Stabilizing GANs Training via Evolutionary Strategy

Presented by

Chaoyue Wang

School of Computer Science

THE UNIVERSITY OF SYDNEY

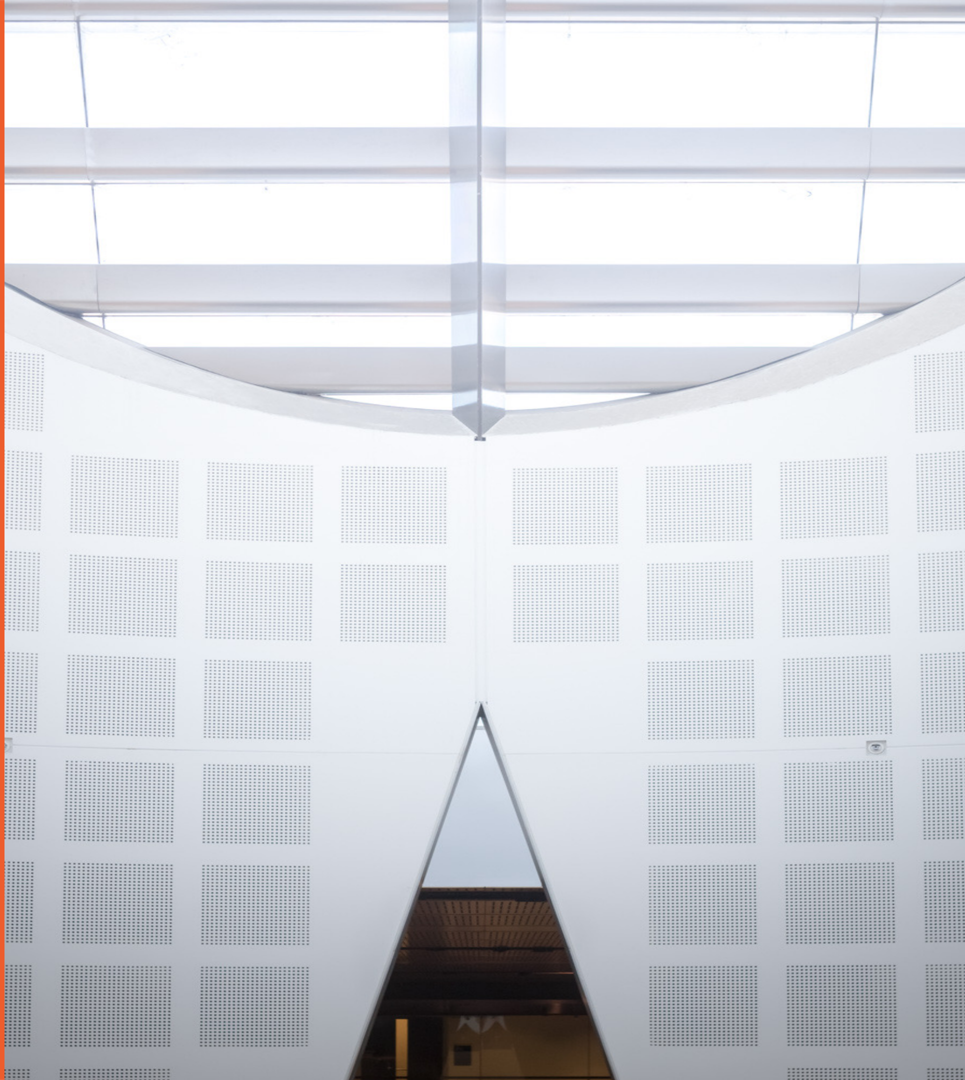
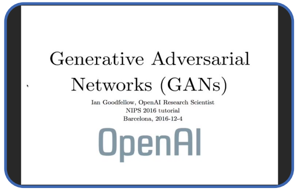
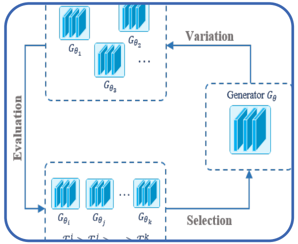


Table of Contents



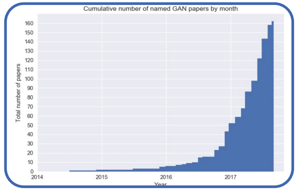
Introduction to GANs

- Generative Modeling
- Generative Adversarial Networks (GANs)
- GAN Losses



Evolutionary Generative adversarial Networks (E-GAN)

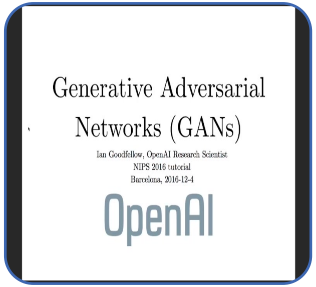
- Two-player Games & Natural Evolution
- Evolutionary Training Framework
- Evaluation Metrics
- E-GAN Experiments & Discussions



GAN related applications & Perceptual adversarial learning

- GANs and related visual applications
- Pix2pix, CycleGAN & Perceptual loss
- Perceptual adversarial networks for image-to-image transformation

Table of Contents



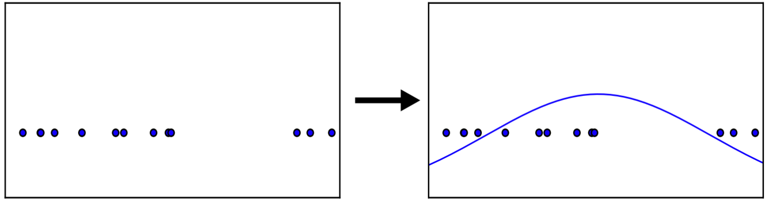
Introduction to GANs

- Generative Modeling
- Generative Adversarial Networks (GANs)
- GAN Losses

Generative Modeling

Density Estimation:

- An estimate, based on observed data, of an unobservable underlying **probability density function**.



Training Data Density Function

Sample Generation:

- Through learning the generative model, the algorithm is capable of producing **reasonable samples** following the target distribution.

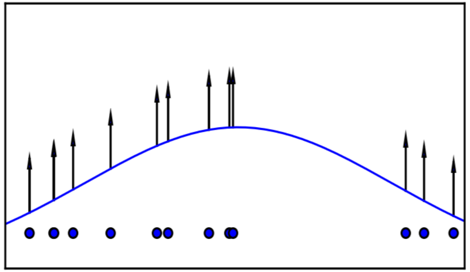
Training Data (CelebA)

Generated samples (Karras et al, 2017)

Maximum Likelihood Estimation:

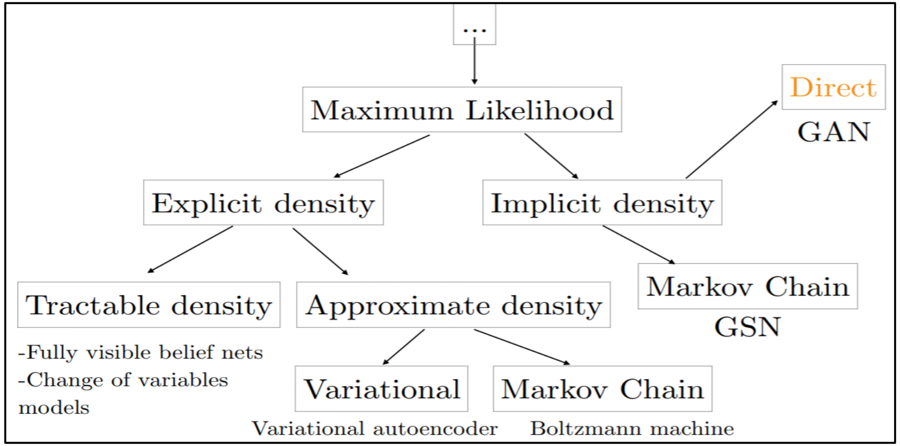
(Goodfellow, 2016)

- Likelihood is the probability that the model assigns to samples. Maximum likelihood estimation aims to choose the parameters for the model that **maximize the likelihood of the training data**.



$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

- Deep generative models that can learn via the principle of maximum likelihood differ with respect to **how they represent or approximate the likelihood.**



(Goodfellow, 2016)

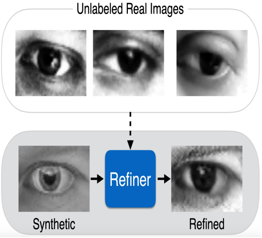
Generative Modeling

What can your do with generative models?

