Distributionally Robust Games: Wasserstein Metric

Jian Gao *, Hamidou Tembine †
Department of Computer Science and Engineering, New York University
Email: * jg4631@nyu.edu, † tembine@nyu.edu

***Abstract*—Deep generative models are powerful but difficult to train due to its instability, saturation problem and high dimensional data distribution. This paper introduces a game theory framework with Wasserstein metric to train generative models, in which the unknown data distribution is learned by dynamically optimizing the worst-case payoff. In the game, two types of players work on opposite objectives to solve a minimax problem. The defenders explore the Wasserstein neighborhood of real data to generate a set of hard samples which have the maximum distance from the model distribution. The attackers update the model to fit for the hard set so as to minimize the discrepancy between model and data distributions. Instead of Kullback-Leibler divergence, we use Wasserstein distance to measure the similarity between distributions. The Wasserstein metric is a true distance with better topology in the parameter space, which improves the stability of training. We provide practical algorithms to train deep generative models, in which an encoder network is designed to learn the feature vector of the high dimensional data. The algorithm is tested on CelebA human face dataset and compared with the state-of-the-art generative models. Performance evaluation shows the training process is stable and converges fast. Our model can produce visual pleasing images which are closer to the real distribution in terms of Wasserstein distance.**

I. Introduction

Deep neural networks have achieved great success in supervised learning. Apart from recognition and classification, people may wish to learn directly from the nature without prior knowledge, i.e., learn the distribution of a set of unlabelled data. In generative learning, new samples produced by the learned model should be indistinguishable from the original data and have enough diversity. Generative models are powerful tools for many tasks such as signal denoising, image inpainting, data synthesis and semi-supervised learning.

However, learning deep generative model is hard and time consuming. The high dimensional training data and extremely complex objective structures lead to many problems in optimization, such as algorithm instability, saturation, and mode collapse. Moreover, the model should have strong generalization power to produce diversiform new examples instead of just memorizing the training set.

An early work of generative learning dates back to the 80s, when restricted Boltzmann machines (RBMs) [1] were proposed to learn probability distributions based on binary input vectors. Just as multi-layer perceptrons are universal function approximators, deep Boltzmann machines are undirected graphical models that can approximate any probability function over discrete variables [2]. To deal with time series data, Conditional Restricted Boltzmann Machines (CRBMs) [3] were proposed, where previous time instances are treated as additional inputs to model short-time temporal dependencies. Later in 2006, Hinton [4] introduced the famous deep belief networks (DBNs). They are hybrid graphic models with both directed and undirected links between latent variables. RBMs and DBNs have historic significance in deep learning, though they are rarely used in the recent years.

Variational Auto-Encoders (VAEs) [5] and Generative Adversarial Networks (GANs) [6] are the most popular deep generative models today. VAE involves an inference network to explicitly formulate the posterior distributions of laten variables, and maximizes a lower bound on the likelihood. It offers a nice theory, but practically, VAE samples suffer from blurry due to the noise term in their model density functions [7]. GAN alternatively trains a generative network and a discriminative network with opposite objectives. The loss functions are defined based on information metric such as Kullback-Leibler and Jensen-Shannon divergence. Despite its great success in producing visual pleasing samples, the training process is unstable and has the risk of 'mode collapse'. It needs to carefully keep the balance between updating those two networks to avoid gradient saturation [8]. To deal with this problem, Martin Arjovsky switches to the Earth Mover (EM) distance and proposed WGAN [9]. Their model involves another neural network to estimate EM, whose weights are clipped to enforce the Lipschitz constraint.

Different from VAE and GAN models, Generative Moment Matching Networks (GMMN) [10] do not need a second network. The generative model is trained by minimizing the maximum mean discrepancy (MMD), where the objective is evaluated by matching all moments of the statistics between real and fake sample distributions. By using kernel tricks, explicit computation of those moments are not required. Recently, Aude Genevay [11] proposed a method to learn with Sinkhorn divergence, which is a mixture of MMD and the optimal transport loss [12]. The references mentioned above do not examine Wasserstein-based distributionally robust games.

This paper introduces a new game-theoretic framework for generative learning. We formulate the problem as a distributionally robust game (DRG) under uncertainty and offer a corresponding distributionally robust equilibrium concept. In this game there are two groups of players with opposite objectives. Each player works on a continuous action space to optimize the distributionally worst-case payoff. Our model differs from the distribution-free robust game framework proposed by [13], [14]. In their approach, the uncertainty set needs to be pre-specified by the decision makers in advance, while in our approach any alternative distribution within a certain

Wasserstein distance from a tractable observed distribution can be tested.

Another issue is how to define the distance between two distributions, i.e., the similarity between real and fake sample sets. Instead of information based metric such as Kullback-Leibler (KL) divergence, we use Wasserstein metric to measure the distance between two distributions. It is a real distance and has finer topology in the parameter space, which provides better gradients and therefore improves the stability of the optimization algorithm. However, computing Wasserstein distance involves solving an optimal transportation problem, which is nontrivial. Marco Cuturi [15] adds an entropic regularization term to the original problem and switch to calculating the Sinkhorn distance. Martin Arjovsky [9] works on the Kantorovich-Rubinstein dual problem, and trains a neural network to estimate the cost. In our framework, this task is given to the defenders, who explore the Wasserstein ball of real data to generate adversarial samples for attackers. Using Moreau-Yosida regularization [16], [17], we transform the Wasserstein-based optimization into an Euclidean distance based optimization, which is much simpler.

To train deep generative models for image data, we implement the attackers and defenders with convolutional neural networks. Since the data has very high dimension, we add an encoder network to learn meaningful feature vectors and embed them into the model.

