

CDE-GAN: Cooperative Dual Evolution Based Generative Adversarial Network

Shiming Chen, Wenjie Wang, Beihao Xia, Xinge You, *Senior Member, IEEE*, Qinmu Peng, Zehong Cao, and Weiping Ding, *Senior Member, IEEE*

Abstract—Generative adversarial networks (GANs) have been a popular deep generative model for real-word applications. Despite many recent efforts on GANs have been contributed, however, mode collapse and instability of GANs are still open problems caused by their adversarial optimization difficulties. In this paper, motivated by the cooperative co-evolutionary algorithm, we propose a *Cooperative Dual Evolution based Generative Adversarial Network* (CDE-GAN) to circumvent these drawbacks. In essence, CDE-GAN incorporates dual evolution with respect to generator(s) and discriminators into a unified evolutionary adversarial framework, thus it exploits the complementary properties and injects dual mutation diversity into training to steadily diversify the estimated density in capturing multi-modes, and to improve generative performance. Specifically, CDE-GAN decomposes the complex adversarial optimization problem into two subproblems (generation and discrimination), and each subproblem is solved with a separated subpopulation (*E-Generators* and *E-Discriminators*), evolved by an individual evolutionary algorithm. Additionally, to keep the balance between *E-Generators* and *E-Discriminators*, we proposed a *Soft Mechanism* to cooperate them to conduct effective adversarial training. Extensive experiments on one synthetic dataset and three real-world benchmark image datasets, demonstrate that the proposed CDE-GAN achieves the competitive and superior performance in generating good quality and diverse samples over baselines. The code and more generated results are available at our project homepage <https://shiming-chen.github.io/CDE-GAN-website/CDE-GAN.html>.

Index Terms—Generative adversarial networks (GANs), evolutionary computation (EC), cooperative co-evolutionary algorithm, cooperative dual evolution, multi-objective optimization.

I. INTRODUCTION AND MOTIVATION

GENERATIVE adversarial networks (GANs) [1] are new popular methods for generative modeling, using game-theoretic training schemes to implicitly learn a given probability density. With the potential power of capturing high dimensional probability distributions, GANs have been successfully

This work was supported in part by the National Natural Science Foundation of China (61571205 and 61772220), the Key Program for International S&T Cooperation Projects of China (2016YFE0121200), the Special Projects for Technology Innovation of Hubei Province (2018ACA135), the Key Science and Technology Innovation Program of Hubei Province (2017AAA017), the Natural Science Foundation of Hubei Province (2018CFB691), fund from Science, Technology and Innovation Commission of Shenzhen Municipality (JCYJ20180305180804836 and JS GG20180507182030600). (Corresponding author: Xinge You.)

S. Chen, W. Wang, Beihao Xia, X. You and Q. Peng are with the Department of Electronic Information and Communication, Huazhong University of Science and Technology, Wuhan 430074, China. (e-mail:gchenshiming@gmail.com; {wangwj54, xbh_hust, youxg, pengqinmu}@hust.edu.cn).

Z. Cao is with the Discipline of ICT, University of Tasmania, TAS 7001, Australia (e-mail:zehong.cao@utas.edu.au).

W. Ding is with the School of Information Science and Technology, Nantong University, Nantong, China (ding.wp@ntu.edu.cn).

deployed for various synthesis tasks, e.g., image generation [2], video prediction [3], [4], text synthesis [5], [6], and image super resolution [7].

A GAN consists of a framework describing the interaction between two different models, i.e., generator(s) and discriminator(s), which are used to solve a min-max game optimization problem. During learning, generator tries to learn real data distribution by generating realistic looking samples that are able to fool discriminator, while discriminator attempts to differentiate between samples from the data distribution and the ones produced by generator. Although the pioneered GAN provided some analysis on the convergence properties of the approach [1], [8], it assumed that updates occurred in pure function space, allowed arbitrarily powerful generator and discriminator networks, and modeled the resulting optimization objective as a convex-concave game, therefore yielding well-defined global convergence properties. In addition, this analysis assumed that the discriminator network is fully optimized between generator updates, an assumption that does not mirror the practice of GAN optimization. In practice, however, there exist a number of well-documented failure models for GANs such as mode collapse¹ or instability².

Therefore, many recent efforts on GANs have contributed to overcome these optimization difficulties by developing various adversarial training approaches, i.e., modifying the optimization objective, training additional discriminators, training multiple generators, and using evolutionary computation. The first method is a typical way to control the optimization gradient of discriminator or generator parameters, which would help GANs to steadily arrive at the equilibrium points of optimization procedure under proper conditions [12]–[16]. Compared to the traditional GANs performed single-objective optimization, the second method revisited the multiple-discriminator setting by framing the simultaneous minimization of losses provided by different models as a multi-objective optimization problem. Thus it would overcome the problem involving the lack of useful gradient signal provided by discriminator [10], [17]–[20]. The third method simultaneously trains multiple generators with the object that mixture of their induced distributions would approximate the data distribution, and thus mode collapse problem can be alleviated [9], [21]–[23]. However, the aforementioned GAN methods are limited by the specified

¹the generator can only learn some limited patterns from the large-scale given datasets, or assigns all of its probability mass to a small region in the space [9]

²the discriminator is able to easily distinguish between real and fake samples during the training phase [10]–[12]

adversarial optimization strategy. The last method introduced evolutionary computation for the optimization of GANs to improve generative performance [6], [24]–[27]. In fact, the existing evolutionary GANs evolve a population of generators (e.g., E-GAN [25], CatGAN [6]) or GANs (Lipizzar [24], Mustangs [27]) to play the adversarial game, which will result that GANs evolve in a relatively static environment and the potential power of GANs is limited. In essential, training GANs is a large-scale optimization problem, which is challenging [20], [25]. This is why the most existing GAN methods usually fall in instability, and mode collapse. To overcome these challenges, we propose a novel insight for GANs, in which cooperative dual evolution paradigm is proposed for dynamic multi-objective optimization of a GAN.

Recently, evolutionary computation (EC), as a powerful method to complex real-word optimization problems, has been used to solve many deep learning challenges, e.g., optimizing deep learning hyper-parameters [28]–[32] and designing network architecture [33], [34]. Meanwhile, researchers also attempt to apply EC on GANs to improve the robust of GANs for mitigating degenerate GAN dynamics [6], [24]–[27]. Among them, Lipizzaner [24] and Mustangs [27] use a spatial distributed competitive co-evolutionary algorithm, to provide diversity in the genome space for GANs. In [25], Wang introduced E-GAN that injects diversity into the training with the evolutionary population. Accordingly, we attempt to train GANs using cooperative dual evolutionary paradigm, which is effective for large-scale optimization task and distribution diversity learning [35]–[42].

