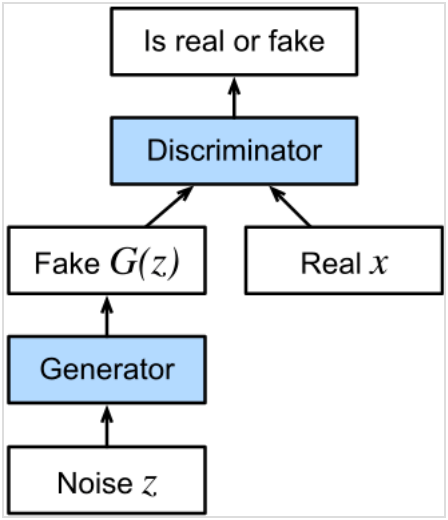
WIKIPEDIA

Generative adversarial network

(Redirected from Generative adversarial networks)

A **generative adversarial network** (**GAN**) is a class of machine learning frameworks and a prominent framework for approaching generative artificial intelligence.[1][2] The concept was initially developed by Ian Goodfellow and his colleagues in June 2014.[3] In a GAN, two neural networks contest with each other in the form of a zero-sum game, where one agent's gain is another agent's loss.



An illustration of how a GAN works

Given a training set, this technique learns to generate new data with the same statistics as the training set. For example, a GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers, having many realistic characteristics. Though originally proposed as a form of generative model for unsupervised learning, GANs have also proved useful for semi-supervised learning,[4] fully supervised learning,[5] and reinforcement learning.[6]

The core idea of a GAN is based on the "indirect" training through the discriminator, another neural network that can tell how "realistic" the input seems, which itself is also being updated dynamically.[7] This means that the generator is not trained to minimize the distance to a specific image, but rather to fool the discriminator. This enables the model to learn in an unsupervised manner.

GANs are similar to mimicry in evolutionary biology, with an evolutionary arms race between both networks.

Definition

Mathematical

The original GAN is defined as the following game:[3]

> Each probability space $(\Omega, \mu_{\text{ref}})$ defines a GAN game.
>
> There are 2 players: generator and discriminator.
>
> The generator's strategy set is $\mathcal{P}(\Omega)$, the set of all probability measures μ_G on Ω.

The discriminator's strategy set is the set of Markov kernels $\mu_D : \Omega \to \mathcal{P}[0,1]$, where $\mathcal{P}[0,1]$ is the set of probability measures on $[0,1]$.

The GAN game is a zero-sum game, with objective function

$$L(\mu_G, \mu_D) := \mathbf{E}_{x\sim\mu_{\text{ref}}, y\sim\mu_D(x)}[\ln y] + \mathbf{E}_{x\sim\mu_G, y\sim\mu_D(x)}[\ln(1-y)].$$

The generator aims to minimize the objective, and the discriminator aims to maximize the objective.

The generator's task is to approach $\mu_G \approx \mu_{\text{ref}}$, that is, to match its own output distribution as closely as possible to the reference distribution. The discriminator's task is to output a value close to 1 when the input appears to be from the reference distribution, and to output a value close to 0 when the input looks like it came from the generator distribution.

In practice

The *generative* network generates candidates while the *discriminative* network evaluates them.[3] The contest operates in terms of data distributions. Typically, the generative network learns to map from a latent space to a data distribution of interest, while the discriminative network distinguishes candidates produced by the generator from the true data distribution. The generative network's training objective is to increase the error rate of the discriminative network (i.e., "fool" the discriminator network by producing novel candidates that the discriminator thinks are not synthesized (are part of the true data distribution)).[3][8]

A known dataset serves as the initial training data for the discriminator. Training involves presenting it with samples from the training dataset until it achieves acceptable accuracy. The generator is trained based on whether it succeeds in fooling the discriminator. Typically, the generator is seeded with randomized input that is sampled from a predefined latent space (e.g. a multivariate normal distribution). Thereafter, candidates synthesized by the generator are evaluated by the discriminator. Independent backpropagation procedures are applied to both networks so that the generator produces better samples, while the discriminator becomes more skilled at flagging synthetic samples.[9] When used for image generation, the generator is typically a deconvolutional neural network, and the discriminator is a convolutional neural network.

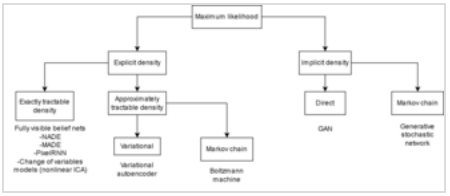
Relation to other statistical machine learning methods

GANs are **implicit generative models**,[10] which means that they do not explicitly model the likelihood function nor provide a means for finding the latent variable corresponding to a given sample, unlike alternatives such as flow-based generative model.

Compared to fully visible belief networks such as WaveNet and PixelRNN and autoregressive models in general, GANs can generate one complete sample in one pass, rather than multiple passes through the network.

Compared to Boltzmann machines and linear ICA, there is no restriction on the type of function used by the network.

Since neural networks are universal approximators, GANs are asymptotically consistent. Variational autoencoders might be universal approximators, but it is not proven as of 2017.[11]



Main types of deep generative models that perform maximum likelihood estimation[11]

Mathematical properties

Measure-theoretic considerations

This section provides some of the mathematical theory behind these methods.

In modern probability theory based on measure theory, a probability space also needs to be equipped with a σ-algebra. As a result, a more rigorous definition of the GAN game would make the following changes:

> Each probability space $(\Omega, \mathcal{B}, \mu_{\text{ref}})$ defines a GAN game.
>
> The generator's strategy set is $\mathcal{P}(\Omega, \mathcal{B})$, the set of all probability measures μ_G on the measure-space $(\Omega, \mathcal{B})$.
>
> The discriminator's strategy set is the set of Markov kernels $\mu_D : (\Omega, \mathcal{B}) \to \mathcal{P}([0,1], \mathcal{B}([0,1]))$, where $\mathcal{B}([0,1])$ is the Borel σ-algebra on $[0,1]$.

Since issues of measurability never arise in practice, these will not concern us further.

Choice of the strategy set

In the most generic version of the GAN game described above, the strategy set for the discriminator contains all Markov kernels $\mu_D : \Omega \to \mathcal{P}[0,1]$, and the strategy set for the generator contains arbitrary probability distributions μ_G on Ω.

However, as shown below, the optimal discriminator strategy against any μ_G is deterministic, so there is no loss of generality in restricting the discriminator's strategies to deterministic functions $D : \Omega \to [0,1]$. In most applications, D is a deep neural network function.

As for the generator, while μ_G could theoretically be any computable probability distribution, in practice, it is usually implemented as a pushforward: $\mu_G = \mu_Z \circ G^{-1}$. That is, start with a random variable $z \sim \mu_Z$, where μ_Z is a probability distribution that is easy to compute (such as the uniform distribution, or the Gaussian distribution), then define a function $G : \Omega_Z \to \Omega$. Then the distribution μ_G is the distribution of $G(z)$.

Consequently, the generator's strategy is usually defined as just G, leaving $z \sim \mu_Z$ implicit. In this formalism, the GAN game objective is

$$L(G, D) := \mathrm{E}_{x \sim \mu_{\mathrm{ref}}}[\ln D(x)] + \mathrm{E}_{z \sim \mu_Z}[\ln(1 - D(G(z)))].$$

Generative reparametrization

The GAN architecture has two main components. One is casting optimization into a game, of form $\min_G \max_D L(G, D)$, which is different from the usual kind of optimization, of form $\min_\theta L(\theta)$. The other is the decomposition of μ_G into $\mu_Z \circ G^{-1}$, which can be understood as a reparametrization trick.

To see its significance, one must compare GAN with previous methods for learning generative models, which were plagued with "intractable probabilistic computations that arise in maximum likelihood estimation and related strategies".[3]

At the same time, Kingma and Welling[12] and Rezende et al.[13] developed the same idea of reparametrization into a general stochastic backpropagation method. Among its first applications was the variational autoencoder.

Move order and strategic equilibria

In the original paper, as well as most subsequent papers, it is usually assumed that the generator *moves first*, and the discriminator *moves second*, thus giving the following minimax game:

$$\min_{\mu_G} \max_{\mu_D} L(\mu_G, \mu_D) := \mathrm{E}_{x \sim \mu_{\mathrm{ref}}, y \sim \mu_D(x)}[\ln y] + \mathrm{E}_{x \sim \mu_G, y \sim \mu_D(x)}[\ln(1 - y)].$$

If both the generator's and the discriminator's strategy sets are spanned by a finite number of strategies, then by the minimax theorem,

$$\min_{\mu_G} \max_{\mu_D} L(\mu_G, \mu_D) = \max_{\mu_D} \min_{\mu_G} L(\mu_G, \mu_D)$$

that is, the move order does not matter.

However, since the strategy sets are both not finitely spanned, the minimax theorem does not apply, and the idea of an "equilibrium" becomes delicate. To wit, there are the following different concepts of equilibrium:

- Equilibrium when generator moves first, and discriminator moves second:

$$\hat{\mu}_G \in \arg\min_{\mu_G} \max_{\mu_D} L(\mu_G, \mu_D), \quad \hat{\mu}_D \in \arg\max_{\mu_D} L(\hat{\mu}_G, \mu_D),$$

- Equilibrium when discriminator moves first, and generator moves second:

$$\hat{\mu}_D \in \arg\max_{\mu_D} \min_{\mu_G} L(\mu_G, \mu_D), \quad \hat{\mu}_G \in \arg\min_{\mu_G} L(\mu_G, \hat{\mu}_D),$$

- Nash equilibrium $(\hat{\mu}_D, \hat{\mu}_G)$, which is stable under simultaneous move order:

$$\hat{\mu}_D \in \arg\max_{\mu_D} L(\hat{\mu}_G, \mu_D), \quad \hat{\mu}_G \in \arg\min_{\mu_G} L(\mu_G, \hat{\mu}_D)$$

For general games, these equilibria do not have to agree, or even to exist. For the original GAN game, these equilibria all exist, and are all equal. However, for more general GAN games, these do not necessarily exist, or agree.[14]

Main theorems for GAN game

The original GAN paper proved the following two theorems:[3]

Theorem (the optimal discriminator computes the Jensen–Shannon divergence) — For any fixed generator strategy μ_G, let the optimal reply be $D^* = \arg\max_D L(\mu_G, D)$, then

$$D^*(x) = \frac{d\mu_{\text{ref}}}{d(\mu_{\text{ref}} + \mu_G)}$$

$$L(\mu_G, D^*) = 2D_{JS}(\mu_{\text{ref}}; \mu_G) - 2\ln 2$$

where the derivative is the Radon–Nikodym derivative, and D_{JS} is the Jensen–Shannon divergence.

Proof

By Jensen's inequality,

$$\mathrm{E}_{x\sim\mu_{\text{ref}}, y\sim\mu_D(x)}[\ln y] \leq \mathrm{E}_{x\sim\mu_{\text{ref}}}[\ln \mathrm{E}_{y\sim\mu_D(x)}[y]]$$

and similarly for the other term. Therefore, the optimal reply can be deterministic, i.e. $\mu_D(x) = \delta_{D(x)}$ for some function $D : \Omega \to [0, 1]$, in which case

$$L(\mu_G, \mu_D) := \mathrm{E}_{x\sim\mu_{\text{ref}}}[\ln D(x)] + \mathrm{E}_{x\sim\mu_G}[\ln(1 - D(x))].$$

To define suitable density functions, we define a base measure $\mu := \mu_{\text{ref}} + \mu_G$, which allows us to take the Radon–Nikodym derivatives

$$\rho_{\text{ref}} = \frac{d\mu_{\text{ref}}}{d\mu} \quad \rho_G = \frac{d\mu_G}{d\mu}$$

with $\rho_{\text{ref}} + \rho_G = 1$.

