

# Generative Adversarial Networks for Multimedia Content Creation

## Summary



**Dr. I. Mademlis, G. Voulgaris, C. Papaioannidis,  
Prof. Ioannis Pitas**  
**Aristotle University of Thessaloniki**  
**[pitass@csd.auth.gr](mailto:pitass@csd.auth.gr)**  
**[www.aiia.csd.auth.gr](http://www.aiia.csd.auth.gr)**  
**Version 3.1.1**

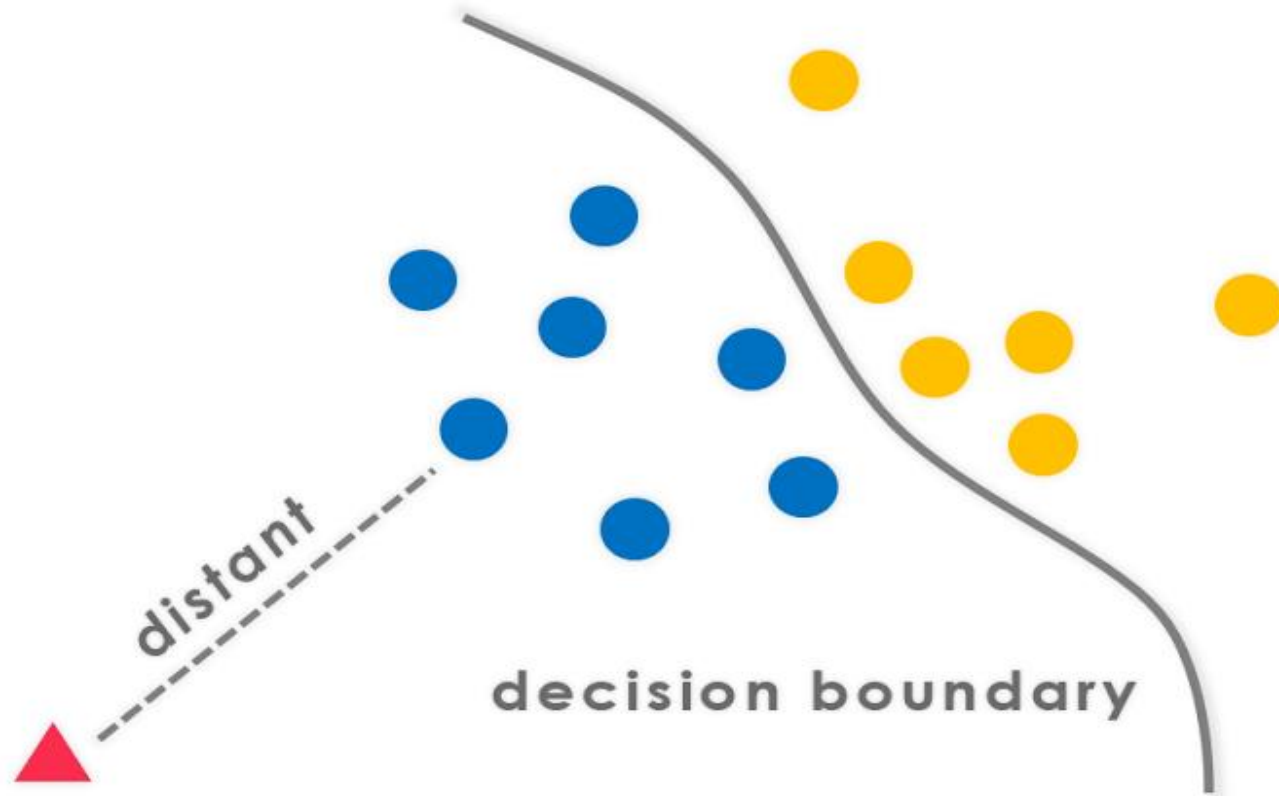


# Introduction

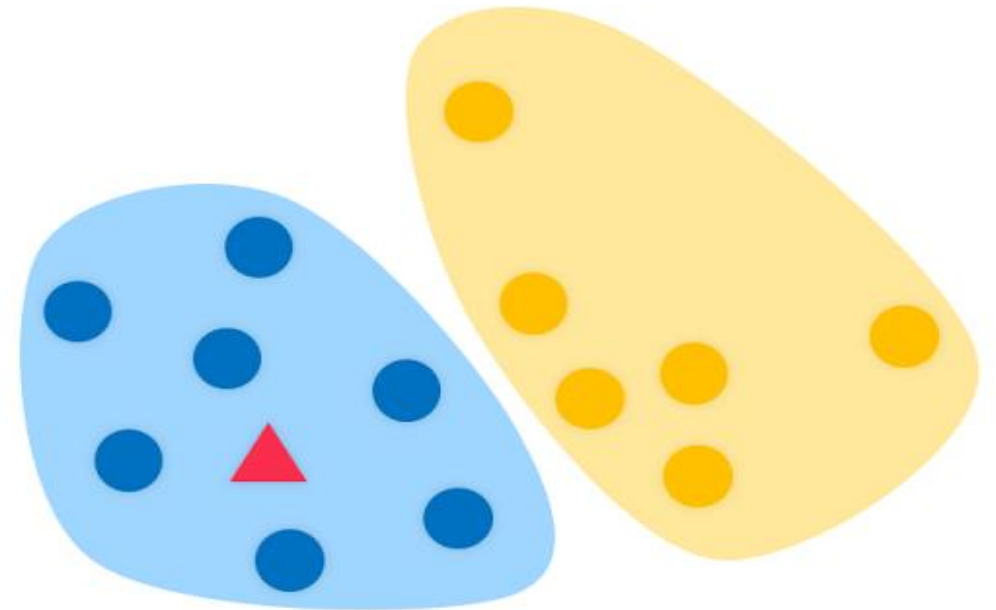
- **Generative learning models** are employed for **approximating the data generating probability density function** (pdf) from which a dataset has been sampled.
- This is meaningful both for supervised and unsupervised learning problems.
- They are contrasted against **discriminative learning models**, used in classification, which are only meaningful for supervised problems.

# Introduction

Discriminative



Generative



# Introduction

- Each generative model is defined by a set of parameters, which are optimized by training on a given dataset.
- A trained generative model approximates the data generating probability density function (***data distribution***).
- ***Generative Adversarial Networks*** (GANs) are Deep Neural Network (DNN)-based generative models. They capture the data distribution implicitly, thus facilitating sampling from data distribution approximation, after training.
- They are typically used for generating realistic novel data samples (“***fakes***”) that meaningfully resemble the training data.



# Introduction

- GAN are a class of generative models that produce promising results in several application domains, ranging from ***image generation*** to ***video captioning***.
- Since it is a DNN-based architecture, it bears all the advantages of DNNs (e.g., ***automatic feature learning***), through proper training.
- Various neural building blocks may be internally employed, depending on the employed data and the target task (e.g., 2D convolutional layers for synthesizing images).

# Introduction

- A trained GAN may be employed to create huge amounts of realistic data for any digital media genre.
- GANs can be used for ***data augmentation***, which can be very useful for:
  - Training/testing other DNNs.
  - DNN ***domain adaptation***.

# Background

- A **generative model** encodes a distribution  $\hat{p}_{\theta}(\mathbf{x})$  defined by parameter vector  $\theta \in \mathbb{R}^c$ , so that  $\hat{p}_{\theta}(\mathbf{x}) \approx p_{\mathbf{x}}(\mathbf{x})$ .
- $\hat{p}_{\theta}(\mathbf{x})$  is a function of random vector  $\mathbf{x}$ .
- After the generative model has been trained,  $\hat{p}_{\theta}(\mathbf{x})$  is a good approximation of  $p_{\mathbf{x}}(\mathbf{x})$ .
- Model complexity is directly related to number of model parameters  $c$ .

# Background

Typically, in a **Maximum Likelihood Estimation** (MLE) setting, training the model on  $\mathcal{D}$ , in order to find a “good”  $\theta$  implies maximizing the likelihood of  $\theta$ , i.e., defined by:

$$\theta^* = \operatorname{argmax}_{\theta} \prod_{i=1}^N \hat{p}_{\theta}(\mathbf{x}_i).$$

- Due to practical considerations (e.g., numerical stability), we obtain an equivalent optimization problem, by computing the logarithm of the likelihood:

$$\theta^* = \operatorname{argmax}_{\theta} \sum_{i=1}^N \log \hat{p}_{\theta}(\mathbf{x}_i).$$



# Background

- The term  $\log p_{\mathbf{x}}(\mathbf{x})$  does not affect the optimization process, since it is not a function of  $\theta$ .
- Thus, the minimization problem may be reformulated equivalently as:

$$\theta^* = \operatorname{argmin}_{\theta} - E_{\mathbf{x}}\{\log \hat{p}_{\theta}(\mathbf{x})\}.$$

- This is called ***negative log-likelihood minimization***.

# Background

**Proposition:** Finding  $\theta^*$  under this formulation is equivalent to minimizing **cross-entropy**  $H(p_{\mathbf{x}}, \hat{p}_{\theta})$  between distributions  $p_{\mathbf{x}}(\mathbf{x})$  and  $\hat{p}_{\theta}(\mathbf{x})$ .

**Proof:** As the entropy of  $p_{\mathbf{x}}(\mathbf{x})$  is given by:

$$H(p_{\mathbf{x}}) = -E_{\mathbf{x}}\{\log p_{\mathbf{x}}(\mathbf{x})\},$$

cross-entropy  $H(p_{\mathbf{x}}, \hat{p}_{\theta})$  takes the form:

$$H(p_{\mathbf{x}}, \hat{p}_{\theta}) = H(p_{\mathbf{x}}) + D_{KL}(p_{\mathbf{x}} || \hat{p}_{\theta}) = -E_{\mathbf{x}}\{\log \hat{p}_{\theta}(\mathbf{x})\}.$$

Q.E.D.

# Background

- The cross-entropy training objective can be easily extended to **discriminative model learning**, where a label  $\mathbf{y}_i$  is available for each  $\mathbf{x}_i$  in  $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N\}$ , using conditional probabilities  $p_{\mathbf{y}}(\mathbf{y}|\mathbf{x})$ .
- In classification, the model output  $\hat{\mathbf{y}}_i \in \mathbb{R}^m$  (where  $m$  is the number of classes) given input  $\mathbf{x}_i$  encodes the class assigned to  $\mathbf{x}_i$  by the model.
- Typically, the ground-truth label  $\mathbf{y}_i \in \{0,1\}^m$  is a **one-hot** encoded vector in residing in the label-space ( $|\mathbf{y}_i|_1 = 1$ ).

# Background

- Let us employ a generative model to derive a pdf model  $\hat{p}_{\theta}(\mathbf{x})$  defined by parameter vector  $\theta \in \mathbb{R}^c$ , so that  $\hat{p}_{\theta}(\mathbf{y}|\mathbf{x}) \approx p_{\mathbf{y}}(\mathbf{y}|\mathbf{x})$ .
- The training objective is to find the “best”  $\theta$  that maximizes the likelihood of  $\mathbf{Y}$  given  $\mathbf{X}$ :

$$\theta^* = \operatorname{argmax}_{\theta} \prod_{i=1}^N \hat{p}_{\theta}(\mathbf{y}_i|\mathbf{x}_i).$$

- We can obtain the equivalent negative log-likelihood reformulation:

$$\theta^* = \operatorname{argmin}_{\theta} - E_{\mathbf{y}}\{\log \hat{p}_{\theta}(\mathbf{y}|\mathbf{x})\}.$$



# Background

- By reformulating the optimization problem into an equivalent negative log-likelihood one, we obtain:

$$\theta^* = \operatorname{argmin}_{\theta} - E_{\mathbf{y}} \left\{ \sum_{k=1}^m y^k \log \hat{y}^k \right\},$$