In light of the above observation, we propose a cooperative dual evolution based generative adversarial network, called *CDE-GAN*, to train model effectively and to improve generation performance steadily. In essence, CDE-GAN incorporates dual evolution with respect to generator(s) and discriminators into a unified evolutionary adversarial framework, thus it exploits the complementary properties and injects dual mutation diversity into training to steadily diversify the estimated density in capturing multi-modes. Our strategy towards achieving this goal is to decompose the complex adversarial optimization problem into two subproblems (generation and discrimination), and each subproblem is solved with a separated subpopulation (*E-Generators* and *E-Discriminators*), evolved by an individual evolutionary algorithm (including individual mutations and cooperative fitness functions). Unlike the existing EC based GANs [24], [25], [27], CDE-GAN simultaneously evolves a population of generators (*population*) and an array of discriminators (*environment*) by operating their various objective functions (*mutations*), which are interpretable and complementary. Intuitively, CDE-GAN integrates multi-discriminator and EC into a unified framework, which takes advantage of their complementary pros for adversarial optimization of GANs. During training E-Generators, acting as parents, generators of CDE-GAN undergo different mutations to produce offspring to adapt to the dynamic environment. Meanwhile, we term the quality and diversity of samples generated by the evolved offsprings as fitness score. According to the fitness score, poorly performing offspring are removed and the remaining well-performing offspring are preserved and used for next

generation training. Given current optimal generators, the similar mechanism holds for E-Discriminators with its own mutations and fitness function, and thus discriminators provide more promising gradient to generators for distribution diversity learning. To keep the balance between E-Generators and E-Discriminators, we proposed a *Soft Mechanism* to bridge them to conduct effective and stable adversarial training. In this way, CDE-GAN possesses three key benefits: 1) the cooperative dual evolution (E-Generators and E-Discriminators) injects diversity into training so that CDE-GAN can cover different data modes, which significantly mitigates mode collapse of GANs; 2) CDE-GAN terms adversarial training as a multi-objective dynamic optimization problem, the multiple discriminators provide informative feedback gradient to generators for stabilizing the training process; 3) the complementary mutations in E-Generators and E-Discriminators will help model place fair distribution of probability mass across the modes of the data generating distribution.

To summarize, this study makes the following salient contributions:

- We propose a novel GAN method, cooperative dual evolution based generative adversarial network (CDE-GAN). Benefiting from the proposed E-Generators, E-Discriminators and Soft Mechanism, CDE-GAN incorporates dual evolution with respect to generator(s) and discriminators into a unified evolutionary adversarial framework to overcome adversarial optimization difficulties of GANs, i.e., mode collapse and instability.
- We introduce suitable evaluation and selection mechanisms for E-Generators and E-Discriminators to improve the whole training process more efficient and stable, i.e., complementary mutations and interpretable fitness functions.
- We carry out extensive experiments on several benchmark datasets to verify our claims, and to show that our proposed method achieved obvious advantages over the existing methods.

The remainder of this paper is organized as follows. Section II gives related works in the field of generative adversarial networks. The proposed method CDE-GAN is illustrated in Section III. The performance and evaluation are provided in Section IV. Section V presents the discussion. Section VI provides a summary and the outlook for future research.

II. RELATED WORK

In the following, we provide a comprehensive review of GANs that developed various adversarial training approaches to overcome the optimization difficulties based on different methods, i.e., modifying training objective for GANs, multi-discriminator for GANs, multi-generator for GANs, evolutionary computation for GANs.

A. Modifying Training Objective for GANs

Modifying training objective of GANs is a typical way to improve and stabilize the optimization of GANs. Radford *et al.* [43] introduced DCGAN to improve training stability. In [12], Arjovsky not only proposed Wasserstein-GAN to minimize a reasonable and efficient approximation of the

Earth Mover (EM) distance for promoting the stability of training, he but also theoretically showed the corresponding optimization problem. Meanwhile, Metz *et al.* [13] introduced a method to stabilize GANs and increase diversity by defining the generator objective with respect to an unrolled optimization of the discriminator. Based on idea that gradient signals from Denoising AutoEncoder (DAE) can guide the generator towards producing samples whose activations are close to the manifold of real data activations, Denoising Feature Matching (DFM) is proposed to improve GANs training [14]. Served as a new normalization technique in GANs, SN-GAN called spectral normalization to stabilize the training of the discriminator and achieved promising generative performance [16]. In general, although some of these methods are practically and theoretically well-founded, convergence still remains elusive in practice.

B. Multi-discriminator for GANs

Unlike to the traditional GANs performed single-objective optimization, some works attempt to revisit the multiple discriminators setting by framing the simultaneous minimization of losses provided by different models as a multi-objective optimization problem. Thus it will overcome the problem of lacking of useful gradient signal provided by discriminator. Nguyen *et al.* [17] proposed D2GAN, combing the KL and reverse KL divergences into a unified objective function, to exploits the complementary statistical properties from these divergences to effectively diversify the estimated density in capturing multi-modes. Durugkar *et al.* [18] simultaneously introduced multiple discriminators into the GAN framework to weaken the discriminators of GMAN, which provides informative feedback to generator and better guides generator towards amassing distribution in approximately true data region. In [10], Neyshabur proposed an array of discriminators, each of which looks at a different random low-dimensional projection of the data, to play the adversarial game with a single generator. Thus the individual discriminators are fail to reject generated samples perfectly and provide meaningful gradients to the generator throughout training continuously. In [19], Doan argued that less expressive discriminators are smoother and have a general coarse grained view of the modes map, which enforces the generator to cover a wide region of the true data distribution. To this end, he proposed an on-line adaptative curriculum learning framework that trains the generator against an ensemble of discriminators. Albuquerque *et al.* [20] framed the training of multi-discriminator based GANs as a multi-objective optimization problem and analysed its effectiveness. Overall, although multi-discriminator based GANs perform promising results, they neglect to mine the prominence of generator.

C. Multi-generator for GANs

Multi-generator based GANs simultaneously trained multiple generators with the object that mixture of their induced distributions would approximate the data distribution, and thus mode collapse problem can be alleviated. Motivated by boosting method, Tolstikhin *et al.* [21] trained a mixture of generators by sequentially training and adding new generators to the mixture.

Arora *et al.* [9] introduced a MIX+GAN framework to optimize the minimax game with the reward function being the weighted average reward function between any pair of generator and discriminator. In [23], Ghost proposed MAD-GAN that trains a set of generators using a multi-class discriminator, which predicts which generator produces the sample and detecting whether the sample is fake. Additionally, MGAN [22] was developed to overcome the mode collapsing problem and its theoretical analysis was provided. Indeed, multi-generator based GANs break the balance of generator-discriminator and fail to explore the potential power of discriminator.

