochastic video generation with a learned prior. In *ICML*.

[34] Denton, E. L. and Birodkar, v. (2017). Unsupervised Learning of Disentangled Representations from Video. In *NIPS*.

[35] Dibeklioğlu, H., Salah, A. A., and Gevers, T. (2012). Are you really smiling at me? spontaneous versus posed enjoyment smiles. In *ECCV*.

[36] Dowson, D. and Landau, B. (1982). The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*.

[37] Ebert, F., Finn, C., Lee, A. X., and Levine, S. (2017). Self-supervised visual planning with temporal skip connections. *CoRL*.

[38] Eom, C. and Ham, B. (2019). Learning disentangled representation for robust person re-identification. In *NIPS*.

[39] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*.

[40] Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *CVPR*.

[41] Finn, C., Goodfellow, I., and Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In *NIPS*.

[42] Fréchet, M. (1957). Sur la distance de deux lois de probabilité. *C. R. Acad. Sci. Paris*.

[43] Fu, Y., Wei, Y., Wang, G., Zhou, Y., Shi, H., and Huang, T. S. (2019). Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification. In *ICCV*.

[44] Ge, Y., Chen, D., and Li, H. (2020). Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *ICLR*.

[45] Ge, Y., Li, Z., Zhao, H., Yin, G., Yi, S., Wang, X., and Li, H. (2018). FD-GAN: Pose-guided feature distilling gan for robust person re-identification. In *NIPS*.

[46] Goetschalckx, L., Andonian, A., Oliva, A., and Isola, P. (2019). Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*.

[47] Gonzalez-Garcia, A., van de Weijer, J., and Bengio, Y. (2018). Image-to-image translation for cross-domain disentanglement. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *NIPS*.

[48] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *NIPS*.

[49] Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes. *TPAMI*.

[50] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *NIPS*.

[51] Hara, K., Kataoka, H., and Satoh, Y. (2018). Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *CVPR*.

[52] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *CVPR*.

[53] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

[54] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017a). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*.

[55] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017b). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*.

[56] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*.

[57] Hinton, G. E., Krizhevsky, A., and Wang, S. D. (2011). Transforming auto-encoders. In *ICANN*. Springer.

[58] Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *NIPS*.

[59] Hoai, M. and De la Torre, F. (2014). Max-margin early event detectors. *IJCV*.

[60] Huang, X. and Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*.

[61] Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. (2018). Multimodal unsupervised image-to-image translation. In *ECCV*.

[62] Härkönen, E., Hertzmann, A., Lehtinen, J., and Paris, S. (2020). Ganspace: Discovering interpretable gan controls. In *NeurIPS*.

[63] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

[64] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017a). Image-to-image translation with conditional adversarial networks. *CVPR*.

[65] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017b). Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*.

[66] Jahanian, A., Chai, L., and Isola, P. (2020). On the "steerability" of generative adversarial networks. In *ICLR*.

[67] Jang, Y., Kim, G., and Song, Y. (2018). Video Prediction with Appearance and Motion Conditions. In *ICML*.

[68] Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018). End-to-end recovery of human shape and pose. In *CVPR*.

[69] Kaneko, T., Hiramatsu, K., and Kashino, K. (2017). Generative attribute controller with conditional filtered generative adversarial networks. In *CVPR*.

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

[71] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*.

[72] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *CVPR*.

[73] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. In *CVPR*.

[74] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

[75] Kazeminia, S., Baur, C., Kuijper, A., van Ginneken, B., Navab, N., Albarqouni, S., and Mukhopadhyay, A. (2020). Gans for medical image analysis. *Artificial Intelligence in Medicine*.

[76] Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Niessner, M., Pérez, P., Richardt, C., Zollhöfer, M., and Theobalt, C. (2018). Deep video portraits. *ACM Transactions on Graphics (TOG)*.

[77] Kim, S. W., Zhou, Y., Philion, J., Torralba, A., and Fidler, S. (2020). Learning to simulate dynamic environments with gamegan. In *CVPR*.

[78] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *ICLR*.

[79] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *ICLR*.

[80] Kodali, N., Abernethy, J., Hays, J., and Kira, Z. (2017). On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*.

[81] Lan, T., Chen, T.-C., and Savarese, S. (2014). A hierarchical representation for future action prediction. In *ECCV*.