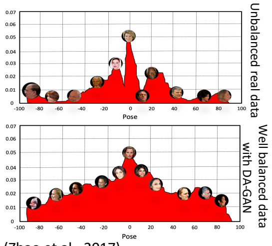
- Simulated training data

  Left: improve the realism of synthetic images from a simulator using unlabeled real data.

  Right: synthesize realistic profile faces for more efficiently training deep pose-invariant models.



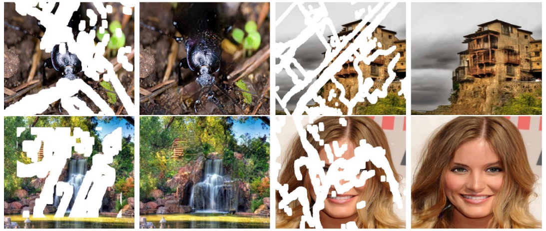
(Shrivastava et al., 2016)



(Zhao et al., 2017)

- Recovery missing or corrupted data

  Image inpainting for irregular holes. The learned generative model learns semantic priors and aim to fill those irregular holes with reasonable and realistic content.



(Liu et al., 2018)

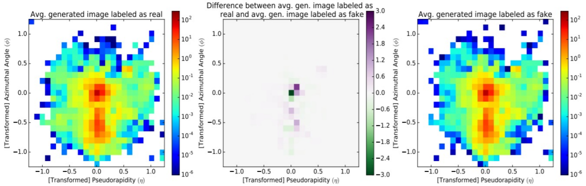
- Realistic generation tasks

  High fidelity natural image synthesis with only a single generative model. Conditioned on object labels, diverse samples from complex datasets, such as ImageNet, are generated.

(Brock et al., 2019)

- Simulation by prediction

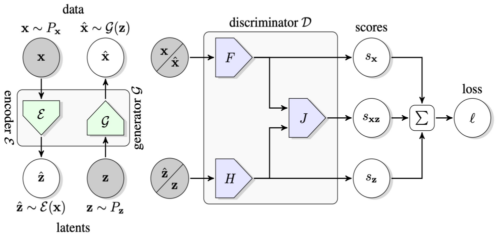
  Produce realistic radiation patterns from simulated high energy particle collisions. Save millions of dollars of CPU time by predicting outcomes of explicit simulations.



(de Oliveira et al., 2017)

- Solve inference problems

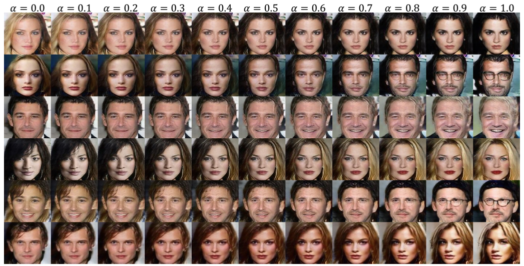
  Progress in image generation quality translates to substantially improved representation learning performance.



(Donahue et al., 2019)

- Learn useful embedding

  Interpolating in latent space. The generative model learns a meaningful projection from latent noisy space to face images.

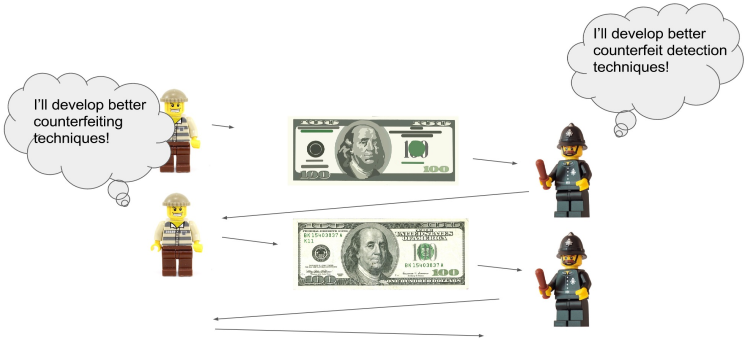


(Wang et al., 2019)

Generative Adversarial Networks (GANs)

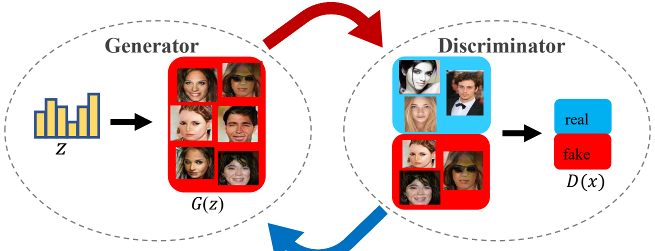
Adversarial Games:

- For example, a game between police and counterfeiters.

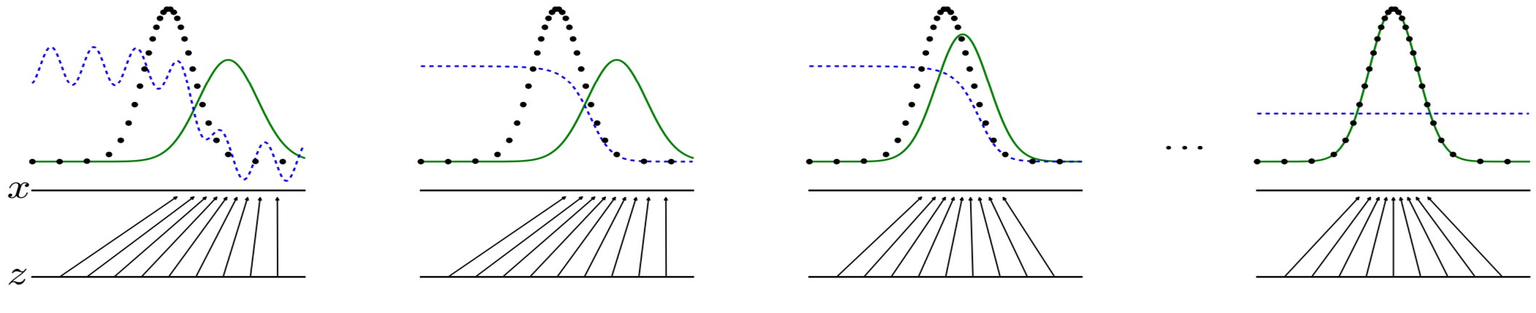


Generative Adversarial Networks (GANs):

- A Generator and a Discriminator are updated alternately to perform the adversarial game.



$$\min_G \max_D \ \mathbb{E}_{x\sim p_{\text{data}}}[\log(D(x))] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$$



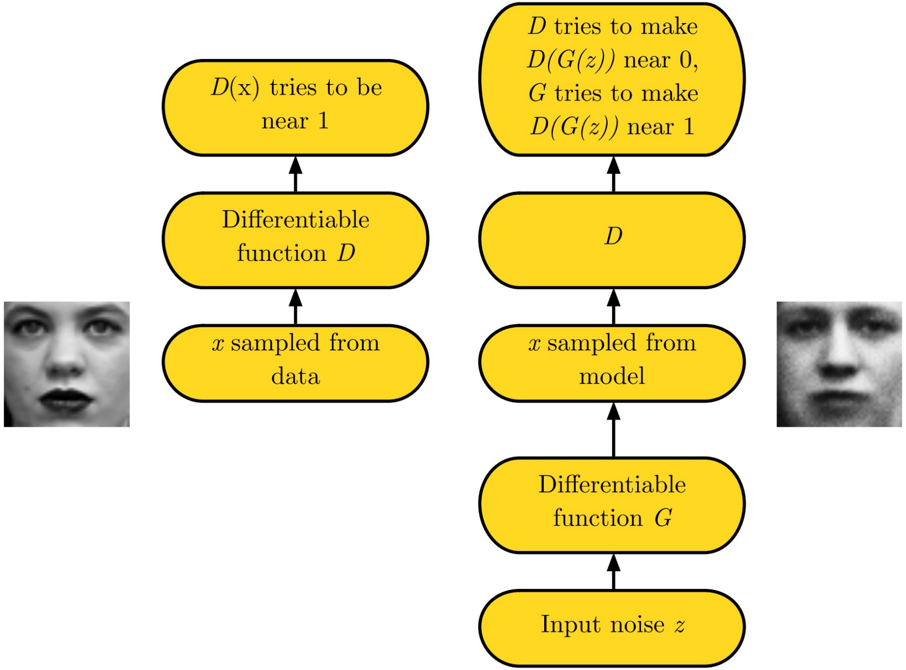
(a) **(b)** **(c)** **(d)**

Figure: Data distribution p_{data} (**black, dotted line**), generative distribution p_g (**green, solid line**), discriminative output $D(x)$(**blue, dashed line**).

