Generative Warfare Nets: Ensemble via Adversaries and Collaborators

Honglun Zhang[1,2], Liqiang Xiao[1,2], Wenqing Chen[1,2], Yongkun Wang[3], Yaohui Jin[1,2]
[1]State Key Lab of Advanced Optical Communication System and Network, Shanghai Jiao Tong University
[2]Artificial Intelligence Institute, Shanghai Jiao Tong University
[3]Network and Information Center, Shanghai Jiao Tong University
{jinyh}@sjtu.edu.cn

Abstract

Generative Adversarial Nets are a powerful method for training generative models of complex data, where a Generator and a Discriminator confront with each other and get optimized in a two-player minmax manner. In this paper, we propose the Generative Warfare Nets (GWN) that involve multiple generators and multiple discriminators from two sides to exploit the advantages of Ensemble Learning. We maintain the authorities for the generators and the discriminators to enhance inter-side interactions, and utilize the mechanisms of imitation and innovation to model intra-side interactions among the generators, where they can not only learn from but also compete with each other. Extensive experiments on three natural image datasets show that GWN can achieve state-of-the-art Inception scores and produce diverse high-quality synthetic results.

1 Introduction

Generative Adversarial Nets (GAN) [Goodfellow *et al.*, 2014] are a powerful method for training generative models of complex data and have been proven effective in a wide range of applications, including natural image generation [Radford *et al.*, 2015], image super-resolution [Zhang *et al.*, 2016] and so on. GAN consists of a *Generator* and a *Discriminator* playing a two-player minmax game, where the generator tries to synthesize fake samples from noises that follow a prior distribution, while the discriminator aims to distinguish those fake samples from the real ones.

However, GAN is particularly difficult to train as the generator cannot learn valid gradients when the discriminator is either under-optimized or over-optimized. Losses of the generator and the discriminator are incapable of explicitly indicating the training progress and a meaningful metric evaluating the synthetic images is lacked [Arjovsky and Bottou, 2017]. Another tough challenge of GAN is the *collapsing mode* problem, where the generated images seem real but resemble each other, namely deficient in diversities.

A large body of literature are devoted to improve GAN and these works can be roughly grouped into three categories. 1) reformulating the original loss functions and exploring alternatives that are more stable, robust and meaningful [Gulrajani *et al.*, 2017; Salimans *et al.*, 2016; Berthelot *et al.*, 2017]. 2) introducing additional auxiliary features to better control the details of the synthetic images [Mirza and Osindero, 2014; Chen *et al.*, 2016]. 3) employing more generators or discriminators to train a mixed model of stronger generative capabilities [Tolstikhin *et al.*, 2017; Hoang *et al.*, 2017; Durugkar *et al.*, 2016; Nguyen *et al.*, 2017].

For the last category, it is intuitive to utilize multiple generators or discriminators as *Ensemble Learning* [Rokach, 2010] has proven to be an effective method where multiple weaker models are combined into a stronger model, such as Decision Tree *v.s.* Random Forest. These are existing literature on multiple generators [Tolstikhin *et al.*, 2017; Hoang *et al.*, 2017] or multiple discriminators [Durugkar *et al.*, 2016; Nguyen *et al.*, 2017], but a more complicated and interesting situation, which involves multiple generators and discriminators, has not been researched much yet.

In this paper, we propose the **Generative Warfare Nets** (**GWN**) analogous to a number of counterfeiter teams and police teams [Goodfellow *et al.*, 2014], where the generators and the discriminators (we refer both of them as **agents**) can interact with both their adversaries and collaborators. The authorities of all agents are maintained to enhance inter-side interactions and the one with higher authority obtains greater influences. We also design the mechanisms of imitation and innovation to model the intra-side interactions among the generators, as they can learn from and compete with each other to achieve stronger generative capabilities.

We conduct extensive experiments on three natural image datasets to evaluate GWN. Compared to single generator and discriminator, a multi-player game for both sides results in stronger capabilities of generation and discrimination.

Our contributions are three-folds:

- We propose an ensemble extension of GAN where multiple generators and discriminators interact with each other and get optimized via both adversaries and collaborators.
- We design several mechanisms to model both inter-side and intra-side interactions, including agent authority, imitation and innovation.
- Extensive experiments demonstrate that GWN can

achieve state-of-the-art Inception scores and produce diverse high-quality synthetic images.

2 Methodology

There are lots of different variants of the vanilla GAN and we choose the Improved WGAN [Gulrajani *et al.*, 2017] as a building block to implement our model, as it proposes a few minor modifications that effectively solve the problems of training instability and mode collapsing.

