Fictitious GAN: Training GANs with Historical Models

Hao Ge, Yin Xia, Xu Chen, Randall Berry, Ying Wu [⋆]

Northwestern University, Evanston, IL, USA
{haoge2013,yinxia2012,chenx}@u.northwestern.edu
{rberry,yingwu}@northwestern.edu

Abstract. Generative adversarial networks (GANs) are powerful tools
for learning generative models. In practice, the training may suffer from
lack of convergence. GANs are commonly viewed as a two-player zero-
sum game between two neural networks. Here, we leverage this game
theoretic view to study the convergence behavior of the training process.
Inspired by the fictitious play learning process, a novel training method,
referred to as Fictitious GAN, is introduced. Fictitious GAN trains the
deep neural networks using a mixture of historical models. Specifically,
the discriminator (resp. generator) is updated according to the best-
response to the *mixture* outputs from a sequence of previously trained
generators (resp. discriminators). It is shown that Fictitious GAN can
effectively resolve some convergence issues that cannot be resolved by the
standard training approach. It is proved that asymptotically the average
of the generator outputs has the same distribution as the data samples.

1 Introduction

1.1 Generative Adversarial Networks

Generative adversarial networks (GANs) are a powerful framework for learning
generative models. They have witnessed successful applications in a wide range of
fields, including image synthesis [1,2], image super-resolution [3,4], and anomaly
detection [5]. A GAN maintains two deep neural networks: the discriminator and
the generator. The generator aims to produce samples that resemble the data
distribution, while the discriminator aims to distinguish the generated samples
and the data samples.

Mathematically, the standard GAN training aims to solve the following op-
timization problem:

$$\min_{G} \max_{D} V(G, D) = \mathsf{E}_{\boldsymbol{x} \sim p_d(\boldsymbol{x})}\{\log D(\boldsymbol{x})\} + \mathsf{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}\{\log(1 - D(G(\boldsymbol{z})))\}. \quad (1)$$

The global optimum point is reached when the generated distribution p_g, which
is the distribution of $G(\boldsymbol{z})$ given $\boldsymbol{z} \sim p_z(\boldsymbol{z})$, is equal to the data distribution.
The optimal point is reached based on the assumption that the discriminator and

---

⋆ First three authors have equal contributions.

generator are *jointly* optimized. Practical training of GANs, however, may not satisfy this assumption. In some training process, instead of ideal joint optimization, the discriminator and generator seek for best response by turns, namely the discriminator (resp. generator) is alternately updated with the generator (resp. discriminator) fixed.

Another conventional training methods are based on a gradient descent form of GAN optimization. In particular, they simultaneously take small gradient steps in both generator and discriminator parameters in each training iteration [6]. There have been some studies on the convergence behaviors of gradient-based training. The local convergence behavior has been studied in [7, 8]. The gradient-based optimization is proved to converge assuming that the discriminator and the generator is convex over the network parameters [9]. The inherent connection between gradient-based training and primal-dual subgradient methods for solving convex optimizations is built in [10].

Despite the promising practical applications, a lot of works still witness the lack of convergence behaviors in training GANs. Two common failure modes are oscillation and mode collapse, where the generator only produces a small family of samples [6, 11, 12]. One important observation in [13] is that such non convergence behaviors stem from the fact that each generator update step is a partial collapse towards a delta function, which is the best response to the objective function. This motivates the study of this paper on the dynamics of best-response training and the proposal of a novel training method to address these convergence issues.

1.2   Contributions

In this paper, we view GANs as a two-player zero-sum game and the training process as a repeated game. For the optimal solution to Eq. (1), the corresponding generated distribution and discriminator (p_g^*, D^*) is shown to be the unique Nash equilibrium in the game. Inspired by the well-established fictitious play mechanism in game theory, we propose a novel training algorithm to resolve the convergence issue and find this Nash equilibrium.

The proposed training algorithm is referred to as Fictitious GAN, where the discriminator (resp. generator) is updated based on the the mixed outputs from the sequence of historical trained generators (resp. discriminators). The previously trained models actually carry important information and can be utilized for the updates of the new model. We prove that Fictitious GAN achieves the optimal solution to Eq. (1). In particular, the discriminator outputs converge to the optimum discriminator function and the mixed output from the sequence of trained generators converges to the data distribution.

Moreover, Fictitious GAN can be regarded as a meta-algorithm that can be applied on top of existing GAN variants. Both synthetic data and real-world image datasets are used to demonstrate the improved performance due to the fictitious training mechanism.

2   Related Works

The idea of training using multiple GAN models have been considered in other works. In [14, 15], the mixed outputs of multiple generators is used to approximate the data distribution. The multiple generators with a modified loss function have been used to alleviate the mode collapse problem [16]. In [13], the generator is updated based on a sequence of unrolled discriminators. In [17], dual discriminators are used to combine the Kullback-Leibler (KL) divergence and reverse KL divergences into a unified objective function. Using an ensemble of discriminators or GAN models has shown promising performance [18, 19]. One distinguishing difference between the above-mentioned methods and our proposed method is that in our method only a *single* deep neural network is trained at each training iteration, while multiple generators (resp. discriminators) only provide inputs to a single discriminator (resp. generators) at each training stage. Moreover, the outputs from multiple networks is simply uniformly averaged and serves as input to the target training network, while other works need to train the optimal weights to average the network models. The proposed method thus has a much lower computational complexity.