D. Evolutionary Computation for GANs

In fact, the aforementioned GAN methods are limited by the specified adversarial optimization strategy, which heavily limits optimization performance during training. Since EC has been successfully applied to solve many deep learning challenges [28]–[32], some researchers attempt to overcome different training problems of GANs using EC technique. In [24], Schmiedlechner proposed Lipizzaner, which provides population diversity by training a two-dimensional grid (each cell contains a pair of generator-discriminator) of GANs with a distributed evolutionary algorithm. Wang *et al.* [25] introduced E-GAN to inject mutation diversity into adversarial optimization of GANs by training the generator with three independent objective functions then selecting the resulting best performing generator for the next batch. Based on E-GAN, Liu *et al.* developed CatGAN with hierarchical evolutionary learning for category text generation. In [27], Toutouh proposed Mustangs, hybridizing E-GAN and Lipizzaner, to combine mutation and population approaches to diversity improvement of GANs. Costa *et al.* [26] developed COEGAN, using neuro-evolution and coevolution in the GAN training, to provide a more stable training method and the automatic design of neural network architectures. However, the aforementioned evolutionary GANs evolve a population of generators or GANs to play the adversarial game, which will result that GANs evolve in a relatively static environment and the potential power of GANs is limited. To this end, we propose a cooperative dual evolution based GAN, which simultaneously evolves a set of generators and an array of discriminators by operating different objective functions (mutations) to explore better balance between generator and discriminator of GANs.

III. PROPOSED METHOD

Motivated by the success of cooperative co-evolutionary algorithm in large-scale optimization and diversity learning [35]–[42], in this section, we propose a *Cooperative Dual Evolution based Generative Adversarial Network* (CDE-GAN) to circumvent drawbacks (i.e., instability and mode collapse) of GANs. In essence, CDE-GAN incorporates dual evolution with respect to generator and discriminator into a unified evolutionary adversarial framework, thus it exploits the complementary properties and injects dual mutation diversity into training to steadily diversify the estimated density in capturing multi-modes. Therefore, it will improve generative performance. As

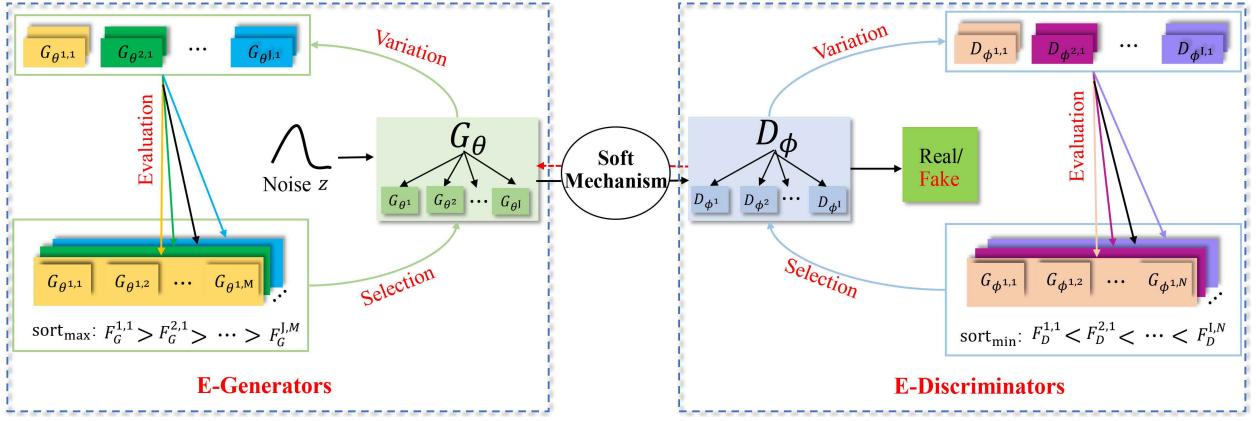


Fig. 1: The pipeline of CDE-GAN. In brief, CDE-GAN decomposes the complex adversarial optimization problem into two subproblems (generation and discrimination), and each subproblem is solved with a separated subpopulation (i.e., *E-Generators* and *E-Discriminators*), evolved by an individual evolutionary algorithm (including individual *Variations*, *Evaluations* and *Selections*). The best offspring of E-Generators and E-Discriminator are served as new parents to produce the next generation's individuals (i.e., children). Furthermore, a *Soft Mechanism* is proposed to cooperate E-Generators and E-Discriminators to conduct effective adversarial training.

shown in Figure 1, CDE-GAN decomposes the complex adversarial optimization problem into two subproblems (generation and discrimination), and each subproblem is solved with a separated subpopulation (i.e., *Evolutionary Generators* and *Evolutionary Discriminators*, which are termed as *E-Generators* and *E-Discriminators* respectively), evolved by an individual evolutionary algorithm. To this end, we first propose a *Soft Mechanism* to keep the balance between E-Generators and E-Discriminators and to cooperative them to conduct effective adversarial training. Then, we introduce E-Generators and E-Discriminators, including their own *Variations*, *Evaluations* and *Selections*.

A. Soft Mechanism

For the sake of keeping the balance between E-Generators and E-Discriminators, we proposed a *Soft Mechanism* to bridge them to conduct effective adversarial training. In practice, generator's learning will be impeded when it trains over a far superior discriminator. That is, generator is unlikely to generate any samples considered "realistic" according to the discriminator's standards, and thus the generator will receive uniformly negative feedback [18]. To this end, we use a *Soft Mechanism* of the classical Pythagorean method parameterized by δ to weaken the discriminators of CDE-GAN. Specifically, the generator(s) G_θ trains against a softmax weighted arithmetic average of I different discriminators, while each discriminator D_{ϕ^i} maximizes its own objective function. It can be formulated as

$$\min \mathcal{L}_{G_\theta}(\delta) = \sum_{i=1}^I w_i l_i, \quad (1)$$

where $w_i = \frac{e^{\delta l_i}}{\sum_{t=1}^I e^{\delta l_t}}$, $l_i = \mathcal{M}_G(D_{\phi^i})$, and $\delta \geq 0$. When $\delta \rightarrow \infty$, G_θ trains against the best discriminator (i.e., a single weak discriminator); when $\delta = 0$, G_θ trains against an ensemble with equal weights. These two scenes are not desired.

Equation 1 is a formulation of multi-objective optimization [44], thus CDE-GAN terms adversarial training as a multi-objective dynamic problem, and the multiple discriminators provide informative feedback gradient to generators for stabilizing the training process. According to [18], we set $\delta = 1$ for our all experiments.