2.1 Improved WGAN

In the Improved WGAN, the generator G produces fake samples $\tilde{\boldsymbol{x}}$ from noises $\boldsymbol{z}$ that follow a prior distribution $p_{\boldsymbol{z}}$, so the generative distribution p_g is implicitly defined by $\tilde{\boldsymbol{x}} = G(\boldsymbol{z}), \boldsymbol{z} \sim p_{\boldsymbol{z}}$. For the input samples, either real ones $\boldsymbol{x}$ from the data distribution p_r or fake ones $\tilde{\boldsymbol{x}}$ synthesized by $G(\boldsymbol{z})$, the discriminator D outputs scalar values indicating the probabilities that each sample comes from the real distribution p_r.

Both G and D are differentiable functions, which can be implemented by either multi-layer perceptron or deep convolutional architectures. The generator and the discriminator are trained to minimize the two losses, L_g and L_d, respectively.

$$L_d = \underset{\tilde{\boldsymbol{x}} \sim p_g}{\mathbb{E}}[D(\tilde{\boldsymbol{x}})] - \underset{\boldsymbol{x} \sim p_r}{\mathbb{E}}[D(\boldsymbol{x})] + GP(\hat{\boldsymbol{x}}) \tag{1}$$

$$GP(\hat{\boldsymbol{x}}) = \lambda \underset{\hat{\boldsymbol{x}} \sim p_{\hat{\boldsymbol{x}}}}{\mathbb{E}}[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2] \tag{2}$$

$$\hat{\boldsymbol{x}} = \epsilon \boldsymbol{x} + (1-\epsilon)\tilde{\boldsymbol{x}}, \epsilon \sim U[0,1] \tag{3}$$

$$L_g = -\underset{\tilde{\boldsymbol{x}} \sim p_g}{\mathbb{E}}[D(\tilde{\boldsymbol{x}})] \tag{4}$$

where $GP(\hat{\boldsymbol{x}})$ is the gradient penalty on D to enforce the 1-Lipschitz constraint, and $\hat{\boldsymbol{x}}$ are the random samples uniformly along straight lines between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$.

The loss function of the generator, which correlates with the qualities of the synthetic images and converges toward a minimum, can be utilized as an indicator of the training progress. Given the appealing advantages of the Improve WGAN, we apply it as a building block to implement our ensemble model.

2.2 Generative Warfare Nets

Based on the Improved WGAN, we propose the Generative Warfare Nets (GWN). Figure 1 illustrates the architecture of GWN, where M generators (the left green blocks) and N discriminators (the right yellow blocks) confront with each other and get optimized via both adversaries and collaborators.

The M generators, $G^{(m)}, m = 1, 2, ..., M$, produce fake samples $\tilde{\boldsymbol{x}}^{(m)}$ from noises $\boldsymbol{z}$ that follow a prior distribution $p_{\boldsymbol{z}}$, so the generative distributions $p_g^{(m)}$ is implicitly defined by $\tilde{\boldsymbol{x}}^{(m)} = G^{(m)}(\boldsymbol{z}), \boldsymbol{z} \sim p_{\boldsymbol{z}}$. The N discriminators, $D^{(n)}, n = 1, 2, ..., N$, accept real samples $\boldsymbol{x}$ as well as fake ones $\tilde{\boldsymbol{x}}^{(m)}$ from each generator and evaluate the probabilities that they come from the real distribution p_r.

We redefine Eq.(1-4) so that both $G^{(m)}$ and $D^{(n)}$ can interact with all the agents from the other side. $D^{(n)}$ needs to distinguish the real samples $\boldsymbol{x}$ from all the fake ones $\tilde{\boldsymbol{x}}^{(m)}$, so the corresponding loss function $L_d^{(n)}$ is calculated as follows.

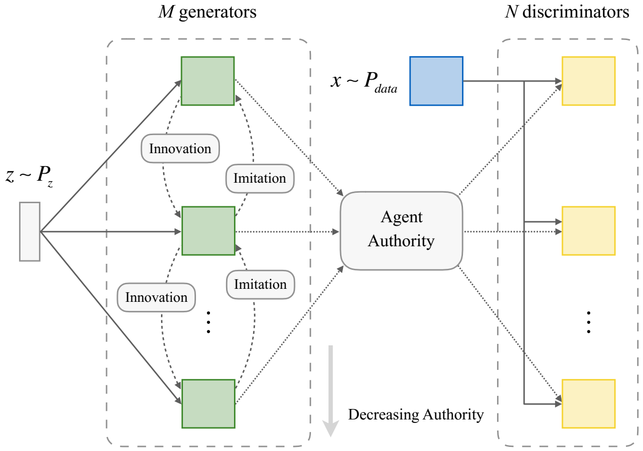


Figure 1: Network Architecture of the Generative Warfare Nets

$$\begin{aligned} L_d^{(n)} =& \frac{1}{M}\sum_{m=1}^{M} \underset{\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}}{\mathbb{E}}[D^{(n)}(\tilde{\boldsymbol{x}}^{(m)})] \\ &- \underset{\boldsymbol{x} \sim p_r}{\mathbb{E}}[D^{(n)}(\boldsymbol{x})] + GP^{(n)}(\hat{\boldsymbol{x}}) \end{aligned} \tag{5}$$