The main contributions of our work as listed below:

- We proposed a new game theory framework to learn generative models. To the best of our knowledge, this is the first work connecting distributionally robust game with deep generative learning.
- We analyzed the properties of Wasserstein distance from both theoretical and empirical perspectives and offer a toy example to illustrate its advantage over KL-divergence.
- We provide practical implementation of our framework to train a deep generative model. The algorithm has been tested on CelebA human face dataset. Both qualitative and quantitative evaluation results are reported. Our model can produce high quality images which are closer to the real data distribution than existing methods.

The rest of the paper is organized as follows. Section II introduces the game theory framework and define the distributionally robust Nash equilibrium. Generative model learning is formulated as a game in which two groups of players iteratively optimizing their worst-case objective. Section III discuses the properties of Wasserstein metric and illustrates its advantage over traditional information-based loss functions. Section IV provides detailed learning procedure of our approach. Practical implementations for training deep generative models is summarized in Algorithm 1. Experiments and performance evaluation are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Distributionally Robust Games

A. From unsupervised learning to Generative Model

Deep learning has earned great success in supervised learning. However, well-labelled data is expensive. People wish to learn directly from unlabelled data. Suppose the real data samples are drawn from an unknown distribution m, and we want to train a model to generate similar fake samples. Let $\tilde{m}$ be the fake sample distribution, then the objective is to minimize the discrepancy between real and fake distributions $D(m, \tilde{m})$. In this paper we use Wasserstein metric to measure the discrepancy and formulate the training problem as a distributionally robust game.

B. Game Theoretic Framework for Learning

Distributionally robust game (DRG) is a game with incomplete information. Instead of assuming an underlying mean-field or exactly known probability distribution, one acts with an uncertainty set, which could be distributions chosen by other players. The set of distributions should be chosen to fit for the applications at hand. In robust best-response problems, the uncertain sets are represented by deterministic models. The opponent players have a bounded capability to change the uncertain parameters, and therefore affects the objective function that the decision maker seeks to optimize. Each player has his own robust best response optimization problem to solve. Thus, the standard best response problem of player j: $\inf_{a_j \in \mathcal{A}_j} l_j(a_j, a_{-j}, \omega)$ becomes the minimax robust best response:

$$\inf_{a_j \in \mathcal{A}_j} \sup_{\omega \in \Omega} l_j(a_j, a_{-j}, \omega) \tag{1}$$

where l is the objective functional evaluated at uncertain state ω. This kind of approach on uncertainty has a long history in optimization, control and games [18], [19], [20]. A credible alternative to this set-based uncertainty is to use a stochastic model, in which the uncertain state ω is a random variable with distribution m. If we assume the generating mean-field distribution, m, is known, it becomes a standard stochastic optimal control paradigm. If m is not known and the only known is a set of distributions lie in some neighborhood of m: $m' \in B_\rho(m)$, the resulting best response to mean-field formulation is the so-called distributionally robust best response:

$$\inf_{a_j \in \mathcal{A}_j} \sup_{m' \in B_\rho(m)} \mathbb{E}_{\omega' \sim m'} l_j(a_j, a_{-j}, \omega') \tag{2}$$

We choose the uncertain set as probability distributions within a Wasserstein ball of radius ρ from m.

$$B_\rho(m) = \{m' \mid W(m, m') \leq \rho\} \tag{3}$$

C. Problem formulation

In distributionally robust games, each agent j adjusts $a_j \in \mathcal{A}_j$ to optimize the worst-case payoff functional $\mathbb{E}_{m'} l_j(a_j, \omega')$. Throughout the paper we assume that the function $l_j(\cdot, \omega')$ is proper and upper semi-continuous for $m'-$almost all

$\omega' \in \Omega$, and either the domain A_j is nonempty compact or $\mathbb{E}_{m'} l_j(a_j, \omega')$ is coercive.

Definition 1 (Robust Game)**.** *The robust game $\mathcal{G}(m)$ is given by*

- *The set of agents: $\mathcal{J} = \{1, 2, \ldots\}$*
- *The action profile of player j: $\mathcal{A}_j$, $j \in \mathcal{J}$*
- *The uncertainty set of probability distributions: $B_\rho(m)$*
- *The objective function of player j: $\mathbb{E}_{m' \in B_\rho(m)} l_j(a, \omega')$, where m' is an alternative probability distribution of m within some bounded distance.*

Then the robust stochastic optimization of agent j given the uncertain set and the action of other players is

$$(P_j): \quad \inf_{a_j \in \mathcal{A}_j} \sup_{m' \in B_\rho(m)} \mathbb{E}_{m'} l_j(a, \omega') \tag{4}$$

We introduce a distributionally robust equilibrium concept for the game $\mathcal{G}(m)$.

Definition 2 (Distributionally Robust Equilibrium)**.** *Denote by a_j^* the optimal configuration of player j and by $a_{-j}^* := (a_k^*)_{k \neq j}$ the action profile of the other players than j. A strategy profile $a^* = (a_1^*, \ldots, a_n^*)$ satisfying*

$$\sup_{m' \in B_\rho(m)} \mathbb{E}_{m'} l_j(a^*, \omega') \leq \sup_{m' \in B_\rho(m)} \mathbb{E}_{m'} l_j(a_j, a_{-j}^*, \omega')$$

for every $a_j \in \mathcal{A}_j$ and every player j, is a distributionally robust pure Nash equilibrium of the game $\mathcal{G}(m)$.