We then have

$$L(\mu_G, \mu_D) := \int \mu(dx) \left[\rho_{\text{ref}}(x) \ln(D(x)) + \rho_G(x) \ln(1 - D(x))\right].$$

The integrand is just the negative cross-entropy between two Bernoulli random variables with parameters $\rho_{\text{ref}}(x)$ and $D(x)$. We can write this as $-H(\rho_{\text{ref}}(x)) - D_{KL}(\rho_{\text{ref}}(x) \parallel D(x))$, where H is the binary entropy function, so

$$L(\mu_G, \mu_D) = -\int \mu(dx)(H(\rho_{\text{ref}}(x)) + D_{KL}(\rho_{\text{ref}}(x) \parallel D(x))).$$

This means that the optimal strategy for the discriminator is $D(x) = \rho_{\text{ref}}(x)$, with

$$L(\mu_G, \mu_D^*) = -\int \mu(dx) H(\rho_{\text{ref}}(x)) = D_{JS}(\mu_{\text{ref}} \parallel \mu_G) - 2\ln 2$$

after routine calculation.

Interpretation: For any fixed generator strategy μ_G, the optimal discriminator keeps track of the likelihood ratio between the reference distribution and the generator distribution:

$$\frac{D(x)}{1 - D(x)} = \frac{d\mu_{\text{ref}}}{d\mu_G}(x) = \frac{\mu_{\text{ref}}(dx)}{\mu_G(dx)}; \quad D(x) = \sigma(\ln \mu_{\text{ref}}(dx) - \ln \mu_G(dx))$$

where σ is the logistic function. In particular, if the prior probability for an image x to come from the reference distribution is equal to $\frac{1}{2}$, then $D(x)$ is just the posterior probability that x came from the reference distribution:

$$D(x) = \Pr(x \text{ came from reference distribution} \mid x).$$

Theorem (the unique equilibrium point) — For any GAN game, there exists a pair $(\hat{\mu}_D, \hat{\mu}_G)$ that is both a sequential equilibrium and a Nash equilibrium:

$$L(\hat{\mu}_G, \hat{\mu}_D) = \min_{\mu_G} \max_{\mu_D} L(\mu_G, \mu_D) = \qquad \max_{\mu_D} \min_{\mu_G} L(\mu_G, \mu_D) = -2\ln 2$$

$$\hat{\mu}_D \in \arg\max_{\mu_D} \min_{\mu_G} L(\mu_G, \mu_D), \qquad \hat{\mu}_G \in \arg\min_{\mu_G} \max_{\mu_D} L(\mu_G, \mu_D)$$

$$\hat{\mu}_D \in \arg\max_{\mu_D} L(\hat{\mu}_G, \mu_D), \qquad \hat{\mu}_G \in \arg\min_{\mu_G} L(\mu_G, \hat{\mu}_D)$$

$$\forall x \in \Omega, \hat{\mu}_D(x) = \delta_{\frac{1}{2}}, \qquad \hat{\mu}_G = \mu_{\text{ref}}$$

That is, the generator perfectly mimics the reference, and the discriminator outputs $\frac{1}{2}$ deterministically on all inputs.

Proof

From the previous proposition,

$$\arg\min_{\mu_G}\max_{\mu_D} L(\mu_G, \mu_D) = \mu_{\text{ref}}; \quad \min_{\mu_G}\max_{\mu_D} L(\mu_G, \mu_D) = -2\ln 2.$$

For any fixed discriminator strategy μ_D, any μ_G concentrated on the set

$$\{x \mid \mathrm{E}_{y\sim\mu_D(x)}[\ln(1-y)] = \inf_x \mathrm{E}_{y\sim\mu_D(x)}[\ln(1-y)]\}$$

is an optimal strategy for the generator. Thus,

$$\arg\max_{\mu_D}\min_{\mu_G} L(\mu_G, \mu_D) = \arg\max_{\mu_D} \mathrm{E}_{x\sim\mu_{\text{ref}}, y\sim\mu_D(x)}[\ln y] + \inf_x \mathrm{E}_{y\sim\mu_D(x)}[\ln(1-y)].$$

By Jensen's inequality, the discriminator can only improve by adopting the deterministic strategy of always playing $D(x) = \mathrm{E}_{y\sim\mu_D(x)}[y]$. Therefore,

$$\arg\max_{\mu_D}\min_{\mu_G} L(\mu_G, \mu_D) = \arg\max_{D} \mathrm{E}_{x\sim\mu_{\text{ref}}}[\ln D(x)] + \inf_x \ln(1-D(x))$$

By Jensen's inequality,

$$\begin{aligned}
&\ln \mathrm{E}_{x\sim\mu_{\text{ref}}}[D(x)] + \inf_x \ln(1-D(x)) \\
&= \ln \mathrm{E}_{x\sim\mu_{\text{ref}}}[D(x)] + \ln(1-\sup_x D(x)) \\
&= \ln[\mathrm{E}_{x\sim\mu_{\text{ref}}}[D(x)](1-\sup_x D(x))] \leq \ln[\sup_x D(x))(1-\sup_x D(x))] \leq \ln\frac{1}{4},
\end{aligned}$$

with equality if $D(x) = \frac{1}{2}$, so

$$\forall x \in \Omega, \hat{\mu}_D(x) = \delta_{\frac{1}{2}}; \quad \max_{\mu_D}\min_{\mu_G} L(\mu_G, \mu_D) = -2\ln 2.$$

Finally, to check that this is a Nash equilibrium, note that when $\mu_G = \mu_{\text{ref}}$, we have

$$L(\mu_G, \mu_D) := \mathrm{E}_{x\sim\mu_{\text{ref}}, y\sim\mu_D(x)}[\ln(y(1-y))]$$

which is always maximized by $y = \frac{1}{2}$.

When $\forall x \in \Omega, \mu_D(x) = \delta_{\frac{1}{2}}$, any strategy is optimal for the generator.

Training and evaluating GAN

Training

Unstable convergence

While the GAN game has a unique global equilibrium point when both the generator and discriminator have access to their entire strategy sets, the equilibrium is no longer guaranteed when they have a restricted strategy set.[14]

In practice, the generator has access only to measures of form $\mu_Z \circ G_\theta^{-1}$, where G_θ is a function computed by a neural network with parameters θ, and μ_Z is an easily sampled distribution, such as the uniform or normal distribution. Similarly, the discriminator has access only to functions of form D_ζ, a function computed by a neural network with parameters ζ. These restricted strategy sets take up a *vanishingly small proportion* of their entire strategy sets.[15]

Further, even if an equilibrium still exists, it can only be found by searching in the high-dimensional space of all possible neural network functions. The standard strategy of using gradient descent to find the equilibrium often does not work for GAN, and often the game "collapses" into one of several failure modes. To improve the convergence stability, some training strategies start with an easier task, such as generating low-resolution images[16] or simple images (one object with uniform background),[17] and gradually increase the difficulty of the task during training. This essentially translates to applying a curriculum learning scheme.[18]

Mode collapse

GANs often suffer from **mode collapse** where they fail to generalize properly, missing entire modes from the input data. For example, a GAN trained on the MNIST dataset containing many samples of each digit might only generate pictures of digit 0. This was termed "the Helvetica scenario".[3]

One way this can happen is if the generator learns too fast compared to the discriminator. If the discriminator D is held constant, then the optimal generator would only output elements of $\arg\max_x D(x)$.[19] So for example, if during GAN training for generating MNIST dataset, for a few epochs, the discriminator somehow prefers the digit 0 slightly more than other digits, the generator may seize the opportunity to generate only digit 0, then be unable to escape the local minimum after the discriminator improves.

Some researchers perceive the root problem to be a weak discriminative network that fails to notice the pattern of omission, while others assign blame to a bad choice of objective function. Many solutions have been proposed, but it is still an open problem.[20][21]

Even the state-of-the-art architecture, BigGAN (2019), could not avoid mode collapse. The authors resorted to "allowing collapse to occur at the later stages of training, by which time a model is sufficiently trained to achieve good results".[22]

Two time-scale update rule

The **two time-scale update rule (TTUR)** is proposed to make GAN convergence more stable by making the learning rate of the generator lower than that of the discriminator. The authors argued that the generator should move slower than the discriminator, so that it does not "drive the discriminator steadily into new regions without capturing its gathered information".

They proved that a general class of games that included the GAN game, when trained under TTUR, "converges under mild assumptions to a stationary local Nash equilibrium".[23]

They also proposed using the Adam stochastic optimization[24] to avoid mode collapse, as well as the Fréchet inception distance for evaluating GAN performances.

Vanishing gradient

Conversely, if the discriminator learns too fast compared to the generator, then the discriminator could almost perfectly distinguish $\mu_{G_\theta}, \mu_{\text{ref}}$. In such case, the generator G_θ could be stuck with a very high loss no matter which direction it changes its θ, meaning that the gradient $\nabla_\theta L(G_\theta, D_\zeta)$ would be close to zero. In such case, the generator cannot learn, a case of the **vanishing gradient** problem.[15]

Intuitively speaking, the discriminator is too good, and since the generator cannot take any small step (only small steps are considered in gradient descent) to improve its payoff, it does not even try.

One important method for solving this problem is the Wasserstein GAN.

Evaluation

GANs are usually evaluated by Inception score (IS), which measures how varied the generator's outputs are (as classified by an image classifier, usually Inception-v3), or Fréchet inception distance (FID), which measures how similar the generator's outputs are to a reference set (as classified by a learned image featurizer, such as Inception-v3 without its final layer). Many papers that propose new GAN architectures for image generation report how their architectures break the state of the art on FID or IS.

Another evaluation method is the Learned Perceptual Image Patch Similarity (LPIPS), which starts with a learned image featurizer $f_\theta : \text{Image} \to \mathbb{R}^n$, and finetunes it by supervised learning on a set of $(x, x', \text{perceptual difference}(x, x'))$, where x is an image, x' is a perturbed version of it, and $\text{perceptual difference}(x, x')$ is how much they differ, as reported by human subjects. The model is finetuned so that it can approximate $\|f_\theta(x) - f_\theta(x')\| \approx \text{perceptual difference}(x, x')$. This finetuned model is then used to define $\text{LPIPS}(x, x') := \|f_\theta(x) - f_\theta(x')\|$.[25]

Other evaluation methods are reviewed in.[26]

Variants

There is a veritable zoo of GAN variants.[27] Some of the most prominent are as follows:

Conditional GAN

Conditional GANs are similar to standard GANs except they allow the model to conditionally generate samples based on additional information. For example, if we want to generate a cat face given a dog picture, we could use a conditional GAN.

The generator in a GAN game generates μ_G, a probability distribution on the probability space Ω. This leads to the idea of a conditional GAN, where instead of generating one probability distribution on Ω, the generator generates a different probability distribution $\mu_G(c)$ on Ω, for each given class label c.

For example, for generating images that look like ImageNet, the generator should be able to generate a picture of cat when given the class label "cat".

In the original paper,[3] the authors noted that GAN can be trivially extended to conditional GAN by providing the labels to both the generator and the discriminator.

Concretely, the conditional GAN game is just the GAN game with class labels provided:

$$L(\mu_G, D) := \mathbf{E}_{c\sim\mu_C, x\sim\mu_{\text{ref}}(c)}[\ln D(x,c)] + \mathbf{E}_{c\sim\mu_C, x\sim\mu_G(c)}[\ln(1 - D(x,c))]$$

where μ_C is a probability distribution over classes, $\mu_{\text{ref}}(c)$ is the probability distribution of real images of class c, and $\mu_G(c)$ the probability distribution of images generated by the generator when given class label c.

In 2017, a conditional GAN learned to generate 1000 image classes of ImageNet.[28]

GANs with alternative architectures

The GAN game is a general framework and can be run with any reasonable parametrization of the generator G and discriminator D. In the original paper, the authors demonstrated it using multilayer perceptron networks and convolutional neural networks. Many alternative architectures have been tried.

Deep convolutional GAN (DCGAN):[29] For both generator and discriminator, uses only deep networks consisting entirely of convolution-deconvolution layers, that is, fully convolutional networks.[30]

Self-attention GAN (SAGAN):[31] Starts with the DCGAN, then adds residually-connected standard self-attention modules to the generator and discriminator.