The use of historical models have been proposed as a heuristic method to increase the diversity of generated samples [20], while the theoretical convergence guarantee is lacking. Game theoretic approaches have been utilized to achieve a resource-bounded Nash equilibrium in GANs [21]. Another closely related work to this paper is the recent work [22] that applies the Follow-the-Regularized-Leader (FTRL) algorithm to train GANs. In their work, the historical models are also utilized for online learning. There are at least two distinct features in our work. First, we borrow the idea of fictitious play from game theory to prove convergence to the Nash equilibrium for any GAN architectures assuming that networks have enough capacity, while [22] only proves convergence for semi-shallow architectures. Secondly, we prove that a *single* discriminator, instead of a mixture of multiple discriminators, asymptotically converges to the optimal discriminator. This provides important design guidelines for the training, where asymptotically a single discriminator needs to be maintained. [1]

3   Toy Examples

In this section, we use two toy examples to show that both the best-response approach and the gradient-based training approach may oscillate for simple minimax optimization problems.

Take the GAN framework for instance, for the best-response training approach, the discriminator and the generator are updated to the optimum point at each iteration. Mathematically, the discriminator and the generator is alter-

---

[1] Due to space constraints, all the proofs in the paper are omitted and can be found in the Supplementary materials.

nately updated according to the following rules:

$$\max_{D} \mathsf{E}_{\boldsymbol{x} \sim p_d(\boldsymbol{x})}\{\log D(\boldsymbol{x})\} + \mathsf{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}\{\log(1 - D(G(\boldsymbol{z}))\} \tag{2}$$

$$\min_{G} \mathsf{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}\{\log(1 - D(G(\boldsymbol{z})))\} \tag{3}$$

Example 1. Let the data follow the Bernoulli distribution $p_d \sim$ Bernoulli (a), where $0 < a < 1$. Suppose the initial generated distribution $p_g \sim$ Bernoulli (b), where $b \neq a$. We show that in the best-response training process, the generated distribution oscillates between $p_g \sim$ Bernoulli (1) and $p_g \sim$ Bernoulli (0).

We show the oscillation phenomenon in training using best-response training approach. To minimize (3), it is equivalent to find p_g such that $\mathsf{E}_{\boldsymbol{x} \sim p_g(\boldsymbol{x})}\{\log(1 - D(\boldsymbol{x}))\}$ is minimized. At each iteration, the output distribution of the updated generator would concentrate all the probability mass at $x = 0$ if $D(0) > D(1)$, or at $x = 1$ if $D(0) < D(1)$. Suppose $p_g(x) = 1\{x = 0\}$, where $1\{\cdot\}$ is the indicator function, then by solving (2), the discriminator at the next iteration is updated as

$$D(x) = \frac{p_d(x)}{p_d(x) + p_g(x)}, \tag{4}$$

which yields $D(1) = 1$ and $D(0) < D(1)$. Therefore, the generated distribution at the next iteration becomes $p_g(x) = 1\{x = 1\}$. The oscillation between $p_g \sim$ Bernoulli (1) and $p_g \sim$ Bernoulli (0) continues by induction. A similar phenomenon can be observed for Wasserstein GAN.

The first toy example implies that the oscillation behavior is a fundamental problem to the iterative best-response training. In practical training of GANs, instead of finding the best response, the discriminator and generator are updated based on gradient descent towards the best-response of the objective function. However, the next example adapted from [23] demonstrates the failure of convergence in a simple minimax problem using a gradient-based method.

Example 2. Consider the following minimax problem:

$$\min_{-10 \leq y \leq 10} \max_{-10 \leq x \leq 10} xy. \tag{5}$$

Consider the gradient based training approach with step size $\triangle$. The update rule of x and y is:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & \triangle \\ -\triangle & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}. \tag{6}$$

By using the knowledge of eigenvalues and eigenvectors, we can obtain

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} -c_1^n c_2 \sin(n\theta + \beta) \\ c_1^n c_2 \cos(n\theta + \beta) \end{bmatrix}, \tag{7}$$

where $c_1 = \sqrt{1 + \triangle^2} > 1$ and c_2, θ, β are constants depending on the initial (x_0, y_0). As $n \to \infty$, since $c_1 > 1$, the process will not converge.
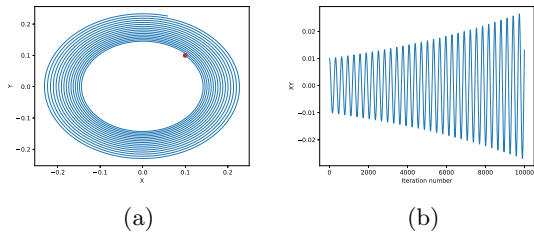
Fig. 1: Performance of gradient method with fixed step size for Example 2. (a) illustrates the choices of x and y as iteration processes, the red point $(0.1, 0.1)$ is the initial value. (b) illustrates the value of xy as a function of iteration numbers.