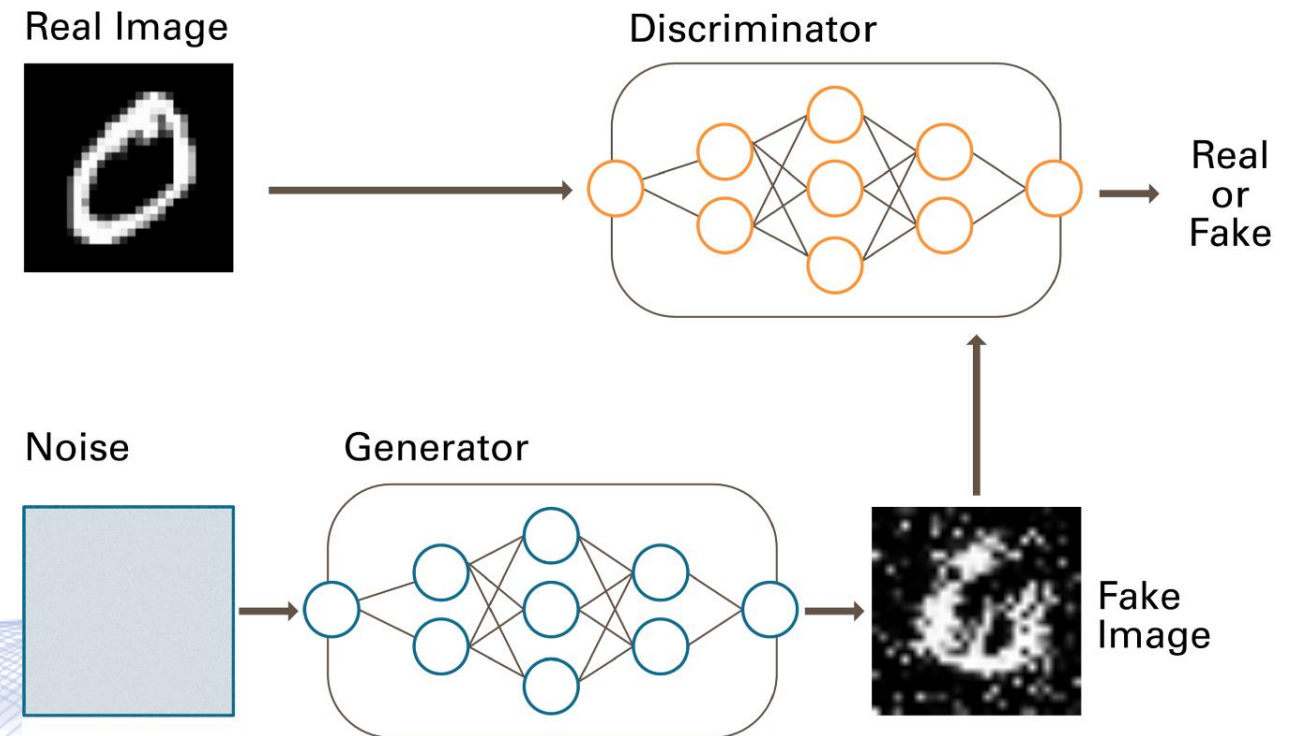
where  $\mathbf{y} \sim \hat{p}(\mathbf{y}|\mathbf{x})$ .

- This is the **multi-class** version of the training objective, for  $m > 2$ .
- In the special **binary** version ( $m = 2$ ), we typically assume that we have a **positive** and a **negative** class.

# GAN theory

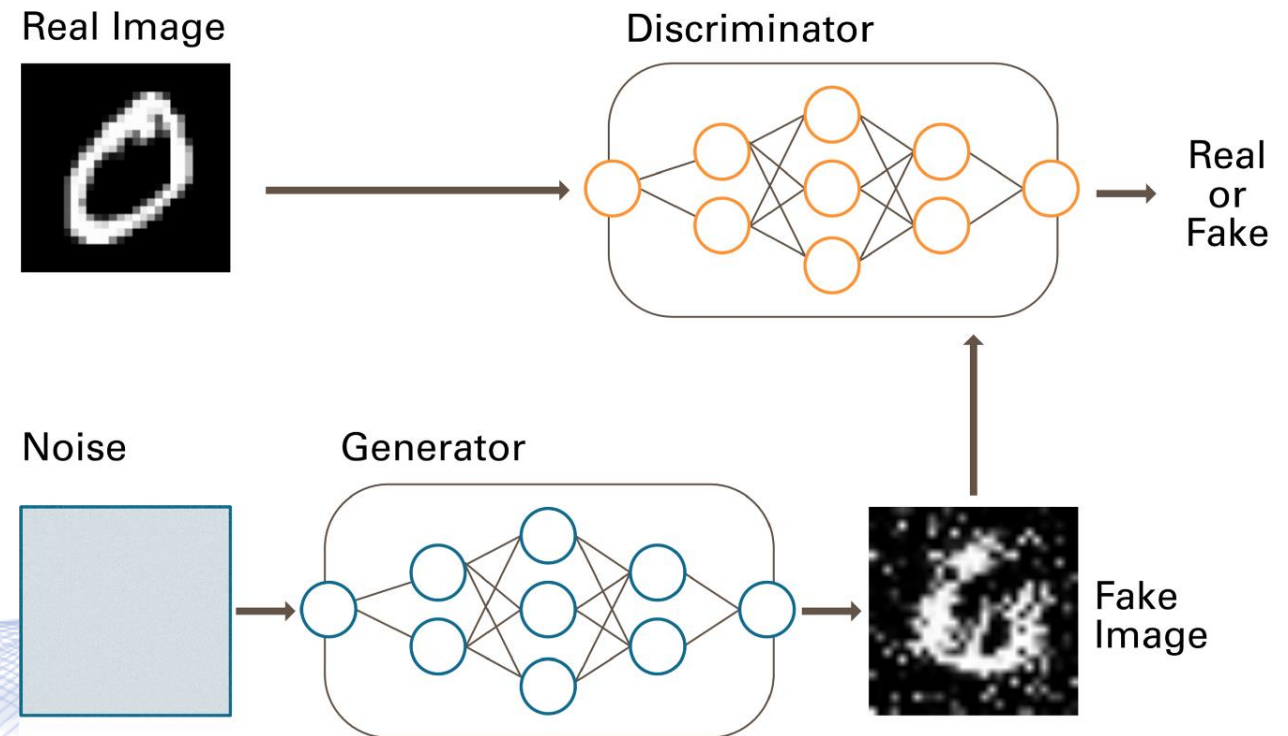
A GAN is composed of two interacting DNNs:

- A **Generator** function  $\hat{y} = G(\mathbf{z}; \boldsymbol{\theta}_G)$ , parameterized by vector  $\boldsymbol{\theta}_G$ .
- A **Discriminator** function  $\hat{y} = D(\mathbf{q}; \boldsymbol{\theta}_D)$  parameterized by vector  $\boldsymbol{\theta}_D$ .
- Optimal  $\boldsymbol{\theta}_D$  and  $\boldsymbol{\theta}_G$  are typically found using iterative gradient descent and error back-propagation.



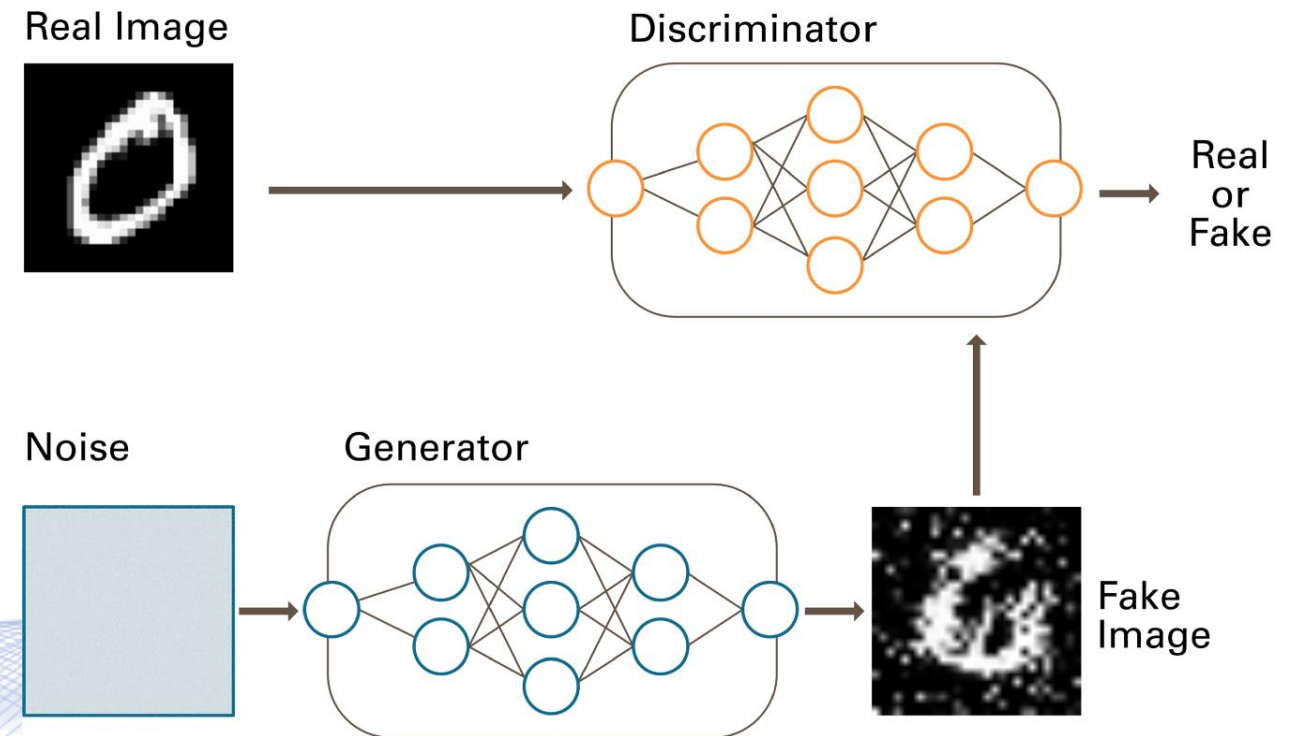
# GAN theory

- $G$  and  $D$  are trained independently in an alternating fashion, but ***the individual loss of each one at each training iteration is a function of the output of the other one.***
- During training, only  $D$  directly accesses the training dataset  $\mathcal{D}$ .



# GAN theory

- Serendipitously, this acts as a built-in safeguard mechanism of  $G$  against **overfitting** to  $p_{\mathbf{x}}(\mathbf{x})$ .
- After training is complete,  $D$  is **typically discarded** and  $G$  can be employed for sampling fake data points from an approximation of  $p_{\mathbf{x}}(\mathbf{x})$ .





# GAN theory

- The training optimization problem for  $D$  can be formulated using the ***cross-entropy loss for binary classification problems***, assuming that positive (real) examples are drawn from the training dataset and negative ones (fakes) from the output of  $G$ :

$$\begin{aligned} \theta_D^{*m} &= \operatorname{argmin}_{\theta_D} -E_{\mathbf{x}}\{\log D(\mathbf{x})\} - E_{\mathbf{z}}\{\log (1 - D(G(\mathbf{z})))\} = \\ &= \operatorname{argmax}_{\theta_D} E_{\mathbf{x}}\{\log D(\mathbf{x})\} + E_{\mathbf{z}}\{\log (1 - D(G(\mathbf{z})))\}, \end{aligned}$$

where  $\mathbf{x} \sim \hat{p}(\mathbf{x})$ ,  $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ .

# GAN theory

- This is slightly simplified relatively to the typical binary cross-entropy loss, because we know that:
  - $y = 1$ , for the first expectation (when  $\mathbf{q}$  is drawn from  $\mathbf{X}$ ), and
  - $y = 0$ , for the second expectation (when  $\mathbf{q}$  is drawn from the output of  $G$ ).

# GAN theory

The training optimization for  $G$  may be formulated in different ways.

- **Minimax optimization** directly **penalizes the ability of  $D$  to successfully detect fake data points** generated by  $G$ :

$$\theta_G^{*m} = \operatorname{argmin}_{\theta_G} - [-E_{\mathbf{x}} \{\log D(\mathbf{x})\} - E_{\mathbf{z}} \{\log (1 - D(G(\mathbf{z})))\}] = \operatorname{argmin}_{\theta_G} E_{\mathbf{z}} \{\log (1 - D(G(\mathbf{z})))\},$$

where  $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ .

- It allows us to theoretically summarize overall GAN training as a minimax game, at a high level of abstraction:

$$\{\theta_G^{*m}, \theta_D^{*m}\} = \operatorname{arg min}_{\theta_G} \max_{\theta_D} E_{\mathbf{x}} \{\log D(\mathbf{x})\} + E_{\mathbf{z}} \{\log (1 - D(G(\mathbf{z})))\},$$

# GAN theory

- Minimax optimization function takes the following numerical form:

$$\{\boldsymbol{\theta}_G^{*m}, \boldsymbol{\theta}_D^{*m}\} = \arg \min_{\boldsymbol{\theta}_G} \max_{\boldsymbol{\theta}_D} \frac{1}{N} \sum_{i=1}^N (J_{iD} + J_{iG}) =$$

$$\arg \min_{\boldsymbol{\theta}_G} \max_{\boldsymbol{\theta}_D} \frac{1}{N} \sum_{i=1}^N \log D(\mathbf{x}_i) + \frac{1}{N} \sum_{i=1}^N \log (1 - D(G(\mathbf{z}_i))),$$



# GAN theory

- An alternative training optimization objective for  $G$  that does not suffer from this low loss signal problem is the ***heuristic optimization***:

$$\theta_G^{*h} = \operatorname{argmin}_{\theta_G} - E_{\mathbf{z}} \{ \log (D(G(\mathbf{z}))) \},$$

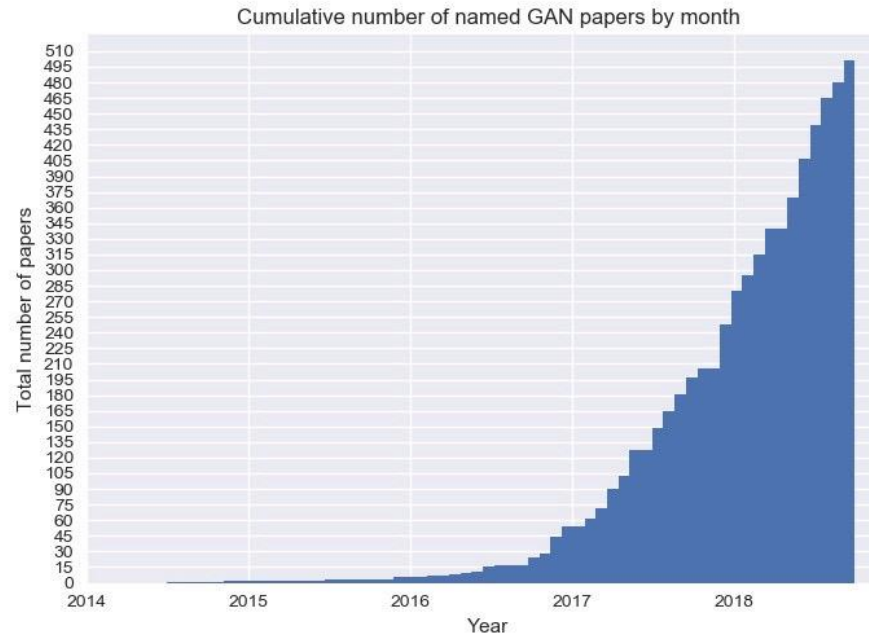
where  $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ .

- Discriminator optimization function remains unaltered:

$$\begin{aligned} \theta_D^{*h} &= \operatorname{argmin}_{\theta_D} - E_{\mathbf{x}} \{ \log D(\mathbf{x}) \} - E_{\mathbf{z}} \{ \log (1 - D(G(\mathbf{z}))) \} \\ &= \operatorname{argmax}_{\theta_D} E_{\mathbf{x}} \{ \log D(\mathbf{x}) \} + E_{\mathbf{z}} \{ \log (1 - D(G(\mathbf{z}))) \}, \end{aligned}$$

# GANs in Multimedia Creation

- GANs appeared in 2014 [GOO2014].
- Their use exploded since 2016.



# GANs in Multimedia Creation



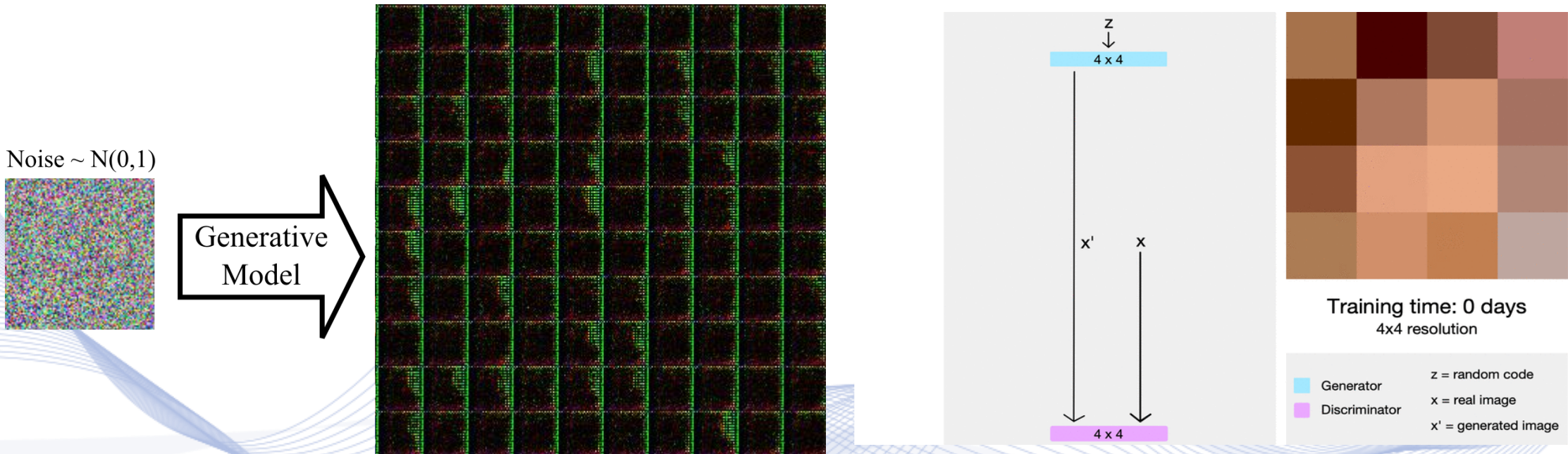
- Over 600 variations exist as of today applied in various domains [HIN].

GAN	C-RNN-GAN	Unim2im	UNIT	PSGAN	AE-GAN	SS-GAN	AttnGAN	Pip-GAN	PPAN	DA-GAN	PM-GAN	NetGAN
CGAN	CC-GAN	VIGAN	DualGAN	RankGAN	AlignGAN	VIGAN	BCGAN	pix2pixHD	RAN	DP-GAN	ProGanSR	OCAN
LAPGAN	DTN	WGAN	FF-GAN	RPGAN	APE-GAN	ARIGAN	BicycleGAN	Sobolev GAN	SGAN	DPGAN	PS-GAN	OT-GAN
CatGAN	GMAN	acGAN	GoGAN	RWGAN	ARDA	CausalGAN	CatGAN	StarGAN	SRPGAN	First Order GAN	ReConNN	PGGAN
DCGAN	lcGAN	ArtGAN	MAD-GAN	SBADA-GAN	DAN	D2GAN	CoAtt-GAN	TGAN	ST-CGAN	GC-GAN	SAGA	Sdf-GAN
VAE-GAN	LSGAN	Bayesian GAN	MAGAN	SD-GAN	I-GAN	ExposureGAN	ConceptGAN	tripletGAN	Super-FAN	LB-GAN	sGAN	Social GAN
GRAN	MV-BiGAN	BS-GAN	SL-GAN	VEEGAN	LD-GAN	ExprGAN	Cover-GAN	VA-GAN	TV-GAN	MAGAN	Sketcher-Refiner GAN	Spike-GAN
S <sup>2</sup> GAN	pix2pix	MalGAN	Softmax GAN	WS-GAN	LeGAN	GAMN	D-GAN	XGAN	UGACH	ND-GAN	SyncGAN	ST-GAN
MGAN	RenderGAN	MaliGAN	TAN	ARAE	MMGAN	GraspGAN	DAGAN	ZipNet-GAN	UV-GAN	PGD-GAN	TGANs-C	Text2Shape
BiGAN	SAD-GAN	McGAN	TP-GAN	BCGAN	MoCoGAN	LDAN	DeblurGAN	ACGAN	VGAN	RadialGAN	UT-SCA-GAN	tiny-GAN
GAN-CLS	SGAN	ST-GAN	VariGAN	CAN	ResGAN	LeakGAN	DNA-GAN	CA-GAN	weGAN	SAR-GAN	AdvEntuRe	VOS-GAN
ALI	SSL-GAN	WaterGAN	VAW-GAN	Chekhov GAN	SisGAN	MD-GAN	DRPAN	ComboGAN	AdvGAN	SCH-GAN	AVID	3D-PhysNet
CoGAN	TGAN	AEGAN	WGAN-GP	crVAE-GAN	ss-InfoGAN	MuseGAN	FIGAN	DF-GAN	CFG-GAN	StainGAN	BourGAN	AF-DCGAN
f-GAN	Unrolled GAN	AM-GAN	$\beta$ -GAN	DelIGAN	SSGAN	OptionGAN	FSEGAN	Dynamics Transfer GAN	CipherGAN	SWGAN	BRE	BEAM
Improved c	VGAN	AnoGAN	Bayesian GAN	DistanceGAN	SteinGAN	PassGAN	FTGAN	EnergyWGAN	Cross-GAN	VoiceGAN	cd-GAN	CorrGAN
InfoGAN	AL-CGAN	BEGAN	CaloGAN	DSP-GAN	VRAL	RefineGAN	GANDI	ExGAN	dp-GAN	WaveGAN	cowboy	D-WCGAN
SketchGAN	MARTA-GAN	CS-GAN	Conditional cycleGAN	Dualing GAN	3D-RecGAN	Splitting GAN	GPU	f-CLSWGAN	ecGAN	Attention-GAN	CSG	Defo-Net
Context-R	MDGAN	CVAE-GAN	Cramèr GAN	Fila-GAN	ABC-GAN	$\Delta$ -GAN	HAN	FusionGAN	FusedGAN	B-DCGAN	Defense-GAN	DSH-GAN
EBGAN	MPM-GAN	CycleGAN	DR-GAN	GANCS	ASDL-GAN	CM-GAN	HP-GAN	G2-GAN	GeoGAN	BAGAN	DialogWAE	DTR-GAN
IAN	PPGN	DiscoGAN	DRAGAN	GMM-GAN	BGAN	GAN-ATV	HR-DCGAN	GAGAN	GLCA-GAN	BranchGAN	DTLC-GAN	DVGAN
iGAN	PrGAN	GP-GAN	ED//GAN	IWGAN	CDcGAN	GAP	IfcVAEGAN	GAN-RS	LAC-GAN	D2IA-GAN	FairGAN	EAR
SeqGAN	SGAN	LR-GAN	EGAN	PAN	CGAN	GP-GAN	In2I	GANG	MaskGAN	DBLRGAN	Fairness GAN	FBGAN
SRGAN	SimGAN	MedGAN	Fisher GAN	Perceptual GAN	contrast-GAN	Progressive GAN	Iterative-GAN	GANosaic	SG-GAN	E-GAN	FakeGAN	FusionGAN
VGAN	StackGAN	MIX+GAN	Flow-GAN	PixelGAN	Coulomb GAN	PS <sup>2</sup> -GAN	IVE-GAN	IdCycleGAN	SketchyGAN	ELEGANT	FBGAN	Graphical-GAN
3D-GAN	textGAN	RTT-GAN	GeneGAN	RCGAN	DM-GAN	SVSGAN	IVGAN	manifold-WGAN	tempoGAN	Fictitious GAN	FC-GAN	IterGAN
AC-GAN	AdaGAN	SEGAN	Geometric GAN	RNN-WGAN	GAN-sep	TGAN	KBGAN	MC-GAN	UGAN	GAAN	GAF	M-AAE
AffGAN	ID-CGAN	SeGAN	IRGAN	SegAN	GAN-VFS	3D-ED-GAN	KGAN	MIL-GAN	AmbientGAI	GONet	GAN Q-learning	MelanoGAN
GAWWN	LAGAN	SGAN	MMD-GAN	TextureGAN	MGGAN	ABC-GAN	LGAN	MS-GAN	ATA-GAN	memoryGAN	GAN-SD	MGGAN
b-GAN	LS-GAN	TAC-GAN	ORGAN	$\alpha$ -GAN	PGAN	ACTUAL	MLGAN	PacGAN	C-GAN	MTGAN	GAN-Word2Vec	ModularGAN
	SalGAN	Triple-GAN	Pose-GAN	3D-IWGAN	SN-GAN	AttGAN	ORGAN	PN-GAN	CapsuleGAN	NCE-GAN	GANAX	NAN



# GANs in Multimedia Creation

- Nowadays, main focus of generative models is generating artificially realistic data or transforming existing [MED].