(a) Initialize; (b) Updated *D to optimal*; (c) Updated *G*; (d) Reach the convergence point: $p_g = p_{data}$; $D(x) = 0.5$. (Goodfellow, 2014)

Generative Adversarial Networks (GANs)

Original GANs:



$$\min_G \max_D \; \mathbb{E}_{x\sim p_{\text{data}}}[\log(D(x))] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$$

(Goodfellow, 2014)

Minimax Game and Nash Equilibrium:

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

- **Proposition 1.** *For G fixed, the optimal discriminator D is*

$$D_G^* = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

- **Theorem 1.** *Giving the optimal D_G^*, the game can be reformulated as $C(G) = \max_D V(D,G)$. The global minimum of the virtual training criterion C(G) is achieved if and only if $p_g = p_{data}$. At that point, C(G) achieves the value* −log4.

- **Proposition 2.** *If G and D have enough capacity, and at each step of adversarial, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion,* $\mathbb{E}_{x\sim p_{\text{data}}}[\log(D_G^*(x))] + \mathbb{E}_{x\sim p_g}[\log(1 - D_G^*(G(z)))]$, then p_g converges to p_{data}.

GANs: Losses

Minimax Game and Jensen-Shannon Divergence:

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

- **Theorem 1.** *Giving the optimal* $D_G^* = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$, *the game can be reformulated as* $C(G) = \max_D V(D,G)$. *The global minimum of the virtual training criterion C(G) is achieved if and only if* $p_g = p_{data}$. *At that point, C(G) achieves the value* −log4.
- ***Proof***:

$$C(G) = \mathbb{E}_{x\sim p_{\text{data}}}\left[\log\left(D_G^*(x)\right)\right] + +\mathbb{E}_{x\sim p_g}\left[\log\left(1 - D_G^*(G(x))\right)\right]$$

$$= \mathbb{E}_{x\sim p_{\text{data}}}\left[\log\left(\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right)\right] + \mathbb{E}_{x\sim p_g}\left[\log\left(\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right)\right]$$

$$= \mathbb{E}_{x\sim p_{\text{data}}}\left[\log\left(\frac{2\cdot p_{data}(x)}{p_{data}(x)+p_g(x)}\right)\right] + \mathbb{E}_{x\sim p_g}\left[\log\left(\frac{2\cdot p_g(x)}{p_{data}(x)+p_g(x)}\right)\right] - \log(4)$$

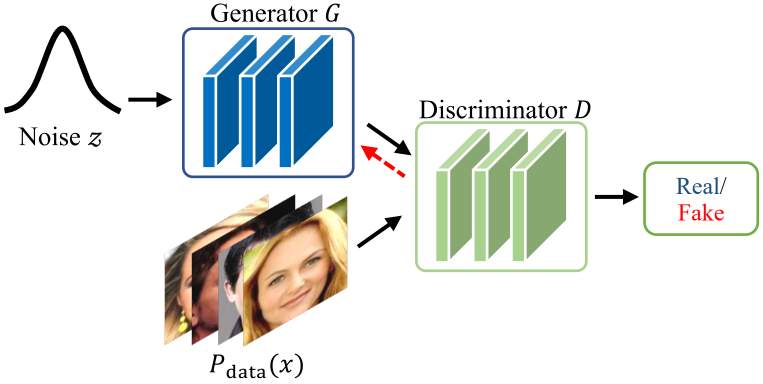
$$= KL(p_{data}|| \frac{p_{data}+p_g}{2}) + KL(p_g|| \frac{p_{data}+p_g}{2}) - \log(4)$$

$$= 2\cdot JSD(p_{data}||p_g) - \log(4)$$

Since the **Jensen–Shannon divergence** between two distributions is always non-negative, and zero *iff* they are equal, we have shown that $C^* = -\log(4)$ is the global minimum and that the only solution is $p_g = p_{data}$.

Generalized Adversarial Games:

- Actually, the adversarial progress can be formulated as **not only** as the original **minimax game**. The adversarial game only asks the generator and the discriminator against each other.



- Therefore, the **generalized adversarial game** is:

$$\max_D \mathbb{E}_{x_r\sim p_{\text{data}}}[f_1(D(x_r))] + \mathbb{E}_{z\sim p_z}[f_2(D(G(z)))]$$

$$\min_G \mathbb{E}_{x_r\sim p_{\text{data}}}[g_1(D(x_r))] + \mathbb{E}_{z\sim p_z}[g_2(D(G(z)))]$$

where f_1, f_2, g_1, g_2 can be defined as different functions.

- Different f_1, f_2, g_1, g_2 represent **different GAN losses**, which theoretically minimize **different distances** between the generative distribution and real data distribution.

GANs: Losses

Different losses and their corresponding distances:

- **Minimax (regular) GAN:**

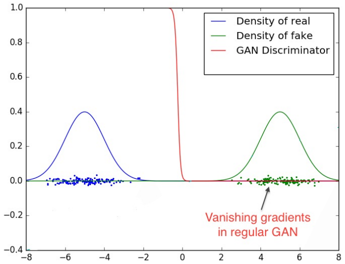
$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$$

$$\Downarrow$$

$JSD(p_G||p_{\text{data}})$ (Jensen-Shannon divergence)

Pros: *1) easy to explain and theoretically analyze*
2) partly avoid mode collapse
3) relative satiability at convergence

Cons:
1) Gradients vanishing



- **Heuristic GAN:**

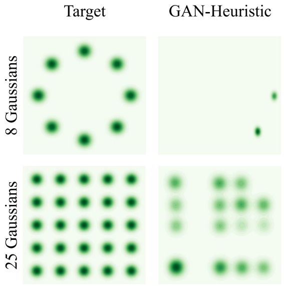
$$\min_G -\mathbb{E}_{z\sim p_z}[\log(D(G(z)))]$$
$$\max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{x\sim p_z}[\log(1 - D(G(z)))]$$

$$\Downarrow$$

$$KL(p_G||p_{\text{data}}) - 2JSD(p_G||p_{\text{data}})$$

Pros: *1) KL divergence largely avoids vanishing gradient.*
2) DCGAN, reasonable performance

Cons:
1) Mode collapse caused by KL divergence
2) Instability caused by -JSD



- **Least-squares GAN:**

$$\min_G \frac{1}{2}\mathbb{E}_{z\sim p_z}[(D(G(z)) - 1)^2]$$
$$\max_D \frac{1}{2}\mathbb{E}_{x\sim p_{\text{data}}}[(D(x) - 1)^2] + \mathbb{E}_{x\sim p_z}[(D(G(z)))^2]$$

$$\Downarrow$$

$$\mathcal{X}^2_{\text{Pearson}}(p_{\text{data}} + p_G||2p_G)$$

Pros: *1) Partly avoid mode collapse*
2) Non-saturating when $D(G(z))\to 0$

Cons:
1) Cannot assign an high cost to generate fake samples.

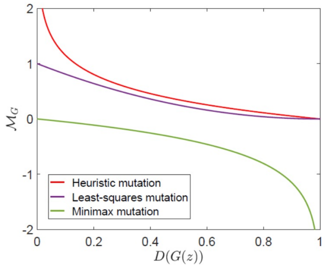
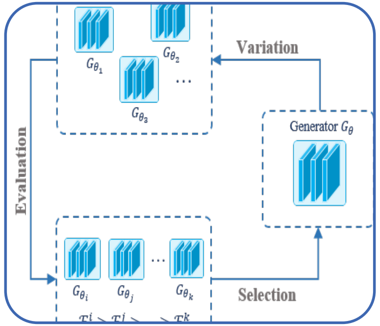


Table of Contents



Evolutionary Generative adversarial Networks (E-GAN)

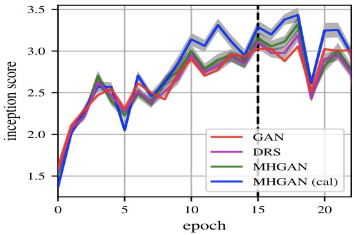
- Two-player Games & Natural Evolution
- Evolutionary Training Framework
- Evaluation Metrics
- E-GAN Experiments & Discussions

Two-player v.s. Natural evolution

Two-player Games (Most current GANs):

- Pre-defined movements and 'fighting' strategy.