In other words, reaching the robust Nash equilibrium means all players achieve the minimum loss in their worst-case scenario. As in classical game theory, sufficient condition for existence of robust equilibrium can be obtained from the standard fixed-point theory: if $\mathcal{A}_j$ are nonempty compact convex sets and l_j are continuous functions such that for any fixed a_{-j}, the function $a_j \mapsto l_j(a, \omega')$ is quasi-convex for each j, then there exists at least one distributionally robust *pure* Nash equilibrium. This result can be easily extended to the coupled-action constraint case for generalized robust Nash equilibria.

Next we formulate the generative learning problem into the DRG framework. As depicted in Figure 1, there are two groups of players in this game. The attackers train the generative model $G_{\theta_a}(z)$ to produce fake samples $\tilde{x}_i$ that are similar to the real ones, where z is a low dimension random variable feed to the generator and θ_a is the model parameter. The defenders explore the neighborhood of m and slightly change the real data to produce hard samples x_i' which have the maximum distance from the model distribution. The attackers again refine the model to fit for those hard samples. The loss function is defined by the discrepancy $D(\tilde{m}, m')$, where m' is the hard sample distribution chosen by the defender. Since the real distribution m is unknown and we only have an observation dataset $\{x_1, \cdots, x_N\} \subset \mathbb{R}^d$, the optimization is performed by iteratively updating the generative model as well as the hard sample distribution m', which is an approximation of m within some bounded uncertain set $B_\rho(m)$.

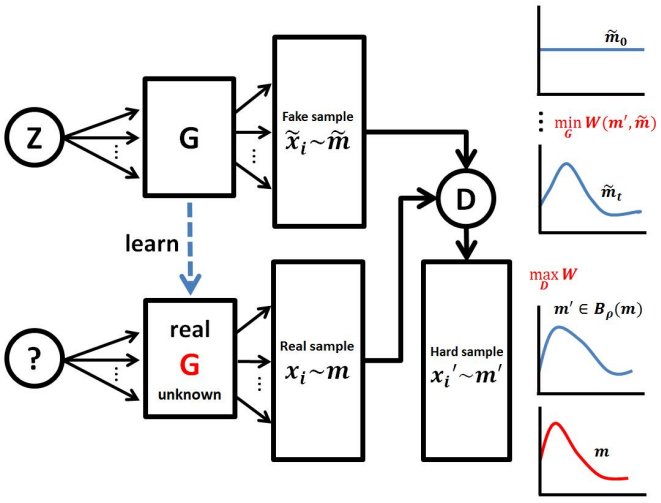

Fig. 1: Distributionally robust game (DRG) framework for generative learning

As displayed in the right column of Figure 1, initially, the fake samples are drawn from an arbitrary distribution, e.g., $\tilde{m}_0$ is a uniform. In each iteration, the attackers refine it closer to the hard sample distribution m', while the defenders keep looking for the worst-case approximation m' within the uncertain set $B_\rho(m)$. Once the algorithm converges, i.e. $D(\tilde{m}, m') \leq \epsilon$, we can ensure that the discrepancy is bounded by a small value if $D(\cdot, \cdot)$ satisfies triangle inequality:

$$D(\tilde{m}, m) \leq D(\tilde{m}, m') + D(m', m) \leq \epsilon + \rho \tag{5}$$

If m' is exactly the worst distribution in $B_\rho(m)$, we can say that $D(\tilde{m}, m) \leq |\rho - \epsilon|$ (see Figure 2). Therefore, the learning task is completed and the fake samples drawn from $\tilde{m}$ will be indistinguishable from the real ones.

Next section will discuss the properties of Wasserstein distance as a metric for $D(\cdot, \cdot)$ and compare it with the popular used KL divergence.

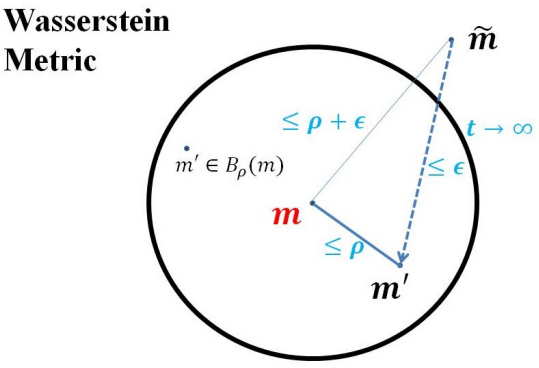

Fig. 2: Wasserstein metric as a true distance

III. Wasserstein Metric

There are various ways to define the divergence between two distributions. The most straightforward way is to sum up the point-wise loss on those two sets, such as L^p distances used in ridge regression (L^2-norm) and Lasso (L^1-norm), KL divergence and its symmetric alternative Jensen-Shannon (JS) divergence. These loss functions are decomposable and widely adopted in both discriminative and generative tasks. Since the evaluation can be conducted on individual parts, they provide convenience for incremental learning and it's easier to develop efficient algorithms. However, they do not take into account the interactions of the individual points within a set. Non-decomposable losses such as F-measure, total variation and Wasserstein distance capture the entire structure of data and provide better topologies for optimization, at the cost of additional computational burden in loss evaluation.

A. Definition

a) Optimal Transport: The optimal transport cost measures the least energy required to move all the mass in the initial distribution f_0 to match the target distribution f_1.

$$C(f_0, f_1) = \inf_{\pi \in \Pi(f_0, f_1)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\pi(x, y) \tag{6}$$

where $c(x, y)$ is the ground cost for moving one unit of mass from x to y, and $\Pi(f_0, f_1)$ denotes the set of all measures on $\mathcal{X} \times \mathcal{Y}$ with marginal distributions f_0, f_1, i.e., the collection of all possible transport plans [12].