[82] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A. P., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*.

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

[84] Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. (2018b). Diverse image-to-image translation via disentangled representations. In *ECCV*.

[85] Lee, H.-Y., Tseng, H.-Y., Mao, Q., Huang, J.-B., Lu, Y.-D., Singh, M. K., and Yang, M.-H. (2020). Drit++: Diverse image-to-image translation viadisentangled representations. *International Journal of Computer Vision*.

[86] Lenc, K. and Vedaldi, A. (2015). Understanding image representations by measuring their equivariance and equivalence. In *CVPR*.

[87] Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., and Yan, S. (2017). Perceptual generative adversarial networks for small object detection. In *CVPR*.

[88] Li, J. and Zhang, S. (2020). Joint visual and temporal consistency for unsupervised domain adaptive person re-identification. In *ECCV*.

[89] Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., and Yang, M.-H. (2018). Flow-grounded spatial-temporal video prediction from still images. In *ECCV*.

[90] Li, Y.-J., Lin, C.-S., Lin, Y.-B., and Wang, Y.-C. F. (2019). Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation. In *ICCV*.

[91] Liang, X., Lee, L., Dai, W., and Xing, E. P. (2017). Dual motion gan for future-flow embedded video prediction. In *ICCV*.

[92] Lin, J., Zhang, R., Ganz, F., Han, S., and Zhu, J.-Y. (2021). Anycost gans for interactive image synthesis and editing. In *CVPR*.

[93] Lin, S., Li, H., Li, C.-T., and Kot, A. C. (2018). Multi-task mid-level feature alignment network for unsupervised cross-dataset person re-identification. In *BMVC*.

[94] Lin, Y., Dong, X., Zheng, L., Yan, Y., and Yang, Y. (2019). A bottom-up clustering approach to unsupervised person re-identification. In *AAAI*.

[95] Lin, Y., Xie, L., Wu, Y., Yan, C., and Tian, Q. (2020). Unsupervised person re-identification via softened similarity learning. In *CVPR*.

[96] Liu, M.-Y., Breuel, T., and Kautz, J. (2017). Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*.

[97] Liu, S., Wang, T., Bau, D., Zhu, J.-Y., and Torralba, A. (2020). Diverse image generation via self-conditioned gans. In *CVPR*.

[98] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *ICCV*.

[99] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *CVPR*.

[100] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*.

[101] Luc, P., Neverova, N., Couprie, C., Verbeek, J., and LeCun, Y. (2017). Predicting deeper into the future of semantic segmentation. *ICCV*.

[102] Luo, C., Song, C., and Zhang, Z. (2020). Generalizing person re-identification by camera-aware invariance learning and cross-domain mixup. In *European Conference on Computer Vision*.

[103] Ma, L., Sun, Q., Georgoulis, S., Van Gool, L., Schiele, B., and Fritz, M. (2018). Disentangled person image generation. In *CVPR*.

[104] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. (2017). Least squares generative adversarial networks. In *ICCV*.

[105] Mathieu, M., Couprie, C., and LeCun, Y. (2016). Deep Multi-Scale Video Prediction Beyond Mean Square Error. In *ICLR*.

[106] Menapace, W., Lathuilière, S., Tulyakov, S., Siarohin, A., and Ricci, E. (2021). Playable video generation. *CVPR*.

[107] Mescheder, L., Geiger, A., and Nowozin, S. (2018a). Which training methods for gans do actually converge? In *ICML*.

[108] Mescheder, L., Nowozin, S., and Geiger, A. (2018b). Which training methods for gans do actually converge? In *ICML*.

[109] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[110] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *ICLR*.

[111] Miyato, T. and Koyama, M. (2018). cGANs with projection discriminator. In *ICLR*.

[112] Nagrani, A., Chung, J. S., Xie, W., and Zisserman, A. (2019). Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*.

[113] Odena, A., Olah, C., and Shlens, J. (2017). Conditional Image Synthesis With Auxiliary Classifier GANs. In *ICML*.

[114] Ohnishi, K., Yamamoto, S., Ushiku, Y., and Harada, T. (2018). Hierarchical video generation from orthogonal information: Optical flow and texture. In *AAAI*.