- **Consequence:** Carefully balance the capabilities of both *generator* and *discriminator*, otherwise, training problems (K.O.!)

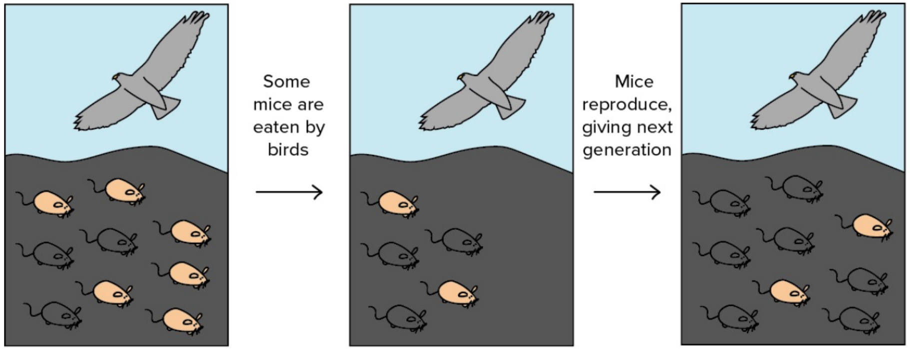


Training instability

Mode collapse

Theory of Evolution & Natural Selection

- In nature, species get evolved by fighting against their natural enemy. Survival of the fittest.

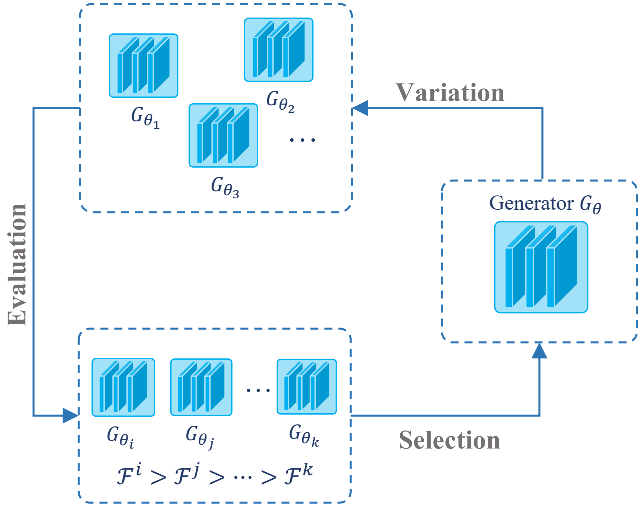


- The adversarial progress may also be regarded as the adversarial game between **the population of generators** (*i.e. Mouse*) and the **natural enemy discriminator** (*i.e. Birds*).

- Three most important components in an evolution process:
  1) *Genetic variation*
  2) *Individual evaluation*
  3) *Natural selection*

Evolutionary training framework

Framework: *Evolutionary Generative Adversarial Networks (E-GAN)*



Variation: Given an individual G_θ in the population, several copies of each individual, or parent, are created, each of which are modified by different mutations. Then, each modified copy in $\{G_{\theta_1}, G_{\theta_2}, \dots\}$ is regarded as one child

- ***Gene*** is the *parameters* of the network.
- **Mutations** correspond to *different generator losses* (*i.e.* updating strategies)

Evaluation: For each child, its performance, or individual's quality, is evaluated by a fitness function $\mathcal{F}(\cdot)$ that depends on the current enemy (i.e., discriminator D).

- ***Fitness function***: select the suitable/desired individuals for the following adversarial game.
- Or, we hope to evaluate the performance of current generator individual. Two properties are focused, 1) the quality and 2) the diversity of generated samples. Thus,

$$\mathcal{F} = \mathcal{F}_q + \lambda \mathcal{F}_d$$

- ***Quality fitness score***:

$$\mathcal{F}_q = \mathbb{E}_z[D\left(G_{\theta_n}(z)\right)]$$

- ***Diversity fitness score***:

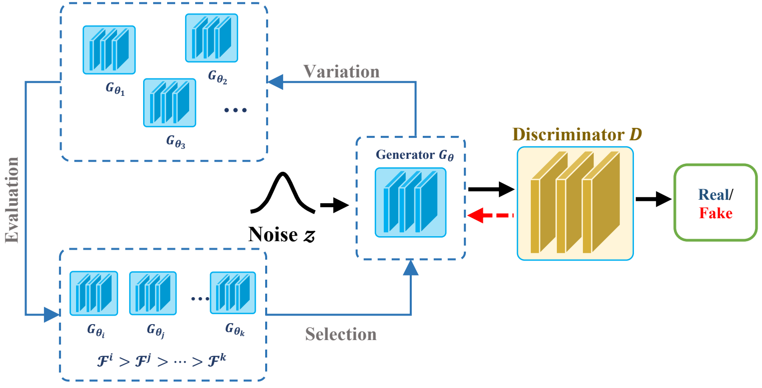
$$\mathcal{F}_d = -\log\left\|\nabla_D - \mathbb{E}_x[\log D(x)] - \mathbb{E}_z[\log(1 - D(G_{\theta_n}(z)))]\right\|$$

Selection: We employ a simple yet useful *survivor selection strategy* to determine the next generation based on the $\mathcal{F}$ of existing individuals.

- Fitness scores evaluated in different generations cannot compare with each other.
- Select desired offspring is equivalent to selecting the effective adversarial strategies.
- (μ, λ)-selection: sorting $\{\mathbf{x}_i\}_{i=1}^{\lambda}$, the μ-best individuals are selected.

Evolutionary training framework

How about the discriminator *D* ?



- Updating strategy (vanilla training loss)

$$\mathcal{L}_D = -\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] - \mathbb{E}_{y\sim p_g}[\log(1 - D(y))]$$

- Proof of the Optimal Discriminator

Based on our proving, though different two-player GANs have different training losses for both *G* and *D*, discriminator's losses considered in the proposed E-GAN lead to the same optimal discriminator:

$$D^*_{E-GAN} = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

Complete E-GAN Algorithm:

Algorithm 1 E-GANs. Default Values $\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.99$, $n_D = 3$, $n_m = 3$, and $m = 32$

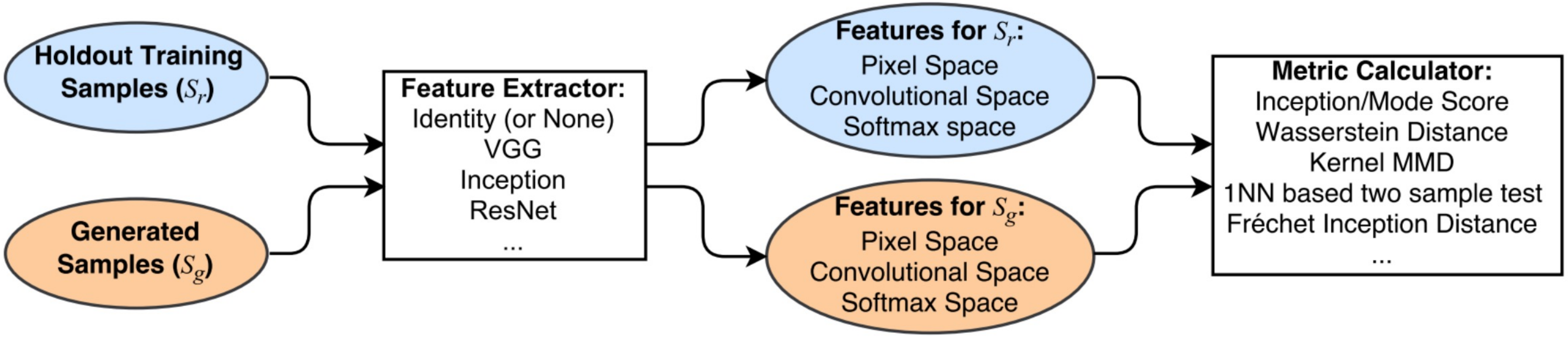
Require: the batch size m. the discriminator's updating steps per iteration n_D. the number of parents μ. the number of mutations n_m. Adam hyper-parameters α, β_1, β_2, the hyper-parameter γ of evaluation function.