Figure 1 shows the performance of gradient based approach, the initial value $(x_0, y_0) = (0.1, 0.1)$ and step size is 0.01. It can be seen that both players' actions do not converge. This toy example shows that even the gradient based approach with arbitrarily small step size may not converge.

We will revisit the convergence behavior in the context of game theory. A well-established learning mechanism in game theory naturally leads to a training algorithm that resolves the non-convergence issues of these two toy examples.

4 Nash Equilibrium in Zero-Sum Games

In this section, we introduce the two-player zero-sum game and describe the learning mechanism of fictitious play, which provably achieves a Nash equilibrium of the game. We will show that the minimax optimization of GAN can be formulated as a two-player zero-sum game, where the optimal solution corresponds to the unique Nash equilibrium in the game. In the next section we will propose a training algorithm which simulates the fictitious play mechanism and provably achieves the optimal solution.

4.1 Zero-Sum Games

We start with some definitions in game theory. A game consists of a set of n players, who are rational and take actions to maximize their own utilities. Each player i chooses a pure strategy s_i from the strategy space $S_i = \{s_{i,0}, \cdots, s_{i,m-1}\}$. Here player i has m strategies in her strategy space. A utility function $u_i(s_i, s_{-i})$, which is defined over all players' strategies, indicates the outcome for player i, where the subscript $-i$ stands for all players excluding player i. There are two kinds of strategies, pure and mixed strategy. A pure strategy provides a specific action that a player will follow for any possible situation in a game, while a mixed strategy $\mu_i = (p_i(s_{i,0}), \cdots, p_i(s_{i,m-1}))$ for player i is a probability distribution over the m pure strategies in her strategy space with $\sum_j p_i(s_{i,j}) = 1$. The set of possible mixed strategies available to player i is denoted by ΔS_i. The expected

utility of mixed strategy (μ_i, μ_{-i}) for player i is

$$\mathsf{E}\{u_i(\mu_i, \mu_{-i})\} = \sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} u_i(s_i, s_{-i}) p_i(s_i) p_{-i}(s_{-i}). \tag{8}$$

For ease of notation, we write $u_i(\mu_i, \mu_{-i})$ as $\mathsf{E}\{u_i(\mu_i, \mu_{-i})\}$ in the following. Note that a pure strategy can be expressed as a mixed strategy that places probability 1 on a single pure strategy and probability 0 on the others. A game is referred to as a finite game or a continuous game, if the strategy space is finite or nonempty and compact, respectively. In a continuous game, the mixed strategy indicates a probability density function (pdf) over the strategy space.

Definition 1. *For player i, a strategy μ_i^* is called a best response to others' strategy μ_{-i} if $u_i(\mu_i^*, \mu_{-i}) \geq u_i(\mu_i, \mu_{-i})$ for any $\mu_i \in \Delta S_i$.*

Definition 2. *A set of mixed strategies $\mu^* = (\mu_1^*, \mu_2^*, \cdots, \mu_n^*)$ is a Nash equilibrium if, for every player i, μ_i^* is a best response to the strategies μ_{-i}^* played by the other players in this game.*

Definition 3. *A zero-sum game is one in which each player's gain or loss is exactly balanced by the others' loss or gain and the sum of the players' payoff is always zero.*

Now we focus on a continuous two-player zero-sum game. In such a game, given the strategy pair (μ_1, μ_2), player 1 has a utility of $u(\mu_1, \mu_2)$, while player 2 has a utility of $-u(\mu_1, \mu_2)$. In the framework of GAN, the training objective (1) can be regarded as a two-player zero-sum game, where the generator and discriminator are two players with utility functions $-V(G, D)$ and $V(G, D)$, respectively. Both of them aim to maximize their utility and the sum of their utilities is zero.

Knowing the opponent is always seeking to maximize its utility, Player 1 and 2 choose strategies according to

$$\mu_1^* = \operatorname*{argmax}_{\mu_1 \in \Delta S_1} \min_{\mu_2 \in \Delta S_2} u(\mu_1, \mu_2) \tag{9}$$

$$\mu_2^* = \operatorname*{argmin}_{\mu_2 \in \Delta S_2} \max_{\mu_1 \in \Delta S_1} u(\mu_1, \mu_2). \tag{10}$$

Define $\underline{v} = \max_{\mu_1 \in \Delta S_1} \min_{\mu_2 \in \Delta S_2} u(\mu_1, \mu_2)$ and $\bar{v} = \min_{\mu_2 \in \Delta S_2} \max_{\mu_1 \in \Delta S_1} u(\mu_1, \mu_2)$ as the lower value and upper value of the game, respectively. Generally, $\underline{v} \leq \bar{v}$. Sion [24] showed that these two values coincide under some regularity conditions:

Theorem 1 (Sion's Minimax Theorem [24]). *Let X and Y be convex, compact spaces, and $f: X \times Y \to \mathbb{R}$. If for any $x \in X$, $f(x, \cdot)$ is upper semi-continuous and quasi-concave on Y and for any $y \in Y$, $f(\cdot, y)$ is lower semi-continuous and quasi-convex on X, then $\inf_{x \in X} \sup_{y \in Y} f(x, y) = \sup_{y \in Y} \inf_{x \in X} f(x, y)$.*

Hence, in a zero-sum game, if the utility function $u(\mu_1, \mu_2)$ satisfies the conditions in Theorem 1, then $\underline{v} = \bar{v}$. We refer to $v = \underline{v} = \bar{v}$ as the *value of the game*. We further show that a Nash equilibrium of the zero-sum game achieves the value of the game.

Corollary 1. *In a two-player zero-sum game with the utility function satisfying the conditions in Theorem 1, if a strategy (μ_1^*, μ_2^*) is a Nash equilibrium, then $u(\mu_1^*, \mu_2^*) = v$.*

Corollary 1 implies that if we have an algorithm that achieves a Nash equilibrium of a zero-sum game, we may utilize this algorithm to optimally train a GAN. We next describe a learning mechanism to achieve a Nash equilibrium.

4.2   Fictitious Play

Suppose the zero-sum game is played repeatedly between two rational players, then each player may try to infer her opponent's strategy. Let $s_i^n \in S_i$ denote the action taken by player i at time n. At time n, given the previous actions $\{s_2^0, s_2^1, \cdots, s_2^{n-1}\}$ chosen by player 2, one good hypothesis is that player 2 is using stationary mixed strategies and chooses strategy s_2^t, $0 \leq t \leq n-1$, with probability $\frac{1}{n}$. Here we use the empirical frequency to approximate the probability in mixed strategies. Under this hypothesis, the best response for player 1 at time n is to choose the strategy μ_1^* satisfying:

$$\mu_1^* = \underset{\mu_1 \in \Delta S_1}{\operatorname{argmax}} \, u(\mu_1, \mu_2^n), \tag{11}$$

where μ_2^n is the empirical distribution of player 2's historical actions. Similarly, player 2 can choose the best response assuming player 1 is choosing its strategy according to the empirical distribution of the historical actions.

Notice that the expected utility is a linear combination of utilities under different pure strategies, hence for any hypothesis μ_{-i}^n, player i can find a pure strategy s_i^n as a best response. Therefore, we further assume each player plays the best pure response at each round. In game theory this learning rule is called *fictitious play*, proposed by Brown [25].

Danskin [26] showed that for any continuous zero-sum games with any initial strategy profile, fictitious play will converge. This important result is summarized in the following theorem.

Theorem 2. *Let $u(s_1, s_2)$ be a continuous function defined on the direct product of two compact sets S_1 and S_2. The pure strategy sequences $\{s_1^n\}$ and $\{s_2^n\}$ are defined as follows: s_1^0 and s_2^0 are arbitrary, and*

$$s_1^n \in \underset{s_1 \in S_1}{\operatorname{argmax}} \frac{1}{n} \sum_{k=0}^{n-1} u(s_1, s_2^k), \quad s_2^n \in \underset{s_2 \in S_2}{\operatorname{argmin}} \frac{1}{n} \sum_{k=0}^{n-1} u(s_1^k, s_2), \tag{12}$$

then

$$\lim_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} u(s_1^n, s_2^k) = \lim_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} u(s_1^k, s_2^n) = v, \tag{13}$$

where v is the value of the game.

<div style="text-align:center">(a)   (b)   (c)</div>
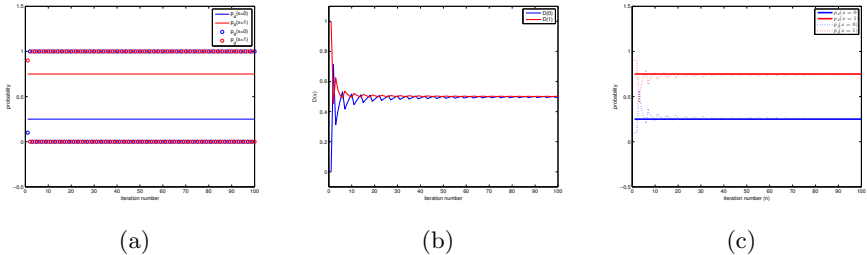
Fig. 2: Performance of best-response training for Example 1. (a) is Bernoulli distribution of p_g assuming best-response updates. (b) illustrates $D(x)$ in Fictitious GAN assuming best response at each training iteration. (c) illustrates the average of $p_g(x)$ in Fictitious GAN assuming best response at each training iteration.