[115] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[116] Pan, J., Wang, C., Jia, X., Shao, J., Sheng, L., Yan, J., and Wang, X. (2019). Video generation from single semantic label map. In *CVPR*.

[117] Park, T., Efros, A. A., Zhang, R., and Zhu, J.-Y. (2020). Contrastive learning for unpaired image-to-image translation. In *ECCV*.

[118] Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *CVPR*.

[119] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

[120] Peebles, W., Peebles, J., Zhu, J.-Y., Efros, A. A., and Torralba, A. (2020). The hessian penalty: A weak prior for unsupervised disentanglement. In *ECCV*.

[121] Pintea, S. L., van Gemert, J. C., and Smeulders, A. W. (2014). Déja vu. In *ECCV*.

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

[123] Rao, K., Harris, C., Irpan, A., Levine, S., Ibarz, J., and Khansari, M. (2020). Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *CVPR*.

[124] Reda, F. A., Liu, G., Shih, K. J., Kirby, R., Barker, J., Tarjan, D., Tao, A., and Catanzaro, B. (2018). Sdc-net: Video prediction using spatially-displaced convolution. In *ECCV*.

[125] Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV Workshops*.

[126] Romero, A., Arbeláez, P., Van Gool, L., and Timofte, R. (2019). Smit: Stochastic multi-label image-to-image translation. In *ICCV Workshops*.

[127] Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. (2017). Stabilizing training of generative adversarial networks through regularization. In *NIPS*.

[128] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252.

[129] Ryoo, M. S. (2011). Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*.

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

[131] Saito, M., Saito, S., Koyama, M., and Kobayashi, S. (2020). Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan.

[132] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016a). Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*.

[133] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X. (2016b). Improved techniques for training gans. In *NIPS*.

[134] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X. (2016c). Improved techniques for training GANs. In *NIPS*.

[135] Shen, Y., Gu, J., Tang, X., and Zhou, B. (2020). Interpreting the latent space of gans for semantic face editing. In *CVPR*.

[136] Shen, Y. and Zhou, B. (2021). Closed-form factorization of latent semantics in gans. In *CVPR*.

[137] Siddiquee, M. M. R., Zhou, Z., Tajbakhsh, N., Feng, R., Gotway, M. B., Bengio, Y., and Liang, J. (2019). Learning fixed points in generative adversarial networks: From image-to-image translation to disease detection and localization. In *ICCV*.

[138] Singh, K. K., Ojha, U., and Lee, Y. J. (2019). Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*.

[139] Smith, L., Dhawan, N., Zhang, M., Abbeel, P., and Levine, S. (2019). Avid: Learning multi-stage tasks via pixel-level translation of human videos. *RSS*.

[140] Song, Y., Demirdjian, D., and Davis, R. (2011). Tracking Body and Hands For Gesture Recognition: NATOPS Aircraft Handling Signals Database. In *FG*.

[141] Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

[142] Sun, Y., Zheng, L., Yang, Y., Tian, Q., and Wang, S. (2018). Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*.

[143] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *NIPS*.

[144] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *CVPR*.

[145] Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., and Niessner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[146] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

[147] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *CVPR*.

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

[149] TURING, A. M. (1950). I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460.

[150] Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. (2018). Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*.

[151] Varol, G., Laptev, I., Schmid, C., and Zisserman, A. (2019). Synthetic humans for action recognition from unseen viewpoints. *CoRR*, abs/1912.04070.

[152] Varol, G., Laptev, I., Schmid, C., and Zisserman, A. (2021). Synthetic humans for action recognition from unseen viewpoints. *International Journal of Computer Vision*, pages 1–24.

[153] Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017a). Learning from synthetic humans. In *CVPR*, pages 109–117.

[154] Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017b). Learning from synthetic humans. In *CVPR*.

[155] Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. (2017a). Decomposing motion and content for natural video sequence prediction. *ICLR*.

[156] Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., and Lee, H. (2017b). Learning to Generate Long-term Future via Hierarchical Prediction. In *ICML*.

[157] Vondrick, C., Pirsiavash, H., and Torralba, A. (2016). Generating videos with scene dynamics. In *NIPS*.

[158] Voynov, A. and Babenko, A. (2020). Unsupervised discovery of interpretable directions in the gan latent space. *ICML*.