where $GP^{(n)}(\hat{\boldsymbol{x}})$ is the gradient penalty on $D^{(n)}$ to enforce the 1-Lipschitz constraint.

$$GP^{(n)}(\hat{\boldsymbol{x}}) = \lambda \underset{\hat{\boldsymbol{x}} \sim p_{\hat{\boldsymbol{x}}}}{\mathbb{E}}[(\left\|\nabla_{\hat{\boldsymbol{x}}} D^{(n)}(\hat{\boldsymbol{x}})\right\|_2 - 1)^2] \tag{6}$$

$$\hat{\boldsymbol{x}} = \epsilon \boldsymbol{x} + \frac{1-\epsilon}{M}\sum_{m=1}^{M} \tilde{\boldsymbol{x}}^{(m)}, \epsilon \sim U[0,1] \tag{7}$$

The generator $G^{(m)}$ tries to fool all the discriminators and learn valid gradients from them to get improved, so the corresponding loss function $L_g^{(m)}$ is calculated as follows.

$$L_g^{(m)} = -\frac{1}{N}\sum_{n=1}^{N} \underset{\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}}{\mathbb{E}}[D^{(n)}(\tilde{\boldsymbol{x}}^{(m)})] \tag{8}$$

2.3 Agent Authority

In Eq.(5-8), each agent contributes equally to the other side, which should be replaced by a more reasonable and flexible strategy. We propose the idea of **Authority** for each agent and higher authorities mean greater influences and stronger capabilities of generation or discrimination.

We denote the authorities of $D^{(n)}$ and $G^{(m)}$ at the training step t by $ad_t^{(n)}$ and $ag_t^{(m)}$. We assume that the agent authorities of each side should add up to 1.

$$\sum_{n=1}^{N} ad_t^{(n)} = \sum_{m=1}^{M} ag_t^{(m)} = 1 \tag{9}$$

Initially, agents of the same side have equal influences so $ad_0^{(n)} = \frac{1}{N}$ and $ag_0^{(m)} = \frac{1}{M}$. After each training iteration, we update the authorities according to the performances of $D^{(n)}$ and $G^{(m)}$. For the discriminators, the one that successfully identifies more fake samples should gain higher authority, which can be calculated as a weighted combination of the expected discriminative accuracies on all the generators.

$$\widehat{ad}_t^{(n)} = \sum_{m=1}^{M} ag_{t-1}^{(m)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}} [1 - D^{(n)}(\tilde{\boldsymbol{x}}^{(m)})] \tag{10}$$

Similarly, we update $ag_t^{(m)}$ so that the generator that successfully fools more discriminators obtains higher authority.

$$\widehat{ag}_t^{(m)} = \sum_{n=1}^{N} ad_{t-1}^{(n)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}} [D^{(n)}(\tilde{\boldsymbol{x}}^{(m)})] \tag{11}$$

In order to meet the requirements of Eq.(9), we apply the softmax transformation to conduct scaling on $\widehat{ad}_t^{(n)}$ and $\widehat{ag}_t^{(m)}$.

$$ad_t^{(n)} = \frac{\exp(\widehat{ad}_t^{(n)})}{\sum_{j=1}^{N} \exp(\widehat{ad}_t^{(j)})} \tag{12}$$

$$ag_t^{(m)} = \frac{\exp(\widehat{ag}_t^{(m)})}{\sum_{i=1}^{M} \exp(\widehat{ag}_t^{(i)})} \tag{13}$$

Given the agent authorities, we reformulate Eq.(5) so that the generators with higher authorities demonstrate greater influences on the computations of $L_d^{(n)}$.

$$\begin{aligned} L_d^{(n)} = & \sum_{m=1}^{M} ag_t^{(m)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}} [D^{(n)}(\tilde{\boldsymbol{x}}^{(m)})] \\ & - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_r} [D^{(n)}(\boldsymbol{x})] + GP^{(n)}(\hat{\boldsymbol{x}}) \end{aligned} \tag{14}$$

$\hat{\boldsymbol{x}}$ and $L_g^{(m)}$ are redefined similarly as follows.

$$\hat{\boldsymbol{x}} = \epsilon \boldsymbol{x} + (1 - \epsilon) \sum_{m=1}^{M} ag_t^{(m)} \tilde{\boldsymbol{x}}^{(m)}, \epsilon \sim U[0, 1] \tag{15}$$

$$L_g^{(m)} = - \sum_{n=1}^{N} ad_t^{(n)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}} [D^{(n)}(\tilde{\boldsymbol{x}}^{(m)})] \tag{16}$$

The authorities get updated dynamically during training iterations and make it possible to distinguish stronger agents from weaker ones, thereby producing more reasonable comprehensions when evaluating the agents of the other side.