Require: initial discriminator's parameters w_0. initial generators' parameters $\{\theta_0^1, \theta_0^2, \ldots, \theta_0^\mu\}$.

```
for number of training iterations do
    for k = 0, ..., n_D do
        Sample a batch of {x^(i)}_{i=1}^m ~ p_data (training data), and a batch of {z^(i)}_{i=1}^m ~ p_z (noise samples).
        g_w <- ∇_w[ (1/m) Σ_{i=1}^m log D_w(x^(i))
            + (1/m) Σ_{j=1}^μ Σ_{i=1}^{m/μ} log(1 - D_w(G_{θ^j}(z^(i))))]
        w <- Adam(g_w, w, α, β1, β2)
    end for
    for j = 0, ..., μ do
        for h = 0, ..., n_m do
            Sample a batch of {z^(i)}_{i=1}^m ~ p_z
            g_{θ^j,h} <- ∇_{θ^j} M_G^h({z^(i)}_{i=1}^m, θ^j)
            θ_child^{j,h} <- Adam(g_{θ^j,h}, θ^j, α, β1, β2)
            F^{j,h} <- F_q^{j,h} + γ F_d^{j,h}
        end for
    end for
    {F^{j1,h1}, F^{j2,h2}, ...} <- sort({F^{j,h}})
    θ^1, θ^2, ..., θ^μ <- θ_child^{j1,h1}, θ_child^{j2,h2}, ..., θ_child^{jμ,hμ}
end for
```

E-GANs: Experiments

Evaluation Metrics:



- **The Inception Score (IS):**

$$\mathrm{IS}(\mathbb{P}_g) = e^{\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_g}[KL(p_{\mathcal{M}}(y|\mathbf{x})||p_{\mathcal{M}}(y))]}$$

where $p_{\mathcal{M}}(y|\mathbf{x})$ denotes the label distribution of $\mathbf{x}$ as predicted by $\mathcal{M}$, *and* $p_{\mathcal{M}}(y) = \int_{\mathbf{x}} p_{\mathcal{M}}(y|\mathbf{x}) d\mathbb{P}_g$, *i.e.*, the marginal of $p_{\mathcal{M}}(y|\mathbf{x})$ over the probability measure $\mathbb{P}_g$.

A higher IS happens when the Inception network is very **confident that the image belongs to a particular ImageNet category**, and has $p_{\mathcal{M}}(y)$ close to uniform, *i.e.*, **all categories are equally represented**.

- **The Fréchet Inception Distance (FID):**

$$\mathrm{FID}(\mathbb{P}_r, \mathbb{P}_g) = \|\mu_r - \mu_g\| + \mathrm{Tr}(\boldsymbol{C}_r + \boldsymbol{C}_g - 2(\boldsymbol{C}_r\boldsymbol{C}_g)^{1/2})$$

where μ_r (μ_g) and $\boldsymbol{C}_r$ ($\boldsymbol{C}_g$) are the **mean** and **covariance** of the real (generated) distribution, respectively. Note the variables (or distributions $\mathbb{P}_r, \mathbb{P}_g$) correspond to **features** extracted from real (generated) samples.

Under the **Gaussian assumption** on both $\mathbb{P}_r$ and $\mathbb{P}_g$, *i.e.*, extracted features, the Fréchet distance is equivalent to the **Wasserstein-2 distance**.

E-GANs: Experiments

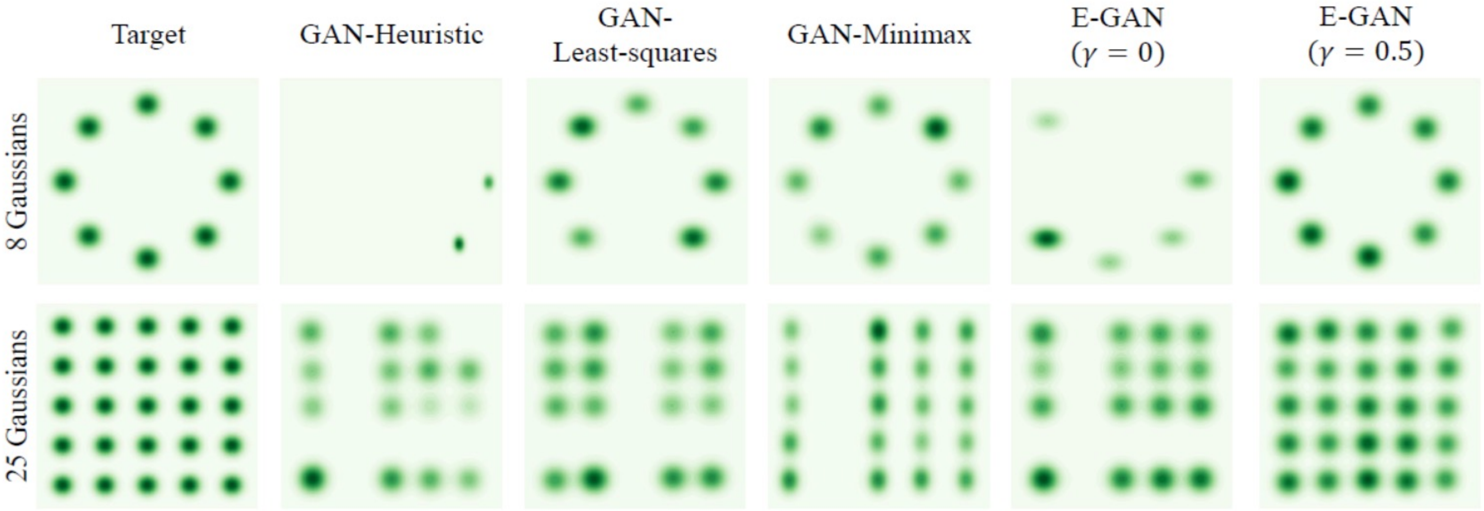
Synthetic Datasets and Mode Collapse:

- Target/real distribution: *2D Gaussian mixture distributions* (8 Gaussians and 25 Gaussians)
- Quantitative metric: *Maximum Mean Discrepancy* (MMD).
- *Two-player GANs* are more or less troubled by the mode collapse issue. Meanwhile, *E-GAN without diversity fitness function* (i.e. γ=0) also shows model collapse. Yet, the *basic E-GAN model*, which only keeps the best children each iteration, achieved much better performance.

TABLE II

MMD ($\times 10^{-2}$) WITH MIXED GAUSSIAN DISTRIBUTIONS ON OUR TOY DATASETS. WE RAN EACH METHOD FOR 10 TIMES, AND REPORT THEIR AVERAGE AND BEST RESULTS. THE METHOD WITH LOWER MMD VALUE IMPLIES THE GENERATED DISTRIBUTION IS CLOSER TO THE TARGET ONE.

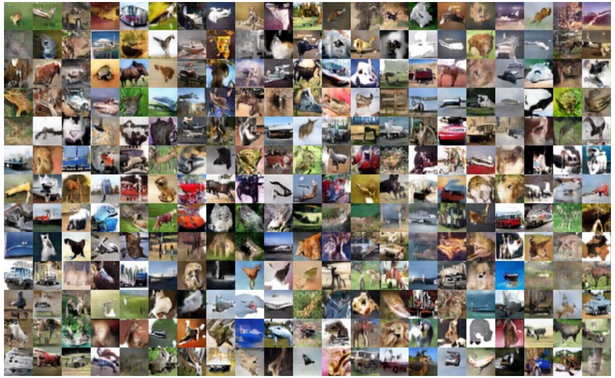
Methods	8 Gaussians		25 Gaussians	
	Average	Best	Average	Best
GAN-Heuristic	45.27	33.2	2.80	2.19
GAN-Least-squares	3.99	3.16	1.83	1.72
GAN-Minimax	2.94	1.89	1.65	1.55
E-GAN ($\lambda = 0$, without GP)	11.54	7.31	1.69	1.60
E-GAN ($\lambda = 0.5$, without GP)	**2.36**	**1.17**	**1.20**	**1.04**



E-GANs: Experiments

CIFAR-10 Dataset and Training Stability:

- *Convergence speed*:
  Much less training iterations, and reasonable training time
- *Training stability and inception score*:
  GAN-Heuristic: instability at convergence
  GAN-Minimax: invalid
  Fitness function: limited performance
- *Selected adversarial strategies*:
  More GAN-Heuristic in the beginning, yet Minimax became more and more
- *# of survived parents*:
  Long-term, bigger population brings better performance