4.3   Effectiveness of Fictitious Play

In this section, we show that fictitious play enables the convergence of learning to the optimal solution for the two counter-examples in Section 3.

   Example 1: Fig. 2 shows the performance of the best-response approach, where the data follows a Bernoulli distribution $p_d \sim$ Bernoulli (0.25), the initialization is $D(x) = x$ for $x \in [0, 1]$ and the initial generated distribution $p_g \sim$ Bernoulli (0.1). It can be seen that the generated distribution based on best responses oscillates between $p_g(x = 0) = 1$ and $p_g(x = 1) = 1$.

   Assuming best response at each iteration n, under fictitious play, the discriminator is updated according to $D_n = \arg\max_D \frac{1}{n} \sum_{w=0}^{n-1} V(p_{g,w}, D)$ and the generated distribution is updated according to $p_{g,n} = \arg\max_{p_g} \frac{1}{n} \sum_{w=0}^{n-1} V(p_g, D_w)$. Fig 2 shows the change of D_n and the empirical mean of the generated distributions $\bar{p}_{g,n} = \frac{1}{n} \sum_{w=0}^{n-1} p_{g,w}$ as training proceeds. Although the best-response generated distribution at each iteration oscillates as in Fig. 2a, the learning mechanism of fictitious play makes the empirical mean $\bar{p}_{g,n}$ converge to the data distribution.

   Example 2: At each iteration n, player 1 chooses $x = \arg\max_x \frac{1}{n} \sum_{i=0}^{n-1} x y_i$, which is equal to $10 * \text{sign}(\sum_{i=0}^{n-1} y_i)$. Similarly, player 2 chooses y according to $y = -10 * \text{sign}(\sum_{i=0}^{n-1} x_i)$. Hence regardless of what the initial condition is, both players will only choose 10 or -10 at each iteration. Consequently, as iteration goes to infinity, the empirical mixed strategy only proposes density on 10 and -10. It is proved in the Supplementary material that the mixed strategy (σ_1^*, σ_2^*) that both players choose 10 and -10 with probability $\frac{1}{2}$ is a Nash equilibrium for this game. Fig 3 shows that under fictitious play, both players' empirical mixed strategy converges to the Nash equilibrium and the expected utility for each player converges to 0.

   One important observation is fictitious play can provide the Nash equilibrium if the equilibrium is unique in the game. However, if there exist multiple Nash equilibriums, different initialization may yield different solutions. In the above
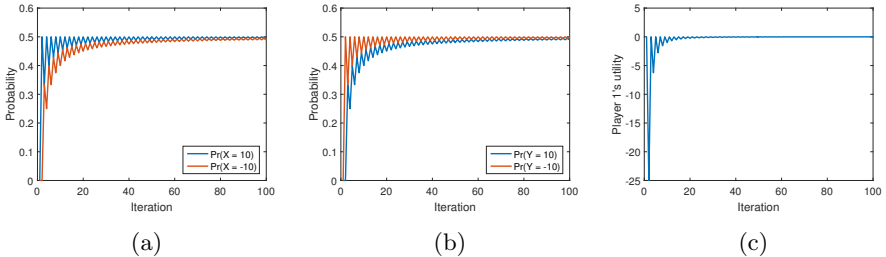
Fig. 3: (a) and (b) illustrate the empirical distribution of x and y at 10 and -10, respectively. (c) illustrates the expected utility for player 1 under fictitious play.

example, it is easy to check $(0,0)$ is also a Nash equilibrium, which means both players always choose 0, but fictitious play can lead to this solution only when the initialization is $(0,0)$. The good thing we show in the next section is, due to the special structure of GAN (the utility function is linear over generated distribution), fictitious play can help us find the desired Nash equilibrium.

5   Fictitious GAN

5.1   Algorithm Description

As discussed in the last section, the competition between the generator and discriminator in GAN can be modeled as a two-player zero-sum game. The following theorem proved in the supplementary material shows that the optimal solution of (1) is actually a unique Nash equilibrium in the game.

Theorem 3. *Consider (1) as a two-player zero-sum game. The optimal solution of (1) with $p_g^* = p_d$ and $D^*(\boldsymbol{x}) = 1/2$ is a unique Nash equilibrium in this game. The value of the game is $-\log 4$.*

By relating GAN with the two-player zero-sum game, we can design a training algorithm to simulate the fictitious play such that the training outcome converges to the Nash equilibrium

Fictitious GAN, as described in Algorithm 1, adapts the fictitious play learning mechanism to train GANs. We use two queues $\mathcal{D}$ and $\mathcal{G}$ to store the historically trained models of the discriminator and the generator, respectively. At each iteration, the discriminator (resp. generator) is updated according to the best response to $V(G, D)$ assuming that the generator (resp. discriminator) chooses a historical strategy uniformly at random. Mathematically, the discriminator and generator are updated according to (14) and (15), where the outputs due to the generator and the discriminator is mixed uniformly at random from the previously trained models. Note the the back-propagation is still performed on a single neural network at each training step. Different from standard training approaches, we perform k_0 gradient descent updates when training the discriminator and the generator in order to achieve the best response. In practical

learning, queues $\mathcal{D}$ and $\mathcal{G}$ are maintained with a fixed size. The oldest model is discarded if the queue is full when we update the discriminator or the generator.