2.4 Imitation and Innovation

Eq.(14-16) define the inter-side interactions for the agents with their adversaries. However, the agents can also cooperate with their collaborators, namely intra-side interactions.

For the generators, $G^{(m)}$ can interact and learn from those with higher authorities. We use $G_{sup}^{(m)}$ to denote the set of superiors for $G^{(m)}$, where $G^{(i)}, i \in G_{sup}^{(m)}$ are the generators with higher authorities than $G^{(m)}$, namely $ag_t^{(i)} > ag_t^{(m)}$. $G^{(m)}$ learns from the synthetic samples $\tilde{\boldsymbol{x}}^{(i)}$ of $G^{(i)}$ and tries to chase $G^{(i)}$ by approaching $\tilde{\boldsymbol{x}}^{(m)}$ with $\tilde{\boldsymbol{x}}^{(i)}$. We define this kind of behaviors as **Imitation**, which are modeled by adding another loss term to $L_g^{(m)}$.

$$\begin{aligned} L_g^{(m)} = & - \sum_{n=1}^{N} ad_t^{(n)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}} [D^{(n)}(\tilde{\boldsymbol{x}}^{(m)})] \\ & + \gamma \sum_{i \in G_{sup}^{(m)}} (ag_t^{(i)} - ag_t^{(m)}) \, \mathbb{E}(\left\| \tilde{\boldsymbol{x}}^{(i)} - \tilde{\boldsymbol{x}}^{(m)} \right\|_2) \end{aligned} \tag{17}$$

where γ is the imitation coefficient and the generators with higher rankings demonstrate stronger influences with greater values of $ag_t^{(i)} - ag_t^{(m)}$.

Imitation is effective as there are no concrete criteria to judge whether a sample is real, so the generators need to learn from their superiors and absorb beneficial patterns. In contrast, it is less necessary to apply imitation on the discriminators, as we know the specific answers for each sample, real or fake, so it is adequate for the discriminators to refer to the exact answers and get optimized accordingly.

However, a potential disadvantage of imitation is that all generators end up producing similar samples with each other, which violates the objective of ensemble. To alleviate the problem, we propose the mechanism of **Innovation**, where the superiors try to generate samples different from the inferiors and explore possibilities of new patterns. We use $G_{inf}^{(m)}$ to denote the set of inferiors for $G^{(m)}$, where $G^{(i)}, i \in G_{inf}^{(m)}$ are the generators with lower authorities than $G^{(m)}$, namely $ag_t^{(i)} < ag_t^{(m)}$. $G^{(m)}$ tries to distance themselves from inferiors by maximizing the variances between $\tilde{\boldsymbol{x}}^{(m)}$ and $\tilde{\boldsymbol{x}}^{(i)}$, so the innovation loss is calculated as follows.

$$\sum_{i \in G_{inf}^{(m)}} (ag_t^{(i)} - ag_t^{(m)}) \, \mathbb{E}(\left\| \tilde{\boldsymbol{x}}^{(i)} - \tilde{\boldsymbol{x}}^{(m)} \right\|_2) \tag{18}$$

Given the facts that $G_{sup}^{(m)} \cap G_{inf}^{(m)} = \varnothing$ and $G_{sup}^{(m)} \cup G_{inf}^{(m)} = \{1, ..., M\}/m$, we merge Eq.(17-18) into a more compact expression, where γ is shared by both imitation and innovation. For the same reason discussed above, the innovation loss is not applied to the discriminators.

$$\begin{aligned} L_g^{(m)} = & - \sum_{n=1}^{N} ad_t^{(n)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}} [D^{(n)}(\tilde{\boldsymbol{x}}^{(m)})] \\ & + \gamma \sum_{i=1}^{M} (ag_t^{(i)} - ag_t^{(m)}) \, \mathbb{E}(\left\| \tilde{\boldsymbol{x}}^{(i)} - \tilde{\boldsymbol{x}}^{(m)} \right\|_2) \end{aligned} \tag{19}$$

The ideas of imitation and innovation are intuitive and properly model the cases in real life. For a group of technology companies, the inferiors aim to imitate and duplicate the products of the superiors, while the superiors attempt to exploit new technologies that distinguish them from others. The intra-side interactions can be both collaborative and adversarial, thus leading to a more comprehensive and competitive situation.