Variational autoencoder GAN (VAEGAN):[32] Uses a variational autoencoder (VAE) for the generator.

Transformer GAN (TransGAN):[33] Uses the pure transformer architecture for both the generator and discriminator, entirely devoid of convolution-deconvolution layers.

Flow-GAN:[34] Uses flow-based generative model for the generator, allowing efficient computation of the likelihood function.

GANs with alternative objectives

Many GAN variants are merely obtained by changing the loss functions for the generator and discriminator.

Original GAN:

We recast the original GAN objective into a form more convenient for comparison:

$$\begin{cases} \min_D L_D(D, \mu_G) = -\mathbf{E}_{x\sim\mu_G}[\ln D(x)] - \mathbf{E}_{x\sim\mu_{\text{ref}}}[\ln(1 - D(x))] \\ \min_G L_G(D, \mu_G) = -\mathbf{E}_{x\sim\mu_G}[\ln(1 - D(x))] \end{cases}$$

Original GAN, non-saturating loss:

This objective for generator was recommended in the original paper for faster convergence.[3]

$$L_G = \mathbf{E}_{x\sim\mu_G}[\ln D(x)]$$

The effect of using this objective is analyzed in Section 2.2.2 of Arjovsky et al.[35]

Original GAN, maximum likelihood:

$$L_G = \mathbf{E}_{x\sim\mu_G}[(\exp \circ \sigma^{-1} \circ D)(x)]$$

where σ is the logistic function. When the discriminator is optimal, the generator gradient is the same as in maximum likelihood estimation, even though GAN cannot perform maximum likelihood estimation *itself*.[36][37]

Hinge loss GAN:[38]

$$L_D = -\mathbf{E}_{x\sim p_{\text{ref}}}[\min(0, -1 + D(x))] - \mathbf{E}_{x\sim\mu_G}[\min(0, -1 - D(x))]$$

$$L_G = -\mathbf{E}_{x\sim\mu_G}[D(x)]$$

Least squares GAN:[39]

$$L_D = \mathbf{E}_{x\sim\mu_{\text{ref}}}[(D(x) - b)^2] + \mathbf{E}_{x\sim\mu_G}[(D(x) - a)^2]$$

$$L_G = \mathbf{E}_{x \sim \mu_G}[(D(x) - c)^2]$$

where a, b, c are parameters to be chosen. The authors recommended $a = -1, b = 1, c = 0$.

Wasserstein GAN (WGAN)

The Wasserstein GAN modifies the GAN game at two points:

- The discriminator's strategy set is the set of measurable functions of type $D : \Omega \to \mathbb{R}$ with bounded Lipschitz norm: $\|D\|_L \leq K$, where K is a fixed positive constant.
- The objective is

$$L_{WGAN}(\mu_G, D) := \mathbf{E}_{x \sim \mu_G}[D(x)] - \mathbb{E}_{x \sim \mu_{\text{ref}}}[D(x)]$$

One of its purposes is to solve the problem of mode collapse (see above).[15] The authors claim "In no experiment did we see evidence of mode collapse for the WGAN algorithm".

GANs with more than two players

Adversarial autoencoder

An adversarial autoencoder (AAE)[40] is more autoencoder than GAN. The idea is to start with a plain autoencoder, but train a discriminator to discriminate the latent vectors from a reference distribution (often the normal distribution).

InfoGAN

In conditional GAN, the generator receives both a noise vector z and a label c, and produces an image $G(z, c)$. The discriminator receives image-label pairs (x, c), and computes $D(x, c)$.

When the training dataset is unlabeled, conditional GAN does not work directly.

The idea of InfoGAN is to decree that every latent vector in the latent space can be decomposed as (z, c): an incompressible noise part z, and an informative label part c, and encourage the generator to comply with the decree, by encouraging it to maximize $I(c, G(z, c))$, the mutual information between c and $G(z, c)$, while making no demands on the mutual information z between $G(z, c)$.

Unfortunately, $I(c, G(z, c))$ is intractable in general, The key idea of InfoGAN is Variational Mutual Information Maximization:[41] indirectly maximize it by maximizing a lower bound

$$\hat{I}(G, Q) = \mathbb{E}_{z \sim \mu_Z, c \sim \mu_C}[\ln Q(c \mid G(z, c))]; \quad I(c, G(z, c)) \geq \sup_Q \hat{I}(G, Q)$$

where Q ranges over all Markov kernels of type $Q : \Omega_Y \to \mathcal{P}(\Omega_C)$.

The InfoGAN game is defined as follows:[42]

> Three probability spaces define an InfoGAN game:

- $(\Omega_X, \mu_{\text{ref}})$, the space of reference images.
- (Ω_Z, μ_Z), the fixed random noise generator.
- (Ω_C, μ_C), the fixed random information generator.

There are 3 players in 2 teams: generator, Q, and discriminator. The generator and Q are on one team, and the discriminator on the other team.

The objective function is

$$L(G, Q, D) = L_{GAN}(G, D) - \lambda \hat{I}(G, Q)$$

where $L_{GAN}(G, D) = \mathbf{E}_{x \sim \mu_{\text{ref}}}[\ln D(x)] + \mathbf{E}_{z \sim \mu_Z}[\ln(1 - D(G(z, c)))]$ is the original GAN game objective, and $\hat{I}(G, Q) = \mathbb{E}_{z \sim \mu_Z, c \sim \mu_C}[\ln Q(c \mid G(z, c))]$

Generator-Q team aims to minimize the objective, and discriminator aims to maximize it:

$$\min_{G,Q} \max_{D} L(G, Q, D)$$

Bidirectional GAN (BiGAN)

The standard GAN generator is a function of type $G : \Omega_Z \to \Omega_X$, that is, it is a mapping from a latent space Ω_Z to the image space Ω_X. This can be understood as a "decoding" process, whereby every latent vector $z \in \Omega_Z$ is a code for an image $x \in \Omega_X$, and the generator performs the decoding. This naturally leads to the idea of training another network that performs "encoding", creating an autoencoder out of the encoder-generator pair.

Already in the original paper,[3] the authors noted that "Learned approximate inference can be performed by training an auxiliary network to predict z given x". The bidirectional GAN architecture performs exactly this.[43]

The BiGAN is defined as follows:

Two probability spaces define a BiGAN game:

- (Ω_X, μ_X), the space of reference images.
- (Ω_Z, μ_Z), the latent space.

There are 3 players in 2 teams: generator, encoder, and discriminator. The generator and encoder are on one team, and the discriminator on the other team.

The generator's strategies are functions $G : \Omega_Z \to \Omega_X$, and the encoder's strategies are functions $E : \Omega_X \to \Omega_Z$. The discriminator's strategies are functions $D : \Omega_X \to [0, 1]$.

The objective function is

$$L(G, E, D) = \mathbb{E}_{x \sim \mu_X}[\ln D(x, E(x))] + \mathbb{E}_{z \sim \mu_Z}[\ln(1 - D(G(z), z))]$$

Generator-encoder team aims to minimize the objective, and discriminator aims to maximize it:

$$\min_{G,E} \max_{D} L(G, E, D)$$

In the paper, they gave a more abstract definition of the objective as:

$$L(G, E, D) = \mathbb{E}_{(x,z)\sim\mu_{E,X}}[\ln D(x, z)] + \mathbb{E}_{(x,z)\sim\mu_{G,Z}}[\ln(1 - D(x, z))]$$

where $\mu_{E,X}(dx, dz) = \mu_X(dx) \cdot \delta_{E(x)}(dz)$ is the probability distribution on $\Omega_X \times \Omega_Z$ obtained by pushing μ_X forward via $x \mapsto (x, E(x))$, and $\mu_{G,Z}(dx, dz) = \delta_{G(z)}(dx) \cdot \mu_Z(dz)$ is the probability distribution on $\Omega_X \times \Omega_Z$ obtained by pushing μ_Z forward via $z \mapsto (G(x), z)$.

Applications of bidirectional models include semi-supervised learning,[44] interpretable machine learning,[45] and neural machine translation.[46]

CycleGAN

CycleGAN is an architecture for performing translations between two domains, such as between photos of horses and photos of zebras, or photos of night cities and photos of day cities.

The CycleGAN game is defined as follows:[47]

There are two probability spaces $(\Omega_X, \mu_X), (\Omega_Y, \mu_Y)$, corresponding to the two domains needed for translations fore-and-back.

There are 4 players in 2 teams: generators $G_X : \Omega_X \to \Omega_Y, G_Y : \Omega_Y \to \Omega_X$, and discriminators $D_X : \Omega_X \to [0, 1], D_Y : \Omega_Y \to [0, 1]$.

The objective function is

$$L(G_X, G_Y, D_X, D_Y) = L_{GAN}(G_X, D_X) + L_{GAN}(G_Y, D_Y) + \lambda L_{cycle}(G_X, G_Y)$$

where λ is a positive adjustable parameter, L_{GAN} is the GAN game objective, and L_{cycle} is the *cycle consistency loss*:

$$L_{cycle}(G_X, G_Y) = E_{x\sim\mu_X}\|G_X(G_Y(x)) - x\| + E_{y\sim\mu_Y}\|G_Y(G_X(y)) - y\|$$

The generators aim to minimize the objective, and the discriminators aim to maximize it:

$$\min_{G_X,G_Y} \max_{D_X,D_Y} L(G_X, G_Y, D_X, D_Y)$$

Unlike previous work like pix2pix,[48] which requires paired training data, cycleGAN requires no paired data. For example, to train a pix2pix model to turn a summer scenery photo to winter scenery photo and back, the dataset must contain pairs of the same place in summer and winter, shot at the same angle; cycleGAN would only need a set of summer scenery photos, and an unrelated set of winter scenery photos.

GANs with particularly large or small scales

BigGAN

The BigGAN is essentially a self-attention GAN trained on a large scale (up to 80 million parameters) to generate large images of ImageNet (up to 512 x 512 resolution), with numerous engineering tricks to make it converge.[22][49]

Invertible data augmentation

When there is insufficient training data, the reference distribution μ_{ref} cannot be well-approximated by the empirical distribution given by the training dataset. In such cases, data augmentation can be applied, to allow training GAN on smaller datasets. Naïve data augmentation, however, brings its problems.

Consider the original GAN game, slightly reformulated as follows:

$$\begin{cases} \min_D L_D(D, \mu_G) = -\mathbf{E}_{x\sim\mu_{\text{ref}}}[\ln D(x)] - \mathbf{E}_{x\sim\mu_G}[\ln(1 - D(x))] \\ \min_G L_G(D, \mu_G) = -\mathbf{E}_{x\sim\mu_G}[\ln(1 - D(x))] \end{cases}$$

Now we use data augmentation by randomly sampling semantic-preserving transforms $T : \Omega \to \Omega$ and applying them to the dataset, to obtain the reformulated GAN game:

$$\begin{cases} \min_D L_D(D, \mu_G) = -\mathbf{E}_{x\sim\mu_{\text{ref}}, T\sim\mu_{\text{trans}}}[\ln D(T(x))] - \mathbf{E}_{x\sim\mu_G}[\ln(1 - D(x))] \\ \min_G L_G(D, \mu_G) = -\mathbf{E}_{x\sim\mu_G}[\ln(1 - D(x))] \end{cases}$$

This is equivalent to a GAN game with a different distribution μ'_{ref}, sampled by $T(x)$, with $x \sim \mu_{\text{ref}}, T \sim \mu_{\text{trans}}$. For example, if μ_{ref} is the distribution of images in ImageNet, and μ_{trans} samples identity-transform with probability 0.5, and horizontal-reflection with probability 0.5, then μ'_{ref} is the distribution of images in ImageNet and horizontally-reflected ImageNet, combined.

The result of such training would be a generator that mimics μ'_{ref}. For example, it would generate images that look like they are randomly cropped, if the data augmentation uses random cropping.