Algorithm 1 Fictitious GAN training algorithm.

Initialization: Set $\mathcal{D}$ and $\mathcal{G}$ as the queues to store the historical models of the discriminators and the generators, respectively.
while the stopping criterion is not met **do**
    for $k = 1, \cdots, k_0$ **do**
        Sample data via minibatch $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$.
        Sample noise via minibatch $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_m$.
        Update the discriminator via gradient ascent:

$$\nabla_{\boldsymbol{\theta}_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log(D(\boldsymbol{x}_i)) + \frac{1}{|\mathcal{G}|} \sum_{G_w \in \mathcal{G}} \log(1 - D(G_w(\boldsymbol{z}_i))) \right]. \tag{14}$$

    end for
    for $k = 1, \cdots, k_0$ **do**
        Sample noise via minibatch $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_m$.
        Update the generator via gradient descent:

$$\nabla_{\boldsymbol{\theta}_g} \left[\frac{1}{m|\mathcal{G}|} \sum_{i=1}^{m} \sum_{D_w \in \mathcal{D}} \log(1 - D_w(G(\boldsymbol{z}_i))) \right]. \tag{15}$$

    end for
    Insert the updated discriminator and the updated generator into $\mathcal{D}$ and $\mathcal{G}$, respectively.
end while

The following theorem provides the theoretical convergence guarantee for Fictitious GAN. It shows that assuming best response at each update in Fictitious GAN, the distribution of the mixture outputs from the generators converge to the data distribution. The intuition of the proof is that fictitious play achieves a Nash equilibrium in two-player zero-sum games. Since the optimal solution of GAN is a unique equilibrium in the game, fictitious GAN achieves the optimal solution.

Theorem 4. *Suppose the discriminator and the generator are updated according to the best-response strategy at each iteration in Fictitious GAN, then*

$$\lim_{n \to \infty} \frac{1}{n} \sum_{w=0}^{n-1} p_{g,w}(\boldsymbol{x}) = p_d(\boldsymbol{x}), \tag{16}$$

$$\lim_{n \to \infty} D_n(\boldsymbol{x}) = \frac{1}{2}, \tag{17}$$

where $D_w(\boldsymbol{x})$ is the output from the w-th trained discriminator model and $p_{g,w}$ is the generated distribution due to the w-th trained generator.

5.2   Fictitious GAN as a Meta-Algorithm

One advantage of Fictitious GAN is that it can be applied on top of existing GANs. Consider the following minimax problem:

$$\min_G \max_D V(G, D) = \mathsf{E}_{\boldsymbol{x} \sim p_d(\boldsymbol{x})}\{f_0(D(\boldsymbol{x}))\} + \mathsf{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}\{f_1(D(G(\boldsymbol{z})))\}, \qquad (18)$$

where $f_0(\cdot)$ and $f_1(\cdot)$ are some quasi-concave functions depending on the GAN variants. Table 1 shows the family of f-GAN [9, 10] and Wasserstein GAN.

We can model these GAN variants as two-player zero-sum games and the training algorithms for these variants of GAN follow by simply changing $f_0(\cdot)$ and $f_1(\cdot)$ in the updating rule accordingly in Algorithm 1. Following the proof in Theorem 4, we can show that the time average of generated distributions will converge to the data distribution and the discriminator will converge to D^* as shown in Table 1.

Table 1: Variants of GANs under the zero-sum game framework.

Divergence metric	$f_0(D)$	$f_1(D)$	D^*	value of the game
Kullback-Leibler	$\log(D)$	$1 - D$	1	0
Reverse KL	$-D$	$\log D$	1	-1
Pearson χ^2	D	$-\frac{1}{4}D^2 - D$	0	0
Squared Hellinger χ^2	$1 - D$	$1 - 1/D$	1	0
Jensen-Shannon	$\log(D)$	$\log(1 - D)$	$\frac{1}{2}$	$-\log 4$
WGAN	D	$-D$	0	0

6   Experiments

Our Fictitious GAN is a meta-algorithm that can be applied on top of existing GANs. To demonstrate the merit of using Fictitious GAN, we apply our meta-algorithm on DCGAN [27] and its extension conditional DCGAN. Conditional DCGAN allows DCGAN to use external label information to generate images of some particular classes. We evaluate the performance on a synthetic dataset and three widely adopted real-world image datasets. Our experiment results show that Fictitious GAN could improve visual quality of both DCGAN and conditional GAN models.

Image dataset. (1) **MNIST**: contains 60,000 labeled images of 28×28 grayscale digits. (2) **CIFAR-10**: consists of colored natural scene images sized at 32×32 pixels. There are 50,000 training images and 10,000 test images in 10 classes. (3) **CelebA**: is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations.