B. E-Generators

In fact, the most existing GAN methods (e.g., modifying training objective based GANs, multi-discriminator based GANs and multi-generator based GANs) are limited by the specified adversarial optimization strategy, which heavily affects optimization performance during training. In addition, multi-generator based GANs simultaneously trained multiple generators with the object that mixture of their induced distributions would approximate the data distribution, and thus mode collapse problem can be alleviated. To this end, we build *E-Generators* that evolves a set of generators $\{G_\theta^1, G_\theta^2, \dots, G_\theta^J\}$ with number of J in a given dynamic environment (*E-Discriminators*) based on an evolutionary algorithm. E-Generators is a individual subpopulation to solved a subproblem (generation), which is cooperative with E-Discriminators to solve adversarial multi-objective optimization problem of GANs. Based on the success of EGAN [25], we take similar variations (mutations), evaluation function (fitness function) and selection for the evolution of E-Generators. The evolutionary process of E-Generators is presented in Algorithm 1.

CDE-GAN applies three complementary mutations corresponding with three different minimization objective w.r.t generator G_θ , i.e., G -Minimax mutation ($\mathcal{M}_G^{minimax}$), G -Heuristic mutation ($\mathcal{M}_G^{heuristic}$), and G -Least-Square mutation (\mathcal{M}_G^{ls}). They are corresponding to original GAN (GAN) [1], non-saturated GAN (NS-GAN) [1], and least square GAN (LSGAN) [45], [46], respectively. In contrast to mutations of E-GAN involving generator(s) against a single specified discriminator,

Algorithm 1 The algotithm of CDE-GAN.

Require: The generator G_θ ; the discriminator D_ϕ ; the batch size B ; the number of iterations T ; the discriminator's updating steps per iteration K ; the number of parents for generator J ; the number of parents for discriminator I ; the number of mutations for generator M ; the number of mutations for discriminator N ; Adam hyper-parameters α, β_1, β_2 ; the hyper-parameter γ of fitness function of generator.*

- 1: Initialize generator's parameter $\{\theta_0^1, \theta_0^2, \dots, \theta_0^J\}$, initialize discriminator's parameter $\{\phi_0^1, \phi_0^2, \dots, \phi_0^I\}$.
- 2: **for** $t = 1, \dots, T$ **do**
- 3: **for** $k = 1, \dots, K$ **do** ▷ E-Discriminators Evolution
- 4: **for** $i = 1, \dots, I$ **do**
- 5: **for** $n = 1, \dots, N$ **do**
- 6: Sample a batch of $\{z^{(b)}\}_{b=1}^B \sim p_{data}$.
- 7: Sample a batch of $\{z^{(b)}\}_{b=1}^B \sim p_z$.
- 8: $g_{\phi^{i,n}} \leftarrow \nabla_{\phi^i} \mathcal{M}_D^n(\{x^{(b)}\}_{b=1}^B, G_\theta(z), \phi^i)$
- 9: $\phi_{child}^{i,n} \leftarrow \text{Adam}(g_{\phi^{i,n}}, \phi^i, \alpha, \beta_1, \beta_2)$
- 10: Caculate $\mathcal{F}_D^{i,n}$.
- 11: **end for**
- 12: **end for**
- 13: $\{\mathcal{F}_D^{1,1}, \mathcal{F}_D^{2,1}, \dots, \mathcal{F}_D^{I,N}\} \leftarrow \text{sort}_{\min}(\{\mathcal{F}_D^{i,n}\})$
- 14: $\phi^1, \phi^2, \dots, \phi^I \leftarrow \phi_{child}^{1,1}, \phi_{child}^{2,1}, \dots, \phi_{child}^{I,1}$
- 15: **end for**
- 16: **for** $j = 1, \dots, J$ **do** ▷ E-Generators Evolution
- 17: **for** $m = 1, \dots, M$ **do**
- 18: Sample a batch of $\{z^{(b)}\}_{b=1}^B \sim p_z$.
- 19: $g_{\theta^{j,m}} \leftarrow \nabla_{\theta^j} \mathcal{M}_G^m(\{z^{(b)}\}_{b=1}^B, \theta^j)$
- 20: $\theta_{child}^{j,m} \leftarrow \text{Adam}(g_{\theta^{j,m}}, \theta^j, \alpha, \beta_1, \beta_2)$
- 21: $\mathcal{F}_G^{j,m} \leftarrow \mathcal{F}_{G^q}^{j,m} + \gamma \mathcal{F}_{G^d}^{j,m}$
- 22: **end for**
- 23: **end for**
- 24: $\{\mathcal{F}_G^{1,1}, \mathcal{F}_G^{2,1}, \dots, \mathcal{F}_G^{J,M}\} \leftarrow \text{sort}_{\max}(\{\mathcal{F}_G^{j,m}\})$
- 25: $\theta^1, \theta^2, \dots, \theta^J \leftarrow \theta_{child}^{1,1}, \theta_{child}^{2,1}, \dots, \theta_{child}^{J,1}$
- 26: **end for**

*Default values: $\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.99$, $B = 32$, $K = 3$, $M = 3$ and $N = 2$.

the mutations of E-Generators train over multiple evolutionary discriminators. The G-Minimax mutation is corresponding to the minimization objective of generator in original GAN [1], defined as

$$\mathcal{M}_G^{minimax} = \frac{1}{2} \sum_{i=1}^I w_i \mathbb{E}_{z \sim p_z} [\log(1 - D_{\phi^i}(G_\theta(z)))] \quad (2)$$

In fact, G-Minimax mutation is to minimize the Jensen-Shannon Divergence (JSD) between the data and model distributions. If discriminators distinguish generated samples with high confidence, the gradient tend to vanish, G-Minimax mutation fails to work; if discriminators cannot completely distinguish real/fake sample, the G-Minimax mutation will provide informative gradient for adversarial training. Thus, G-Minimax mutation typically evolves the best offspring in the latter training process for CDE-GAN. Additionally, G-Heuristic

mutation is non-saturating when the discriminator convincingly rejects the generated samples, and thus it avoids gradient vanish. It is formulated as follow:

$$\mathcal{M}_G^{heuristic} = -\frac{1}{2} \sum_{i=1}^I w_i \mathbb{E}_{z \sim p_z} [\log(D_{\phi^i}(G_\theta(z)))] \quad (3)$$

However, G-Heuristic mutation may direct to training instability and generative quality fluctuations because it pushes the data and model distributions away each other. As for G-Least-Square mutation, it is inspired by LSGAN [45], which applies this criterion to adapt both generator and discriminator. It can be written as

$$\mathcal{M}_G^{ls} = \sum_{i=1}^I w_i \mathbb{E}_{z \sim p_z} [(D_{\phi^i}(G_\theta(z)) - 1)^2] \quad (4)$$

Similar to G-Heuristic, G-Least-Square mutation will effectively avoid gradient vanish when the discriminator easily recognizes the generated samples. Meanwhile, G-Least-Squares mutation will partly avoid mode collapse, because it not assigns an extremely high cost to generate fake samples and also not assigns an extremely low cost to mode dropping. Thus, these different mutations provide various training strategy for E-Generators, which injects mutation diversity into training to diversify the estimated density in capturing multi-modes and constructs a complementary population of generators for steadily training. See [1], [25], [45], [47] for more theoretical analysis of these three objective functions.