2.5 Training GWN

By combining Section 2.2-2.4, we propose the algorithm for training GWN. For each iteration, we sample real data $\boldsymbol{x} \sim p_r$ and fake ones $\tilde{\boldsymbol{x}}^{(m)} \sim p_g^{(m)}$ with the batch size K, tune the parameters of $G^{(m)}$ and $D^{(n)}$ with the *Adam* [Leen and Orr, 1993] optimizers and then update the authorities for the agents accordingly. We follow the ideas of [Gulrajani *et al.*, 2017] to train the discriminators for n_d times in each iteration to produce better gradients for the generators and apply layer normalization [Ba *et al.*, 2016] instead of batch normalization for the discriminators.

Algorithm 1 Training **G**enerative **W**arfare **N**ets

```
Require: real samples $x \sim p_r$, prior noises $z \sim p_z$, $M$ generators and $N$ discriminators, the agent authorities $ad_t^{(n)}$ and $ag_t^{(m)}$, the imitation and innovation coefficient $\gamma$, the batch size $K$, the gradient penalty coefficient $\lambda$, the training times for the discriminators $n_d$, the Adam hyperparameters $\alpha, \beta_1, \beta_2$.
Initialize $G^{(m)}, D^{(n)}, ad_t^{(m)}, ag_t^{(n)}$
while $G^{(m)}, m = 1, ..., M$ has not converged do
    for $d = 1, ..., n_d$ do
        for $k = 1, ..., K$ do
            sample a real data $x \sim p_r$, a noise $z \sim p_z$, a random number $\epsilon \sim U[0, 1]$.
            get each $\tilde{x}^{(m)}$ from $G^{(m)}(z)$
            generate $\hat{x}$ by Eq.(15)
            calculate each $L_d^{(n)}$ by Eq.(14)
        end for
        for $n = 1, ..., N$ do
            train $D^{(n)}$ by Adam($\frac{1}{K} \sum_{k=1}^{K} L_d^{(n)}$)
        end for
    end for
    sample $K$ noises $z \sim p_z$
    for $m = 1, ..., M$ do
        get $\tilde{x}^{(m)}$ from $G^{(m)}(z)$
        calculate $L_g^{(m)}$ by Eq.(19)
        train $G^{(m)}$ by Adam($L_g^{(m)}$)
    end for
    update $ad_t^{(n)}$ for $D^{(n)}$ by Eq.(12)
    update $ag_t^{(m)}$ for $G^{(m)}$ by Eq.(13)
end while
```

3 Experiment

In this section, we conduct detailed discussions about the implementations of GWN, investigate the performances of GWN on three natural image datasets and compare it to state-of-the-art baselines.

3.1 Datasets

We select three natural image datasets with increasing diversities and sizes to conduct experiments for GWN, CIFAR-10 [Krizhevsky and Hinton, 2009], STL-10 [Coates *et al.*, 2011] and ImageNet [Russakovsky *et al.*, 2015]. CIFAR-10 contains 60,000 32 × 32 images of 10 classes including airplane, automobile, bird and so on. STL-10 is a sampled subset of ImageNet and contains 100,000 96 × 96 images of 10 classes with greater diversities than CIFAR-10. Compared to the above two datasets, ImageNet is much larger and more diverse with over 1.2 million images from 1,000 classes, and here we use the downsampled 32 × 32 release.

Dataset	Size	Class	Resolution
CIFAR-10	60K	10	32×32
STL-10	100K	10	96×96
ImageNet	1.28M	1000	32×32

Table 1: Statistics of the three natural image datasets

3.2 Hyperparameters and Training

We utilize the deep convolutional architecture [Radford *et al.*, 2015] for the agents and all the parameters of the neural layers are randomly initialized with the Xavier initializer [Glorot and Bengio, 2010]. Noises $\boldsymbol{z}$ are sampled from a standard Gaussian distribution and we apply the Inception score [Salimans *et al.*, 2016] as the metric for evaluations. After the training progress of GWN, we select the generator with the highest authority and calculate its Inception score as the performance of GWN. The influences of M, N, γ will be further investigated and the settings for other hyperparameters are shown in Table 2.