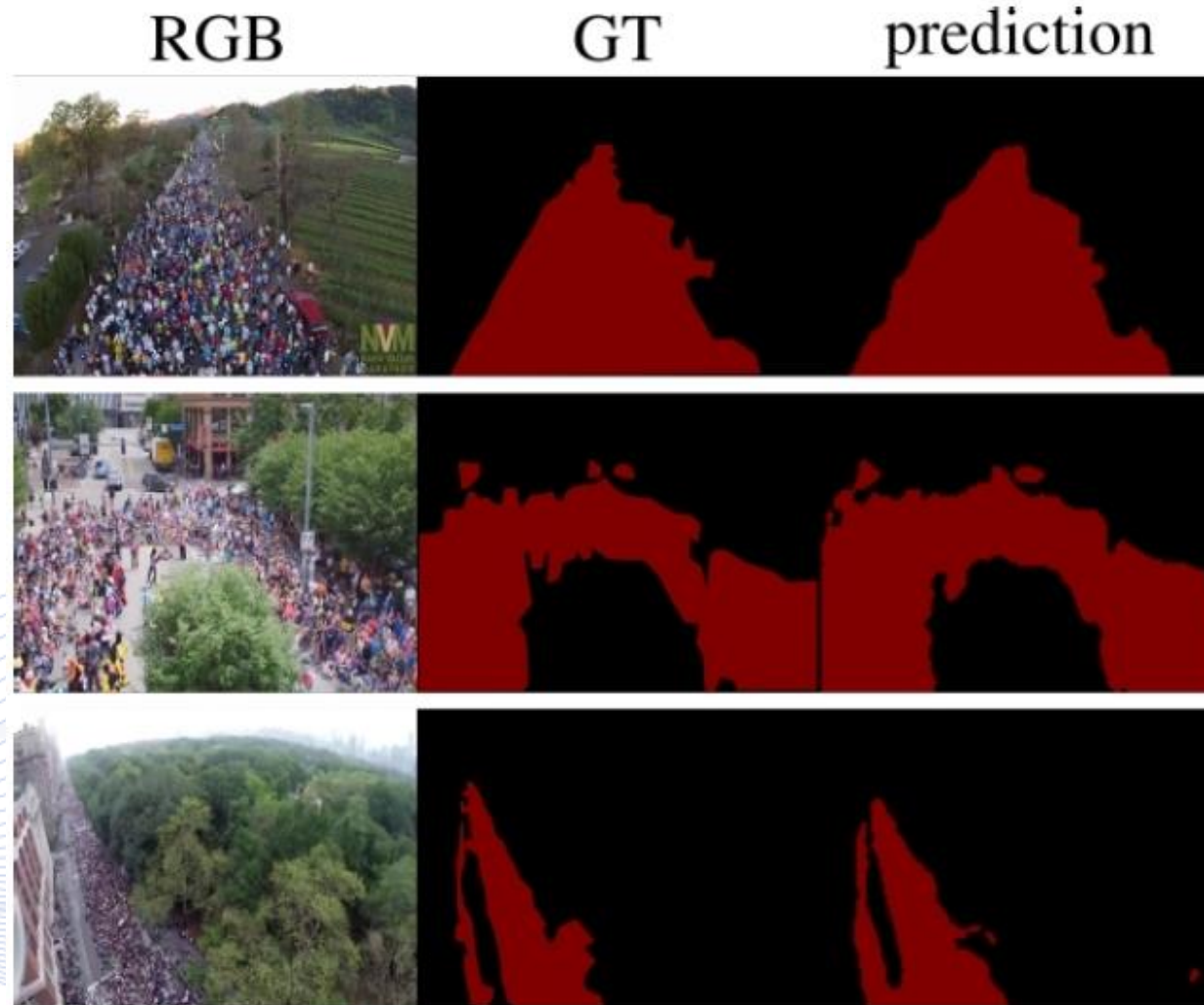
# Conditional GAN



**Conditional GAN (cGAN)** is one of the most important GAN variants.






- It allows **a greater degree of control** over the generated images.
- For instance, when training with multi-class datasets, a vector encoding the class label is given as input both to  $D$  (along with each  $\hat{y}_i$  or  $x_i$ ) and to  $G$  (along with  $z$ ).
- Thus,  $G$  learns to **conditionally map the noise vector to a synthetic image**, given a class label, while  $D$  **is trained more effectively** since it knows the class label of the image that it must recognize as either “fake” or “real”.


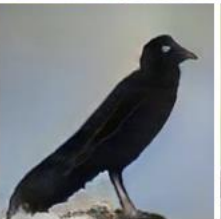


# Conditional GAN



# GAN for Text-to-Image Synthesis



Text description	This flower has petals that are white and has pink shading	This flower has a lot of small purple petals in a dome-like configuration	This flower has long thin yellow petals and a lot of yellow anthers in the center	This flower is pink, white, and yellow in color, and has petals that are striped	This flower is white and yellow in color, with petals that are wavy and smooth	This flower has upturned petals which are thin and orange with rounded edges	This flower has petals that are dark pink with white edges and pink stamen
							

Text description	This bird is red and brown in color, with a stubby beak	The bird is short and stubby with yellow on its body	A bird with a medium orange bill white body gray wings and webbed feet	This small black bird has a short, slightly curved bill and long legs	A small bird with varying shades of brown with white under the eyes	A small yellow bird with a black crown and a short black pointed beak	This small bird has a white breast, light grey head, and black wings and tail
							



# GAN for Video-to-Text Synthesis



**LSTM:** a woman is cooking  
**LSTM-GAN:** a woman is **frying** some food  
**Ground-Truth:** she is cooking on the fish



**LSTM:** a man is dancing  
**LSTM-GAN:** a **group of** men are dancing on the stage  
**Ground-Truth:** people are dancing on stage



**LSTM:** a man is jumping on a motorcycle  
**LSTM-GAN:** a man is riding a motorcycle  
**Ground-Truth:** a man is riding a motorcycle



**LSTM:** a man is pouring tomato into a pot  
**LSTM-GAN:** a man is pouring some **sauce** into a pot  
**Ground-Truth:** a person pours tomato sauce in a pot



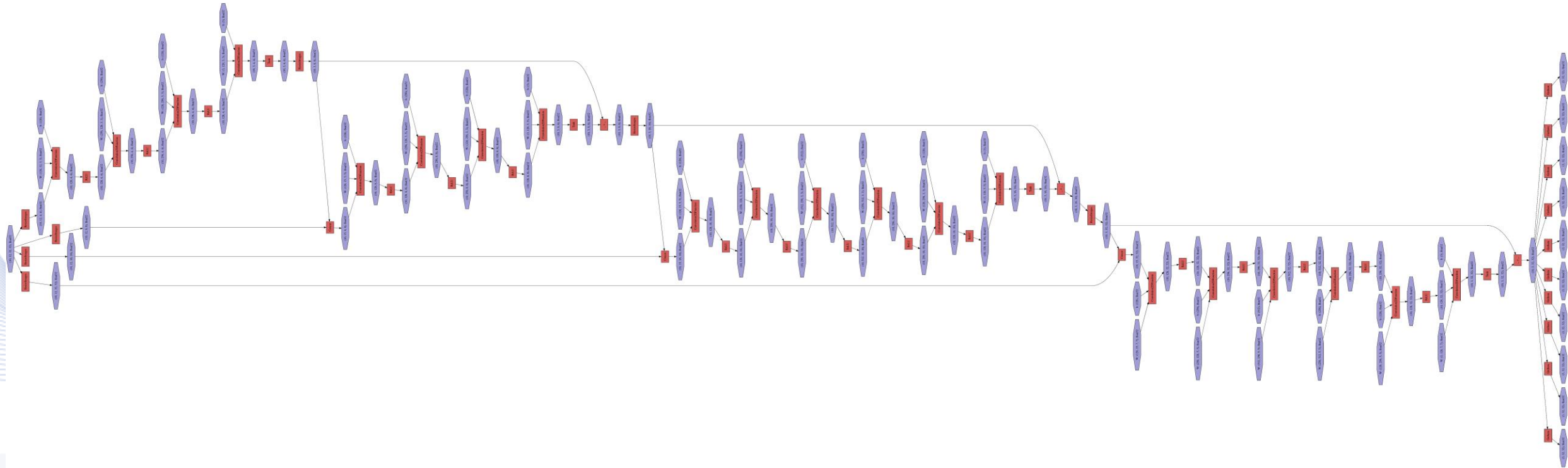
**LSTM:** a man is cutting a bread  
**LSTM-GAN:** a man is cutting a **loaf of** bread  
**Ground-Truth:** a man is cuts a loaf of bread



**LSTM:** a man is cooking a pot  
**LSTM-GAN:** a person is making some food  
**Ground-Truth:** a men is preparing some food

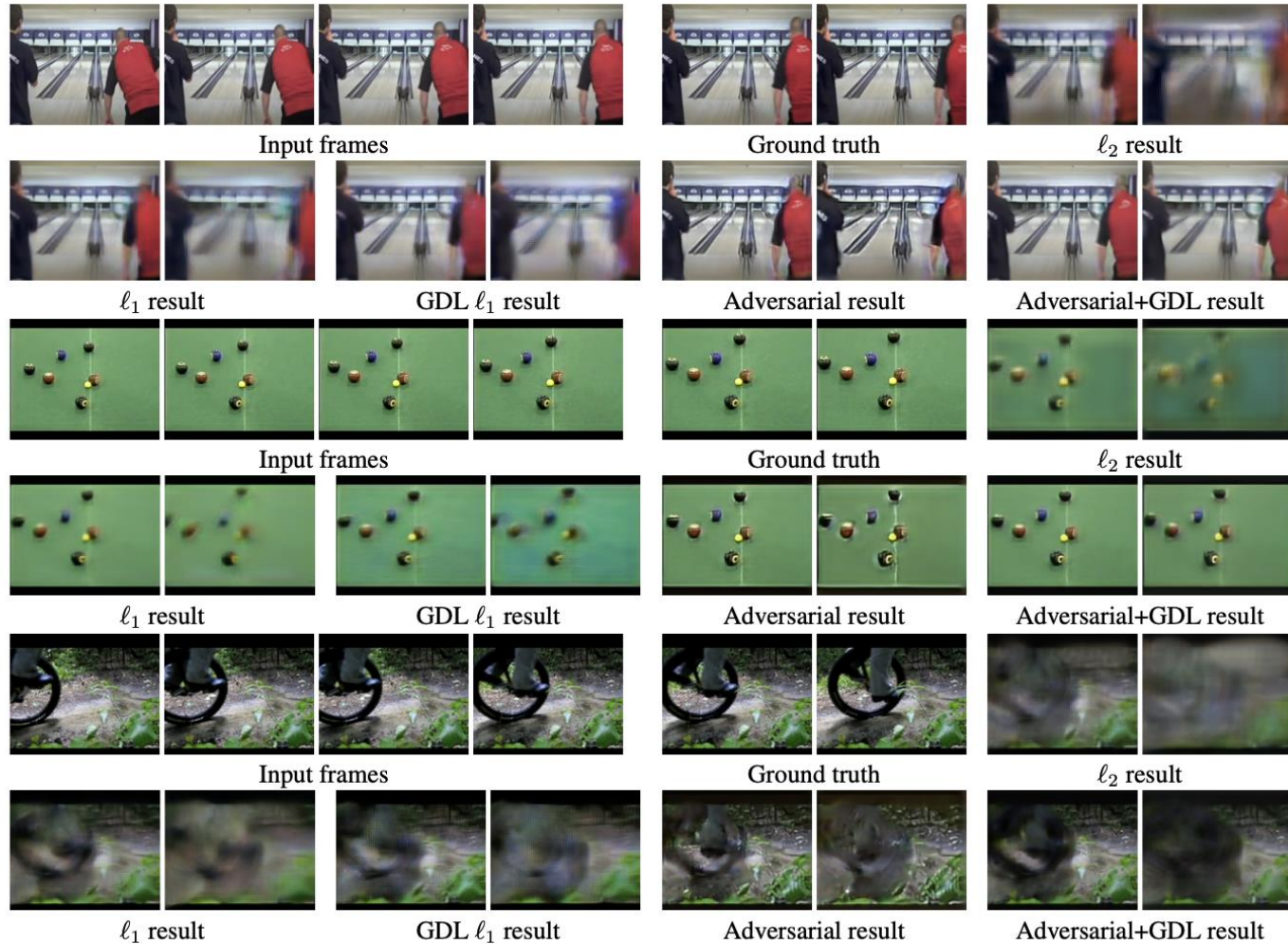


# GAN for Video Frame Prediction



- GAN network can generate future video frames, given an input video sequence [MAT2015].
- It is very useful for video compression and video frame interpolation.

