

# Attribute-Controlled Traffic Data Augmentation Using Conditional Generative Models

Elective in AI Final Project  
Federico Vergallo (1815521)  
Michał Ostyk-Narbut (1854051)

December 16, 2019



SAPIENZA  
UNIVERSITÀ DI ROMA

# Presentation outline

---

## 1 Introduction

- Autonomous vehicles
- Generative Adversarial Networks (GANs)

## 2 Attribute Interpolation with Conditional Generative Models

- Architecture
- Dataset and tools
- Preprocessing and Training

## 3 Results

## 4 Conclusions

# Outline

---

## 1 Introduction

- Autonomous vehicles
- Generative Adversarial Networks (GANs)

## 2 Attribute Interpolation with Conditional Generative Models

- Architecture
- Dataset and tools
- Preprocessing and Training

## 3 Results

## 4 Conclusions

# Introduction

## Autonomous vehicular systems

---

### Use case in self-driving cars

- Require real-world data to train perception systems
- RGB images collected and further manually annotated
- Imbalanced datasets

### Motivation and goals

- Balanced dataset
- Present day solutions:
  - 3D simulations → not realistic and models are susceptible to synthetic artifacts.

# Introduction

## Generative Adversarial Networks (GANs)

### Advantages

- Able to capture the intricacies of natural images
- Generate natural, realistic transformations of input images.
- Allow for latent space interpolation (cheap data generation!)

### What are GANs?

- Introduced in 2014 by Ian Goodfellow et al.
- Class of deep learning methods for unsupervised learning
- The aim is to be able to generate samples with the same features of the examples contained in a dataset

# Introduction

## Generative Adversarial Networks (GANs)

### Basic intuition of generative models

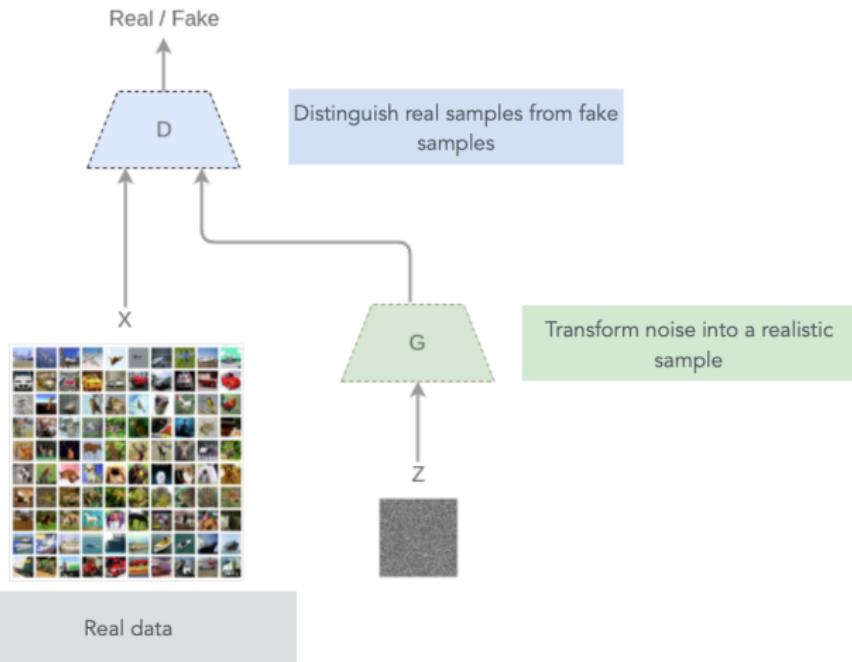
- A model  $P(x; \theta)$  that we can draw samples from i.e. Gaussian Mixture Model.
- GANs: pit a generator (G) and discriminator (D) against each other to play a minimax game.
- Goal is to capture the data distribution  $p_{data}(x)$
- $P(x; \theta) \approx p_{data}(x)$

### Why are they important?

- Model the probability density of images
- Generate novel content
- Artistic applications, Image completion and many more

# Introduction

## Generative Adversarial Networks (GANs)



source: <https://indico.cern.ch/event/655447/contributions/2742176/attachments/1551254/2437141/gan-intro-iml.pdf>

# Outline

---

## 1 Introduction

- Autonomous vehicles
- Generative Adversarial Networks (GANs)