The solution is to apply data augmentation to both generated and real images:

$$\begin{cases} \min_D L_D(D, \mu_G) = -\mathbf{E}_{x\sim\mu_{\text{ref}}, T\sim\mu_{\text{trans}}}[\ln D(T(x))] - \mathbf{E}_{x\sim\mu_G, T\sim\mu_{\text{trans}}}[\ln(1 - D(T(x)))] \\ \min_G L_G(D, \mu_G) = -\mathbf{E}_{x\sim\mu_G, T\sim\mu_{\text{trans}}}[\ln(1 - D(T(x)))] \end{cases}$$

The authors demonstrated high-quality generation using just 100-picture-large datasets.[50]

The StyleGAN-2-ADA paper points out a further point on data augmentation: it must be *invertible*.[51] Continue with the example of generating ImageNet pictures. If the data augmentation is "randomly rotate the picture by 0, 90, 180, 270 degrees with *equal* probability", then there is no way for the generator to know which is the true orientation: Consider two generators G, G', such that for any latent z, the generated image $G(z)$ is a 90-degree rotation of $G'(z)$. They would have exactly the same expected loss, and so neither is preferred over the other.

The solution is to only use invertible data augmentation: instead of "randomly rotate the picture by 0, 90, 180, 270 degrees with *equal* probability", use "randomly rotate the picture by 90, 180, 270 degrees with 0.1 probability, and keep the picture as it is with 0.7 probability". This way, the generator is still rewarded to keep images oriented the same way as un-augmented ImageNet pictures.

Abstractly, the effect of randomly sampling transformations $T : \Omega \to \Omega$ from the distribution μ_{trans} is to define a Markov kernel $K_{\text{trans}} : \Omega \to \mathcal{P}(\Omega)$. Then, the data-augmented GAN game pushes the generator to find some $\hat{\mu}_G \in \mathcal{P}(\Omega)$, such that

$$K_{\text{trans}} * \mu_{\text{ref}} = K_{\text{trans}} * \hat{\mu}_G$$

where $*$ is the Markov kernel convolution. A data-augmentation method is defined to be *invertible* if its Markov kernel K_{trans} satisfies

$$K_{\text{trans}} * \mu = K_{\text{trans}} * \mu' \implies \mu = \mu' \quad \forall \mu, \mu' \in \mathcal{P}(\Omega)$$

Immediately by definition, we see that composing multiple invertible data-augmentation methods results in yet another invertible method. Also by definition, if the data-augmentation method is invertible, then using it in a GAN game does not change the optimal strategy $\hat{\mu}_G$ for the generator, which is still μ_{ref}.

There are two prototypical examples of invertible Markov kernels:

Discrete case: Invertible stochastic matrices, when Ω is finite.

For example, if $\Omega = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ is the set of four images of an arrow, pointing in 4 directions, and the data augmentation is "randomly rotate the picture by 90, 180, 270 degrees with probability p, and keep the picture as it is with probability $(1 - 3p)$", then the Markov kernel K_{trans} can be represented as a stochastic matrix:

$$[K_{\text{trans}}] = \begin{bmatrix} (1-3p) & p & p & p \\ p & (1-3p) & p & p \\ p & p & (1-3p) & p \\ p & p & p & (1-3p) \end{bmatrix}$$

and K_{trans} is an invertible kernel iff $[K_{\text{trans}}]$ is an invertible matrix, that is, $p \neq 1/4$.

Continuous case: The gaussian kernel, when $\Omega = \mathbb{R}^n$ for some $n \geq 1$.

For example, if $\Omega = \mathbb{R}^{256^2}$ is the space of 256x256 images, and the data-augmentation method is "generate a gaussian noise $z \sim \mathcal{N}(0, I_{256^2})$, then add ϵz to the image", then K_{trans} is just convolution by the density function of $\mathcal{N}(0, \epsilon^2 I_{256^2})$. This is invertible, because convolution by a gaussian is just

convolution by the heat kernel, so given any $\mu \in \mathcal{P}(\mathbb{R}^n)$, the convolved distribution $K_{\text{trans}} * \mu$ can be obtained by heating up $\mathbb{R}^n$ precisely according to μ, then wait for time $\epsilon^2/4$. With that, we can recover μ by running the heat equation *backwards in time* for $\epsilon^2/4$.

More examples of invertible data augmentations are found in the paper.[51]

SinGAN

SinGAN pushes data augmentation to the limit, by using only a single image as training data and performing data augmentation on it. The GAN architecture is adapted to this training method by using a multi-scale pipeline.

The generator G is decomposed into a pyramid of generators $G = G_1 \circ G_2 \circ \cdots \circ G_N$, with the lowest one generating the image $G_N(z_N)$ at the lowest resolution, then the generated image is scaled up to $r(G_N(z_N))$, and fed to the next level to generate an image $G_{N-1}(z_{N-1} + r(G_N(z_N)))$ at a higher resolution, and so on. The discriminator is decomposed into a pyramid as well.[52]

StyleGAN series

The StyleGAN family is a series of architectures published by Nvidia's research division.

Progressive GAN

Progressive GAN[16] is a method for training GAN for large-scale image generation stably, by growing a GAN generator from small to large scale in a pyramidal fashion. Like SinGAN, it decomposes the generator as$G = G_1 \circ G_2 \circ \cdots \circ G_N$, and the discriminator as $D = D_1 \circ D_2 \circ \cdots \circ D_N$.

During training, at first only G_N, D_N are used in a GAN game to generate 4x4 images. Then G_{N-1}, D_{N-1} are added to reach the second stage of GAN game, to generate 8x8 images, and so on, until we reach a GAN game to generate 1024x1024 images.

To avoid shock between stages of the GAN game, each new layer is "blended in" (Figure 2 of the paper[16]). For example, this is how the second stage GAN game starts:

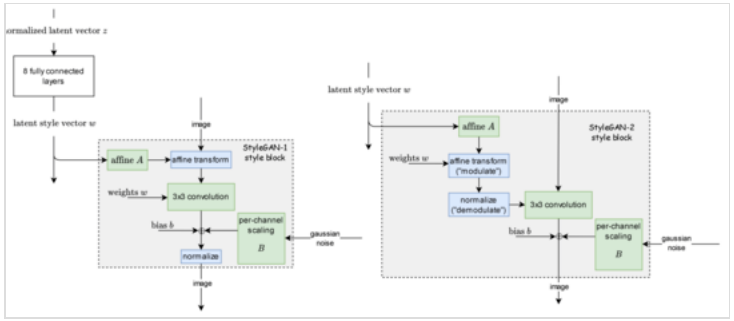
- Just before, the GAN game consists of the pair G_N, D_N generating and discriminating 4x4 images.
- Just after, the GAN game consists of the pair $((1-\alpha) + \alpha \cdot G_{N-1}) \circ u \circ G_N, D_N \circ d \circ ((1-\alpha) + \alpha \cdot D_{N-1})$ generating and discriminating 8x8 images. Here, the functions u, d are image up- and down-sampling functions, and α is a blend-in factor (much like an alpha in image composing) that smoothly glides from 0 to 1.

StyleGAN-1

StyleGAN-1 is designed as a combination of Progressive GAN with neural style transfer.[53]

The key architectural choice of StyleGAN-1 is a progressive growth mechanism, similar to Progressive GAN. Each generated image starts as a constant $4 \times 4 \times 512$ array, and repeatedly passed through style blocks. Each style block applies a "style latent vector" via affine transform ("adaptive instance

normalization"), similar to how neural style transfer uses Gramian matrix. It then adds noise, and normalize (subtract the mean, then divide by the variance).



The main architecture of StyleGAN-1 and StyleGAN-2

At training time, usually only one style latent vector is used per image generated, but sometimes two ("mixing regularization") in order to encourage each style block to independently perform its stylization without expecting help from other style blocks (since they might receive an entirely different style latent vector).

After training, multiple style latent vectors can be fed into each style block. Those fed to the lower layers control the large-scale styles, and those fed to the higher layers control the fine-detail styles.

Style-mixing between two images x, x' can be performed as well. First, run a gradient descent to find z, z' such that $G(z) \approx x, G(z') \approx x'$. This is called "projecting an image back to style latent space". Then, z can be fed to the lower style blocks, and z' to the higher style blocks, to generate a composite image that has the large-scale style of x, and the fine-detail style of x'. Multiple images can also be composed this way.

StyleGAN-2

StyleGAN-2 improves upon StyleGAN-1, by using the style latent vector to transform the convolution layer's weights instead, thus solving the "blob" problem.[54]

This was updated by the StyleGAN-2-ADA ("ADA" stands for "adaptive"),[51] which uses invertible data augmentation as described above. It also tunes the amount of data augmentation applied by starting at zero, and gradually increasing it until an "overfitting heuristic" reaches a target level, thus the name "adaptive".

StyleGAN-3

StyleGAN-3[55] improves upon StyleGAN-2 by solving the "texture sticking" problem, which can be seen in the official videos.[56] They analyzed the problem by the Nyquist–Shannon sampling theorem, and argued that the layers in the generator learned to exploit the high-frequency signal in the pixels they operate upon.

To solve this, they proposed imposing strict lowpass filters between each generator's layers, so that the generator is forced to operate on the pixels in a way faithful to the continuous signals they represent, rather than operate on them as merely discrete signals. They further imposed rotational and translational invariance by using more signal filters. The resulting StyleGAN-3 is able to solve the texture sticking problem, as well as generating images that rotate and translate smoothly.

Other uses

Other than for generative and discriminative modelling of data, GANs have been used for other things.

GANs have been used for transfer learning to enforce the alignment of the latent feature space, such as in deep reinforcement learning.[57] This works by feeding the embeddings of the source and target task to the discriminator which tries to guess the context. The resulting loss is then (inversely) backpropagated through the encoder.

Applications

Science

- Iteratively reconstruct astronomical images[58]
- Simulate gravitational lensing for dark matter research.[59][60][61]
- Model the distribution of dark matter in a particular direction in space and to predict the gravitational lensing that will occur.[62][63]
- Model high energy jet formation[64] and showers through calorimeters of high-energy physics experiments.[65][66][67][68]
- Approximate bottlenecks in computationally expensive simulations of particle physics experiments. Applications in the context of present and proposed CERN experiments have demonstrated the potential of these methods for accelerating simulation and/or improving simulation fidelity.[69][70]
- Reconstruct velocity and scalar fields in turbulent flows.[71][72][73]

GAN-generated molecules were validated experimentally in mice.[74][75]

Medical

One of the major concerns in medical imaging is preserving patient privacy. Due to these reasons, researchers often face difficulties in obtaining medical images for their research purposes. GAN has been used for generating synthetic medical images, such as MRI and PET images to address this challenge. [76]

GAN can be used to detect glaucomatous images helping the early diagnosis which is essential to avoid partial or total loss of vision.[77]

GANs have been used to create forensic facial reconstructions of deceased historical figures.[78]

Malicious

Concerns have been raised about the potential use of GAN-based human image synthesis for sinister purposes, e.g., to produce fake, possibly incriminating, photographs and videos.[79] GANs can be used to generate unique, realistic profile photos of people who do not exist, in order to automate creation of fake social media profiles.[80]

An image generated by a StyleGAN that looks deceptively like a photograph of a real person. This image was generated by a StyleGAN based on an analysis of portraits.