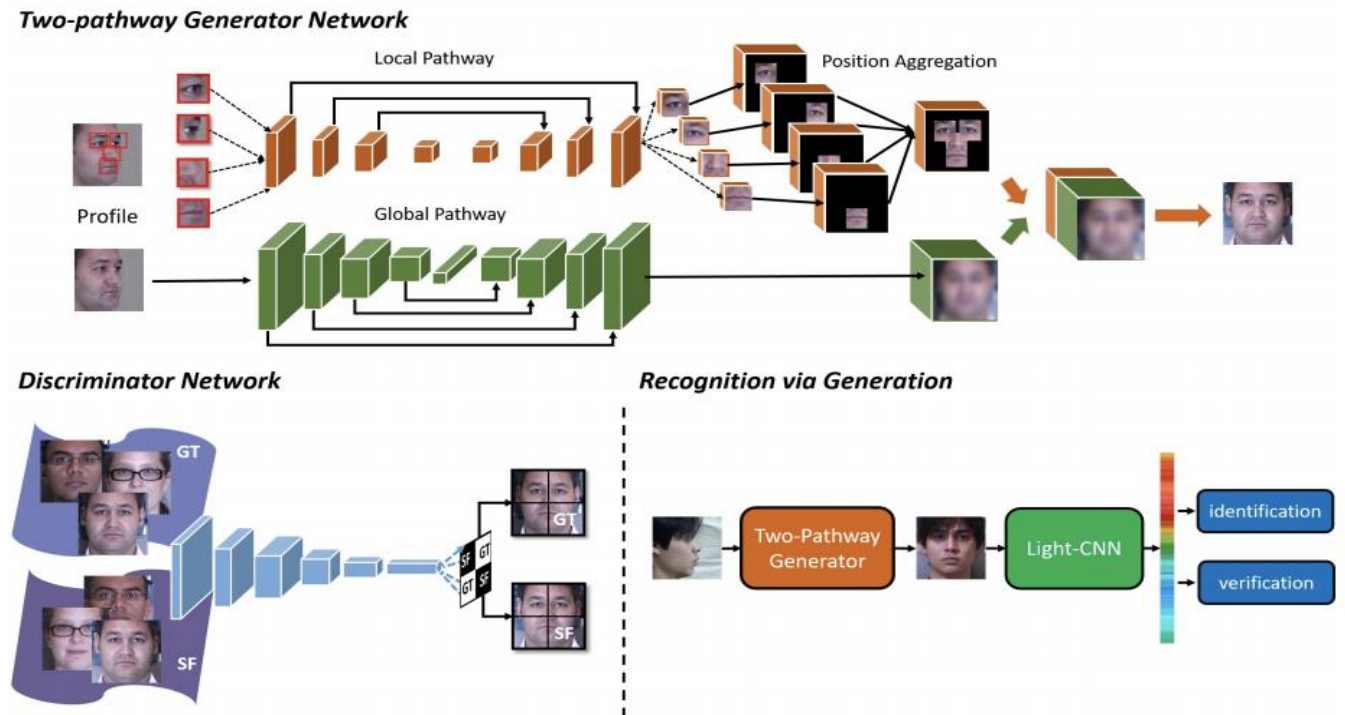
# GAN for Video Frame Prediction



\*Image Gradient Difference Loss (GDL).

# GAN for Face Synthesis

- **TP-GAN** synthesises different views of an input *facial image*.
- **Example:** given a profile image, synthesize a frontal view.
- It can be employed for better face recognition performance [HUA2017].
- The Generator contains **two neural pathways**: one for global face appearance and one for local details. Their results are combined.





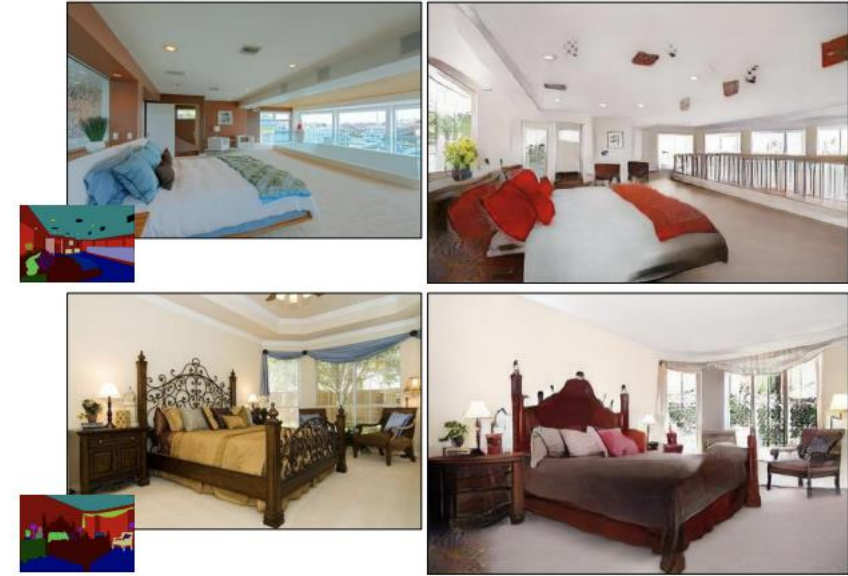
# GAN for Face Synthesis



Synthesis results under various illuminations. The first row is the synthesized image, the second row is the input.



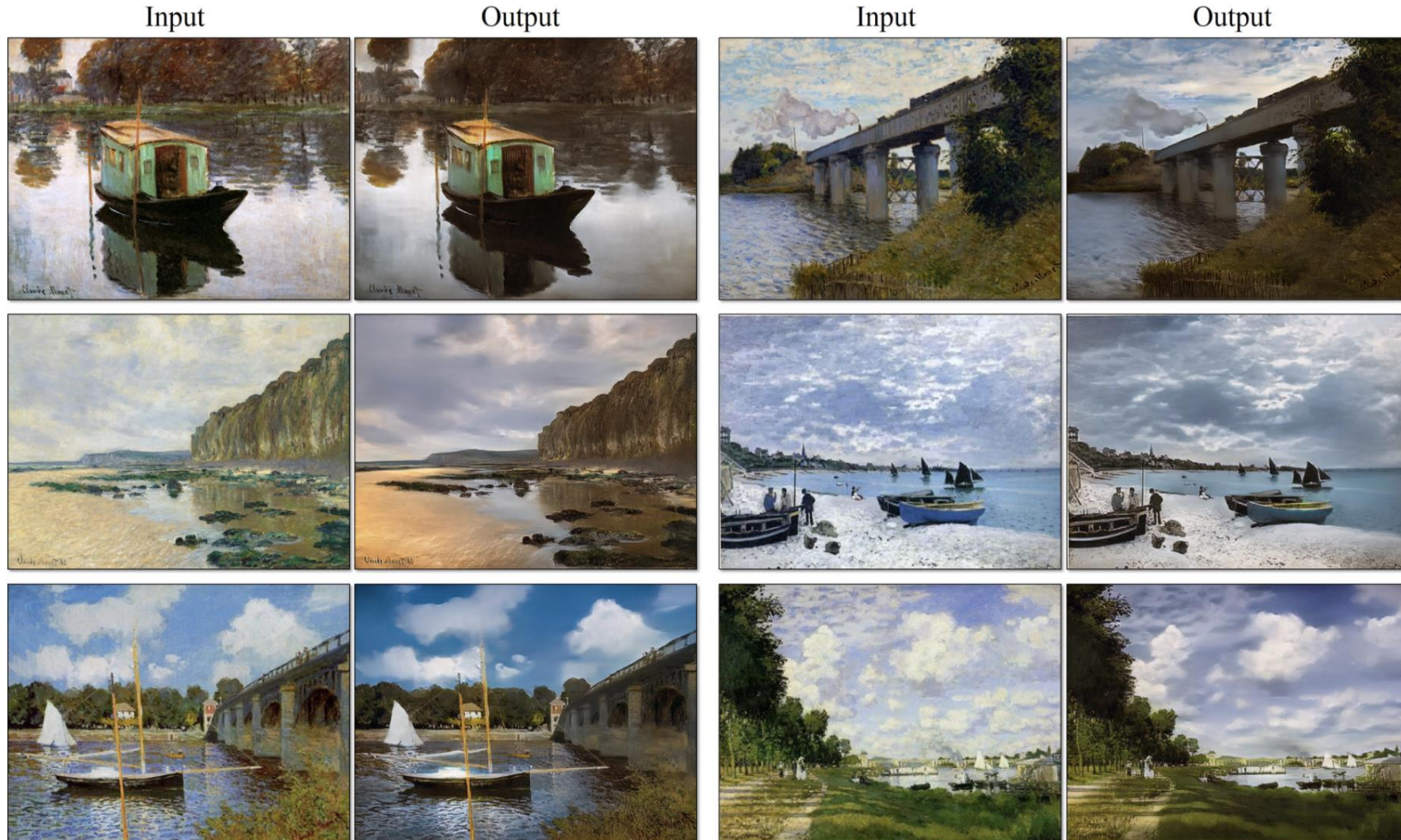
# GAN for Image-to-Image Translation





# GAN for Image-to-Image Translation

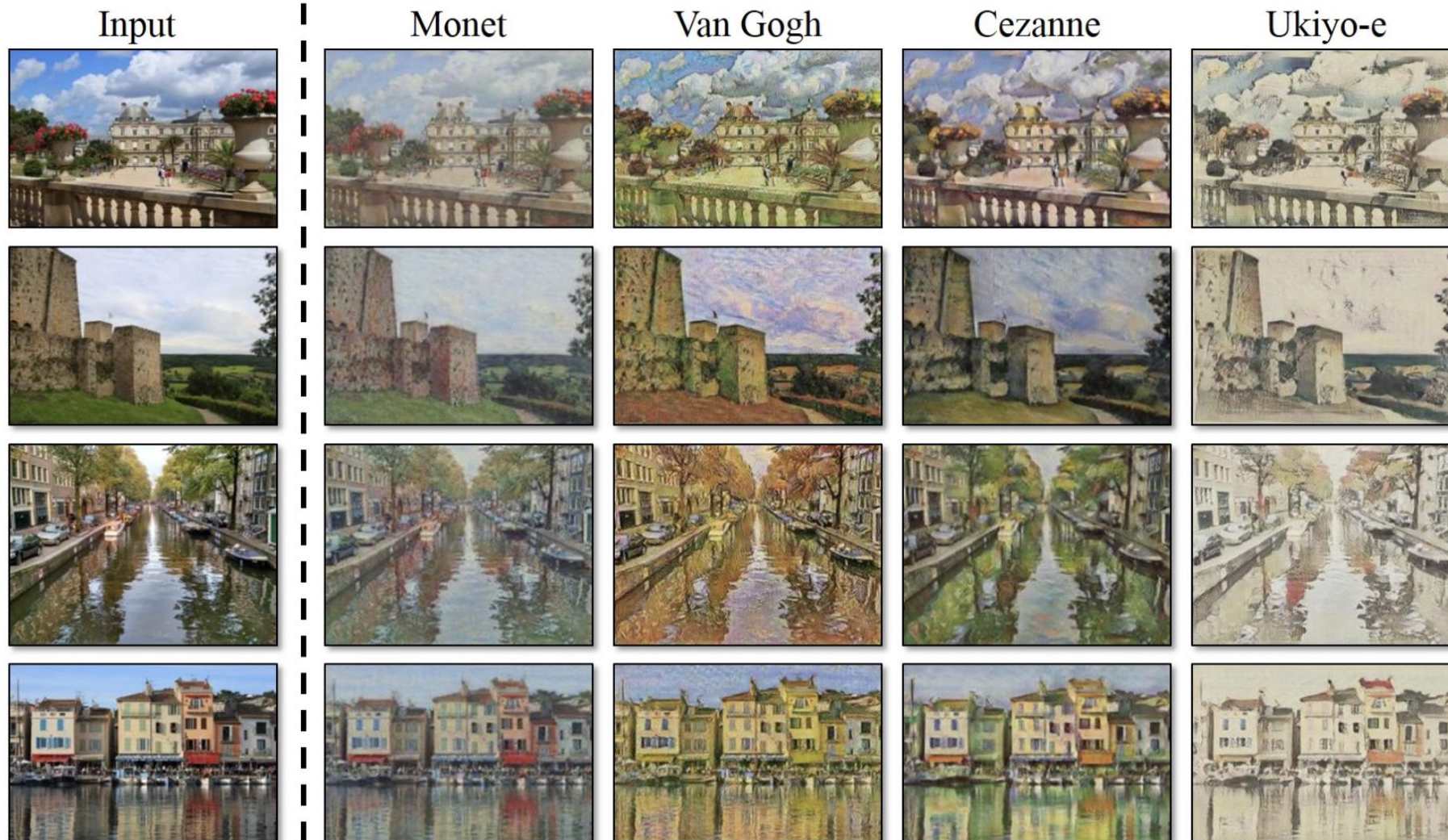
## Monet Paintings to Photos





# GAN for Image-to-Image Translation

Collection Style Transfer



# GAN for Image-to-Image Translation

- **StarGAN** [CHOI2016] is a CycleGAN variant that achieves **multi-domain Image-to-Image translation with a single Generator**.
- Instead of learning a fixed translation (e.g., black-to-blond hair), its input is a pair {image, label}.
- It learns to flexibly translate the image into the label domain (e.g., “happy” or “sad”).