[159] Walker, J., Doersch, C., Gupta, A., and Hebert, M. (2016). An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*.

[160] Walker, J., Gupta, A., and Hebert, M. (2014). Patch to the future: Unsupervised visual prediction. In *CVPR*.

[161] Walker, J., Marino, K., Gupta, A., and Hebert, M. (2017). The pose knows: Video forecasting by generating pose futures. In *ICCV*.

[162] Wang, J., Zhu, X., Gong, S., and Li, W. (2018a). Transferable joint attribute-identity deep learning for unsupervised person re-identification. *CVPR*.

[163] Wang, T.-C., Liu, M.-Y., Tao, A., Liu, G., Kautz, J., and Catanzaro, B. (2019a). Few-shot video-to-video synthesis. In *NeurIPS*.

[164] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. (2018b). Video-to-video synthesis. In *NeurIPS*.

[165] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018c). High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*.

[166] Wang, Y., Bilinski, P., Bremond, F., and Dantcheva, A. (2019b). G3AN: This video does not exist. Disentangling motion and appearance for video generation. *CVPR*.

[167] Wang, Y., Bilinski, P., Bremond, F., and Dantcheva, A. (2020a). G3AN: Disentangling appearance and motion for video generation. In *CVPR*.

[168] Wang, Y., Bilinski, P., Bremond, F., and Dantcheva, A. (2020b). G3an: Disentangling appearance and motion for video generation. In *CVPR*.

[169] Wang, Y., Bilinski, P., Bremond, F., and Dantcheva, A. (2020c). ImaGINator: Conditional spatio-temporal gan for video generation. In *WACV*.

[170] Wang, Y., Bilinski, P., Bremond, F., and Dantcheva, A. (2020d). Imaginator: Conditional spatio-temporal gan for video generation. In *WACV*.

[171] Wang, Y., Bilinski, P., Bremond, F., and Dantcheva, A. (2020). Imaginator: Conditional spatio-temporal gan for video generation. In *WACV*.

[172] Wang, Y., Bremond, F., and Dantcheva, A. (2021). Inmodegan: Interpretable motion decomposition generative adversarial network for video generation. *arXiv preprint arXiv:2101.03049*.

[173] Wang, Y. and Dantcheva, A. (2020). A video is worth more than 1000 lies. comparing 3dcnn approaches for detecting deepfakes. In *FG*.

[174] Wang, Y., Dantcheva, A., and Bremond, F. (2018d). From attribute-labels to faces: face generation using a conditional generative adversarial network. In *ECCV Workshops*.

[175] Wang, Y., Dantcheva, A., and Bremond, F. (2018e). From attribute-labels to faces: face generation using a conditional generative adversarial network. In *ECCV Workshops*.

[176] Wang, Y., Dantcheva, A., and Bremond, F. (2018f). From attributes to faces: a conditional generative adversarial network for face generation. In *BIOSIG*.

[177] Wang, Y., Dantcheva, A., Broutart, J.-C., Robert, P., Bremond, F., and Bilinski, P. (2018g). Comparing methods for assessment of facial dynamics in patients with major neurocognitive disorders. In *ECCV Workshops*.

[178] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

[179] Wang, Z., Vandersteen, C., Demarcy, T., Gnansia, D., Raffaelli, C., Guevara, N., and Delingette, H. (2019c). Deep learning based metal artifacts reduction in post-operative cochlear implant ct imaging. In *MICCAI*.

[180] Wei, L., Zhang, S., Gao, W., and Tian, Q. (2018). Person transfer gan to bridge domain gap for person re-identification. In *CVPR*.

[181] Welander, P., Karlsson, S., and Eklund, A. (2018). Generative adversarial networks for image-to-image translation on multi-contrast mr images-a comparison of cyclegan and unit. *arXiv preprint arXiv:1806.07777*.

[182] Wichers, N., Villegas, R., Erhan, D., and Lee, H. (2018). Hierarchical long-term video prediction without supervision. In *ICML*.

[183] Wu, G., Zhu, X., and Gong, S. (2020). Tracklet self-supervised learning for unsupervised person re-identification. In *AAAI*.

[184] Wu, Y., Lin, Y., Dong, X., Yan, Y., Bian, W., and Yang, Y. (2019). Progressive learning for person re-identification with one example. *TIP*.