In 2019 the state of California considered[81] and passed on October 3, 2019, the bill AB-602 (https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201920200AB602), which bans the use of human image synthesis technologies to make fake pornography without the consent of the people depicted, and bill AB-730 (https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201920200AB730), which prohibits distribution of manipulated videos of a political candidate within 60 days of an election. Both bills were authored by Assembly member Marc Berman and signed by Governor Gavin Newsom. The laws went into effect in 2020.[82]

DARPA's Media Forensics program studies ways to counteract fake media, including fake media produced using GANs.[83]

Another example of a GAN generated portrait

Fashion, art and advertising

GANs can be used to generate art; *The Verge* wrote in March 2019 that "The images created by GANs have become the defining look of contemporary AI art."[84] GANs can also be used to

- inpaint photographs[85]
- generate fashion models,[86] shadows,[87] photorealistic renders of interior design, industrial design, shoes, etc.[88] Such networks were reported to be used by Facebook.[89]

Some have worked with using GAN for artistic creativity, as "creative adversarial network".[90][91] A GAN, trained on a set of 15,000 portraits from WikiArt from the 14th to the 19th century, created the 2018 painting *Edmond de Belamy,* which sold for US$432,500.[92]

GANs were used by the video game modding community to up-scale low-resolution 2D textures in old video games by recreating them in 4k or higher resolutions via image training, and then down-sampling them to fit the game's native resolution (resembling supersampling anti-aliasing).[93]

In 2020, Artbreeder was used to create the main antagonist in the sequel to the psychological web horror series *Ben Drowned*. The author would later go on to praise GAN applications for their ability to help generate assets for independent artists who are short on budget and manpower.[94][95]

In May 2020, Nvidia researchers taught an AI system (termed "GameGAN") to recreate the game of *Pac-Man* simply by watching it being played.[96][97]

In August 2019, a large dataset consisting of 12,197 MIDI songs each with paired lyrics and melody alignment was created for neural melody generation from lyrics using conditional GAN-LSTM (refer to sources at GitHub AI Melody Generation from Lyrics (https://github.com/yy1lab/Lyrics-Conditioned-Neural-Melody-Generation)).[98]

Miscellaneous

GANs have been used to

- show how an individual's appearance might change with age.[99]
- reconstruct 3D models of objects from images,[100]
- generate novel objects as 3D point clouds,[101]
- model patterns of motion in video.[102]
- inpaint missing features in maps, transfer map styles in cartography[103] or augment street view imagery.[104]
- use feedback to generate images and replace image search systems.[105]
- visualize the effect that climate change will have on specific houses.[106]
- reconstruct an image of a person's face after listening to their voice.[107]
- produces videos of a person speaking, given only a single photo of that person.[108]
- recurrent sequence generation.[109]

History

In 1991, Juergen Schmidhuber published "artificial curiosity", neural networks in a zero-sum game.[110] The first network is a generative model that models a probability distribution over output patterns. The second network learns by gradient descent to predict the reactions of the environment to these patterns. GANs can be regarded as a case where the environmental reaction is 1 or 0 depending on whether the first network's output is in a given set.[111]

Other people had similar ideas but did not develop them similarly. An idea involving adversarial networks was published in a 2010 blog post by Olli Niemitalo.[112] This idea was never implemented and did not involve stochasticity in the generator and thus was not a generative model. It is now known as a conditional GAN or cGAN.[113] An idea similar to GANs was used to model animal behavior by Li, Gauci and Gross in 2013.[114]

Another inspiration for GANs was noise-contrastive estimation,[115] which uses the same loss function as GANs and which Goodfellow studied during his PhD in 2010–2014.

Adversarial machine learning has other uses besides generative modeling and can be applied to models other than neural networks. In control theory, adversarial learning based on neural networks was used in 2006 to train robust controllers in a game theoretic sense, by alternating the iterations between a minimizer policy, the controller, and a maximizer policy, the disturbance.[116][117]

In 2017, a GAN was used for image enhancement focusing on realistic textures rather than pixel-accuracy, producing a higher image quality at high magnification.[118] In 2017, the first faces were generated.[119] These were exhibited in February 2018 at the Grand Palais.[120][121] Faces generated by StyleGAN[122] in 2019 drew comparisons with Deepfakes.[123][124][125]

See also

- Artificial intelligence art – Machine application of knowledge of human aesthetic expressions
- Deepfake – Realistic artificially generated media
- Deep learning – Branch of machine learning
- Diffusion model – Deep learning algorithm
- Generative artificial intelligence – AI system capable of generating content in response to prompts
- Synthetic media – Artificial production, manipulation, and modification of data and media by automated means

References

1. "Generative AI and Future" (https://pub.towardsai.net/generative-ai-and-future-c3b1695876f2). November 15, 2022.
2. "CSDL | IEEE Computer Society" (https://www.computer.org/csdl/magazine/co/2022/10/09903869/1H0G6xvtREk).
3. Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). *Generative Adversarial Nets* (https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf) (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.
4. Salimans, Tim; Goodfellow, Ian; Zaremba, Wojciech; Cheung, Vicki; Radford, Alec; Chen, Xi (2016). "Improved Techniques for Training GANs". arXiv:1606.03498 (https://arxiv.org/abs/1606.03498) [cs.LG (https://arxiv.org/archive/cs.LG)].
5. Isola, Phillip; Zhu, Jun-Yan; Zhou, Tinghui; Efros, Alexei (2017). "Image-to-Image Translation with Conditional Adversarial Nets" (https://phillipi.github.io/pix2pix/). *Computer Vision and Pattern Recognition*.
6. Ho, Jonathon; Ermon, Stefano (2016). "Generative Adversarial Imitation Learning" (http://papers.nips.cc/paper/6391-generative-adversarial-imitation-learning). *Advances in Neural Information Processing Systems*. **29**: 4565–4573. arXiv:1606.03476 (https://arxiv.org/abs/1606.03476).
7. "Vanilla GAN (GANs in computer vision: Introduction to generative learning)" (https://theaisummer.com/gan-computer-vision/#vanilla-gan-generative-adversarial-networks-2014). *theaisummer.com*. AI Summer. April 10, 2020. Archived (https://web.archive.org/web/20200603130655/https://theaisummer.com/gan-computer-vision/) from the original on June 3, 2020. Retrieved September 20, 2020.
8. Luc, Pauline; Couprie, Camille; Chintala, Soumith; Verbeek, Jakob (November 25, 2016). "Semantic Segmentation using Adversarial Networks". *NIPS Workshop on Adversarial Training, Dec, Barcelona, Spain*. **2016**. arXiv:1611.08408 (https://arxiv.org/abs/1611.08408).

9. Andrej Karpathy; Pieter Abbeel; Greg Brockman; Peter Chen; Vicki Cheung; Rocky Duan; Ian Goodfellow; Durk Kingma; Jonathan Ho; Rein Houthooft; Tim Salimans; John Schulman; Ilya Sutskever; Wojciech Zaremba, *Generative Models* (https://openai.com/blog/generative-models/), OpenAI, retrieved April 7, 2016
10. Mohamed, Shakir; Lakshminarayanan, Balaji (2016). "Learning in Implicit Generative Models". arXiv:1610.03483 (https://arxiv.org/abs/1610.03483) [stat.ML (https://arxiv.org/archive/stat.ML)].
11. Goodfellow, Ian (April 3, 2017). "NIPS 2016 Tutorial: Generative Adversarial Networks". arXiv:1701.00160 (https://arxiv.org/abs/1701.00160) [cs.LG (https://arxiv.org/archive/cs.LG)].
12. Kingma, Diederik P.; Welling, Max (May 1, 2014). "Auto-Encoding Variational Bayes". arXiv:1312.6114 (https://arxiv.org/abs/1312.6114) [stat.ML (https://arxiv.org/archive/stat.ML)].
13. Rezende, Danilo Jimenez; Mohamed, Shakir; Wierstra, Daan (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models" (https://proceedings.mlr.press/v32/rezende14.html). *Journal of Machine Learning Research*. **32** (2): 1278–1286. arXiv:1401.4082 (https://arxiv.org/abs/1401.4082).
14. Farnia, Farzan; Ozdaglar, Asuman (November 21, 2020). "Do GANs always have Nash equilibria?" (https://proceedings.mlr.press/v119/farnia20a.html). *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. PMLR. pp. 3029–3039.
15. Weng, Lilian (April 18, 2019). "From GAN to WGAN". arXiv:1904.08994 (https://arxiv.org/abs/1904.08994) [cs.LG (https://arxiv.org/archive/cs.LG)].
16. Karras, Tero; Aila, Timo; Laine, Samuli; Lehtinen, Jaakko (October 1, 2017). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". arXiv:1710.10196 (https://arxiv.org/abs/1710.10196) [cs.NE (https://arxiv.org/archive/cs.NE)].
17. Soviany, Petru; Ardei, Claudiu; Ionescu, Radu Tudor; Leordeanu, Marius (October 22, 2019). "Image Difficulty Curriculum for Generative Adversarial Networks (CuGAN)". arXiv:1910.08967 (https://arxiv.org/abs/1910.08967) [cs.LG (https://arxiv.org/archive/cs.LG)].
18. Hacohen, Guy; Weinshall, Daphna (May 24, 2019). "On The Power of Curriculum Learning in Training Deep Networks" (https://proceedings.mlr.press/v97/hacohen19a.html). *International Conference on Machine Learning*. PMLR: 2535–2544. arXiv:1904.03626 (https://arxiv.org/abs/1904.03626).
19. "r/MachineLearning - Comment by u/ian_goodfellow on "[R] [1701.07875] Wasserstein GAN" (https://www.reddit.com/r/MachineLearning/comments/5qxoaz/comment/dd4041v/?context=3). *reddit*. January 30, 2017. Retrieved July 15, 2022.
20. Lin, Zinan; et al. (December 2018). *PacGAN: the power of two samples in generative adversarial networks* (https://dl.acm.org/doi/10.5555/3326943.3327081). 32nd International Conference on Neural Information Processing Systems. pp. 1505–1514. arXiv:1712.04086 (https://arxiv.org/abs/1712.04086).
21. Mescheder, Lars; Geiger, Andreas; Nowozin, Sebastian (July 31, 2018). "Which Training Methods for GANs do actually Converge?". arXiv:1801.04406 (https://arxiv.org/abs/1801.04406) [cs.LG (https://arxiv.org/archive/cs.LG)].
22. Brock, Andrew; Donahue, Jeff; Simonyan, Karen (September 1, 2018). *Large Scale GAN Training for High Fidelity Natural Image Synthesis* (https://ui.adsabs.harvard.edu/abs/2018arXiv180911096B). International Conference on Learning Representations 2019. arXiv:1809.11096 (https://arxiv.org/abs/1809.11096).
23. Heusel, Martin; Ramsauer, Hubert; Unterthiner, Thomas; Nessler, Bernhard; Hochreiter, Sepp (2017). "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium" (https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html). *Advances in Neural Information Processing Systems*. **30**. Curran Associates, Inc. arXiv:1706.08500 (https://arxiv.org/abs/1706.08500).