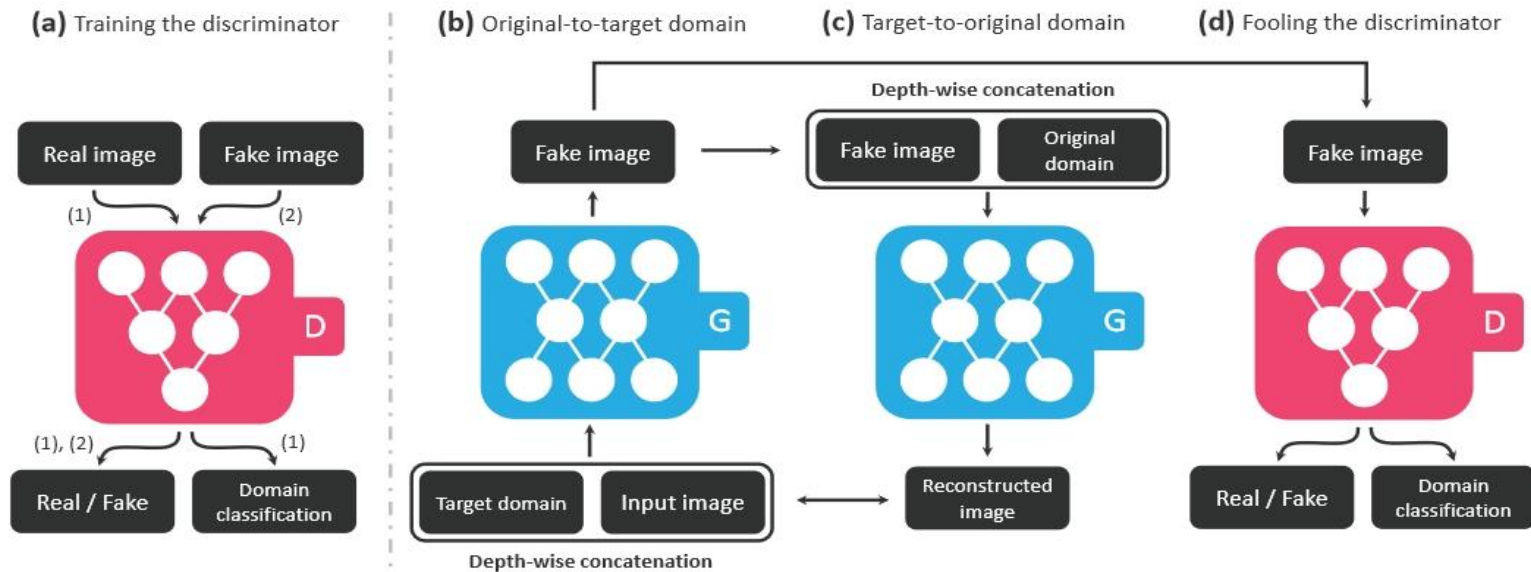
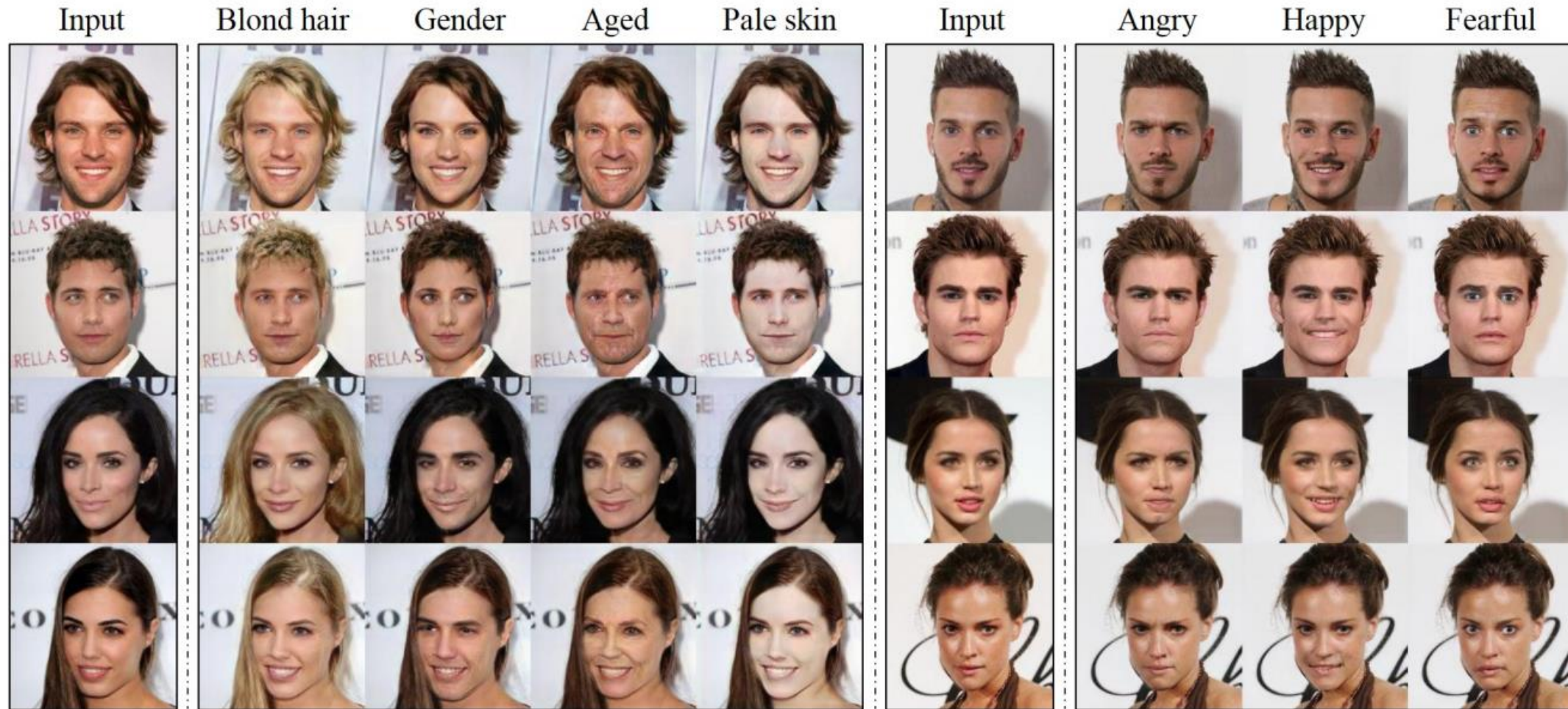


Figure 3. Overview of StarGAN, consisting of two modules, a discriminator  $D$  and a generator  $G$ . (a)  $D$  learns to distinguish between real and fake images and classify the real images to its corresponding domain. (b)  $G$  takes in as input both the image and target domain label and generates an fake image. The target domain label is spatially replicated and concatenated with the input image. (c)  $G$  tries to reconstruct the original image from the fake image given the original domain label. (d)  $G$  tries to generate images indistinguishable from real images and classifiable as target domain by  $D$ .