After producing the next generation's children with different mutations, we evaluate the individual's quality for each child using a *fitness* function \mathcal{F}_G that depends on the current environment (i.e., discriminators D_ϕ). Considering two typical properties (quality³ and diversity⁴) of generated sample, \mathcal{F}_G consists of two fitness scores, i.e., *quality fitness scores* \mathcal{F}_{G^q} and *diversity fitness score* \mathcal{F}_{G^d} , for evaluating the generative performance of CDE-GAN. On the one hand, the generated samples produced by generator are feed into discriminators and the sum value of the output are calculated, which is termed as \mathcal{F}_{G^q}

$$\mathcal{F}_{G^q} = \sum_{i=1}^I w_i \mathbb{E}_{z \sim p_z} [D_{\phi^i}(G_\theta(z))] \quad (5)$$

The higher quality score generators achieved, the more reality generated samples gotten. Reflecting the quality performance of generators G_θ at each evolutionary step, discriminators D_ϕ are constantly upgraded to be optimal during the training process. On the other hand, we also focus on the diversity of generated samples and attempt to gather a better group of generators to circumvent the mode collapse issue in adversarial optimization of GAN. According to [50], a gradient-based regularization term can stabilize the GAN optimization and suppress mode collapse. To this end, the minus log-gradient-

³the generated samples are so realistic enough that it will fool the superior discriminator. [48]

⁴the model distribution is more possible to cover the real data distribution. It could largely avoid mode collapse. [49]

norm of optimizing D_ϕ is used to measure the diversity fitness score of generated samples

$$\begin{aligned} \mathcal{F}_{G^d} = & -\sum_{i=1}^I w_i (\log ||\nabla_{D_{\phi^i}} - \mathbb{E}_{z \sim p_{data}} [\log D_{\phi^i}(x)] \\ & - \mathbb{E}_{z \sim p_z} [\log(1 - D_{\phi^i}(G_\theta(z)))]||). \end{aligned} \quad (6)$$

When an evolved generator obtains a relatively high value, which corresponds to small gradients of D_ϕ , its generated samples tend to spread out enough and to avoid the discriminators from having obvious countermeasures. Therefore, \mathcal{F}_G is formulated as

$$\mathcal{F}_G = \mathcal{F}_{G^q} + \gamma \mathcal{F}_{G^d}, \quad (7)$$

$\gamma \geq 0$ is used for balancing quality and diversity of generated samples. Generally, a higher fitness score \mathcal{F}_G directs to higher training efficiency and better generative performance. Finally, a simple yet useful survivor selection strategy (μ, λ) -selection [51] is employed to select the next generation evolution according to the *highest* fitness score of existing individuals.

C. E-Discriminators

Recently, some works attempt to revisit the multiple discriminators setting by framing the simultaneous minimization of losses provided by different models as a multi-objective optimization problem [10], [17]–[19]. Thus it would overcome the problem of lacking of useful gradient signal provided by discriminators. To this end, we develop E-Discriminators, that evolves a population of discriminators, to provide meaningful gradient for E-Generators of CDE-GAN. In fact, E-Discriminators possesses two advantages to help CDE-GAN steadily achieve promising generative performance: 1) the evolutionary mechanism provides a dynamic strategy to discriminators, thus the trade-off between generator(s) and discriminators is well adjusted during training; 2) individual discriminators are unable to reject generated samples perfectly and continue to provide meaningful gradients to the generator throughout training. Specifically, given the optimal G_θ^* , the similar evolutionary mechanism of E-Generators holds for E-Discriminators with its own mutations and fitness function. The evolutionary process of E-Discriminators is presented in Algorithm 1.

Here, we take various objective functions as the mutations for E-Discriminators. To keep the objective functions of E-Discriminators corresponding to the objective functions of E-Generators, we take two different objective w.r.t discriminators D_ϕ , including *D-Minimax mutation* ($\mathcal{M}_D^{minimax}$), and *D-Least-Square mutation* (\mathcal{M}_D^{LS}). According to [1], D-Minimax mutation is a shared objective function for original GAN and NS-GAN, it can be formulated as

$$\begin{aligned} \mathcal{M}_D^{minimax} = & \mathbb{E}_{x \sim p_{data}} [\log D_{\phi^i}(x)] \\ & + \mathbb{E}_{z \sim p_z} [\log(1 - D_{\phi^i}(G_\theta(z)))] \end{aligned} \quad (8)$$

Indeed, D-Minimax mutation adopts the sigmoid cross-entropy loss for the discriminator, which will lead to the problem of vanishing gradients when updating the generator using the fake samples that are on the correct side of the decision

boundary, but are still far from the real data [46]. Specifically, the non-saturating loss $\mathcal{M}_G^{heuristic}$ will saturate when the input is relatively large; the minimax objective $\mathcal{M}_G^{minimax}$ will saturate when the input is relatively small. Thus, the generative performance of E-GAN [25] will be limited when it bases on single discriminator with minimax objective. In light of this observation, we further adopt least squares objective function for the second mutation of E-Discriminators:

$$\begin{aligned} \mathcal{M}_D^{LS} = & \frac{1}{2} \mathbb{E}_{x \sim p_{data}} [(D_{\phi^i}(x) - 1)^2] \\ & \frac{1}{2} \mathbb{E}_{z \sim p_z} [(D_{\phi^i}(G_\theta(z)))^2] \end{aligned} \quad (9)$$

Benefiting the least square objective function penalizes samples that lie in a long way on the correct side of the decision boundary, \mathcal{M}_D^{LS} is capable of moving the fake samples toward the decision boundary. To this end, \mathcal{M}_D^{LS} will help CDE-GAN generate samples that are closer to the real data. Overall, $\mathcal{M}_D^{minimax}$ and \mathcal{M}_D^{LS} are complementary to each other and will provide a promising optimization direction for the evolution of E-Discriminators, which effectively adjusts the trade-off between generator(s) and discriminators.