## 2 Attribute Interpolation with Conditional Generative Models

- Architecture
- Dataset and tools
- Preprocessing and Training

## 3 Results

## 4 Conclusions

# Architecture

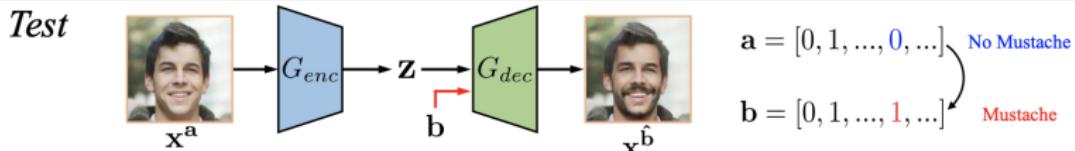
## Formulation part 1

### Binary attributes editing

- Two subnetworks:
  - Encoder  $G_{enc}$  and Decoder  $G_{dec}$
  - Discriminator  $D$  and Attribute Classifier  $C$
  - $a = [a_1 \dots a_n]$  (original) and  $b = [b_1 \dots b_n]$  (desired).

### Desired testing scenario

- $z = G_{enc}(x^a)$  latent representation
- $x^{\hat{b}} = G_{dec}(z, b)$  decoding
- $x^{\hat{b}} = G_{dec}(G_{enc}(x^a), b)$  whole editing process → unsupervised



# Architecture

## Formulation part 2

### Training Roles

- ① Attribute classifier: constrain images  $x^{\hat{b}}$  to desired attributes
- ② Adversarial learning: visual reality of  $x^{\hat{b}}$
- ③ Reconstruction learning:
  - make  $z$  conserve enough information for attribute-excluding details recovery
  - Enable  $G_{dec}$  to restore attribute-excluding details.  $x^{\hat{a}} = G_{dec}(z, a)$  should approximate itself.

# Architecture

## Overview

*Train*

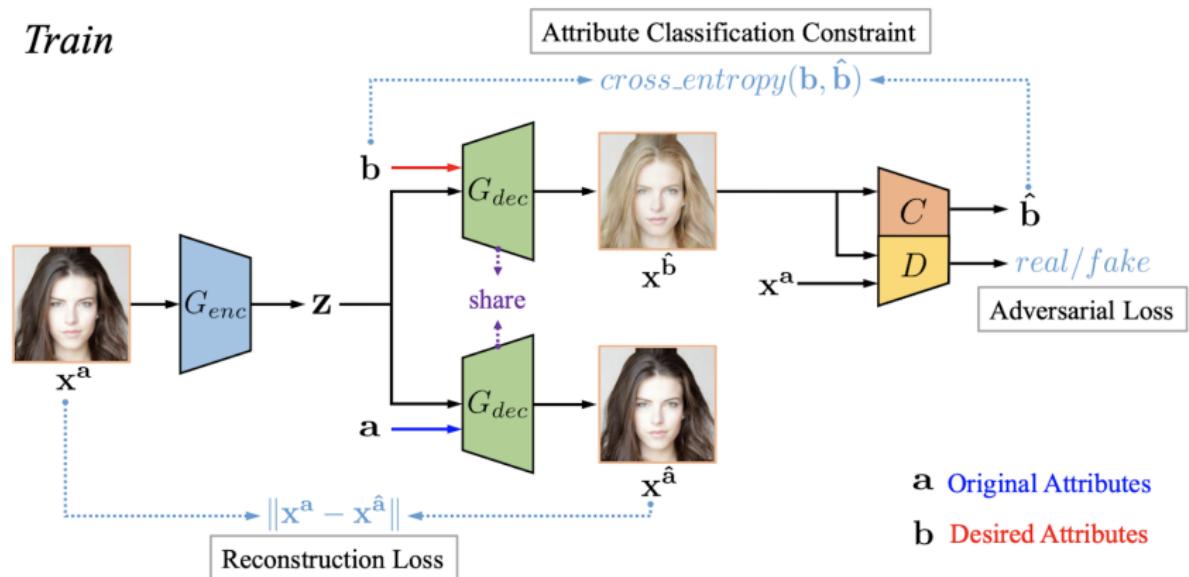


Figure: Architecture from the original Att-GAN paper [1]

# Architecture

## Losses

### Losses

- Attribute Classification Constraint (cls):
  - $\min_{G_{enc}, G_{dec}} \mathcal{L}_{cls_g}$
  - $\min_C \mathcal{L}_{cls_c}$
- Reconstruction (rec):  $\min_{G_{enc}, G_{dec}} \mathcal{L}_{rec}$  ( $l_1$  loss)
- Adversarial (adv) (W-GAN style):
  - $\min_{|D| \leq 1} \mathcal{L}_{adv_d}$
  - $\min_{G_{enc}, G_{dec}} \mathcal{L}_{adv_g}$