Wasserstein distance is a specific kind of optimal transport cost in which $c(x, y)$ is a distance function. The p^{th} Wasserstein distance ($p \geq 1$) is defined on a completely separable metric space $(\mathcal{X}, d)$:

$$W_p(f_0, f_1) := (\inf_{\pi \in \Pi(f_0, f_1)} \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^p d\pi(x, y))^{\frac{1}{p}} \tag{7}$$

Specifically, when $p = 1$, W_1 is called the Kantorovich-Rubinstein distance or Earth-Mover (EM) distance. It has duality as a supremum over all 1-Lipschitz functions ψ:

$$W_1(f_0, f_1) = \sup_{\|\psi\|_{Lip} \leq 1} \mathbb{E}_{f_0}[\psi(x)] - \mathbb{E}_{f_1}[\psi(x)] \tag{8}$$

B. From KL divergence to Wasserstein Metric

Although KL divergence and its generalized version f-divergence are very popular in generative learning literatures, using Wasserstein metric $D(\tilde{m}, m) = W_p(\tilde{m}, m)$ has at least three advantages. First, Wasserstein metric is a true distance: the properties of positivity, symmetry and triangular inequality are fulfilled. Thanks to triangular inequality (Figure 2), the maximum discrepancy $D(\tilde{m}, m)$ is bounded by $\rho + \epsilon$ when the optimizer approaches the distributionally robust equilibrium.

Second, the Wasserstein space has a finer topology where the loss changes smoothly with respect to the model parameters, thus effective gradients are always available during optimization. The information based divergence like KL doesn't recognize the spatial relationship between random variables. $D_{KL}(m, \tilde{m}) = \int m(x) \log(\frac{m(x)}{\tilde{m}(x)}) dx$ is invariant to reversible transformations on $x = (x_1, x_2, \cdots)^T$ because $m(x)dx$ removes the dimensional information. This property is illustrated in Figure 3. Therefore defining the uncertain set in Wasserstein space is more reasonable than using KL. It ensures the hard samples drawn from the Wasserstein neighborhood $B_\rho(m)$ will not deviate too far from the real ones.

Third, f-divergence $D_f(m, \tilde{m}) = \int_\Omega f(\frac{dm}{d\tilde{m}}) d\tilde{m}$ requires the model distribution $\tilde{m}$ to be positive everywhere, which is not possible in many cases. But adding a widespread noise term to enforce this constraint will lead to unwanted blur in generated samples [21]. The Wasserstein metric does not impose such constraint, thus can produce sharp images.

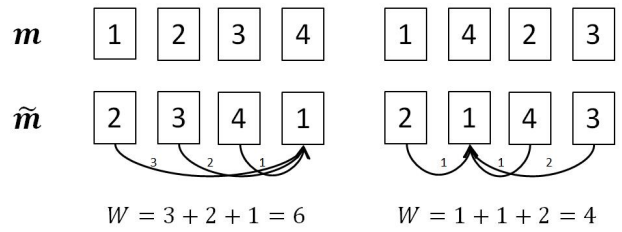

Fig. 3: An example to show Wasserstein metric can recognize permutation changes, while KL divergence outputs the same value.

Thanks to these properties, optimizing with Wasserstein loss can continuously improve the model, while the discontinuity behavior of KL divergence will deteriorate the gradients and make the training process unstable.

IV. Learning algorithm for Distributionally Robust Games

In this section we provide learning procedure to solve the distributionally robust Nash equilibria, and develop practical implementation algorithms to train deep generative models.

A. Learning the Distributional Robust Equilibrium

In DRG, the attacker and defender work against each other to find the robust Nash equilibrium by solving a minimax optimization problem in (4)

$$(P_j): \ \inf_{a_j \in \mathcal{A}_j} \ \sup_{m' \in B_\rho(m)} \ \mathbb{E}_{m'} l_j(a, \omega')$$

Since $B_\rho(m)$ is a subset of Lebesgue space (the set of integrable measurable functions under m), the original problem (P_j) has infinite dimensions, which does not facilitate the computation of robust optimal strategies. It has been proved in [22] that (P_j) can be reduced to a finite dimensional stochastic optimization problem when $\omega' \mapsto l_j(a, \omega')$ is upper semi-continuous and (Ω, d) is a Polish space. We introduce a Lagrangian function for constraint (3),

$$\tilde{l_j}(a, \lambda, m, m') = \int_\omega^{'} l_j(a, \omega') dm' + \lambda(\rho - W(m, m')) \tag{9}$$

the original problem (P_j) becomes

$$(\tilde{P_j}): \ \inf_{a_j \in \mathcal{A}_j, \lambda \geq 0} \sup_{m' \in B_\rho(m)} \tilde{l_j}(a, \lambda, m, m') \tag{10}$$

In robust game $\mathcal{G}(m)$, the defenders search for the worst hard sample distribution m' in the Wasserstein neighborhood of m to maximize its loss against the model $\tilde{m}$. According to the definition of Wasserstein metric with ground distance $d(\cdot,\cdot)$,

$$\begin{aligned}\sup_{m'} \tilde{l_j} &= \lambda\rho + \sup_{m'} \int_{\omega'} [l_j(a,\omega')]\ dm' - \lambda W(m,m') \\ &= \lambda\rho + \int_{\omega} \sup_{\omega'} [l_j(a,\omega') - \lambda d(\omega,\omega')]\ dm\end{aligned} \tag{11}$$

Define the integrand cost as

$$h_j(a,\lambda,\omega) = \lambda\rho + \sup_{\omega'}[l_j(a,\omega') - \lambda d(\omega,\omega')], \tag{12}$$

then $(\tilde{P_j})$ becomes a finite dimension problem on $\mathcal{A}_j \times \mathbb{R}_+ \times \Omega$ if $\mathcal{A}_j$ and Ω have finite dimensions

$$(\tilde{P_j^*}) \inf_{a_j \in \mathcal{A}_j, \lambda \geq 0} \mathbb{E}_m h_j(a,\lambda,\omega) \tag{13}$$

Since m is an unknown distribution observed by the noisy unsupervised dataset $x_1,\ldots,x_N$, it is challenging to compute the expected payoff $\mathbb{E}_{m'} l_j(a,\omega')$, $\mathbb{E}_m h_j(a,\lambda,\omega)$ and their partial derivatives. We need a stochastic learning algorithm to estimate the empirical gradients for the Wasserstein metric.