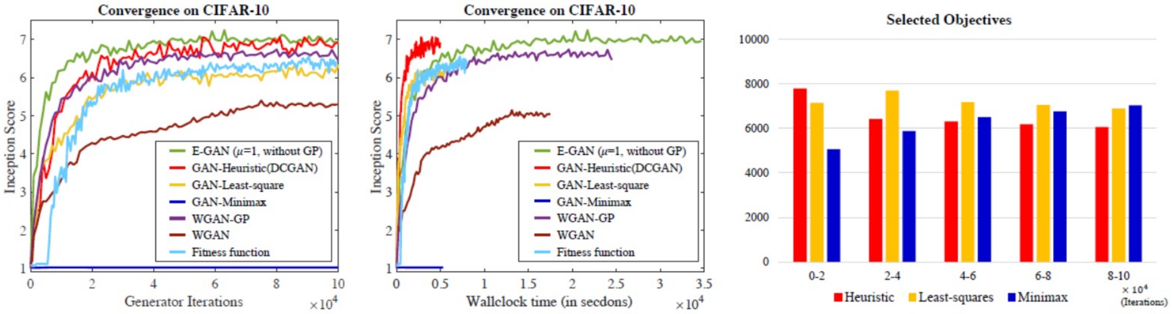


Fig. 4. Experiments on the CIFAR-10 dataset. CIFAR-10 inception score over generator iterations (left), over wall-clock time (middle), and the graph of selected mutations in the E-GAN training process (right).

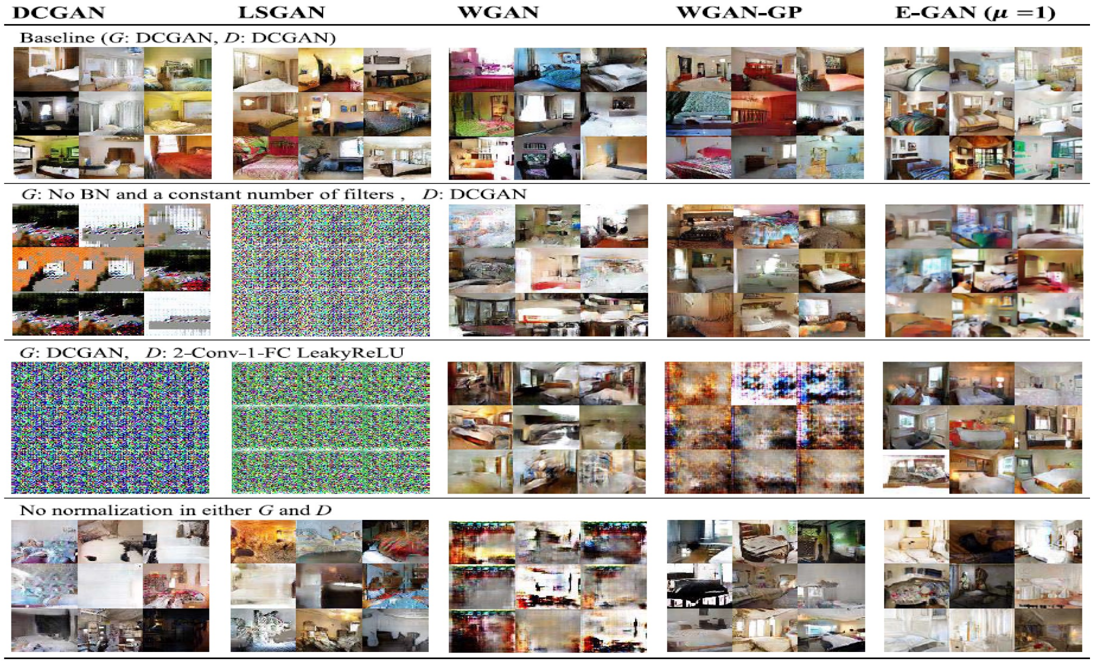
TABLE III
INCEPTION SCORES AND FIDs WITH UNSUPERVISED IMAGE GENERATION ON CIFAR-10. THE METHOD WITH HIGHER IS OR LOWER FID IMPLIES THE GENERATED DISTRIBUTION IS CLOSER TO THE TARGET ONE. † [22], ‡ [6]

Methods	Inception score	FID
Real data	$11.24 \pm .12$	7.8
-Standard CNN-		
(ours) E-GAN-GP ($\mu = 1$)	$7.13 \pm .07$	33.2
(ours) E-GAN-GP ($\mu = 2$)	$7.23 \pm .08$	31.6
(ours) E-GAN-GP ($\mu = 4$)	$7.32 \pm .09$	29.8
(ours) E-GAN-GP ($\mu = 8$)	$\mathbf{7.34 \pm .07}$	**27.3**
(ours) E-GAN ($\mu = 1$, without GP)	$6.98 \pm .09$	36.2
DCGAN (without GP)†	$6.64 \pm .14$	-
GAN-GP‡	$6.93 \pm .08$	37.7
WGAN-GP‡	$6.68 \pm .06$	40.2

E-GANs: Experiments

LSUN and Architecture Robustness:

- Experiments of *architecture robustness*. Different GAN architectures, which correspond to different training challenges.
- According to both visual observation and FID score (FID is better than IS in this case), our E-GAN achieved better results on all of four different configurations.
- 128x128 Bedrooms are generated and reported below.



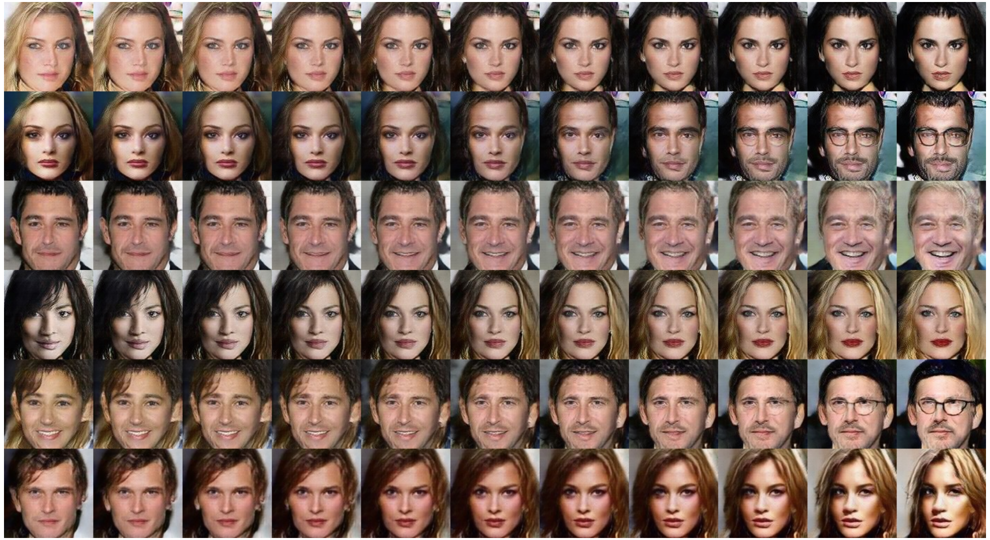
Methods	FID			
	-Baseline-	-Weak G-	-Weak D-	-Weak both-
DCGAN	43.7	187.3	410.6	82.7
LSGAN	46.3	452.9	423.1	126.2
WGAN	51.1	113.6	129.2	115.7
WGAN-GP	38.5	66.7	385.8	73.2
(ours) E-GAN ($\mu = 1$, without GP)	34.2	63.3	64.8	71.9
(ours) E-GAN ($\mu = 4$, without GP)	**29.7**	**59.1**	**55.2**	**60.9**

E-GANs: Experiments

CelebA and Space Continuity:

- We first select pairs of generated faces and record their corresponding latent vectors z_1 and z_2. The two images in one pair have different attributes, such as gender, expression, hairstyle, and age. Then, linear interpolating between them, i.e. $(1-\alpha)z_1 + \alpha z_2$.
- Generated samples can seamlessly change between these semantically meaningful face attributes.