24. Kingma, Diederik P.; Ba, Jimmy (January 29, 2017). "Adam: A Method for Stochastic Optimization". arXiv:1412.6980 (https://arxiv.org/abs/1412.6980) [cs.LG (https://arxiv.org/archive/cs.LG)].
25. Zhang, Richard; Isola, Phillip; Efros, Alexei A.; Shechtman, Eli; Wang, Oliver (2018). "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". pp. 586–595. arXiv:1801.03924 (https://arxiv.org/abs/1801.03924) [cs.CV (https://arxiv.org/archive/cs.CV)].
26. Borji, Ali (February 1, 2019). "Pros and cons of GAN evaluation measures" (https://www.sciencedirect.com/science/article/pii/S1077314218304272). *Computer Vision and Image Understanding*. **179**: 41–65. arXiv:1802.03446 (https://arxiv.org/abs/1802.03446). doi:10.1016/j.cviu.2018.10.009 (https://doi.org/10.1016%2Fj.cviu.2018.10.009). ISSN 1077-3142 (https://search.worldcat.org/issn/1077-3142). S2CID 3627712 (https://api.semanticscholar.org/CorpusID:3627712).
27. Hindupur, Avinash (July 15, 2022), *The GAN Zoo* (https://github.com/hindupuravinash/the-gan-zoo), retrieved July 15, 2022
28. Odena, Augustus; Olah, Christopher; Shlens, Jonathon (July 17, 2017). "Conditional Image Synthesis with Auxiliary Classifier GANs" (https://proceedings.mlr.press/v70/odena17a.html). *International Conference on Machine Learning*. PMLR: 2642–2651. arXiv:1610.09585 (https://arxiv.org/abs/1610.09585).
29. Radford, Alec; Metz, Luke; Chintala, Soumith (2016). "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". *ICLR*. S2CID 11758569 (https://api.semanticscholar.org/CorpusID:11758569).
30. Long, Jonathan; Shelhamer, Evan; Darrell, Trevor (2015). "Fully Convolutional Networks for Semantic Segmentation" (https://openaccess.thecvf.com/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html). *CVF*: 3431–3440.
31. Zhang, Han; Goodfellow, Ian; Metaxas, Dimitris; Odena, Augustus (May 24, 2019). "Self-Attention Generative Adversarial Networks" (https://proceedings.mlr.press/v97/zhang19d.html). *International Conference on Machine Learning*. PMLR: 7354–7363.
32. Larsen, Anders Boesen Lindbo; Sønderby, Søren Kaae; Larochelle, Hugo; Winther, Ole (June 11, 2016). "Autoencoding beyond pixels using a learned similarity metric" (https://proceedings.mlr.press/v48/larsen16.html). *International Conference on Machine Learning*. PMLR: 1558–1566. arXiv:1512.09300 (https://arxiv.org/abs/1512.09300).
33. Jiang, Yifan; Chang, Shiyu; Wang, Zhangyang (December 8, 2021). "TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up". arXiv:2102.07074 (https://arxiv.org/abs/2102.07074) [cs.CV (https://arxiv.org/archive/cs.CV)].
34. Grover, Aditya; Dhar, Manik; Ermon, Stefano (May 1, 2017). "Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models". arXiv:1705.08868 (https://arxiv.org/abs/1705.08868) [cs.LG (https://arxiv.org/archive/cs.LG)].
35. Arjovsky, Martin; Bottou, Léon (January 1, 2017). "Towards Principled Methods for Training Generative Adversarial Networks". arXiv:1701.04862 (https://arxiv.org/abs/1701.04862) [stat.ML (https://arxiv.org/archive/stat.ML)].
36. Goodfellow, Ian J. (December 1, 2014). "On distinguishability criteria for estimating generative models". arXiv:1412.6515 (https://arxiv.org/abs/1412.6515) [stat.ML (https://arxiv.org/archive/stat.ML)].
37. Goodfellow, Ian (August 31, 2016). "Generative Adversarial Networks (GANs), Presentation at Berkeley Artificial Intelligence Lab" (https://www.iangoodfellow.com/slides/2016-08-31-Berkeley.pdf) (PDF). Archived (https://web.archive.org/web/20220508101103/https://www.iangoodfellow.com/slides/2016-08-31-Berkeley.pdf) (PDF) from the original on May 8, 2022.
38. Lim, Jae Hyun; Ye, Jong Chul (May 8, 2017). "Geometric GAN". arXiv:1705.02894 (https://arxiv.org/abs/1705.02894) [stat.ML (https://arxiv.org/archive/stat.ML)].

39. Mao, Xudong; Li, Qing; Xie, Haoran; Lau, Raymond Y. K.; Wang, Zhen; Paul Smolley, Stephen (2017). "Least Squares Generative Adversarial Networks" (https://openaccess.thecvf.com/content_iccv_2017/html/Mao_Least_Squares_Generative_ICCV_2017_paper.html). *2017 IEEE International Conference on Computer Vision (ICCV)*. pp. 2794–2802. arXiv:1611.04076 (https://arxiv.org/abs/1611.04076). doi:10.1109/ICCV.2017.304 (https://doi.org/10.1109%2FICCV.2017.304). ISBN 978-1-5386-1032-9.
40. Makhzani, Alireza; Shlens, Jonathon; Jaitly, Navdeep; Goodfellow, Ian; Frey, Brendan (2016). "Adversarial Autoencoders". arXiv:1511.05644 (https://arxiv.org/abs/1511.05644) [cs.LG (https://arxiv.org/archive/cs.LG)].
41. Barber, David; Agakov, Felix (December 9, 2003). "The IM algorithm: a variational approach to Information Maximization" (https://dl.acm.org/doi/abs/10.5555/2981345.2981371). *Proceedings of the 16th International Conference on Neural Information Processing Systems*. NIPS'03. Cambridge, MA, USA: MIT Press: 201–208.
42. Chen, Xi; Duan, Yan; Houthooft, Rein; Schulman, John; Sutskever, Ilya; Abbeel, Pieter (2016). "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets" (https://proceedings.neurips.cc/paper/2016/hash/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Abstract.html). *Advances in Neural Information Processing Systems*. **29**. Curran Associates, Inc. arXiv:1606.03657 (https://arxiv.org/abs/1606.03657).
43. Donahue, Jeff; Krähenbühl, Philipp; Darrell, Trevor (2016). "Adversarial Feature Learning". arXiv:1605.09782 (https://arxiv.org/abs/1605.09782) [cs.LG (https://arxiv.org/archive/cs.LG)].
44. Dumoulin, Vincent; Belghazi, Ishmael; Poole, Ben; Mastropietro, Olivier; Arjovsky, Alex; Courville, Aaron (2016). "Adversarially Learned Inference". arXiv:1606.00704 (https://arxiv.org/abs/1606.00704) [stat.ML (https://arxiv.org/archive/stat.ML)].
45. Xi Chen; Yan Duan; Rein Houthooft; John Schulman; Ilya Sutskever; Pieter Abeel (2016). "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". arXiv:1606.03657 (https://arxiv.org/abs/1606.03657) [cs.LG (https://arxiv.org/archive/cs.LG)].
46. Zhirui Zhang; Shujie Liu; Mu Li; Ming Zhou; Enhong Chen (October 2018). "Bidirectional Generative Adversarial Networks for Neural Machine Translation" (https://www.aclweb.org/anthology/K18-1019.pdf) (PDF). pp. 190–199.
47. Zhu, Jun-Yan; Park, Taesung; Isola, Phillip; Efros, Alexei A. (2017). "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks". pp. 2223–2232. arXiv:1703.10593 (https://arxiv.org/abs/1703.10593) [cs.CV (https://arxiv.org/archive/cs.CV)].
48. Isola, Phillip; Zhu, Jun-Yan; Zhou, Tinghui; Efros, Alexei A. (2017). "Image-To-Image Translation With Conditional Adversarial Networks". pp. 1125–1134. arXiv:1611.07004 (https://arxiv.org/abs/1611.07004) [cs.CV (https://arxiv.org/archive/cs.CV)].
49. Brownlee, Jason (August 22, 2019). "A Gentle Introduction to BigGAN the Big Generative Adversarial Network" (https://machinelearningmastery.com/a-gentle-introduction-to-the-biggan/). *Machine Learning Mastery*. Retrieved July 15, 2022.
50. Shengyu, Zhao; Zhijian, Liu; Ji, Lin; Jun-Yan, Zhu; Song, Han (2020). "Differentiable Augmentation for Data-Efficient GAN Training" (https://proceedings.neurips.cc/paper/2020/hash/55479c55ebd1efd3ff125f1337100388-Abstract.html). *Advances in Neural Information Processing Systems*. **33**. arXiv:2006.10738 (https://arxiv.org/abs/2006.10738).
51. Tero, Karras; Miika, Aittala; Janne, Hellsten; Samuli, Laine; Jaakko, Lehtinen; Timo, Aila (2020). "Training Generative Adversarial Networks with Limited Data" (https://proceedings.neurips.cc/paper/2020/hash/8d30aa96e72440759f74bd2306c1fa3d-Abstract.html). *Advances in Neural Information Processing Systems*. **33**.

52. Shaham, Tamar Rott; Dekel, Tali; Michaeli, Tomer (October 2019). "SinGAN: Learning a Generative Model from a Single Natural Image" (https://dx.doi.org/10.1109/iccv.2019.00467). *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. pp. 4569–4579. arXiv:1905.01164 (https://arxiv.org/abs/1905.01164). doi:10.1109/iccv.2019.00467 (https://doi.org/10.1109%2Ficcv.2019.00467). ISBN 978-1-7281-4803-8. S2CID 145052179 (https://api.semanticscholar.org/CorpusID:145052179).
53. Karras, Tero; Laine, Samuli; Aila, Timo (June 2019). "A Style-Based Generator Architecture for Generative Adversarial Networks" (https://dx.doi.org/10.1109/cvpr.2019.00453). *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. pp. 4396–4405. arXiv:1812.04948 (https://arxiv.org/abs/1812.04948). doi:10.1109/cvpr.2019.00453 (https://doi.org/10.1109%2Fcvpr.2019.00453). ISBN 978-1-7281-3293-8. S2CID 54482423 (https://api.semanticscholar.org/CorpusID:54482423).
54. Karras, Tero; Laine, Samuli; Aittala, Miika; Hellsten, Janne; Lehtinen, Jaakko; Aila, Timo (June 2020). "Analyzing and Improving the Image Quality of StyleGAN" (https://dx.doi.org/10.1109/cvpr42600.2020.00813). *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. pp. 8107–8116. arXiv:1912.04958 (https://arxiv.org/abs/1912.04958). doi:10.1109/cvpr42600.2020.00813 (https://doi.org/10.1109%2Fcvpr42600.2020.00813). ISBN 978-1-7281-7168-5. S2CID 209202273 (https://api.semanticscholar.org/CorpusID:209202273).
55. Timo, Karras, Tero Aittala, Miika Laine, Samuli Härkönen, Erik Hellsten, Janne Lehtinen, Jaakko Aila (June 23, 2021). *Alias-Free Generative Adversarial Networks* (http://worldcat.org/oclc/1269560084). OCLC 1269560084 (https://search.worldcat.org/oclc/1269560084).
56. Karras, Tero; Aittala, Miika; Laine, Samuli; Härkönen, Erik; Hellsten, Janne; Lehtinen, Jaakko; Aila, Timo. "Alias-Free Generative Adversarial Networks (StyleGAN3)" (https://nvlabs.github.io/stylegan3). *nvlabs.github.io*. Retrieved July 16, 2022.
57. Li, Bonnie; François-Lavet, Vincent; Doan, Thang; Pineau, Joelle (February 14, 2021). "Domain Adversarial Reinforcement Learning". arXiv:2102.07097 (https://arxiv.org/abs/2102.07097) [cs.LG (https://arxiv.org/archive/cs.LG)].
58. Schawinski, Kevin; Zhang, Ce; Zhang, Hantian; Fowler, Lucas; Santhanam, Gokula Krishnan (February 1, 2017). "Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit" (https://doi.org/10.1093%2Fmnrasl%2Fslx008). *Monthly Notices of the Royal Astronomical Society: Letters*. **467** (1): L110–L114. arXiv:1702.00403 (https://arxiv.org/abs/1702.00403). Bibcode:2017MNRAS.467L.110S (https://ui.adsabs.harvard.edu/abs/2017MNRAS.467L.110S). doi:10.1093/mnrasl/slx008 (https://doi.org/10.1093%2Fmnrasl%2Fslx008). S2CID 7213940 (https://api.semanticscholar.org/CorpusID:7213940).
59. Kincade, Kathy. "Researchers Train a Neural Network to Study Dark Matter" (https://www.rdmag.com/news/2019/05/researchers-train-neural-network-study-dark-matter). R&D Magazine.
60. Kincade, Kathy (May 16, 2019). "CosmoGAN: Training a neural network to study dark matter" (https://phys.org/news/2019-05-cosmogan-neural-network-dark.html). *Phys.org*.
61. "Training a neural network to study dark matter" (https://www.sciencedaily.com/releases/2019/05/190516145206.htm). *Science Daily*. May 16, 2019.
62. at 06:13, Katyanna Quach 20 May 2019. "Cosmoboffins use neural networks to build dark matter maps the easy way" (https://www.theregister.co.uk/2019/05/20/neural_networks_dark_matter/). *www.theregister.co.uk*. Retrieved May 20, 2019.