To evaluate the subpopulation evolved by E-Discriminators, we take the minus log-gradient-norm of optimizing D_ϕ as fitness function \mathcal{F}_D , which is corresponding to the diversity fitness score of generated samples in G_θ :

$$\begin{aligned} \mathcal{F}_D = & -\log ||\nabla_{D_{\phi^i}} - \mathbb{E}_{z \sim p_{data}} [\log D_{\phi^i}(x)] \\ & - \mathbb{E}_{z \sim p_z} [\log(1 - D_{\phi^i}(G_\theta(z)))]|| \end{aligned} \quad (10)$$

There are two reasons for this setting: 1) the gradient significantly reveals the train status of GANs. When the generator is able to generate realistic samples, the discriminators will not reject the generated sample confidently (i.e., D_ϕ updated with small gradient); when the generator collapses to a small region, the discriminator will subsequently label collapsed points as fake with obvious countermeasure (i.e., D_ϕ updated with big gradient); 2) the cooperative fitness function (Equation 6 and Equation 10) between E-Generators and E-Discriminators will keep the adversarial consistence of them, which will improve the training stability of CDE-GAN. Therefore, \mathcal{F}_D is effective for representing whether model falls in mode collapse, and thus it will guide E-Discriminators to evolve with the meaningful direction.

After evaluation, the new parents of evolution can be selected following the principle of survival-of-the-fittest, which is similar to the selection of E-Generators. Specifically, we first initialize I discriminators. In one training step, each existing discriminator i will evolve a set of children $\{D_{\phi^{i,1}}, D_{\phi^{i,2}}, \dots, D_{\phi^{i,N}}\}$ with different mutations, thus there are $I \times (N+1)$ individuals (including current parents and children) for competing. After sorting, I individuals $\{D_{\phi^1}, D_{\phi^2}, \dots, D_{\phi^I}\}$ possessing the *lowest* fitness score can be survived for next evolution during adversarial training.

IV. EXPERIMENTS AND EVALUATION

In subsequent sections, we first introduce the implementation details and hyper-parameter analysis of experiments. Furthermore, we qualitatively and quantitatively analyse the generative

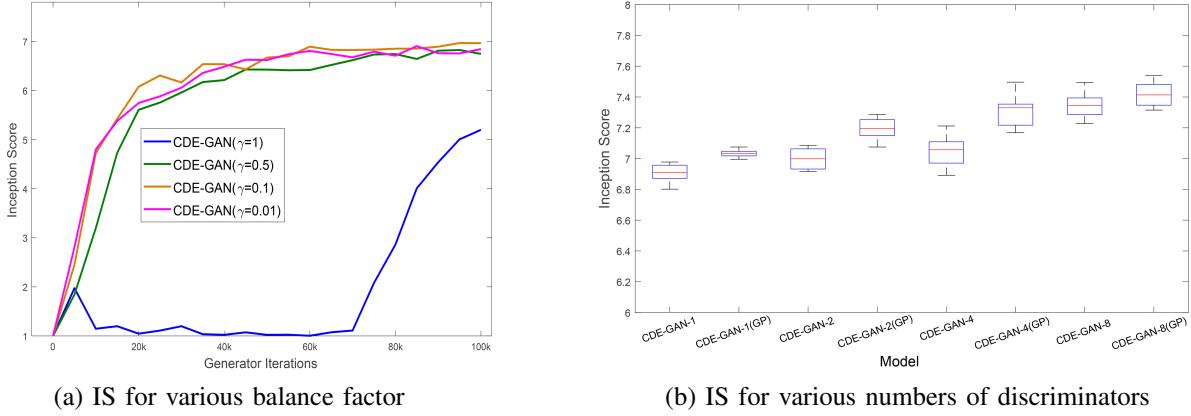


Fig. 2: Experiments on the CIFAR-10 dataset for hyper-parameters analysis. (a) Inception score evaluation for different CDE-GANs with various balance factor $\gamma = \{1, 0.5, 0.1, 0.01\}$. (b) Inception score evaluation for different CDE-GANs with various numbers of discriminators $I = \{1, 2, 4, 8\}$.

TABLE I: ARCHITECTURES OF THE GENERATIVE AND DISCRIMINATIVE NETWORKS USED IN THIS WORK, I.E., DCGAN MODEL, MLP WITH 3 LAYERS, AND MLP WITH 4 LAYERS.

Generative network	Discriminative network
DCGAN [15], [25], [43] Input: Noise $z \sim p_z$, 100 [layer 1] Fully connected and Reshape to $(4 \times 4 \times 512)$; ReLU; [layer 2] Transposed Convolution $(4, 4, 512)$, stride=2; ReLU; [layer 3] Transposed Convolution $(4, 4, 256)$, stride=2; ReLU; [layer 4] Transposed Convolution $(4, 4, 128)$, stride=2; ReLU; [layer 5] Transposed Convolution $(4, 4, 3)$, stride=2; Tanh; Output: Generated Image, $(32 \times 32 \times 3)$	Input: Image, $(32 \times 32 \times 3)$ [layer 1] Convolution $(4, 4, 128)$, stride=2; LeakyReLU; [layer 2] Convolution $(4, 4, 256)$, stride=2; LeakyReLU; [layer 3] Convolution $(4, 4, 512)$, stride=2; LeakyReLU; [layer 4] Fully connected (1); Sigmoid/Least squares; Output: Real or Fake (Probability)
MLP with 3 Layers [45], [46] Input: Noise $z \sim p_z$, 256 [layer 1] Fully connected (128); ReLU; [layer 2] Fully connected (128); ReLU; [layer 3] Fully connected (2); Linear; Output: Generated Point	Input: Point [layer 1] Fully connected (128); LeakyReLU; [layer 2] Fully connected (128); LeakyReLU; [layer 3] Fully connected (1); Sigmoid/Least squares; Output: Real or Fake (Probability)
MLP with 4 Layers [15] Input: Noise $z \sim p_z$, 256 [layer 1] Fully connected (128); ReLU; [layer 2] Fully connected (128); ReLU; [layer 3] Fully connected (128); ReLU; [layer 4] Fully connected (2); Linear; Output: Generated Point	Input: Point [layer 1] Fully connected (128); ReLU; [layer 2] Fully connected (128); ReLU; [layer 3] Fully connected (128); ReLU; [layer 4] Fully connected (1); Sigmoid/Least squares; Output: Real or Fake (Probability)

performance of CDE-GAN to verify our claims. Finally, we demonstrate advantages of our method over 16 state-of-the-art methods, including modifying training objective based GANs, multi-generator based GANs, multi-discriminator based GANs, and evolutionary computation based GANs.