For a single player, the stochastic state ω_j leads to error

$$\varepsilon_j = \nabla_{a,\lambda} h_j(a,\lambda,\omega_j) - \nabla_{a,\lambda}\mathbb{E}_m h_j(a,\lambda,\omega)$$

The variance of ε_j is high and not vanishing. To handle this, we introduce a swarm of players $\omega_j \sim m$, $j \in \mathcal{J}$, then the error term becomes

$$\varepsilon = \frac{1}{|\mathcal{J}|}\sum_j\ \nabla_{a,\lambda} h_j(a,\lambda,\omega_j) - \nabla_{a,\lambda}\mathbb{E}_m h_j(a,\lambda,\omega)$$

It has zero mean and standard deviation as

$$\sqrt{\mathbb{E}[\varepsilon^2]} = \frac{1}{|\mathcal{J}|}\sqrt{var[\nabla_{a,\lambda} h_j(a,\lambda,\cdot)]}$$

For realized $\omega \leftarrow \{x_1,\ldots,x_N\}$, the expected payoff for N players is $\frac{1}{N}\sum_{j=1}^N h_j(a,\lambda,\omega_j)$, and the optimal strategy is

$$(a^*,\lambda^*) \in \arg\min_{a,\lambda} \sum_{j=1}^N h_j(a,\lambda,\omega_j)$$

This provides an accurate robust equilibrium payoff when N is very large.

B. Toy Example

To illustrate the stochastic learning algorithm we consider specific robust games with finite number of players. Each player acts as if he is facing a group of opponents whose randomized control actions are limited to a Wasserstein ball, and tries to optimize the worst case payoff. The random variable ω is distributed over m and we assume it has finite p moments. We choose $|\mathcal{J}| = 2$, $p = 2$, $d(\omega,\omega') = \|\omega - \omega'\|_2^2$ and a convex payoff function $l_j(a,\omega')$ defined on $\mathbb{R}^2 \times \mathbb{R}^2$

$$l_j(a,\omega') = \|\omega' - a\|_2^2 = (\omega_1' - a_1)^2 + (\omega_2' - a_2)^2 \tag{14}$$

The optimal defender state ω'^* is computed through the Moreau-Yosida regularization, and the attacker's action pushes it closer to the destination ω as shown in Figure 4.

$$\begin{aligned}\sup_{m'} \tilde{l_j} &= \lambda\rho + \int_{\omega\in\Omega} \phi_j(a,\lambda,\omega) dm \\ \phi_j(a,\lambda,\omega) &= \sup_{\omega'\in\mathbb{R}^2} [l_j(a,\omega') - \lambda d(\omega,\omega')] \\ &= \sup_{\omega'\in\mathbb{R}^2} (\|\omega' - a\|_2^2 - \lambda\|\omega' - \omega\|_2^2)\end{aligned} \tag{15}$$

$$\omega'^* = \omega + \frac{\omega - a}{\lambda - 1}, \quad (\lambda > 1) \tag{16}$$

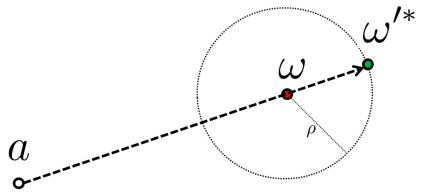

Fig. 4: Action pushes the particle toward ω (which is unknown) given ω'^*

Then $d(\omega,\omega'^*) = \|\frac{\omega - a}{\lambda - 1}\|_2^2$, leads to the worst-case loss

$$l_j(a,\omega'^*) = \|\omega'^* - a\|_2^2 = \frac{\lambda^2}{(\lambda-1)^2}\|\omega - a\|_2^2 \tag{17}$$

The Moreau-Yosida regularization on m' realized at ω'^* is

$$\begin{aligned}\phi_j(a,\lambda,\omega) &= l_j(a,\omega'^*) - \lambda d(\omega,\omega'^*) \\ &= \frac{\lambda}{\lambda-1}\|\omega - a\|_2^2\end{aligned} \tag{18}$$

The integrand cost function $h_j = \lambda\rho^2 + \frac{\lambda}{\lambda-1}\|\omega - a\|_2^2$. Thus, problem $(\tilde{P_j^*})$ becomes

$$\inf_{a,\lambda}\ \mathbb{E}_m h_j = \inf_{a,\lambda} \int_{\omega} \lambda\rho^2 + \frac{\lambda}{\lambda-1}\|\omega - a\|_2^2\ dm, \quad (\lambda > 1) \tag{19}$$

Given N observations, the stochastic robust loss is

$$\begin{aligned}l_N^* &= \frac{1}{N}\sum_{j=1}^N h_j(a,\lambda,\omega_j) \\ &= \lambda\rho^2 + \frac{\lambda}{N(\lambda-1)}\sum_{j=1}^N \|\omega_j - a\|_2^2\end{aligned}$$

We set $\rho = 1$ and m is a dirac distribution where $\omega_j \equiv 1$. Figure 5 plots the trajectories of strategies during learning.