Parameter Settings. We used Tensorflow for our implementation. Due to GPU memory limitation, we limit number of historical models to 5 in real-world image dataset experiments. More architecture details are included in supplementary material.

6.1   2D Mixture of Gaussian

Fig. 4 shows the performance of Fictitious GAN for a mixture of 8 Gaussain data on a circle in 2 dimensional space. We use the network structure in [13] to evaluate the performance of our proposed method. The data is sampled from a mixture of 8 Gaussians uniformly located on a circle of radius 1.0. Each has standard deviation of 0.02. The input noise samples are a vector of 256 independent and identically distributed (i.i.d.) Gaussian variables with mean zero and unit standard deviation.

While the original GANs experience mode collapse [13, 17], Fictitious GAN is able to generate samples over all 8 modes, even with a single discriminator asymptotically.

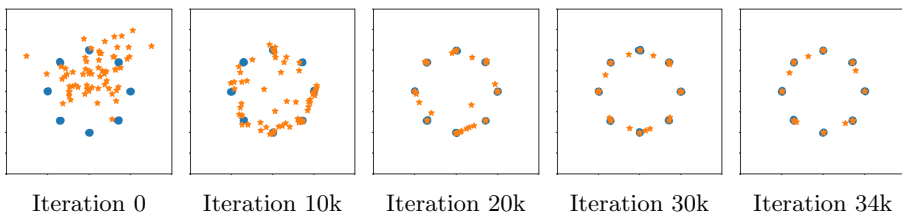|				
Iteration 0	Iteration 10k	Iteration 20k	Iteration 30k	Iteration 34k

Fig. 4: Performance of Fictitious GAN on 2D mixture of Gaussian data. The data samples are marked in blue and the generated samples are marked in orange.

6.2   Qualitative Results for Image Generation

We show visual quality of samples generated by DCGAN and conditional DC-GAN, trained by proposed Fictitious GAN. In Fig. 5 first row corresponds to generated samples. We apply train DCGAN on CelebA dataset, and train conditional DCGAN on MNIST and CIFAR-10. Each image in the first row corresponds to the image in the same grid position in second row of Fig. 5 . The second row shows the nearest neighbor in training dataset computed by Euclidean distance. The samples are randomly drawn without cherry picking, they are representative of model output distribution.

In CelebA, we can generate face images with various genders, skin colors and hairstyles. In MNIST dataset, all generated digits have almost visually identical samples. Also, digit images have diverse visual shapes and fonts. CIFAR-10 dataset is more challenging, images of each object have large visual appearance variance. We observe some visual and label consistency in generated images and the nearest neigbhors, especially in the categories of airplane, horse and ship. Note that though we theoratical proved that Fictitious GAN could improve robustness of training in best response strategy, the visual quality still depends on the baseline GAN architecture and loss design, which in our case is conditional DCGAN.

Fig. 5: Generated images in CelebA, MNIST and CIFAR-10. Top row samples are generated, bottom row images are corresponding nearest neighbors in training dataset.

6.3   Quantitative Results

In this section, we quantitatively show that DCGAN models trained by our Fictitious GAN could gain improvement over traditional training methods. Also, we may have a better performance by applying Fictitious gan on other existing gan models. The results of comparison methods are directly copied as reported.

Metric. The visual quality of generated images is measured by the widely used Inception score metric [20]. It measures visual objectiveness of generated image and correlates well with human scoring of the realism of generated images. Following evaluation scheme of [20] setup, we generate 50,000 images from our model to compute the score.

Table 2: Inception Score on CIFAR-10.

Method	Score
Fictitious cDCGAN*	**7.27 ± 0.10**
DCGAN* [28](best variant)	7.16 ± 0.10
MIX+WGAN* [14]	4.04 ± 0.07
Fictitious DCGAN	6.63 ± 0.06
DCGAN [28]	**6.16 ± 0.07**
GMAN [18]	6.00 ± 0.19
WGAN [14]	3.82 ± 0.06
Real data	11.24 ± 0.12

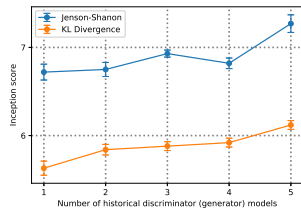Note: * denotes models that use labels for training.

Fig. 6: We show that Fictitious-GAN can improve Inception score as a meta-algorithm with larger number of historical models , We select 2 divergence metrics from Table 1: Jenson-Shanon and KL divergence.

As shown in Table 2, Our method outperforms recent state-of-the-art methods. Specifically, we improve baseline DCGAN from 6.16 to 6.63; and conditional DCGAN model from 7.16 to 7.27. It sheds light on the advantage of training with the proposed learning algorithm. Note that in order to highlight the performance improvement gained from fictitious GAN, the inception score of reproduced DC-GAN model is 6.72, obtained without using tricks as [20]. Also, we did not use any regularization terms such as conditional loss and entropy loss to train DC-GAN, as in [28]. We expect higher inception score when more training tricks are used in addition to Fictitious GAN.