Description	Setting
batch size	$K = 64$
gradient penalty	$\lambda = 10$
discriminator times	$n_d = 5$
Adam hyperparameters	$\alpha = 0.0001, \beta_1 = 0.5, \beta_2 = 0.9$

Table 2: Hyperparameter settings

3.3 Influences of M and N

It is intuitive that the settings of M and N have essential influences on the performances of GWN. We explore different combinations of M and N under the constraint $M + N \leq 6, \gamma = 0.01$ on CIFAR-10 to limit the space for searching and the results are illustrated in Figure 2, where the blue and green cells denote the results of Inception score and iteration time respectively.

GWN reduces to the Improved WGAN when $M = 1$ and $N = 1$ as no extra mechanisms are applied. Compared to the Improved WGAN, GWN achieves improvements of 0.41 with $M = 2, N = 1$ and 0.32 with $M = 1, N = 2$. The Inception score significantly rises with greater M, N and benefits more from an increasing number of generators, as more capabilities of generation are available.

The computation time per iteration in seconds is positively related to M and N, more than $M + N$ but less than $M \times N$. It is always more time-consuming to introduce another discriminator than a generator, as the discriminators need to deal with both the fake samples and the real ones. For a tradeoff between Inception score and iteration time, we choose to apply $M = 4$ and $N = 2$ for the subsequent experiments.

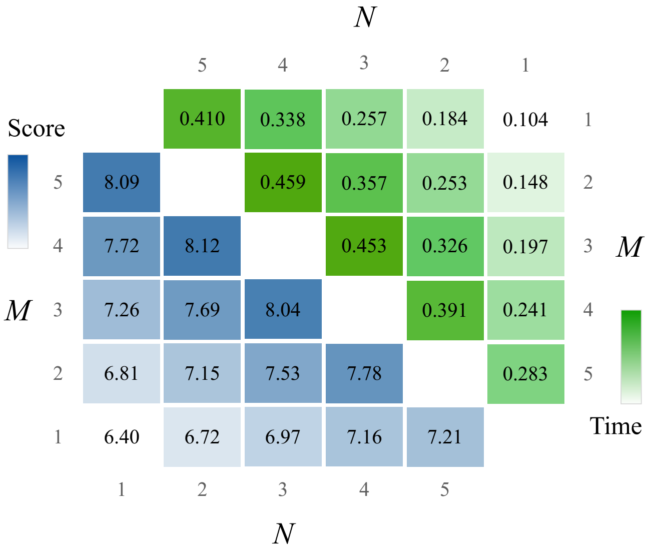
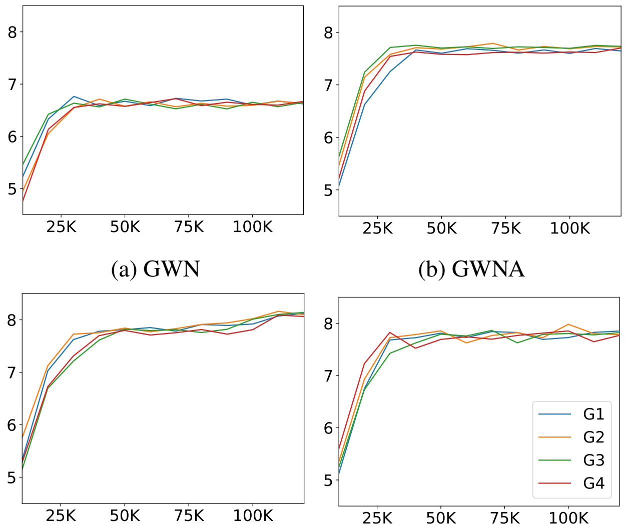


Figure 2: Results under different combinations of M and N

3.4 Ablation Study

We investigate the contributions of agent authority (A) against imitation and innovation (I) on CIFAR-10. Figure 3 illustrates the Inception scores of all the generators every 10K iterations during the training progresses of four scenarios with $M = 4$ and $N = 2$. The mechanisms of imitation and innovation are constructed based on the agent authority, so we cannot build a scenario where only I is applied without A.



(a) GWN

(b) GWNA

(c) GWNAI-0.01

(d) GWNAI-0.1

Figure 3: Comparisons of different scenarios

- GWN neither A nor I is applied.
- GWNA only A is applied.
- GWNAI-0.01 both A and I are applied with $\gamma = 0.01$.
- GWNAI-0.1 both A and I are applied with $\gamma = 0.1$.

In Figure 3a, the generators and the discriminators just interact with the agents from the other side and share equal weights, which result in an average improvement of 0.21 against the Improved WGAN (6.40 in Figure 2). In contrast, the agents in Figure 3b obtain their respective authorities and the ones with higher authorities speak louder, which lead to a more reasonable learning strategy for the agents and promote greater possibilities of improvements.