C. Train a Deep Generative Model

Image generative models such as VAE [5], GAN [6], WGAN [9] have shown great success in recent years. VAE trains a encoder network and a decoder network by minimizing the reconstruction loss, i.e., the negative log-likelihood with a regularizer. It tends to produce blurring images due to the additional noise terms in their model. GAN trains a generator network and a discriminator network by solving a minimax problem based on the KL-divergence. The model is unstable

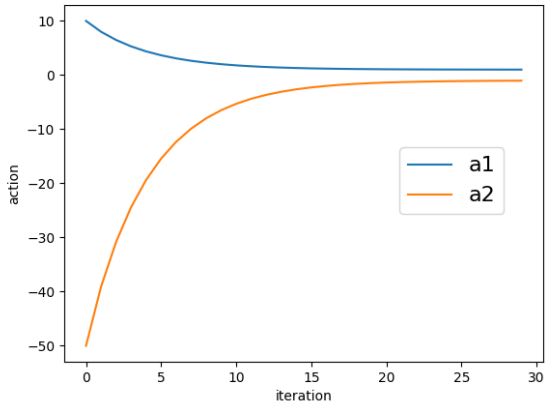

Fig. 5: The optimal strategies converges to $(a_1^*, a_2^*) = (1, -1)$

(Figure 7) due to the discontinuity of the information-based loss functions, and the generator is vulnerable to saturation as the discriminator getting better. [23], [8] gives some empirical solutions to these problems, e.g., keeping balance in training generator and discriminator networks, designing a customized network structure. WGAN [9] defines a GAN model by an efficient approximation (equation 8) of the Earth Mover distance. During training, it simply crops all weights of the discriminator network to maintain the Lipschitz constraint.

In our framework, generative learning is formulated as a distributionally robust game with two competitive groups of players, whose actions are defined on the parameter space $\theta = (\theta_a, \theta_d) \in \Theta$. In stochastic settings, ω, ω' and $\tilde{\omega}$ are instantiated to sample vectors $\{x_1, x_2, \ldots\}$, $\{x'_1, x'_2, \ldots\}$ and $\{\tilde{x}_1, \tilde{x}_2, \ldots\}$. The attacker produces indistinguishable artificial samples $\tilde{x}_i = G_{\theta_a}(z)$ to minimize the discrepancy $\inf_\theta D(\tilde{m}, m')$, where $\tilde{x}_i \sim \tilde{m}$. Meanwhile, the defender produce adversarial samples $x'_i = G_{\theta_d}(x_i)$, which are substitutes of the real ones, to maximize the loss $\sup_{m' \in B_\rho(m)} D(\tilde{m}, m')$, $x_i \sim m$, where $x'_i \sim m'$.

With Moreau-Yosida regularization, the defenders work on the following maximization problem to generate the optimal adversarial samples in Wasserstein ball $B_\rho(m)$,

$$\theta_d^* \in \arg\max_{\theta_d} \; l(\theta_a, \omega') - \lambda d(\omega, \omega')$$

and the attackers work on the minimization problem to find the best generative parameters θ_a^*

$$\theta_a^* \in \arg\min_{\theta_a, \lambda} \; \lambda\rho + l(\tilde{x}, x'^*) - \lambda d(x, x'^*)$$

Given enough observations $\{x_1, x_2, \ldots\}$ from the unknown real distribution m, a similar distribution $\tilde{m}$ can be learned by solving the distributionally robust Nash equilibrium. New samples generated from $\tilde{x}_i \sim \tilde{m}$ should be indistinguishable from the real ones. The DRG algorithm is summarized in Algorithm 1.

Algorithm 1 DRG with Wasserstein metric

Input: real data $(x_i)_{i=1}^N$, batch size n, initial attacker parameters θ_{a0}, Lagrangian multiplier λ_0, initial defender parameters θ_{d0}, number of defender updates per attacker loop n_d, Wasserstein ball radius ρ, learning rate η, low-dimension random noise $z \sim \zeta$
Output: θ_a, θ_d, λ
while θ_a has not converged **do**
  for $t = 1, 2, \ldots, n_d$ **do**
    Sample $(x_i)_{i=1}^n \sim m$ from real dataset
    Sample $(\tilde{x}_i)_{i=1}^n \sim \tilde{m}$ from generator $G_{\theta_a}(z)$
    $y_i \leftarrow E_{\theta_d}(x_i), \quad \tilde{y}_i \leftarrow E_{\theta_d}(\tilde{x}_i)$
    Modify to adversarial samples $y'_i \leftarrow G_{\theta_d}(y_i)$
    $g_d \leftarrow \nabla_{\theta_d}\, l(\tilde{y}_1^n, y'^n_1) - \lambda d(y_1^n, y'^n_1)$
    $\theta_d \leftarrow \theta_d + \eta RMSProp(g_d)$
  end for
  Sample $(x_i)_{i=1}^n \sim m$ from real dataset
  Sample $(\tilde{x}_i)_{i=1}^n \sim \tilde{m}$ from generator $G_{\theta_a}(z)$
  $y_i \leftarrow E_{\theta_d}(x_i), \quad \tilde{y}_i \leftarrow E_{\theta_d}(\tilde{x}_i)$
  Modify to adversarial samples $y'_i \leftarrow G_{\theta_d}(y_i)$
  $g_{a,\lambda} \leftarrow \nabla_{\theta_a,\lambda}\, \lambda\rho + l(\tilde{y}_1^n, y'^n_1) - \lambda d(y_1^n, y'^n_1)$
  $\theta_a \leftarrow \theta_a - \eta RMSProp(g_{a,\lambda})$
  $\lambda \leftarrow \lambda - \eta RMSProp(g_{a,\lambda})$
end while

V. Experiments

A. Dataset

We apply our DRG algorithm on the CelebA [24] dataset to generate artificial human faces. The training set has 202K cropped face images with size 64×64, therefore each real sample $x_i \sim m$ has 12288 dimensions. The artificial samples are generated from low-dimensional noise vectors $z \sim \zeta$, where ζ is a random normal distribution.