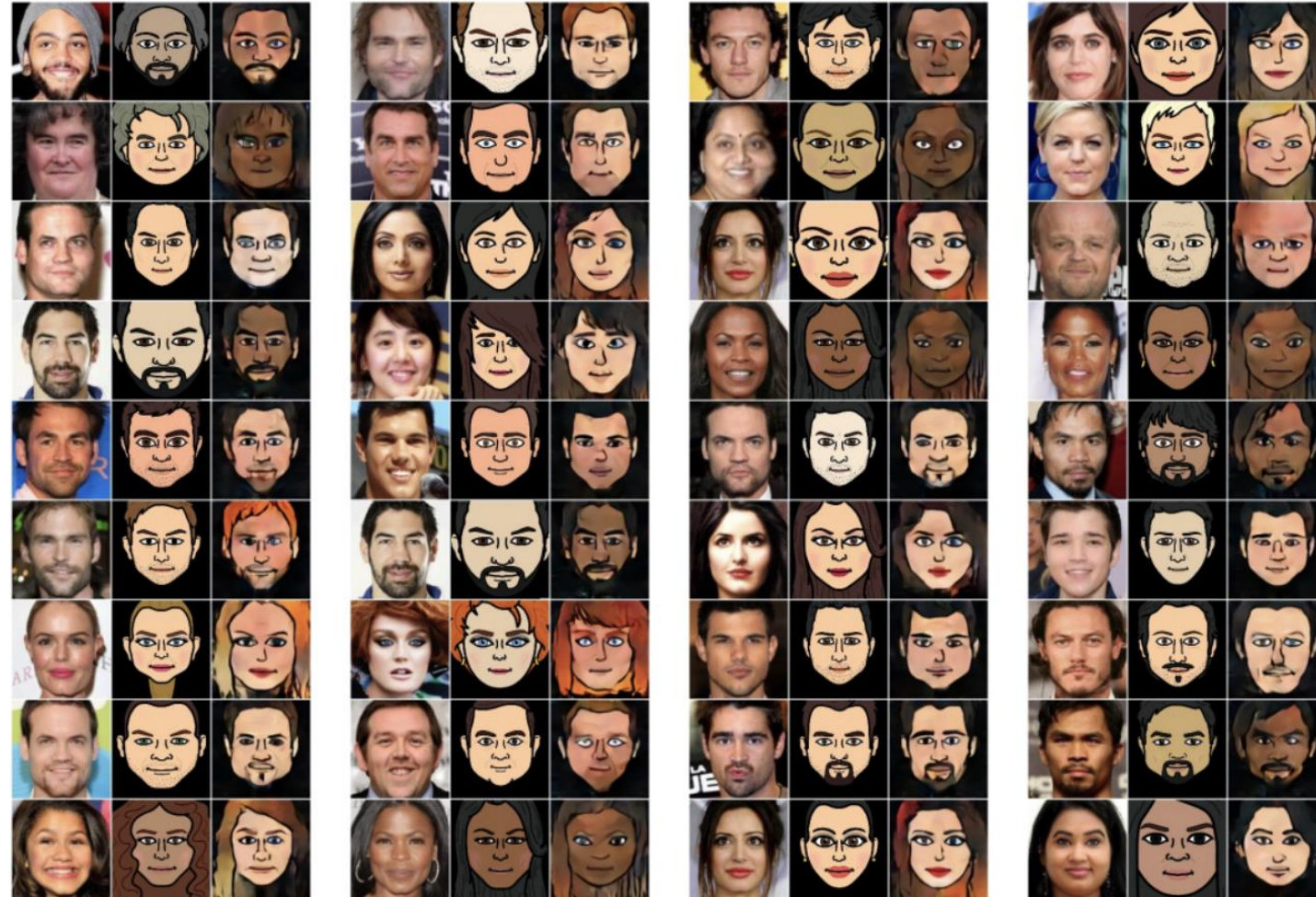


# GAN for Image-to-Image Translation





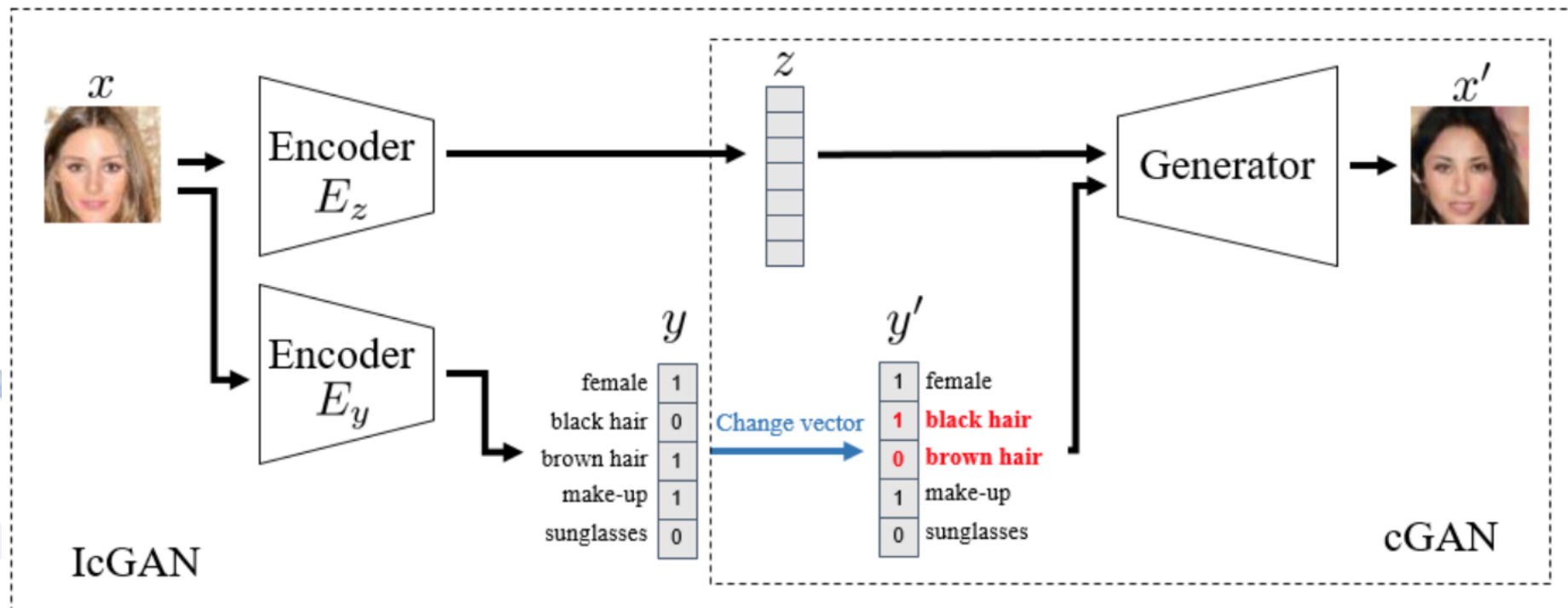
# GAN for Cross-Domain Image Generation



Shown, side by side are sample images from the CelebA dataset, the emoji images created manually using a web interface (for validation only), and the result of the unsupervised DTN. See Tab. 4 for retrieval performance.

# GAN for Image Editing

- **Invertible cGAN (IcGAN)** reconstructs or edits images with specific attributes [PER2016].





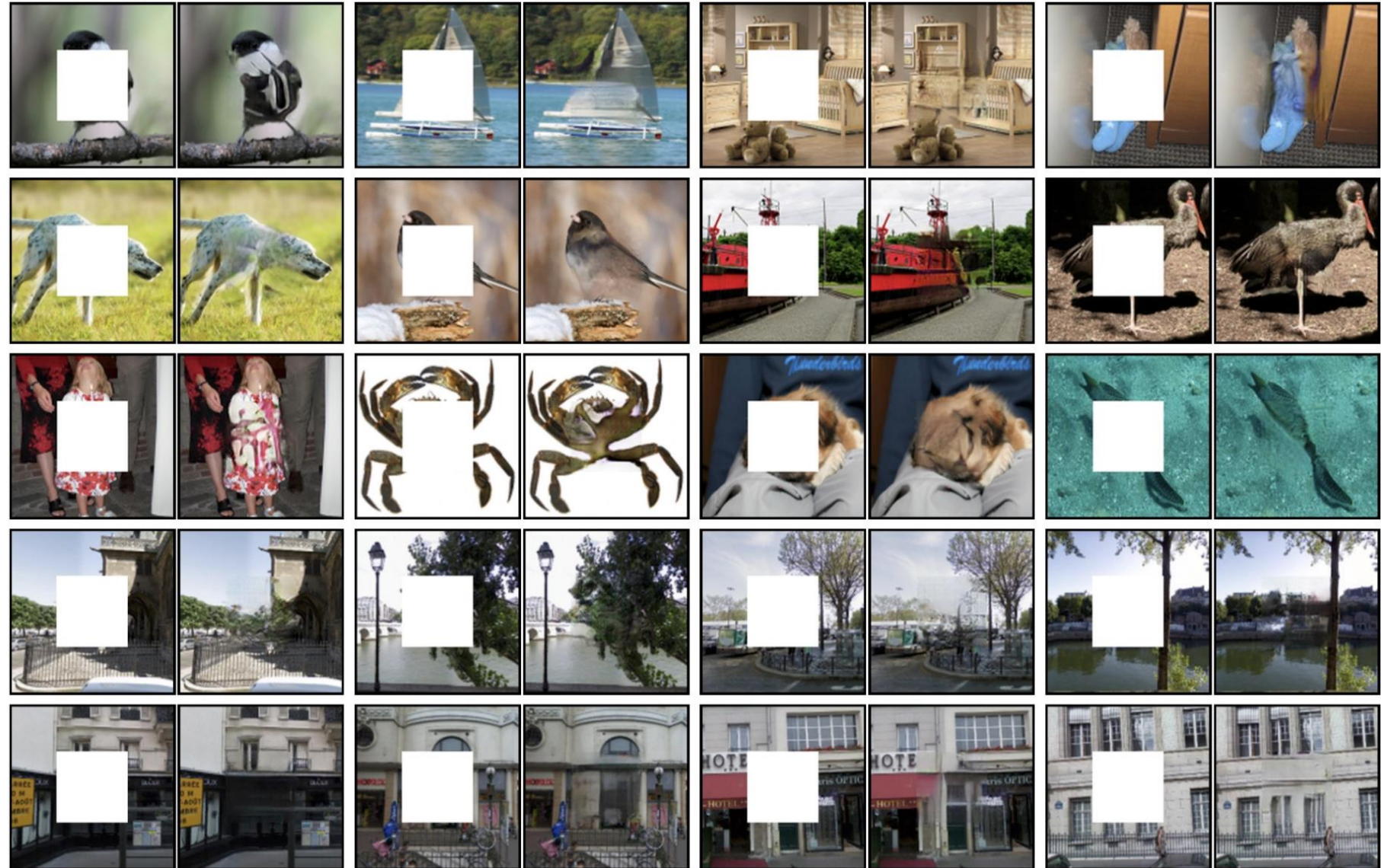
# GAN for Image Inpainting



Input

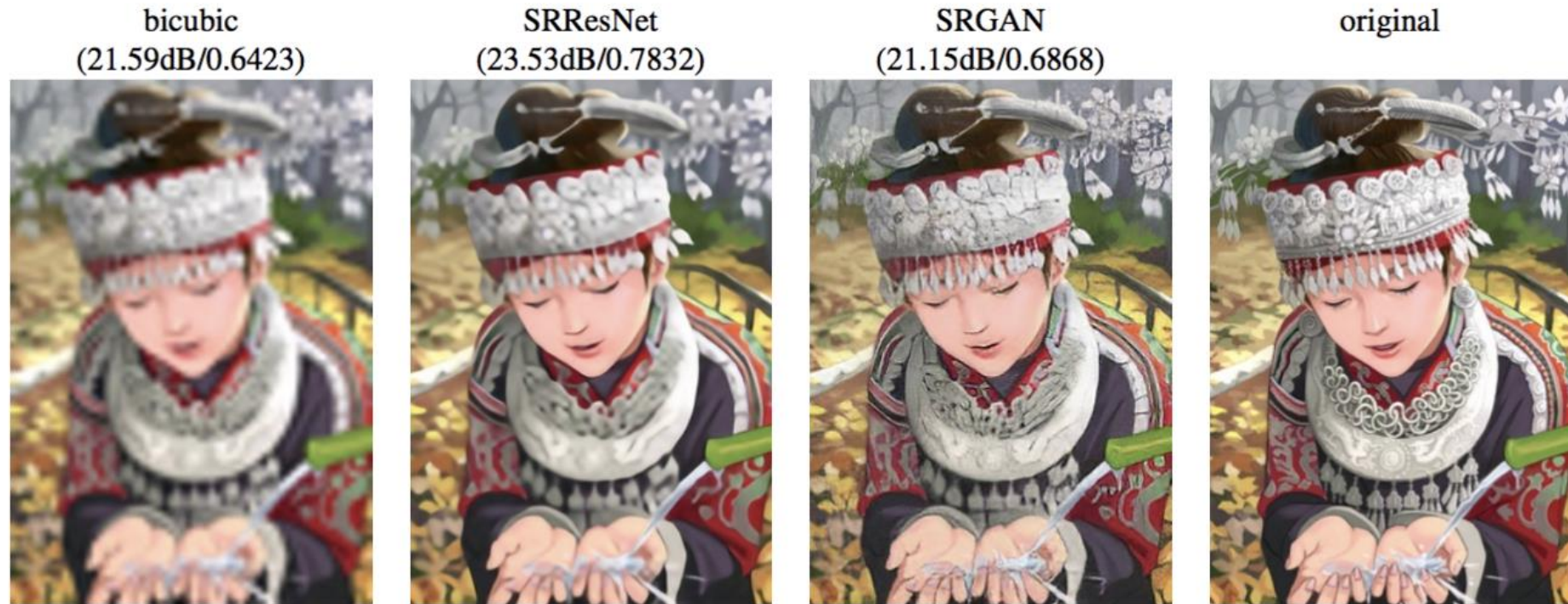
GAN

Photoshop





# GAN for Image Super-resolution



From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

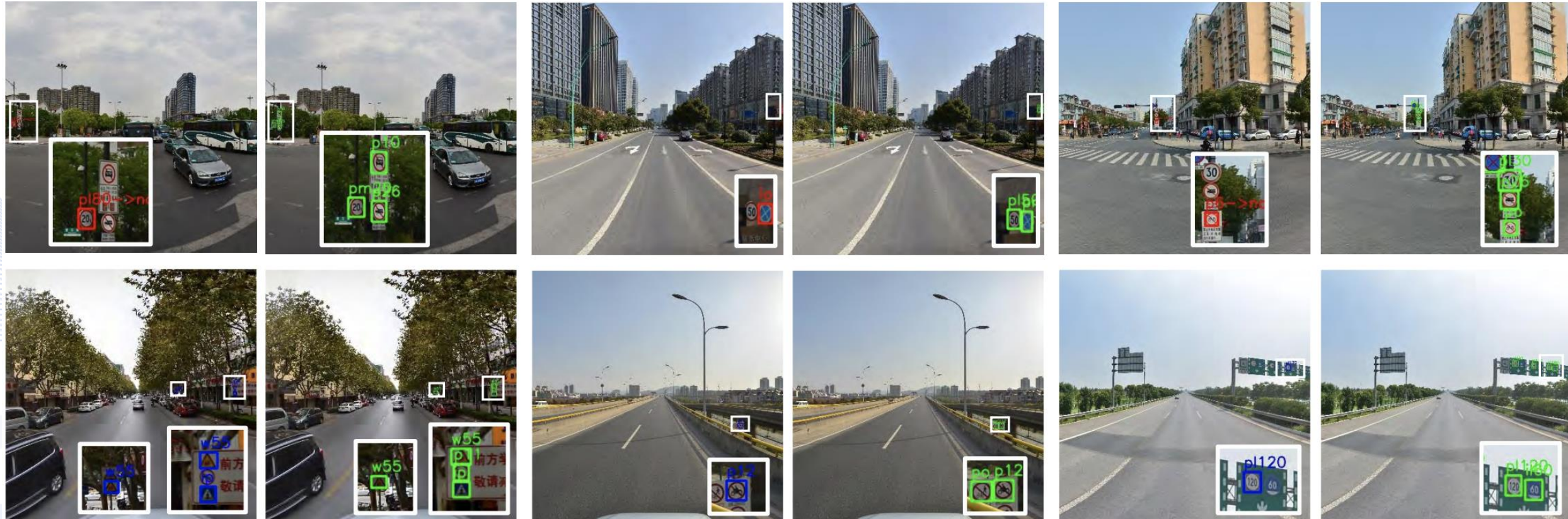
# GAN for Image Deblurring



Blurred – left, DeblurGAN – center, ground truth sharp – right.