A. Implementation Details

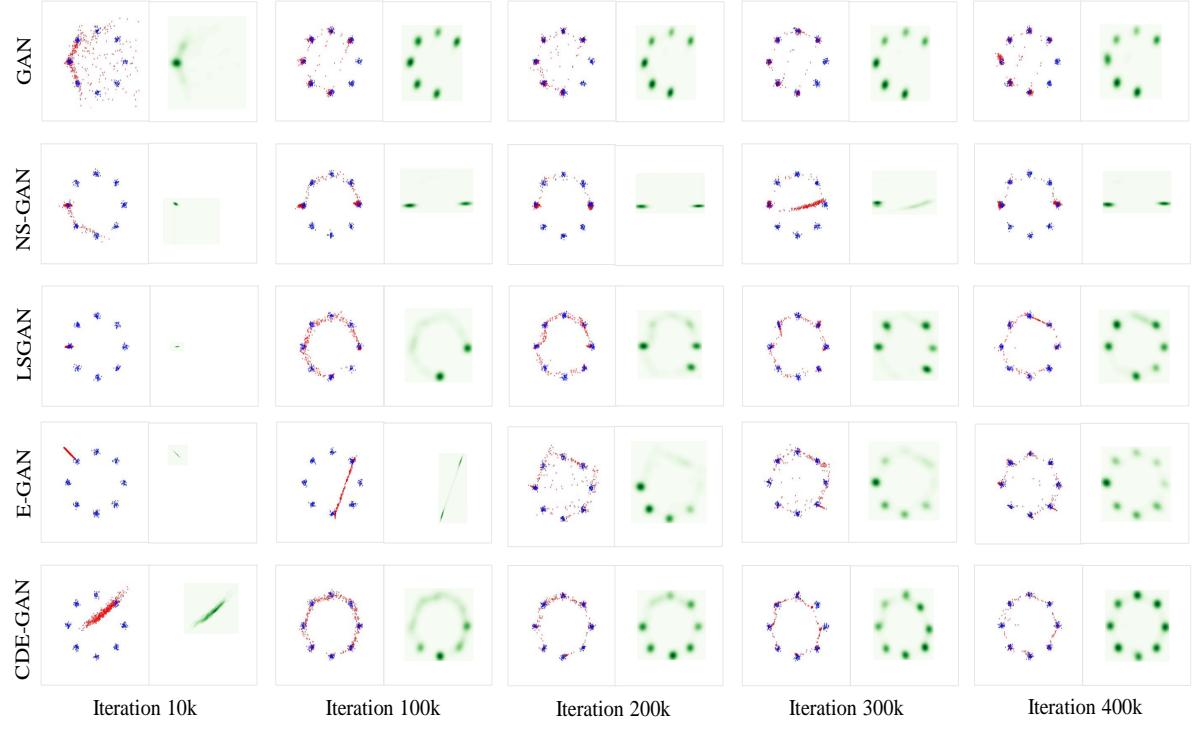
In the following experiments, we use the default hyper-parameter values listed in Algorithm 1. We conduct extensive experiments on one synthetic dataset (i.e., a mixture of 8 Gaussians arranged in a circle) and three real-word benchmark datasets (i.e., CIFAR10 [52], LSUN-Bedrooms [53] and CelebA [54]) to prove the effectiveness of CDE-GAN. Furthermore, we adopt the same network architectures (DCGAN) with existing in works [15], [25], [43] to conduct real data experiments for facilitating direct comparison. For the sake of fair comparison, we select MLP (with 3 layers [45], [46] or 4 layers [15]) as the model architecture of CDE-GAN to conduct toy experiments and generate 512 points to cover modes. The

model architectures are clearly displayed in Table I. The noisy vector z are sampled from uniform distribution p_z with 100 dimensions and 256 dimensions for real-word datasets and synthetic dataset, respectively. All experiments are performed on a single NVIDIA 1080Ti graphic card with 11GB memory. We use PyTorch for the implementation of all the experiments.

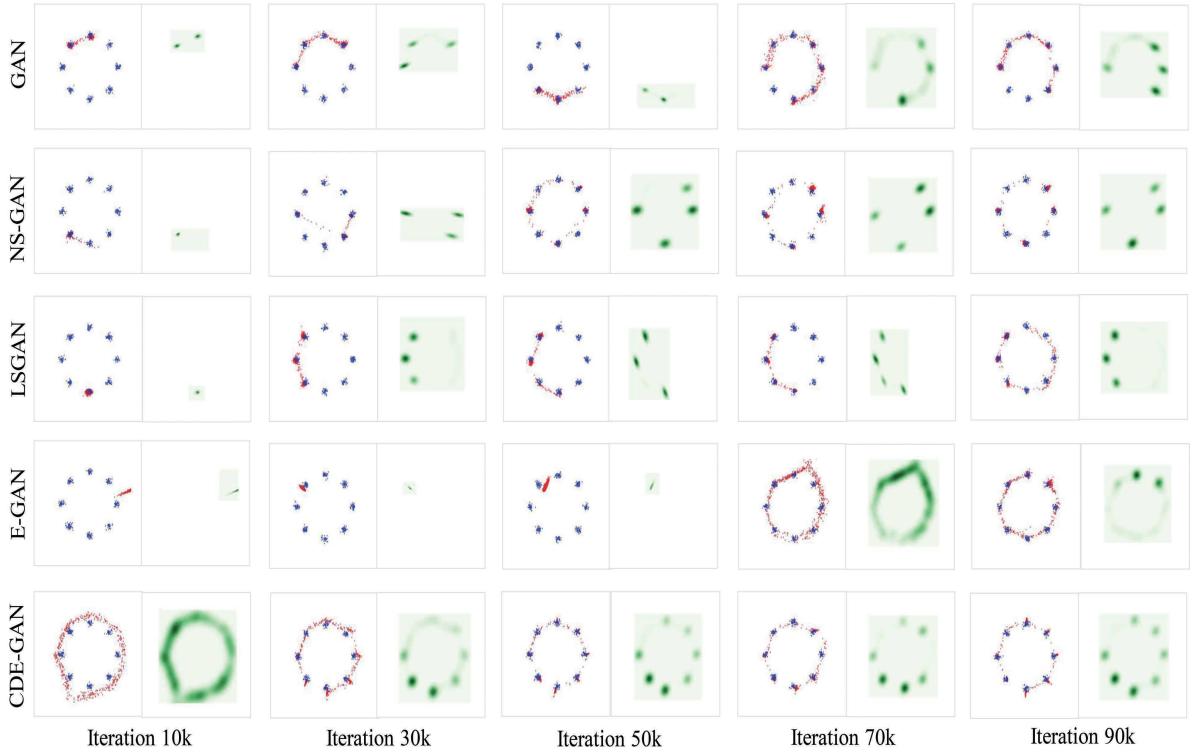
Additionally, we use inception score (IS) [48] to quantitatively evaluate the performance of the proposed method. It is a common quantitative evaluation metrics in image generation of GANs. The higher IS is got, the better quality of samples is generated. Note that IS is calculated by Tensorflow code version using randomly generated 50k samples in this paper. Meanwhile, we also qualitatively evaluate our method.