B. Network Structure

In this paper, the generative network $x = G_{\theta_a}(z)$ follows the DCGAN [25] architecture. We design $y' = G_{\theta_d}(y)$ as a single layer neural network to perform modification. For the encoding network $y = E_{\theta_d}(x)$, we use one CNN-ReLU layer followed by 3 CNN-BatchNorm layers and a fully connected layer to produce code vectors. Both networks have about 5 million training parameters.

C. Loss Functions

In DRG algorithm, the Wasserstein distance $l(\tilde{x}^n, x'^n)$ is implemented by Sinkhorn-Knopp's algorithm [26], and the ground cost $d(x^n, x'^n) = \frac{1}{n}\sum_{i=1}^n \|x_i - x'_i\|_2^2$. Instead of directly computing the L2-norm on raw data vectors, Algorithm 1 uses an encoder network $y = E_{\theta_d}(x)$ to learn useful features.

D. Hyperparameters

The encoder maps the original data into a 100-dimension feature space, which matches the dimension of the random noise z. In all experiments, the cost based on Wasserstein metric is normalized to $[0, 1]$, where the supremum indicates

the cost between images that are all black and all white. The hyperparameters listed in Algorithm 1 are chosen by validation and listed in table I; others are set as the default values in their references. For training we choose the RMSProp optimizer [27] because it doesn't involve a momentum term. Empirically, we found momentum-related optimizers may deteriorate the training. The reason is, in robust games the payoff function is dynamic and changes every time the other players take actions. Since the structure of the objective surface is not stationary, it's meaningless to follow the velocity of the previous optimization steps.

TABLE I: Hyper parameters

parameters	n	ρ	λ_0	n_d	η	$\theta_{a0}, \theta_{d0}, \eta$
values	64	0.1	10	1	0.00005	random normal

E. Evaluation

Experimental results are demonstrated in Figure 6, in which the last line shows the most similar samples in the real dataset. The training curve for DRG is plotted in Figure 8. It means the Wasserstein loss is highly related to the sample quality. By optimizing the worst-case loss function, the DRG model converges very quickly to the real data distribution and successfully produce sharp and meaningful images. In experiments we found that the original GAN generator [25] suffers from unpredictable quality deterioration at iteration $5.3K$, $7.8K$, $10.2K$ (Figure 7), etc, while our algorithm keeps improving the sample quality. This problem is caused by the discontinuity of the KL-divergence.

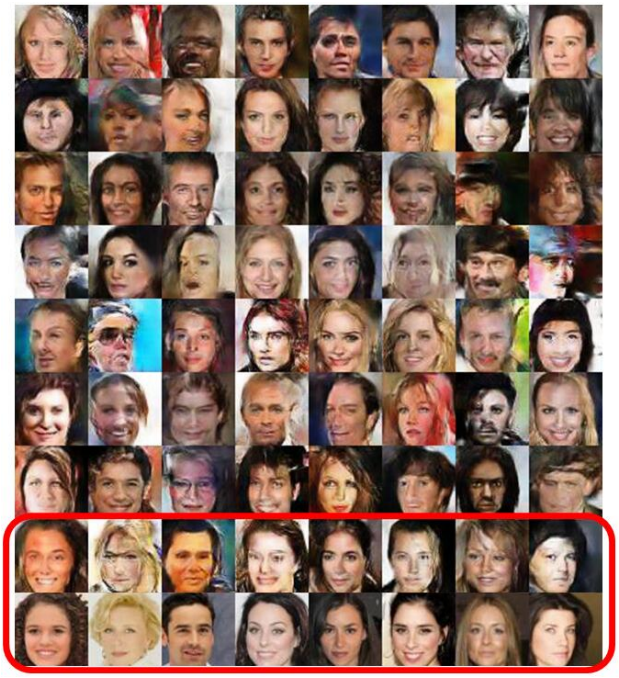

Fig. 6: DRG results on CelebA, attacker iteration = 300K

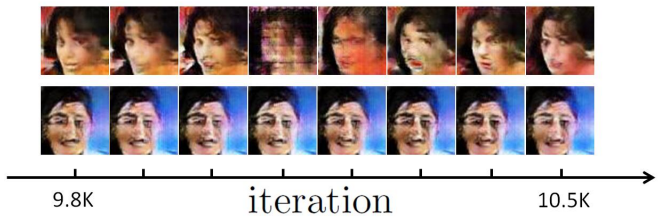

Fig. 7: Stability of the generated models. Upper: DCGAN, image quality suddenly becomes worse. Bottom: DRG

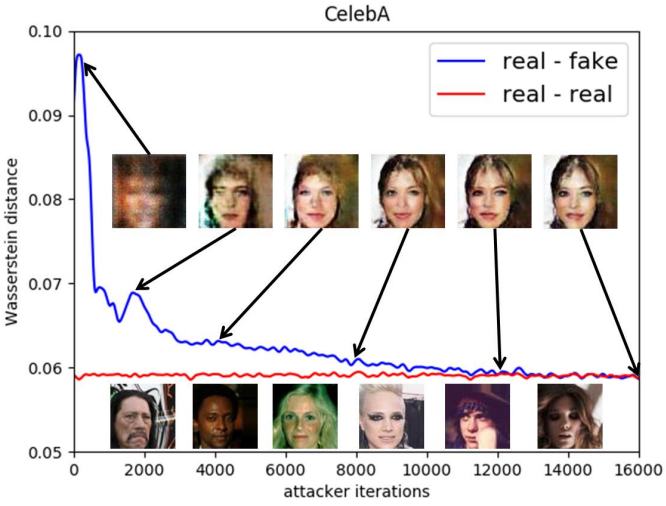

Fig. 8: Training curve for DRG algorithm with Wasserstein metric. The loss goes down as generated samples getting better, and converges to the Wasserstein distance between two real data sets. The curves are smoothed for visualization purpose.