63. Mustafa, Mustafa; Bard, Deborah; Bhimji, Wahid; Lukić, Zarija; Al-Rfou, Rami; Kratochvil, Jan M. (May 6, 2019). "CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks" (https://doi.org/10.1186%2Fs40668-019-0029-9). *Computational Astrophysics and Cosmology*. **6** (1): 1. arXiv:1706.02390 (https://arxiv.org/abs/1706.02390). Bibcode:2019ComAC...6....1M (https://ui.adsabs.harvard.edu/abs/2019ComAC...6....1M). doi:10.1186/s40668-019-0029-9 (https://doi.org/10.1186%2Fs40668-019-0029-9). ISSN 2197-7909 (https://search.worldcat.org/issn/2197-7909). S2CID 126034204 (https://api.semanticscholar.org/CorpusID:126034204).
64. Paganini, Michela; de Oliveira, Luke; Nachman, Benjamin (2017). "Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis". *Computing and Software for Big Science*. **1**: 4. arXiv:1701.05927 (https://arxiv.org/abs/1701.05927). Bibcode:2017arXiv170105927D (https://ui.adsabs.harvard.edu/abs/2017arXiv170105927D). doi:10.1007/s41781-017-0004-6 (https://doi.org/10.1007%2Fs41781-017-0004-6). S2CID 88514467 (https://api.semanticscholar.org/CorpusID:88514467).
65. Paganini, Michela; de Oliveira, Luke; Nachman, Benjamin (2018). "Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multi-Layer Calorimeters". *Physical Review Letters*. **120** (4): 042003. arXiv:1705.02355 (https://arxiv.org/abs/1705.02355). Bibcode:2018PhRvL.120d2003P (https://ui.adsabs.harvard.edu/abs/2018PhRvL.120d2003P). doi:10.1103/PhysRevLett.120.042003 (https://doi.org/10.1103%2FPhysRevLett.120.042003). PMID 29437460 (https://pubmed.ncbi.nlm.nih.gov/29437460). S2CID 3330974 (https://api.semanticscholar.org/CorpusID:3330974).
66. Paganini, Michela; de Oliveira, Luke; Nachman, Benjamin (2018). "CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks". *Phys. Rev. D*. **97** (1): 014021. arXiv:1712.10321 (https://arxiv.org/abs/1712.10321). Bibcode:2018PhRvD..97a4021P (https://ui.adsabs.harvard.edu/abs/2018PhRvD..97a4021P). doi:10.1103/PhysRevD.97.014021 (https://doi.org/10.1103%2FPhysRevD.97.014021). S2CID 41265836 (https://api.semanticscholar.org/CorpusID:41265836).
67. Erdmann, Martin; Glombitza, Jonas; Quast, Thorben (2019). "Precise Simulation of Electromagnetic Calorimeter Showers Using a Wasserstein Generative Adversarial Network". *Computing and Software for Big Science*. **3** (1): 4. arXiv:1807.01954 (https://arxiv.org/abs/1807.01954). Bibcode:2019CSBS....3....4E (https://ui.adsabs.harvard.edu/abs/2019CSBS....3....4E). doi:10.1007/s41781-018-0019-7 (https://doi.org/10.1007%2Fs41781-018-0019-7). S2CID 54216502 (https://api.semanticscholar.org/CorpusID:54216502).
68. Musella, Pasquale; Pandolfi, Francesco (2018). "Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks". *Computing and Software for Big Science*. **2**: 8. arXiv:1805.00850 (https://arxiv.org/abs/1805.00850). Bibcode:2018arXiv180500850M (https://ui.adsabs.harvard.edu/abs/2018arXiv180500850M). doi:10.1007/s41781-018-0015-y (https://doi.org/10.1007%2Fs41781-018-0015-y). S2CID 119474793 (https://api.semanticscholar.org/CorpusID:119474793).
69. "Deep generative models for fast shower simulation in ATLAS" (https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2018-001/). 2018.
70. SHiP, Collaboration (2019). "Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks". *Journal of Instrumentation*. **14** (11): 11028. arXiv:1909.04451 (https://arxiv.org/abs/1909.04451). Bibcode:2019JInst..14P1028A (https://ui.adsabs.harvard.edu/abs/2019JInst..14P1028A). doi:10.1088/1748-0221/14/11/P11028 (https://doi.org/10.1088%2F1748-0221%2F14%2F11%2FP11028). S2CID 202542604 (https://api.semanticscholar.org/CorpusID:202542604).

71. Nista, Ludovico; Pitsch, Heinz; Schumann, Christoph D. K.; Bode, Mathis; Grenga, Temistocle; MacArt, Jonathan F.; Attili, Antonio (June 4, 2024). "Influence of adversarial training on super-resolution turbulence reconstruction" (https://link.aps.org/doi/10.1103/PhysRevFluids.9.064601). *Physical Review Fluids*. **9** (6): 064601. arXiv:2308.16015 (https://arxiv.org/abs/2308.16015). Bibcode:2024PhRvF...9f4601N (https://ui.adsabs.harvard.edu/abs/2024PhRvF...9f4601N). doi:10.1103/PhysRevFluids.9.064601 (https://doi.org/10.1103%2FPhysRevFluids.9.064601).
72. Nista, L.; Schumann, C. D. K.; Grenga, T.; Attili, A.; Pitsch, H. (January 1, 2023). "Investigation of the generalization capability of a generative adversarial network for large eddy simulation of turbulent premixed reacting flows" (https://www.sciencedirect.com/science/article/pii/S1540748922002851). *Proceedings of the Combustion Institute*. **39** (4): 5279–5288. Bibcode:2023PComI..39.5279N (https://ui.adsabs.harvard.edu/abs/2023PComI..39.5279N). doi:10.1016/j.proci.2022.07.244 (https://doi.org/10.1016%2Fj.proci.2022.07.244). ISSN 1540-7489 (https://search.worldcat.org/issn/1540-7489).
73. Fukami, Kai; Fukagata, Koji; Taira, Kunihiko (August 1, 2020). "Assessment of supervised machine learning methods for fluid flows" (https://doi.org/10.1007/s00162-020-00518-y). *Theoretical and Computational Fluid Dynamics*. **34** (4): 497–519. arXiv:2001.09618 (https://arxiv.org/abs/2001.09618). Bibcode:2020ThCFD..34..497F (https://ui.adsabs.harvard.edu/abs/2020ThCFD..34..497F). doi:10.1007/s00162-020-00518-y (https://doi.org/10.1007%2Fs00162-020-00518-y). ISSN 1432-2250 (https://search.worldcat.org/issn/1432-2250).
74. Zhavoronkov, Alex (2019). "Deep learning enables rapid identification of potent DDR1 kinase inhibitors". *Nature Biotechnology*. **37** (9): 1038–1040. doi:10.1038/s41587-019-0224-x (https://doi.org/10.1038%2Fs41587-019-0224-x). PMID 31477924 (https://pubmed.ncbi.nlm.nih.gov/31477924). S2CID 201716327 (https://api.semanticscholar.org/CorpusID:201716327).
75. Barber, Gregory. "A Molecule Designed By AI Exhibits "Druglike" Qualities" (https://www.wired.com/story/molecule-designed-ai-exhibits-druglike-qualities/). *Wired*.
76. Moradi, M; Demirel, H (2024). "Alzheimer's disease classification using 3D conditional progressive GAN-and LDA-based data selection". *Signal, Image and Video Processing*. **18** (2): 1847–1861. doi:10.1007/s11760-023-02878-4 (https://doi.org/10.1007%2Fs11760-023-02878-4).
77. Bisneto, Tomaz Ribeiro Viana; de Carvalho Filho, Antonio Oseas; Magalhães, Deborah Maria Vieira (February 2020). "Generative adversarial network and texture features applied to automatic glaucoma detection". *Applied Soft Computing*. **90**: 106165. doi:10.1016/j.asoc.2020.106165 (https://doi.org/10.1016%2Fj.asoc.2020.106165). S2CID 214571484 (https://api.semanticscholar.org/CorpusID:214571484).
78. *Reconstruction of the Roman Emperors: Interview with Daniel Voshart* (https://www.youtube.com/watch?v=5mr6-JbuLrQ), November 16, 2020, retrieved June 3, 2022
79. msmash (February 14, 2019). " 'This Person Does Not Exist' Website Uses AI To Create Realistic Yet Horrifying Faces" (https://tech.slashdot.org/story/19/02/14/199200/this-person-does-not-exist-website-uses-ai-to-create-realistic-yet-horrifying-faces). *Slashdot*. Retrieved February 16, 2019.
80. Doyle, Michael (May 16, 2019). "John Beasley lives on Saddlehorse Drive in Evansville. Or does he?" (https://www.courierpress.com/story/news/crime/2019/05/16/john-beasley-lives-saddlehorse-drive-evansville-does-he/3700111002/). Courier and Press.
81. Targett, Ed (May 16, 2019). "California moves closer to making deepfake pornography illegal". Computer Business Review.
82. Mihalcik, Carrie (October 4, 2019). "California laws seek to crack down on deepfakes in politics and porn" (https://www.cnet.com/news/california-laws-seek-to-crack-down-on-deepfakes-in-politics-and-porn/). *cnet.com*. CNET. Retrieved October 13, 2019.
83. Knight, Will (August 7, 2018). "The Defense Department has produced the first tools for catching deepfakes" (https://www.technologyreview.com/s/611726/the-defense-department-has-produced-the-first-tools-for-catching-deepfakes/). *MIT Technology Review*.

84. Vincent, James (March 5, 2019). "A never-ending stream of AI art goes up for auction" (https://www.theverge.com/2019/3/5/18251267/ai-art-gans-mario-klingemann-auction-sothebys-technology). *The Verge*. Retrieved June 13, 2020.
85. Yu, Jiahui, et al. "Generative image inpainting with contextual attention (http://openaccess.thecvf.com/content_cvpr_2018/papers/Yu_Generative_Image_Inpainting_CVPR_2018_paper.pdf)." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
86. Wong, Ceecee (May 27, 2019). "The Rise of AI Supermodels" (https://www.cdotrends.com/story/14300/rise-ai-supermodels). *CDO Trends*.
87. Taif, K.; Ugail, H.; Mehmood, I. (2020). "Cast Shadow Generation Using Generative Adversarial Networks". *Computational Science – ICCS 2020*. Lecture Notes in Computer Science. Vol. 12141. pp. 481–495. doi:10.1007/978-3-030-50426-7_36 (https://doi.org/10.1007%2F978-3-030-50426-7_36). ISBN 978-3-030-50425-0. PMC 7302543 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7302543).
88. Wei, Jerry (July 3, 2019). "Generating Shoe Designs with Machine Learning" (https://towardsdatascience.com/generating-shoe-designs-with-deep-learning-5dde432a23b8). *Medium*. Retrieved November 6, 2019.
89. Greenemeier, Larry (June 20, 2016). "When Will Computers Have Common Sense? Ask Facebook" (https://www.scientificamerican.com/article/when-will-computers-have-common-sense-ask-facebook/). *Scientific American*. Retrieved July 31, 2016.
90. Elgammal, Ahmed; Liu, Bingchen; Elhoseiny, Mohamed; Mazzone, Marian (2017). "CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms". arXiv:1706.07068 (https://arxiv.org/abs/1706.07068) [cs.AI (https://arxiv.org/archive/cs.AI)].
91. Mazzone, Marian; Ahmed Elgammal (February 21, 2019). "Art, Creativity, and the Potential of Artificial Intelligence" (https://doi.org/10.3390%2Farts8010026). *Arts*. **8**: 26. doi:10.3390/arts8010026 (https://doi.org/10.3390%2Farts8010026).
92. Cohn, Gabe (October 25, 2018). "AI Art at Christie's Sells for $432,500" (https://www.nytimes.com/2018/10/25/arts/design/ai-art-sold-christies.html). *The New York Times*.
93. Tang, Xiaoou; Qiao, Yu; Loy, Chen Change; Dong, Chao; Liu, Yihao; Gu, Jinjin; Wu, Shixiang; Yu, Ke; Wang, Xintao (September 1, 2018). "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks". arXiv:1809.00219 (https://arxiv.org/abs/1809.00219) [cs.CV (https://arxiv.org/archive/cs.CV)].
94. Allen, Eric Van (July 8, 2020). "An Infamous Zelda Creepypasta Saga Is Using Artificial Intelligence to Craft Its Finale" (https://web.archive.org/web/20221107030028/https://www.usgamer.net/articles/ben-drowned-zelda-creepypasta-finale-artificial-intelligence). *USgamer*. Archived from the original (https://www.usgamer.net/articles/ben-drowned-zelda-creepypasta-finale-artificial-intelligence) on November 7, 2022. Retrieved November 7, 2022.
95. arcadeattack (September 28, 2020). "Arcade Attack Podcast – September (4 of 4) 2020 - Alex Hall (Ben Drowned) - Interview" (https://www.arcadeattack.co.uk/podcast-september-4-2020/). *Arcade Attack*. Retrieved November 7, 2022.
96. "Nvidia's AI recreates Pac-Man from scratch just by watching it being played" (https://www.theverge.com/2020/5/22/21266251/nvidia-ai-gamegan-recreate-pac-man-virutal-environment). *The Verge*. May 22, 2020.
97. Seung Wook Kim; Zhou, Yuhao; Philion, Jonah; Torralba, Antonio; Fidler, Sanja (2020). "Learning to Simulate Dynamic Environments with GameGAN". arXiv:2005.12126 (https://arxiv.org/abs/2005.12126) [cs.CV (https://arxiv.org/archive/cs.CV)].
98. Yu, Yi; Canales, Simon (2021). "Conditional LSTM-GAN for Melody Generation from Lyrics". *ACM Transactions on Multimedia Computing, Communications, and Applications*. **17**: 1–20. arXiv:1908.05551 (https://arxiv.org/abs/1908.05551). doi:10.1145/3424116 (https://doi.org/10.1145%2F3424116). ISSN 1551-6857 (https://search.worldcat.org/issn/1551-6857). S2CID 199668828 (https://api.semanticscholar.org/CorpusID:199668828).