$\alpha = 0.0$ $\alpha = 0.1$ $\alpha = 0.2$ $\alpha = 0.3$ $\alpha = 0.4$ $\alpha = 0.5$ $\alpha = 0.6$ $\alpha = 0.7$ $\alpha = 0.8$ $\alpha = 0.9$ $\alpha = 1.0$

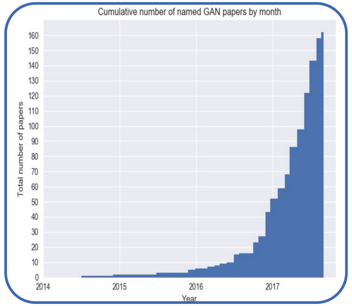


$G((1-\alpha)z_1 + \alpha z_2)$

Conclusion:

To reduce training difficulties and improve generative performance, we devised an evolutionary algorithm to evolve a population of generators to adapt to the natural enemy (*i.e.* the discriminator *D*). In contrast to two-player GANs, the evolutionary paradigm allows the proposed E-GAN to overcome the limitations of individual adversarial objectives and preserve the well-performing offspring after each iteration. Experiments showed that E-GAN improves the training stability of GAN models and achieves convincing generative performance.

Table of Contents

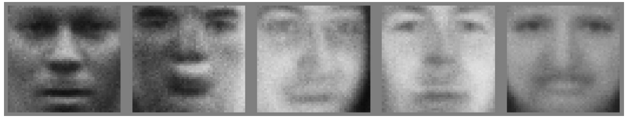


GAN Related Applications & Perceptual Adversarial Nets

- GANs and related visual applications
- Pix2pix, CycleGAN & Perceptual loss
- Perceptual adversarial networks for image-to-image transformation

GANs: Visual Applications

The first GAN *in 2014:*



(Goodfellow et al., 2014)

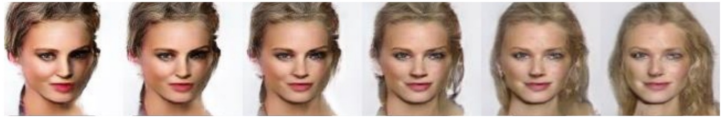
- The generator and discriminator employ basic **fully-connected nets.**

DCGAN *in 2015:*

(Radford et al., 2015)

- The generator and discriminator employ **deep convolutional nets.**

CoGAN *in 2016:*



(Liu et al., 2015)

- **Multiple** generator and discriminator with **weight sharing** layers.

StyleGAN *in 2018:*

(Karras et al., 2015)

- Inject the input into the each convolutional layer of the generator through **adaptive instance normalization (AdaIN).**

SNGAN & projection discriminator *in 2018:*

(Miyato et al., 2014)

- **Spectral normalization** is devised to control the **Lipschitz constant** of the discriminator function, which can stabilize training process.
- **Projection *D*** structure is devised for better conditional generation.

SAGAN *in later 2018:*

(Zhang et al., 2018)

- **Self-attention module** is introduced to improve generation capability and performance.

GauGAN *in 2019:*

(Park et al., 2019)

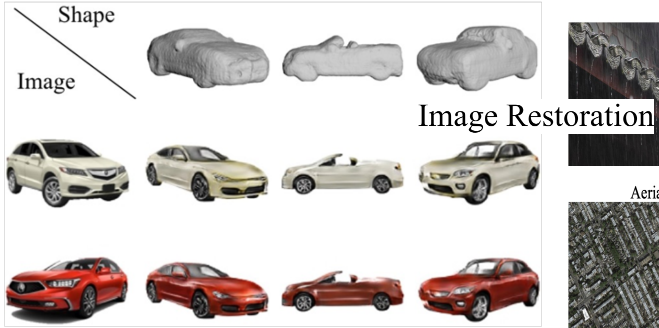
- **Spatially-Adaptive Normalization** is proposed to involve semantic information during the generation.

GANs: Visual Applications

Image Generation

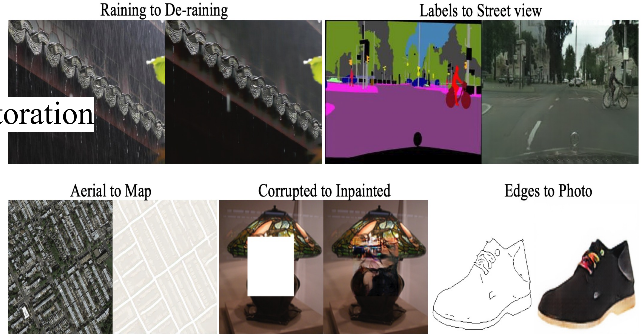
Brock. *et al.* “Large scale GAN training for high fidelity natural image synthesis.” In *ICLR*, 2019.

3D Object Generation



Zhu. *et al.* "Visual object networks: image generation with disentangled 3D representations." In *NeurIPS*, 2018.

Image Restoration

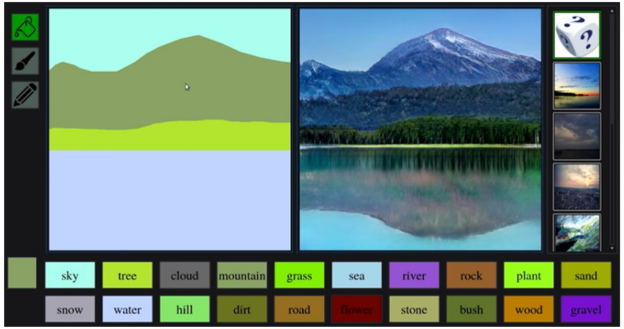


Wang, *et al.* “Perceptual adversarial networks for image-to-image transformation”, In *IEEE T-IP*, 2018.

Video Generation

Zakharov. *et al.* "Few-Shot adversarial learning of realistic neural talking head models." In *arXiv*, 2019.

Image-to-Image translation



Park. *et al.* "Semantic image synthesis with spatially-adaptive normalization." In *CVPR*, 2019.

Style Transfer

Chen. *et al.* "Gated-gan: Adversarial gated networks for multi-collection style transfer." IEEE T-IP, 2018.

Pix2pix, CycleGAN & Perceptual loss

Input | Pixel loss | Pixel + GAN loss | Pixel + GAN + Perceptual loss

Zhang, He, Vishwanath Sindagi, and Vishal M. Patel. "Image De-raining Using a Conditional Generative Adversarial Network." arXiv. 2017.

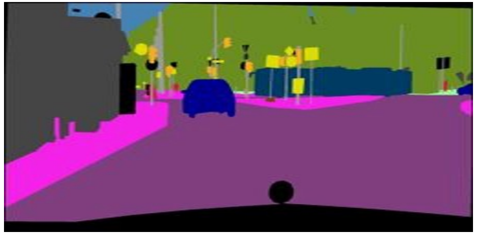
Pix2pix, CycleGAN & Perceptual loss


Pixel loss


Pixel + GAN loss


Pixel + Perceptual loss


Pixel + GAN
+ Perceptual loss


Input

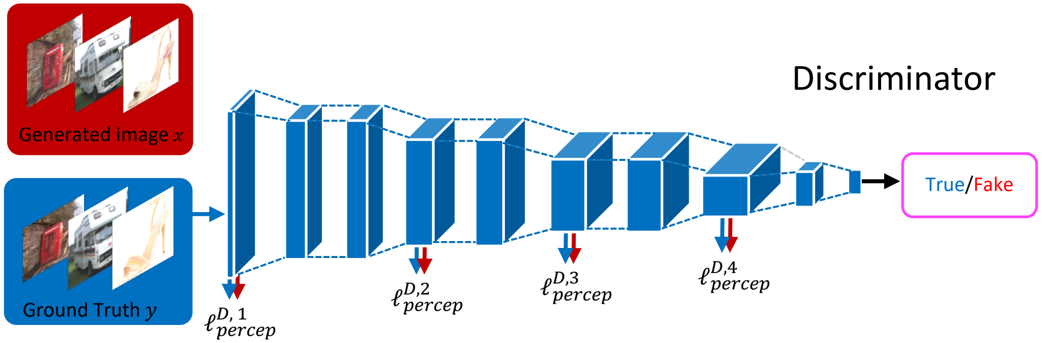

Output


Ground truth

Pixel loss
Perceptual loss
⋮

Perceptual adversarial networks (PAN)

Perceptual adversarial loss (feature matching loss)