The mechanisms of imitation and innovation are applied in Figure 3c, where the inferiors try to chase the superiors while the superiors aim to explore new patterns. The effects of imitation can be observed in Figure 3c, where the generators learn from their superiors and achieve higher authorities (see the peaks of $G^{(4)}$). The mechanism of innovation may fail due to exploring a wrong direction, analogy to an unsuccessful new product (see the valleys of $G^{(1)}$ and $G^{(2)}$), but can also discover useful new patterns with the right directions (see the sustained increasing trend in Figure 3c compared to Figure 3b).

By comparing Figure 3c and 3d, it can be concluded that a γ with properly smaller values is better, which alleviates the costs of trial and error for the superiors and encourages them to discover new patterns of greater diversities. In Figure 3d, the performances of the superiors may drop severely due to wrong trials with $\gamma = 0.1$ (see the sharp valleys of the generators with the highest authorities).

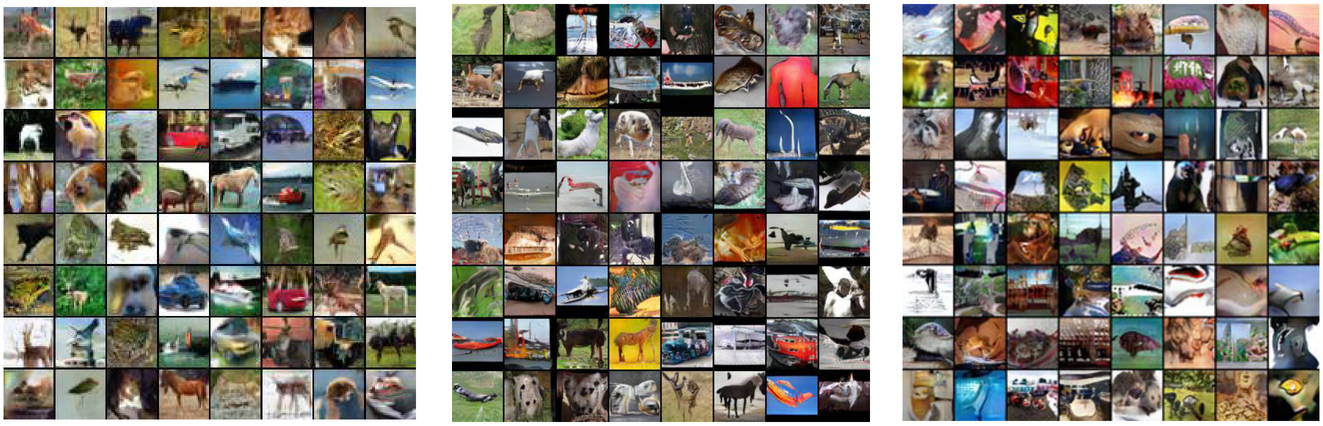
3.5 Image Generation

We present the image samples generated by GWN with $M = 4, N = 2, \gamma = 0.01$ on the three datasets in Figure 4 and samples are randomly drawn rather than cherry-picked, where meaningful and recognizable objects can be observed with increasing varieties.

3.6 Comparisons with State-of-the-art Models

We configure GWN with $M = 4, N = 2, \gamma = 0.01$ and compare it against the following state-of-the-art models, which are all trained in a completely unsupervised manner without label information. The images of STL-10 are resized down to 48×48 following the procedure of [Krizhevsky *et al.*, 2012] for fair comparisons with [Nguyen *et al.*, 2017] and [Hoang *et al.*, 2017].

- **DCGAN** GAN with a deep convolutional architecture trained with the standard procedure using the $-log(D)$ trick [Radford *et al.*, 2015]
- **IWGAN** The Improved WGAN trained with a deep ResNet architecture [Gulrajani *et al.*, 2017]
- **MIX+GAN** GAN with a mixture of multiple generators and discriminators in deep convolutional architectures [Arora *et al.*, 2017]
- **D2GAN** GAN with two discriminators that combines the KL and reverse KL divergences into a unified objective function [Nguyen *et al.*, 2017]
- **GMAN** GAN with multiple discriminators acting as formidable adversaries or forgiving teachers [Durugkar *et al.*, 2016]



(a) CIFAR-10 32 × 32

(b) STL-10 48 × 48

(c) ImageNet 32 × 32

Figure 4: Image samples generated by GWN on different datasets

- **MGAN** GAN with a mixture of multiple generators and a classifier [Hoang *et al.*, 2017]

As Table 3 shows, GWN achieves competitive or better Inception scores on all the three natural image datasets. MGAN outperforms our model on CIFAR-10 and STL-10, as it trains a mixture of 10 generators each specialized in different patterns of the data distribution. However, MGAN requires the number of generators to approach the class number, which cannot be satisfied for the ImageNet and other datasets that are highly diverse. We only explore different combinations under the constraint $M + N \leq 6$ due to limited resources, but the results demonstrate that it is of high chances for GWN to achieve better performances with greater M and N.