99. Antipov, Grigory; Baccouche, Moez; Dugelay, Jean-Luc (2017). "Face Aging With Conditional Generative Adversarial Networks". arXiv:1702.01983 (https://arxiv.org/abs/1702.01983) [cs.CV (https://arxiv.org/archive/cs.CV)].
100. "3D Generative Adversarial Network" (http://3dgan.csail.mit.edu/). *3dgan.csail.mit.edu*.
101. Achlioptas, Panos; Diamanti, Olga; Mitliagkas, Ioannis; Guibas, Leonidas (2018). "Learning Representations and Generative Models for 3D Point Clouds". arXiv:1707.02392 (https://arxiv.org/abs/1707.02392) [cs.CV (https://arxiv.org/archive/cs.CV)].
102. Vondrick, Carl; Pirsiavash, Hamed; Torralba, Antonio (2016). "Generating Videos with Scene Dynamics" (https://www.cs.columbia.edu/~vondrick/tinyvideo/). *carlvondrick.com*. arXiv:1609.02612 (https://arxiv.org/abs/1609.02612). Bibcode:2016arXiv160902612V (https://ui.adsabs.harvard.edu/abs/2016arXiv160902612V).
103. Kang, Yuhao; Gao, Song; Roth, Rob (2019). "Transferring Multiscale Map Styles Using Generative Adversarial Networks" (https://geods.geography.wisc.edu/archives/1192). *International Journal of Cartography*. **5** (2–3): 115–141. arXiv:1905.02200 (https://arxiv.org/abs/1905.02200). Bibcode:2019IJCar...5..115K (https://ui.adsabs.harvard.edu/abs/2019IJCar...5..115K). doi:10.1080/23729333.2019.1615729 (https://doi.org/10.1080%2F23729333.2019.1615729). S2CID 146808465 (https://api.semanticscholar.org/CorpusID:146808465).
104. Wijnands, Jasper; Nice, Kerry; Thompson, Jason; Zhao, Haifeng; Stevenson, Mark (2019). "Streetscape augmentation using generative adversarial networks: Insights related to health and wellbeing". *Sustainable Cities and Society*. **49**: 101602. arXiv:1905.06464 (https://arxiv.org/abs/1905.06464). Bibcode:2019SusCS..4901602W (https://ui.adsabs.harvard.edu/abs/2019SusCS..4901602W). doi:10.1016/j.scs.2019.101602 (https://doi.org/10.1016%2Fj.scs.2019.101602). S2CID 155100183 (https://api.semanticscholar.org/CorpusID:155100183).
105. Ukkonen, Antti; Joona, Pyry; Ruotsalo, Tuukka (2020). "Generating Images Instead of Retrieving Them" (https://doi.org/10.1145/3397271.3401129). *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 1329–1338. doi:10.1145/3397271.3401129 (https://doi.org/10.1145%2F3397271.3401129). hdl:10138/328471 (https://hdl.handle.net/10138%2F328471). ISBN 9781450380164. S2CID 220730163 (https://api.semanticscholar.org/CorpusID:220730163).
106. "AI can show us the ravages of climate change" (https://www.technologyreview.com/f/613547/ai-can-show-us-the-ravages-of-climate-change/). *MIT Technology Review*. May 16, 2019.
107. Christian, Jon (May 28, 2019). "ASTOUNDING AI GUESSES WHAT YOU LOOK LIKE BASED ON YOUR VOICE" (https://futurism.com/the-byte/ai-guesses-appearance-voice). Futurism.
108. Kulp, Patrick (May 23, 2019). "Samsung's AI Lab Can Create Fake Video Footage From a Single Headshot" (https://www.adweek.com/digital/samsungs-ai-lab-can-create-fake-video-footage-from-a-single-headshot/). *AdWeek*.
109. Mohammad Navid Fekri; Ananda Mohon Ghosh; Katarina Grolinger (2020). "Generating Energy Data for Machine Learning with Recurrent Generative Adversarial Networks" (https://doi.org/10.3390%2Fen13010130). *Energies*. **13** (1): 130. doi:10.3390/en13010130 (https://doi.org/10.3390%2Fen13010130).
110. Schmidhuber, Jürgen (1991). "A possibility for implementing curiosity and boredom in model-building neural controllers". *Proc. SAB'1991*. MIT Press/Bradford Books. pp. 222–227.
111. Schmidhuber, Jürgen (2020). "Generative Adversarial Networks are Special Cases of Artificial Curiosity (1990) and also Closely Related to Predictability Minimization (1991)". *Neural Networks*. **127**: 58–66. arXiv:1906.04493 (https://arxiv.org/abs/1906.04493). doi:10.1016/j.neunet.2020.04.008 (https://doi.org/10.1016%2Fj.neunet.2020.04.008). PMID 32334341 (https://pubmed.ncbi.nlm.nih.gov/32334341). S2CID 216056336 (https://api.semanticscholar.org/CorpusID:216056336).

112. Niemitalo, Olli (February 24, 2010). "A method for training artificial neural networks to generate missing data within a variable context" (http://yehar.com:80/blog/?p=167). *Internet Archive (Wayback Machine)*. Archived (https://web.archive.org/web/20120312111546/http://yehar.com/blog/?p=167) from the original on March 12, 2012. Retrieved February 22, 2019.
113. "GANs were invented in 2010?" (https://www.reddit.com/r/MachineLearning/comments/bnqm0p/d_gans_were_invented_in_2010/). *reddit r/MachineLearning*. 2019. Retrieved May 28, 2019.
114. Li, Wei; Gauci, Melvin; Gross, Roderich (July 6, 2013). "Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference - GECCO '13". *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO 2013)*. Amsterdam, the Netherlands: ACM. pp. 223–230. doi:10.1145/2463372.2465801 (https://doi.org/10.1145%2F2463372.2465801). ISBN 9781450319638.
115. Gutmann, Michael; Hyvärinen, Aapo. "Noise-Contrastive Estimation" (http://proceedings.mlr.press/v9/gutmann10a/gutmann10a.pdf) (PDF). *International Conference on AI and Statistics*.
116. Abu-Khalaf, Murad; Lewis, Frank L.; Huang, Jie (July 1, 2008). "Neurodynamic Programming and Zero-Sum Games for Constrained Control Systems". *IEEE Transactions on Neural Networks*. **19** (7): 1243–1252. doi:10.1109/TNN.2008.2000204 (https://doi.org/10.1109%2FTNN.2008.2000204). S2CID 15680448 (https://api.semanticscholar.org/CorpusID:15680448).
117. Abu-Khalaf, Murad; Lewis, Frank L.; Huang, Jie (December 1, 2006). "Policy Iterations on the Hamilton–Jacobi–Isaacs Equation for H_∞ State Feedback Control With Input Saturation". *IEEE Transactions on Automatic Control*. doi:10.1109/TAC.2006.884959 (https://doi.org/10.1109%2FTAC.2006.884959). S2CID 1338976 (https://api.semanticscholar.org/CorpusID:1338976).
118. Sajjadi, Mehdi S. M.; Schölkopf, Bernhard; Hirsch, Michael (December 23, 2016). "EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis". arXiv:1612.07919 (https://arxiv.org/abs/1612.07919) [cs.CV (https://arxiv.org/archive/cs.CV)].
119. "This Person Does Not Exist: Neither Will Anything Eventually with AI" (https://medium.com/@alagraphy/this-person-does-not-exist-neither-will-anything-if-artificial-intelligence-keeps-learning-1a9fcba728f). March 20, 2019.
120. "ARTificial Intelligence enters the History of Art" (https://www.issuewire.com/artificial-intelligence-enters-the-history-of-art-1620667772563815). December 28, 2018.
121. Tom Février (February 17, 2019). "Le scandale de l'intelligence ARTificielle" (https://link.medium.com/MYqBrGHIKV).
122. "StyleGAN: Official TensorFlow Implementation" (https://github.com/NVlabs/stylegan). March 2, 2019 – via GitHub.
123. Paez, Danny (February 13, 2019). "This Person Does Not Exist Is the Best One-Off Website of 2019" (https://www.inverse.com/article/53280-this-person-does-not-exist-gans-website=website=Inverse). Retrieved February 16, 2019.
124. Beschizza, Rob (February 15, 2019). "This Person Does Not Exist" (https://boingboing.net/2019/02/15/this-person-does-not-exist.html). *Boing-Boing*. Retrieved February 16, 2019.
125. Horev, Rani (December 26, 2018). "Style-based GANs – Generating and Tuning Realistic Artificial Faces" (https://web.archive.org/web/20201105101517/https://www.lyrn.ai/2018/12/26/a-style-based-generator-architecture-for-generative-adversarial-networks/). *Lyrn.AI*. Archived from the original (https://www.lyrn.ai/2018/12/26/a-style-based-generator-architecture-for-generative-adversarial-networks/) on November 5, 2020. Retrieved February 16, 2019.

External links

- Knight, Will. "5 Big Predictions for Artificial Intelligence in 2017" (https://www.technologyreview.com/s/603216/5-big-predictions-for-artificial-intelligence-in-2017/). *MIT Technology Review*. Retrieved January 5, 2017.

Art portal

- Karras, Tero; Laine, Samuli; Aila, Timo (2018). "A Style-Based Generator Architecture for Generative Adversarial Networks". arXiv:1812.04948 (https://arxiv.org/abs/1812.04948) [cs.NE (https://arxiv.org/archive/cs.NE)].
- This Person Does Not Exist (https://www.thispersondoesnotexist.com/) – photorealistic images of people who do not exist, generated by StyleGAN
- This Cat Does Not Exist (https://thiscatdoesnotexist.com/) Archived (https://web.archive.org/web/20190305040119/https://thiscatdoesnotexist.com/) March 5, 2019, at the Wayback Machine – photorealistic images of cats who do not exist, generated by StyleGAN
- Wang, Zhengwei; She, Qi; Ward, Tomas E. (2019). "Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy". arXiv:1906.01529 (https://arxiv.org/abs/1906.01529) [cs.LG (https://arxiv.org/archive/cs.LG)].