6.4   Ablation studies

One hyperparameter that affects the performance of Fictitious GAN is the number of historical generator (discriminator) models. We evaluate the performance of Fictitious GAN with different number of historical models, and report the inception scores on the 150-th epoch in CIFAR-10 dataset in Fig. 6. We keep the number of historical discriminators the same as the number of historical generators. We observe a trend of performance boost with an increasing number of historical models in 2 baseline GAN models. The mean of inception score slightly drops for Jenson-Shannon divergence metric when the copy number is 4, due to random initialization and random noise generation in training.

7   Conclusion

In this paper, we relate the minimax game of GAN to the two-player zero-sum game. This relation enables us to leverage the mechanism of fictitious play to design a novel training algorithm, referred to as fictitious GAN. In the training algorithm, the discriminator (resp. generator) is alternately updated as best response to the mixed output of the stale generator models (resp. discriminator). This novel training algorithm can resolve the oscillation behavior due to the pure best response strategy and the inconvergence issue of gradient based training in some cases. Real world image datasets show that applying fictitious GAN on top of the existing DCGAN models yields a performance gain of up to 8%.

References

1. Arora, S., Ge, R., Liang, Y., Ma, T., Zhang, Y.: Generalization and equilibrium in generative adversarial nets (gans). In: International Conference on Machine Learning. pp. 224–232 (2017)
2. Brown, G.W.: Iterative solution of games by fictitious play. Activity analysis of production and allocation **13**(1), 374–376 (1951)
3. Che, T., Li, Y., Jacob, A.P., Bengio, Y., Li, W.: Mode regularized generative adversarial networks. In: Proc. Int. Conf. Learn. Representations (2017)
4. Chen, X., Wang, J., Ge, H.: Training generative adversarial networks via primal-dual subgradient methods: A lagrangian perspective on gan. In: Proc. Int. Conf. Learn. Representations (2018)
5. Danskin, J.M.: Fictitious play for continuous games. Naval Research Logistics (NRL) **1**(4), 313–320 (1954)
6. Durugkar, I., Gemp, I., Mahadevan, S.: Generative multi-adversarial networks. arXiv preprint arXiv:1611.01673 (2016)
7. Ghosh, A., Kulharia, V., Namboodiri, V., Torr, P.H., Dokania, P.K.: Multi-agent diverse generative adversarial networks. arXiv preprint arXiv:1704.02906 (2017)
8. Goodfellow, I.: Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160 (2016)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
10. Grnarova, P., Levy, K.Y., Lucchi, A., Hofmann, T., Krause, A.: An online learning approach to generative adversarial networks. In: Proc. Int. Conf. Learn. Representations (2018)
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems. pp. 6626–6637 (2017)
12. Hoang, Q., Nguyen, T.D., Le, T., Phung, D.: Multi-generator gernerative adversarial nets. arXiv preprint arXiv:1708.02556 (2017)
13. Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S.: Stacked generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, p. 4 (2017)
14. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision. pp. 694–711. Springer (2016)
15. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 105–114 (2017)
16. Li, J., Madry, A., Peebles, J., Schmidt, L.: Towards understanding the dynamics of generative adversarial networks. arXiv preprint arXiv:1706.09884 (2017)
17. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. In: Proc. Int. Conf. Learn. Representations (2017)
18. Nagarajan, V., Kolter, J.Z.: Gradient descent gan optimization is locally stable. In: Advances in Neural Information Processing Systems. pp. 5585–5595 (2017)
19. Nguyen, T., Le, T., Vu, H., Phung, D.: Dual discriminator generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2667–2677 (2017)

20. Nowozin, S., Cseke, B., Tomioka, R.: f-GAN: Training generative neural samplers using variational divergence minimization. In: Advances in Neural Information Processing Systems. pp. 271–279 (2016)
21. Oliehoek, F.A., Savani, R., Gallego, J., van der Pol, E., Gross, R.: Beyond local nash equilibria for adversarial networks. arXiv preprint arXiv:1806.07268 (2018)
22. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proc. Int. Conf. Learn. Representations (2016)
23. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: International Conference on Machine Learning. pp. 1060–1069 (2016)
24. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems. pp. 2234–2242 (2016)
25. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 3, p. 6 (2017)
26. Sion, M.: On general minimax theorems. Pacific Journal of mathematics $\mathbf{8}$(1), 171–176 (1958)
27. Tolstikhin, I.O., Gelly, S., Bousquet, O., Simon-Gabriel, C.J., Schölkopf, B.: Adagan: Boosting generative models. In: Advances in Neural Information Processing Systems. pp. 5424–5433 (2017)
28. Zhai, S., Cheng, Y., Lu, W., Zhang, Z.: Deep structured energy based models for anomaly detection. In: International Conference on Machine Learning. pp. 1100–1109 (2016)