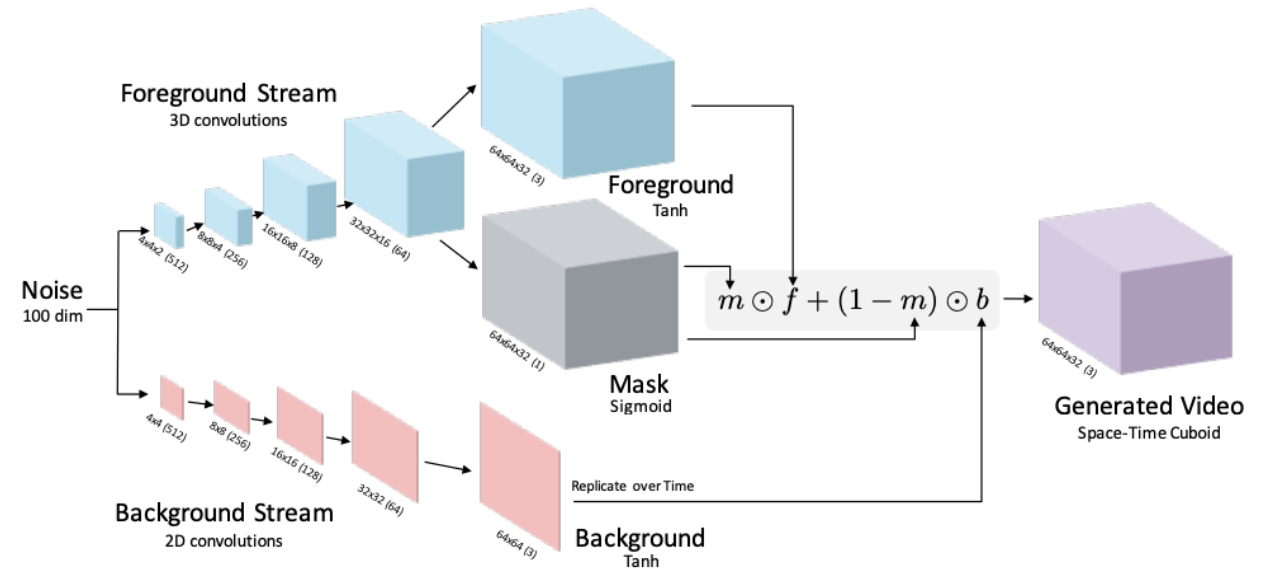
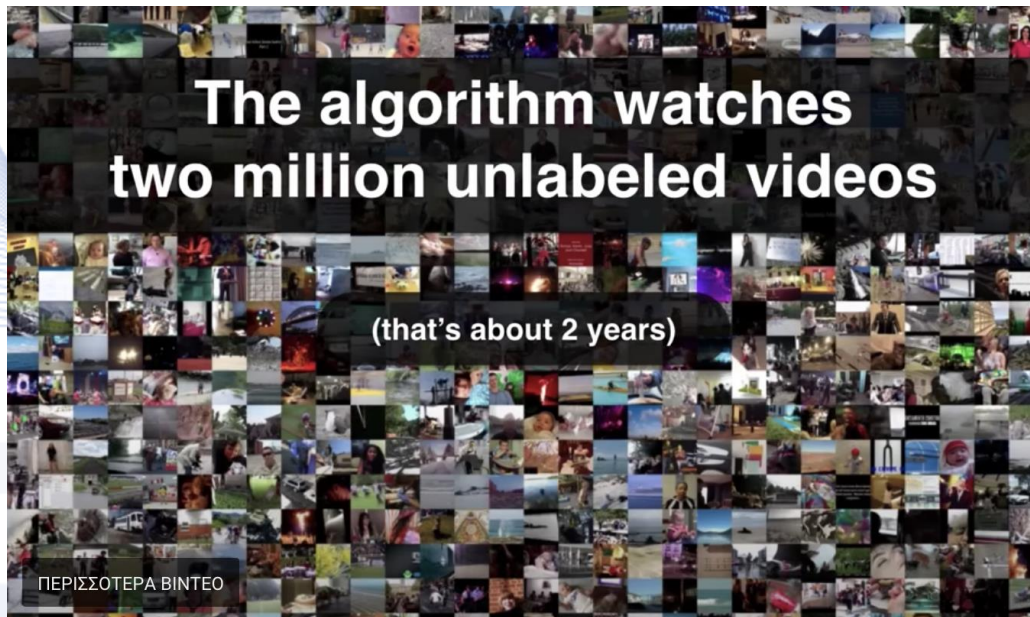


# GAN for Object Detection



# GAN for Video Generation

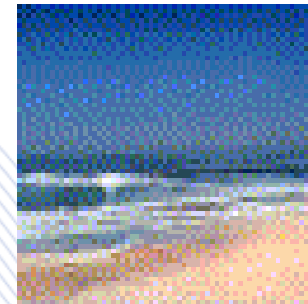
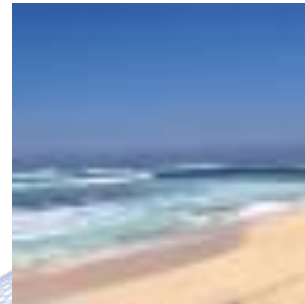
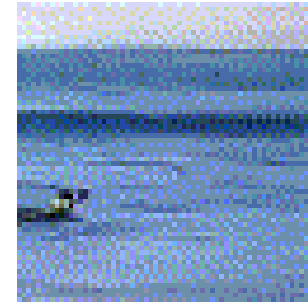
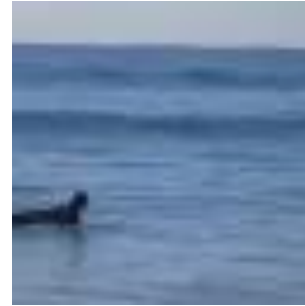
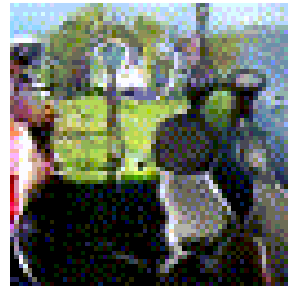
- **Conditional Video Generation.** GAN produces static image animations by prediction training [VON2016].



input is 100 dimensional (Gaussian noise). There are two independent streams: a moving foreground pathway of fractionally-strided spatio-temporal convolutions, and a static background pathway of fractionally-strided spatial convolutions, both of which up-sample. These two pathways are combined to create the generated video using a mask from the motion pathway. Below each volume is its size and the number of channels in parenthesis.

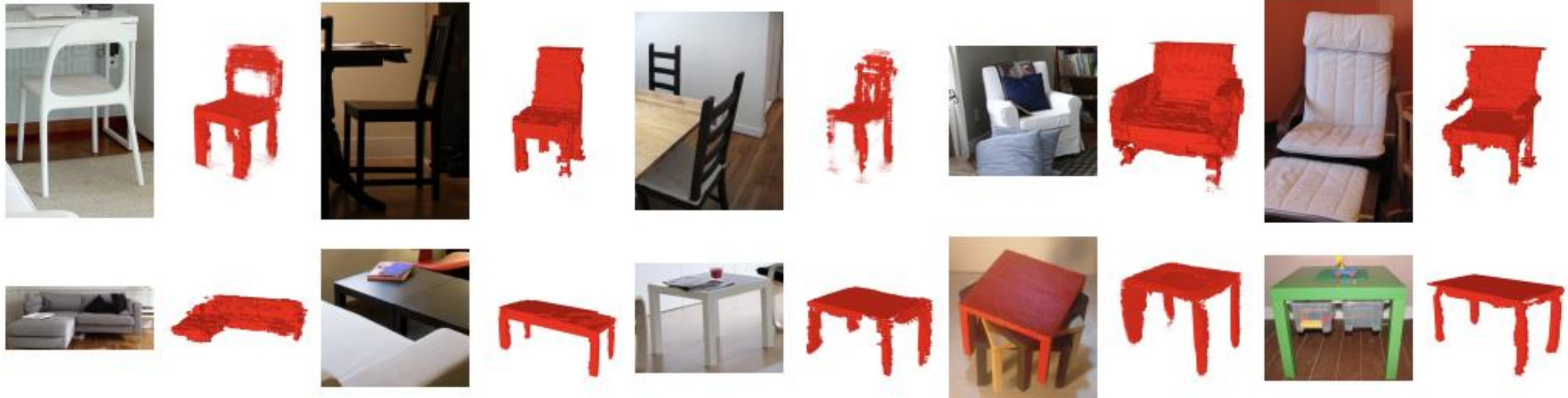


# GAN for Video Generation



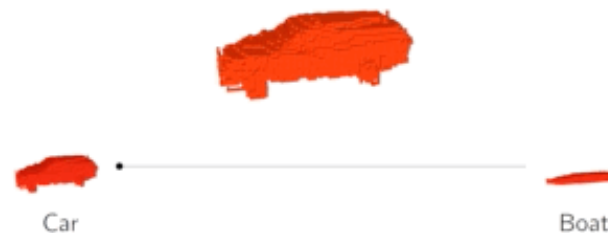


# GAN for 3D Object Creation



Source: [Wu and Zhang et al. 2017]

## Interpolation in Latent Space



# GAN for Music Generation

- During training, the Discriminator learns to differentiate between synthesized and real melodies.
- This architecture is an alternative to RNNs/LSTMs that are typically used in similar problems.



(a) MidiNet model 1



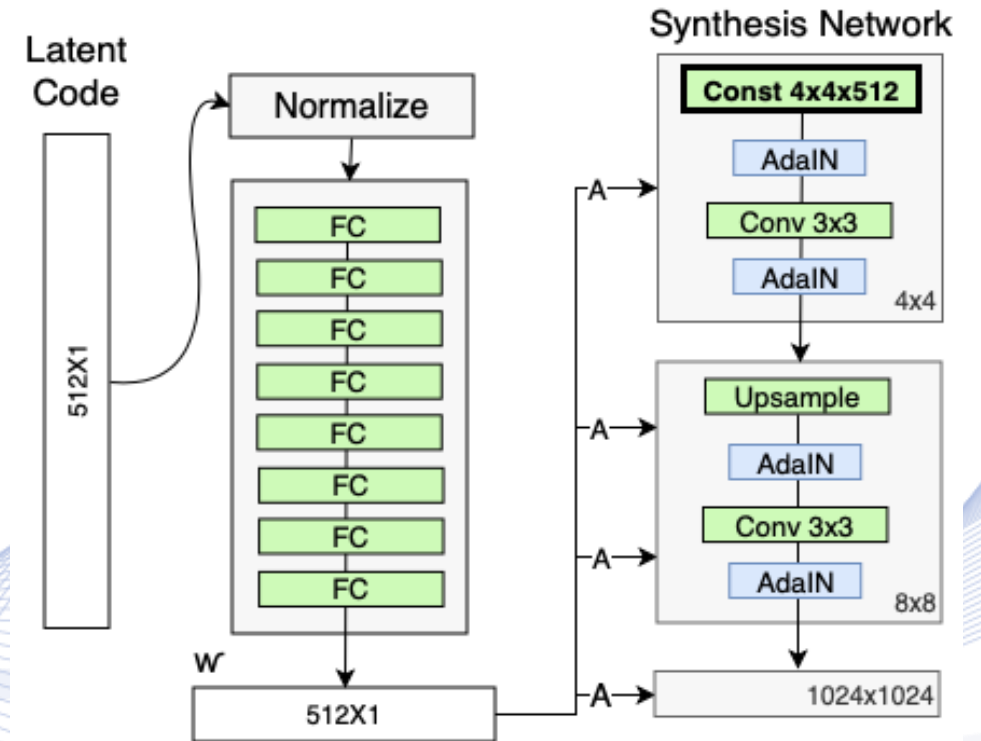
(b) MidiNet model 2



(c) MidiNet model 3

# GAN for Image Synthesis

- **Style-based GAN (StyleGAN)** produces good results in data-driven unconditional generative image modelling [KAR2019a][KAR2019b].
- In GAN, the **feature entanglement problem** is present:
  - small changes to the input latent vector makes the output image/face look drastically different.
- StyleGAN attempts to solve this problem, using a NN that **maps an input vector to a second, intermediate latent vector** to be used by GAN.





# Q & A

**Thank you very much for your attention!**

**Contact: Prof. I. Pitas**  
**[pitas@csd.auth.gr](mailto:pitas@csd.auth.gr)**