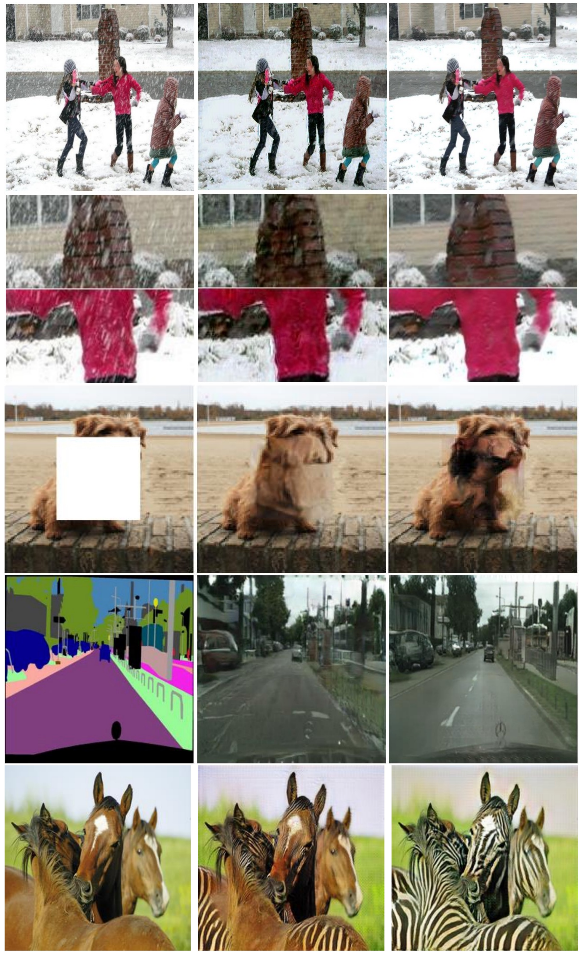


$$\ell^{D,j}_{percep}(x,y) = ||\phi_i(T(x)) - \phi_i(y)||_{\ell_p}$$

$$\min_G -\mathbb{E}_{x\in\mathcal{X}}\left[\log D\big(G(x)\big)\right] + \frac{1}{N}\sum_j \sum_{(x,y)} \lambda_j\, \ell^{D,j}_{percep}(x,y)$$

$$\max_D \mathbb{E}_{y\in\mathcal{Y}}[\log D(y)] - \mathbb{E}_{x\in\mathcal{X}}\left[\log(1 - D(G(x)))\right] + \left[m - \frac{1}{N}\sum_j \sum_{(x,y)} \lambda_j\, \ell^{D,j}_{percep}(x,y)\right]^+$$

- The perceptual adversarial loss is devised to form dynamic measurements based on the hidden layers of the discriminator *D*.
- It minimizes differences between generated images and real images from dynamic semantic feature levels.
- The subsequently proposed **feature matching loss** (pix2pixHD) shares a similar idea with us and is widely used in existing algorithms.



(Wang et al., 2018)

PAN: Experiments

TABLE III
DE-RAINING

	PSNR(dB)	SSIM	UQI	VIF
L2	22.77	0.7959	0.6261	0.3570
cGAN	21.87	0.7306	0.5810	0.3173
L2+cGAN	22.19	0.8083	0.6278	0.3640
ID-CGAN	22.91	0.8198	0.6473	0.3885
PAN	**23.35**	**0.8303**	**0.6644**	**0.4050**
PA Loss+cGAN	23.22	0.8078	0.6375	0.3904

TABLE IV
IN-PAINTING

	PSNR(dB)	SSIM	UQI	VIF
Context-Encoder	21.74	0.8242	0.7828	0.5818
PAN	**21.85**	**0.8307**	**0.7956**	**0.6104**

TABLE V
COMPARISON WIHT PIX2PIX-CGAN

Senmatic labels → Cityscapes imags

	PSNR(dB)	SSIM	UQI	VIF
pix2pix-cGAN	15.74	0.4275	0.07315	0.05208
PAN	**16.06**	**0.4820**	**0.1116**	**0.06581**

Edges → Shoes

	PSNR(dB)	SSIM	UQI	VIF
ID-cGAN	**20.07**	0.7504	0.2724	0.2268
PAN	19.51	**0.7816**	**0.3442**	**0.2393**

Edges → Handbags

	PSNR(dB)	SSIM	UQI	VIF
ID-cGAN	**16.50**	0.6307	0.3978	0.1723
PAN	15.90	**0.6570**	**0.4042**	**0.1841**

Cityscapes images → Semantic labels

	PSNR(dB)	SSIM	UQI	VIF
ID-cGAN	19.46	0.7270	0.1555	0.1180
PAN	**20.67**	**0.7725**	**0.1732**	**0.1638**

Aerial photos → Maps

	PSNR(dB)	SSIM	UQI	VIF
ID-cGAN	26.10	0.6465	0.09125	0.02913
PAN	**28.32**	**0.7520**	**0.3372**	**0.1617**

PAN: Experiments

Input $\ell_{percep}^{D,1}$ $\ell_{percep}^{D,2}$ $\ell_{percep}^{D,3}$ $\ell_{percep}^{D,4}$ Ground-truth

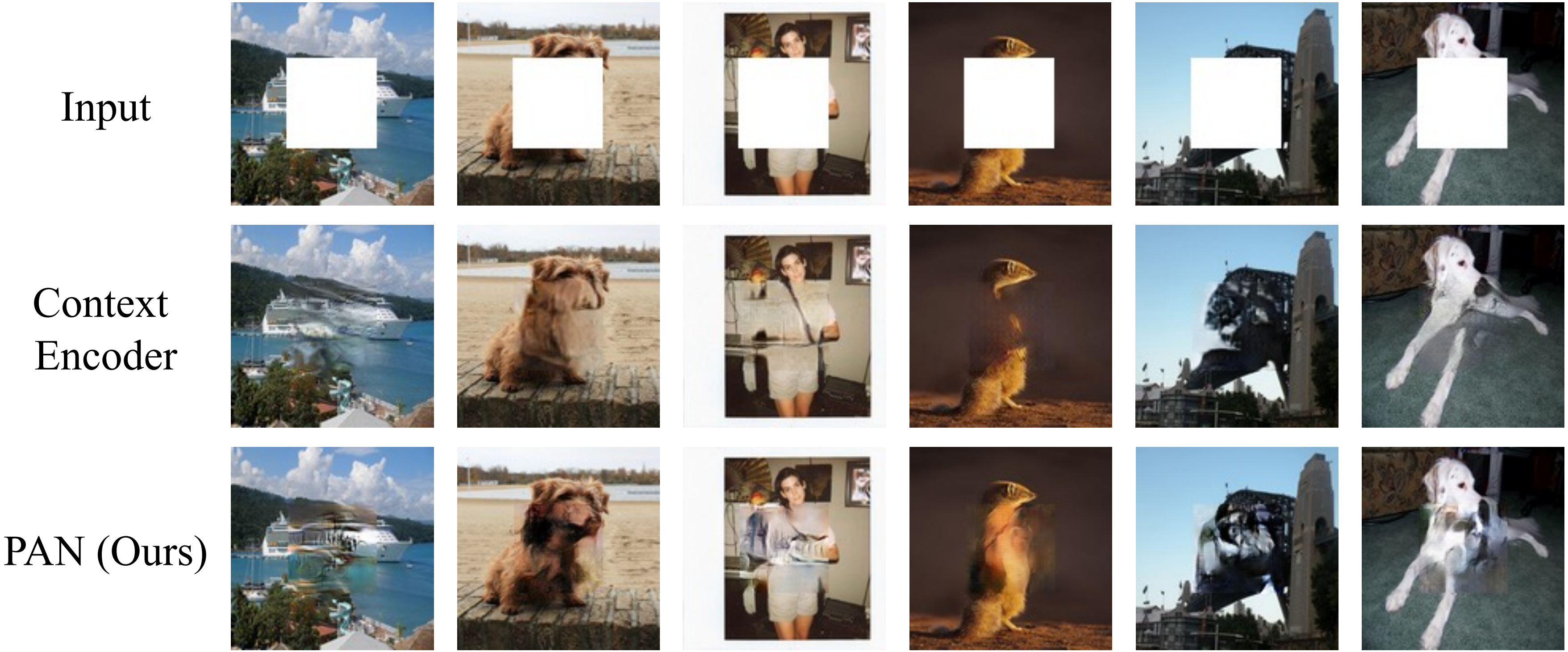
PAN: Experiments

Input | *L2* | *cGAN* | *L2+cGAN* | *ID-cGAN* | *PAN* | *PA Loss+cGAN*

PAN: Experiments

Input ID-CGAN PAN Input ID-CGAN PAN

PAN: Experiments



Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." *CVPR*. 2016.

PAN: Experiments

PAN: Experiments