Model	**CIFAR-10**	**STL-10**	**ImageNet**
Real data	11.24±0.16	26.08±0.26	25.78±0.47
DCGAN	6.40±0.05	7.54	7.89
IWGAN	7.86±0.07	-	-
MIX+GAN	7.72±0.09	-	-
GMAN	6.00±0.19	-	-
D2GAN	7.15±0.07	7.98	8.25
MGAN	**8.33±0.10**	**9.22±0.11**	9.32±0.10
GWN	8.12±0.08	9.16±0.10	**9.40±0.11**

Table 3: Comparisons with state-of-the-art models

4 Related Work

There are a large body of literature about different variants of the vanilla GAN [Goodfellow *et al.*, 2014].

The Improved WGAN [Gulrajani *et al.*, 2017] utilizes the Wasserstein Distance as a loss indicator for the training progress, applies Gradient Penalty instead of weight clipping to enforce a Lipschitz constraint on the discriminator, and propose a few other minor modifications against the vanilla GAN to solve the problems of training difficulties and mode collapsing.

[Tolstikhin *et al.*, 2017] borrows the idea of AdaBoost [Freund and Schapire, 1997] and proposes AdaGAN, where at every step a new generator is added and trained on reweighted samples in an incremental procedure. [Hoang *et al.*, 2017] introduces MGAN where a mixture of generators are trained, mainly to solve the mode collapsing problem. In contrast, [Durugkar *et al.*, 2016] proposes GMAN, a framework that extends GAN to multiple discriminators. They explore two different roles of the discriminators, formidable adversaries or forgiving teachers, and evaluate their influences on the generator.

Other well-known variants include DCGAN [Radford *et al.*, 2015], CGAN [Mirza and Osindero, 2014], StackGAN [Zhang *et al.*, 2016], InfoGAN [Chen *et al.*, 2016], CycleGAN [Zhu *et al.*, 2017] and so on. These works either modify the network structures (DCGAN, StackGAN, CycleGAN) or introduce auxiliary features (CGAN, InfoGAN) in order to better control the details, generate images of higher qualities and resolutions, or conduct other tasks like image translation.

Different from the above works, our models focuses on ensemble via both adversaries and collaborators, where multiple generators and discriminators interact with each other and survive for optimizations. We maintain and update the authorities for these agents to enhance inter-side interaction, and utilize the mechanisms of imitation and innovation to model the intra-side interactions among the generators.

5 Conclusion and Future Work

In this paper, we propose the Generative Warfare Nets where multiple generators and multiple discriminators interact with each other and benefit from ensemble via both adversaries and collaborators. We propose several mechanisms to guarantee a comprehensive and competitive situation, including the agent authorities, the imitation and the innovation among the generators. Extensive experiments show that GWN can achieve competitive Inception scores and produce diverse high-quality synthetic images.

In future work, we would like to investigate more interactions among the generators and the discriminators, and explore other possible variants as the building block.

References

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

[Arora *et al.*, 2017] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*, pages 224–232, 2017.

[Ba *et al.*, 2016] Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[Berthelot *et al.*, 2017] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017.

[Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180, 2016.

[Coates *et al.*, 2011] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *ICAIS*, pages 215–223, 2011.

[Durugkar *et al.*, 2016] Ishan P. Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *CoRR*, abs/1611.01673, 2016.

[Freund and Schapire, 1997] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, pages 5769–5779, 2017.

[Hoang *et al.*, 2017] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Q. Phung. Multi-generator generative adversarial nets. *CoRR*, abs/1708.02556, 2017.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[Leen and Orr, 1993] Todd K. Leen and Genevieve B. Orr. Optimal stochastic search and adaptive momentum. In *NIPS*, pages 477–484, 1993.

[Mirza and Osindero, 2014] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[Nguyen *et al.*, 2017] Tu Nguyen, Trung Le, Hung Vu, and Dinh Q. Phung. Dual discriminator generative adversarial nets. In *NIPS*, pages 2667–2677, 2017.

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

[Rokach, 2010] Lior Rokach. Ensemble-based classifiers. *Artif. Intell. Rev.*, 33(1-2):1–39, 2010.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[Salimans *et al.*, 2016] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, pages 2226–2234, 2016.

[Tolstikhin *et al.*, 2017] Ilya O. Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. In *NIPS*, pages 5430–5439, 2017.

[Zhang *et al.*, 2016] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1612.03242, 2016.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.