### Overall objective

- Generator:  $\min_{G_{enc}, G_{dec}} = \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{cls_g} + \mathcal{L}_{adv_g}$
- Discriminator:  $\min_D, C = \lambda_1 \mathcal{L}_{cls_c} + \mathcal{L}_{adv_d}$

# Architecture

## network

---

Encoder ( $G_{enc}$ )	Decoder ( $G_{dec}$ )	Discriminator ( $D$ )	Classifier ( $C$ )
Conv(64,4,2), BN, Leaky ReLU	DeConv(1024,4,2), BN, ReLU	Conv(64,4,2), LN/IN, Leaky ReLU	
Conv(128,4,2), BN, Leaky ReLU	DeConv(512,4,2), BN, ReLU	Conv(128,4,2), LN/IN, Leaky ReLU	
Conv(256,4,2), BN, Leaky ReLU	DeConv(256,4,2), BN, ReLU	Conv(256,4,2), LN/IN, Leaky ReLU	
Conv(512,4,2), BN, Leaky ReLU	DeConv(128,4,2), BN, ReLU	Conv(512,4,2), LN/IN, Leaky ReLU	
Conv(1024,4,2), BN, Leaky ReLU	DeConv(3,4,2), Tanh	Conv(1024,4,2), LN/IN, Leaky ReLU	
		FC(1024), LN/IN, Leaky ReLU	FC(1024), LN/IN, Leaky ReLU
		FC(1)	FC(2), Sigmoid

Figure: Network details [1]

# Dataset and tools

## description

---

### Datasets

- BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling [3]

### Tools

Python 3.6 with Tensorflow 1.14 GPU

# Preprocessing

## Data storage

### TF-records

- very useful tool for storing large data efficiently for training
- reinitializable iterator
- optimizing performance through parallel data transformation

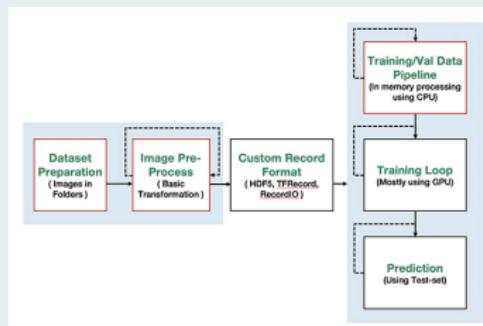


Figure: pipeline source: <http://www.adeveloperdiary.com/>

# Outline

---

## 1 Introduction

- Autonomous vehicles
- Generative Adversarial Networks (GANs)

## 2 Attribute Interpolation with Conditional Generative Models

- Architecture
- Dataset and tools
- Preprocessing and Training

## 3 Results

## 4 Conclusions

# Results

from the paper

---



Figure: Results from the re-implemented paper [2]

# Results

## Our implementation

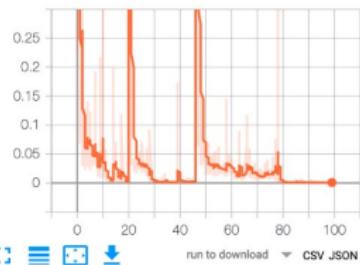
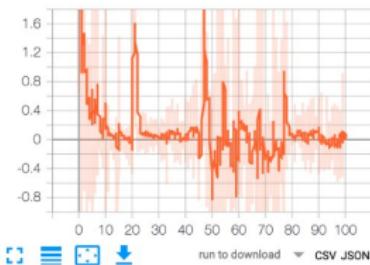
---



# Results

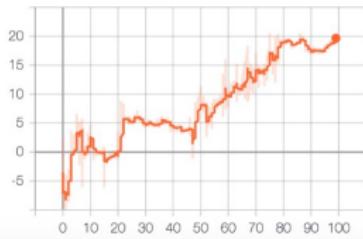
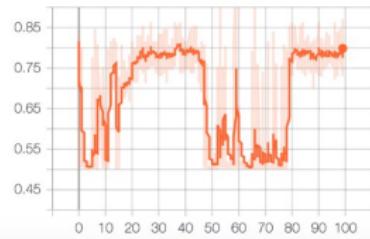
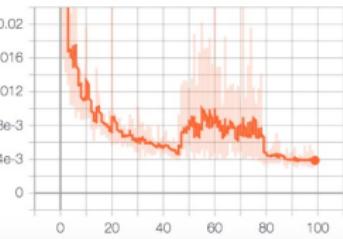
## Tensorboard