The evaluation of generative models is itself a research topic. [28] figured out that different evaluation metrics favor different models. For example, a high log-likelihood doesn't mean good visual quality, and vice versa. Therefore, the metric used in training and evaluation should fit for the specific application. In our case, the learned fake data distribution should be as close as to the real one. So we measure the discrepancy between these distributions using Wasserstein metric. We compare our algorithm with DCGAN [25] and WGAN [9], and report the quantitative results in Table II.

The computation complexity per attacker iteration is linear $O(n)$ with respect to the batch size. We use a Titan Xp to train the model and plot the computation time in Figure 9. When $n = 64$, it takes 0.2 seconds for an attacker update. Our algorithm has smaller constant factor than WGAN.

VI. Conclusion

We proposed a new game theory model with Wasserstein loss to train generative models. In this game, two competing groups of players work on a minimax problem to optimize the discrepancy between model and data. The defenders change slightly the data to produce a set of hard examples that has

TABLE II: Performance evaluation

$W(m, \tilde{m})$ $(\times 10^{-5})$	1K samples	10K samples
real - real	12.9	1.74
real - DRG	**22.6**	**15.9**
real - DCGAN	37.3	16.4
real - WGAN	31.0	17.2

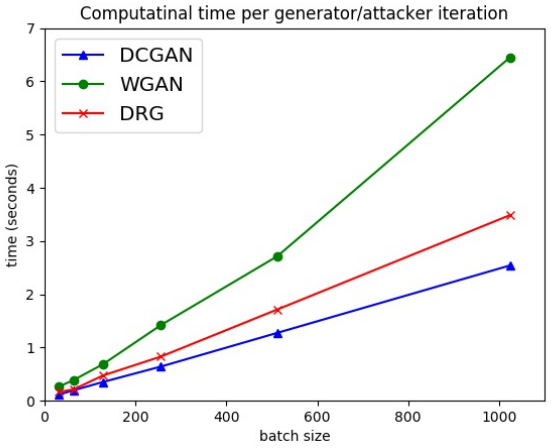

Fig. 9: Computation time with respect to batch size.

the maximum distance from the model distribution, while the attackers take action to push the model toward the unknown real by fitting for the hard set. Instead of prevalent information-based loss functions such as KL-divergence, we use Wasserstein distance to measure the similarity between distributions. Its advantages have been analyzed from both theoretical and empirical perspectives. We offered a practical realization on neural networks and applied our model in deep generative learning. The algorithm was tested on large-scale human face dataset, and it can produce artificial samples with good visual quality and high diversity. The learning process is stable and converges fast. Experiment evaluation shows our algorithm achieving better performance than DCGAN and WGAN in terms of the statistical distance between the real and fake sample distributions.

To our knowledge, this is the first work connecting distributionally robust game with deep generative learning. In the future, we plan to extend this framework to learn sequential data, such as speech synthesis and video generation. Another direction is to study the properties of Wasserstein space and develop more efficient algorithms for robust optimization.

ACKNOWLEDGMENT

This research is supported by U.S. Air Force Office of Scientific Research under grant number FA9550-17-1-0259.

REFERENCES

[1] G. E. Hinton and T. J. Sejnowski. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning and Relearning in Boltzmann Machines, pages 282–317. MIT Press, Cambridge, MA, USA, 1986.

[2] N. Le Roux and Y. Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural Computation*, 20(6):1631–1649, June 2008.

[3] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.

[4] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.

[5] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[7] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *CoRR*, abs/1606.04838, 2016.

[8] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *CoRR*, abs/1701.04862, 2017.

[9] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017.

[10] Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. *CoRR*, abs/1502.02761, 2015.

[11] Aude Genevay, Gabriel Peyrhé, and Marco Cuturi. Learning generative models with sinkhorn divergences, 2017.

[12] Cédric Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer, 2009 edition, September 2008.

[13] Michele Aghassi and Dimitris Bertsimas. Robust game theory. *Mathematical Programming*, 107(1):231–273, Jun 2006.

[14] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Rev.*, 53(3):464–501, August 2011.

[15] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2292–2300, 2013.

[16] J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

[17] K. Yoshida. *Functional Analysis*. Classics in mathematics / Springer. Springer-Verlag Berlin Heidelberg, 6 edition, 1995.

[18] H.E. Scarf. *A Min-max Solution of an Inventory Problem*. P (Rand Corporation). Rand Corporation, 1957.

[19] Maurice Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958.

[20] Allen L. Soyster. Technical note - convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.

[21] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger B. Grosse. On the quantitative analysis of decoder-based generative models. *CoRR*, abs/1611.04273, 2016.

[22] Dario Bauso, Jian Gao, and Hamidou Tembine. Distributionally robust games: f-divergence and learning. *VALUETOOLS, Venice, Italy*, 2017.

[23] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016.

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[25] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[26] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.*, 21(2):343–348, 1967.

[27] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[28] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *CoRR*, abs/1511.01844, 2015.