[185] Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. *CVPR*.

[186] Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. (2018). Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*.

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

[188] Yang, C., Wang, Z., Zhu, X., Huang, C., Shi, J., and Lin, D. (2018). Pose guided human video generation. In *ECCV*.

[189] Yang, C., Xu, Y., Shi, J., Dai, B., and Zhou, B. (2020a). Temporal pyramid network for action recognition. In *CVPR*.

[190] Yang, D., Dai, R., Wang, Y., Mallick, R., Minciullo, L., Francesca, G., and Bremond, F. (2021a). Selective spatio-temporal aggregation based pose refinement system: Towards understanding human activities in real-world videos. In *WACV*.

[191] Yang, D., Wang, Y., Dantcheva, A., Garattoni, L., Francesca, G., and Brémond, F. (2021b). Self-supervised video pose representation learning for occlusion-robust action recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*.

[192] Yang, D., Wang, Y., Dantcheva, A., Garattoni, L., Francesca, G., and Bremond, F. (2021c). Unik: A unified framework for real-world skeleton-based action recognition. *BMVC*.

[193] Yang, F., Li, K., Zhong, Z., Luo, Z., Sun, X., Cheng, H., Guo, X., Huang, F., Ji, R., and Li, S. (2020b). Asymmetric co-teaching for unsupervised cross-domain person re-identification. In *AAAI*.

[194] Yi, X., Walia, E., and Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical image analysis*, 58:101552.

[195] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., and Sang, N. (2018). Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*.

[196] Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

[197] Yu, S., Han, H., Shan, S., Dantcheva, A., and Chen, X. (2019). Improving face sketch recognition via adversarial sketch-photo transformation. In *FG*.

[198] Yuen, J. and Torralba, A. (2010). A data-driven approach for event prediction. In *ECCV*.

[199] Zakharov, E., Shysheya, A., Burkov, E., and Lempitsky, V. (2019). Few-shot adversarial learning of realistic neural talking head models. In *ICCV*.

[200] Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019a). Self-attention generative adversarial networks. In *ICML*.

[201] Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019b). Self-Attention Generative Adversarial Networks. *ICML*.

[202] Zhang, J. Y., Felsen, P., Kanazawa, A., and Malik, J. (2019c). Predicting 3d human dynamics from video. In *ICCV*.

[203] Zhao, B., Meng, L., Yin, W., and Sigal, L. (2019). Image generation from layout. In *CVPR*.

[204] Zhao, L., Peng, X., Tian, Y., Kapadia, M., and Metaxas, D. (2018a). Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*.

[205] Zhao, L., Peng, X., Tian, Y., Kapadia, M., and Metaxas, D. (2018b). Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*.

[206] Zhao, S., Liu, Z., Lin, J., Zhu, J.-Y., and Han, S. (2020). Differentiable augmentation for data-efficient gan training. In *NeurIPS*.

[207] Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., and Tian, Q. (2015). Scalable person re-identification: A benchmark. *ICCV*.

[208] Zheng, Z., Yang, X., Yu, Z., Zheng, L., Yang, Y., and Kautz, J. (2019). Joint discriminative and generative learning for person re-identification. In *CVPR*.

[209] Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2020a). Random erasing data augmentation. In *AAAI*.

[210] Zhong, Z., Zheng, L., Li, S., and Yang, Y. (2018a). Generalizing a person retrieval model hetero- and homogeneously. In *ECCV*.

[211] Zhong, Z., Zheng, L., Luo, Z., Li, S., and Yang, Y. (2019). Invariance matters: Exemplar memory for domain adaptive person re-identification. In *CVPR*.

[212] Zhong, Z., Zheng, L., Luo, Z., Li, S., and Yang, Y. (2020b). Learning to adapt invariance in memory for person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[213] Zhong, Z., Zheng, L., Zheng, Z., Li, S., and Yang, Y. (2018b). Camera style adaptation for person re-identification. *CVPR*.

[214] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017a). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*.

[215] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017b). Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *ICCV*.

[216] Zou, Y., Yang, X., Yu, Z., Kumar, B. V. K. V., and Kautz, J. (2020). Joint disentangling and adaptation for cross-domain person re-identification. In *ECCV*.