---

gp  
tag: D\_/gp[run to download](#) [CSV](#) [JSON](#)loss\_adv\_D  
tag: D\_/loss\_adv\_D[run to download](#) [CSV](#) [JSON](#)loss\_cls\_C  
tag: D\_/loss\_cls\_C[run to download](#) [CSV](#) [JSON](#)

G\_

3

loss\_adv\_G  
tag: G\_/loss\_adv\_G[run to download](#) [CSV](#) [JSON](#)loss\_cls\_G  
tag: G\_/loss\_cls\_G[run to download](#) [CSV](#) [JSON](#)loss\_rec  
tag: G\_/loss\_rec[run to download](#) [CSV](#) [JSON](#)

# Results

## Training

---

### hyperparameters

- Adam optimizer with a decaying learning rate
- batch size 64, 128, 100+ epochs

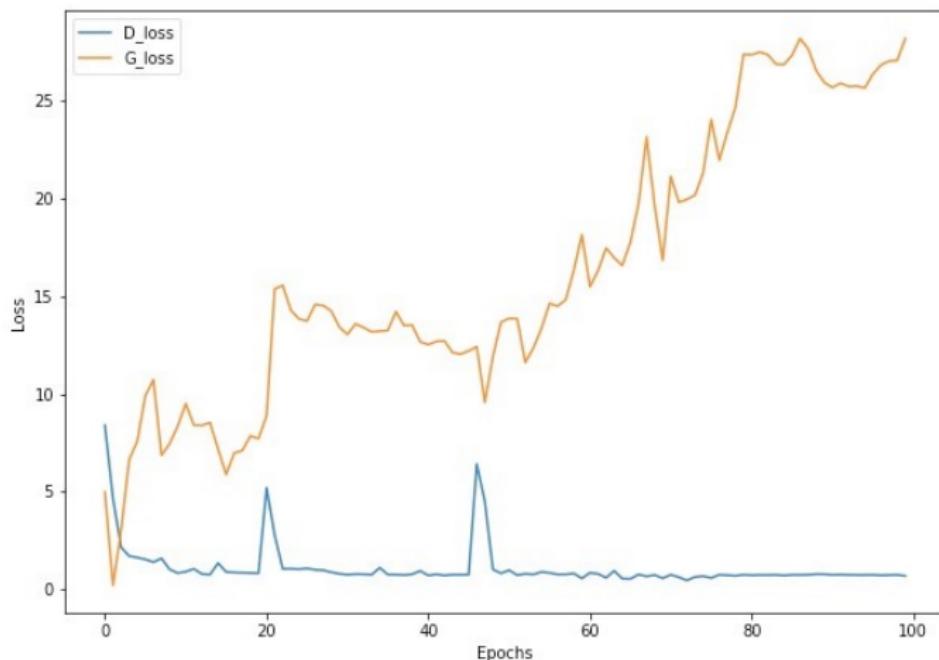
### Overcoming issues

- Balancing Generator and Discriminator weight updates
- Minibatch discrimination
- Adding noise
- Two Time-Scale Update Rule (different learning rates for  $G$  and  $D$ )

# Results

## overall losses

---



# Outline

---

## 1 Introduction

- Autonomous vehicles
- Generative Adversarial Networks (GANs)

## 2 Attribute Interpolation with Conditional Generative Models

- Architecture
- Dataset and tools
- Preprocessing and Training

## 3 Results

## 4 Conclusions

# Conclusions

## Conclusions and possible improvements

- ① Use cropped images
- ② Very unstable training phase, needs multiple overhauls to see results
- ③ Data Augmentation idea is a success.

Attributes	Day		Night	
	Type	Original	Generated	Original
Count	54563	19178	19178	54563

# References

-  Original Paper: Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. arxiv preprint, 2017.
-  Re-implemented paper: Mukherjee, Amitangshu et al. "Attribute-Controlled Traffic Data Augmentation Using Conditional Generative Models." CVPR Workshops (2019).
-  Dataset: Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, Trevor Darrell. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling arXiv, 2018
-  Dataset link: <https://bdd-data.berkeley.edu/>
-  Code: <https://github.com/Ostyk/self-driving-AttGAN>

# How training GANs looks like from reading papers vs how it actually is