B. Experiment 1: Hyper-Parameters Analysis

In principle, there are two hyper-parameters closely corresponding to the performance of CDE-GAN, i.e., balance factor γ (see Equation 7) and number of discriminators I . Since the hyper-paramter γ is used for balancing the



(a) Gaussian kernel estimation with MLP of 3 layers



(b) Gaussian kernel estimation with MLP of 4 layers

Fig. 3: Dynamic results of Gaussian kernel estimation over generator iteration for different GANs. For each pair of images, the left one is data distribution (real data is represented in blue, generated data is represented in red), and the right one is KDE plots of the generated data corresponded to its left generated data. From top to bottom, the rows are the results of original GAN, NS-GAN, LSGAN, E-GAN, and CDE-GAN (Ours).

measurement of sample quality (\mathcal{F}_{G^q}) and diversity (\mathcal{F}_{G^d}), it directs generator selection of E-Generators and thus it will affect the effectiveness of CDE-GAN. Meanwhile, we analyze how the number of discriminators I affects the sample diversity of the corresponding generator and select proper discriminators for balancing time consuming and generative performance of CDE-GAN.

1) *Balance Factor γ :* In fact, quality and diversity of synthesis object are two key goals of generative task. Analogously, we also consider these two measurements for CDE-GAN evaluation in image generation task. Here, we embed a balance factor γ into fitness score of generator G_θ to balance the quality and diversity of generated samples during generator updation. Similar to [25], $\gamma > 0$ is considered. If γ is set as too small, the diversity fitness score is almost not considered during the training process; while γ is set too large, the model is not stable since the gradient-norm of discriminators D_ϕ could vary largely during training. To this end, $0 < \gamma \leq 1$ is considered for setting of our experiments.

To select a proper γ for CDE-GAN, we run grid search to find its value on CIFAR-10. As shown in Fig. 2(a), we take various balance factor $\gamma = \{1, 0.5, 0.1, 0.01\}$ to conduct experiments. Results show that CDE-GAN is out of work at the beginning and achieves convergence with slow speed, when γ is set as 1; while it gains promising generative performance and comparable convergence speed, if γ is set as a relative small value, i.e., 0.1 and 0.01. Based on these observations, we take $\gamma = 0.1$ to conduct later experiments on real-word datasets.

2) *Number of Discriminators I :* Multi-discriminator based GANs frame the training of GAN as multi-objective optimization problem, which will overcome the problem of lacking of meaningful gradient signal provided by discriminator. Here, we can also extend CDE-GAN as a multi-discriminator based GAN. Specifically, we take E-Discriminators to evolve multiple discriminators ($I \geq 1$) with various mutations. According to [18], [20], [55], the number of discriminators is closely corresponding to the diversity of generated sample. Therefore, we take experiments in aiming for analyzing how the number of discriminators affects the generative performance of CDE-GAN and selecting a promising I for later experiments.

In Fig. 2(b), we report the box-plots of inception score evaluation for different CDE-GANs with different numbers of discriminators on CIFAR-10 across 3 independent runs. Results clearly show that increasing the number of discriminators yields better generative performance for CDE-GAN. Note that CDE-GANs is more stable when more numbers of discriminator are survived for evolution. This is benefiting from effectiveness of multi-objective optimization methods [20]. Moreover, we convince that CDE-GAN can further improve generative performance if more discriminators are survived during training. To take trade-off between time consuming and efficacy, we conduct latter experiments with two discriminators for CDE-GAN, unless otherwise claimed.

C. Experiment 2: Generative Performance Evaluation

In this section, we qualitatively and quantitatively analyse the generative performance of CDE-GAN to support our

claims. Here, we take original GAN (GAN) [1], non-saturated GAN (NS-GAN) [1], and least square GAN (LSGAN) [45], [46], wasserstein GAN (WGAN) [12], and evolutionary GAN (EGAN) [25] as baselines for comparison and discussion, because these GAN models are closed to our method.

1) *Qualitative Evaluation:* Learning on a Gaussian mixture distribution to evaluate the diversity of GANs is a popular experiment setting, which intuitively reveals GANs whether suffer from mode collapse. When the model suffers from mode collapse problem, it will generate samples only around few of modes. To validate the effectiveness of our proposed method, analogously, we first qualitatively compare CDE-GAN with different baselines on synthesis dataset of 2-D mixture of 8 gaussian mixture distributions. For conducting a fair comparison, we adopt the experimental design proposed in [45], [46], which trains GANs with 3 layers MLP network architecture. Meanwhile, the survived parents number of E-Discriminators (I) and E-Generators (J) of CDE-GAN are set as 1, i.e., during each evolutionary step, only the best one candidature is kept. We train each method over 400k generator iterations with same network architecture. As shown in Fig. 3(a), the dynamic results of data distribution and the Kernel Density Estimation (KDE) plots on different baselines is reported. We can see that all of baselines only generate samples around a few of valid modes of the data distribution, i.e., 6 modes for original GAN and LSGAN, 2 modes for NS-GAN, 8 for E-GAN (parts of modes are weakly covered), which shows that they suffer from mode collapse to a greater or lesser degree. But CDE-GAN can successfully learn the Gaussian mixture distribution for all modes. These experiments demonstrate that the cooperative dual evolutionary strategy well circumvents the mode collapse of GANs training.

Furthermore, we also try 4 layers MLP network architecture to further conduct same experiment to evaluate the stability of CDE-GAN. In Fig. 3(b), The results show that all methods can speed up their convergence speed compared to CED-GAN with MLP of 3 layers. Notably, all baselines tend to generate few of modes, while CDE-GAN are less prone to this problem and achieve better convergence speed. It reveals that CDE-GAN possesses another advantage of architecture robustness.

For the sake of proving the potential power of generative performance of CDE-GAN, we show several samples generated by our proposed model trained on three datasets with 32×32 resolution in Fig. 4, i.e., CIFAR-10, LSUN-Bedrooms, and CelebA. Note that the presented samples are fair random drawn, not cherry-picked. It can be seen, on CIFAR-10, that our proposed CDE-GAN is capable of generating visually recognizable images of frogs, airplanes, horses, etc. CDE-GAN can produce bedrooms with various styles and different views, and bed and window in the rooms are clear displayed on LSUN-Bedrooms, and it is able to synthesize face images possessing various attributes (e.g., gender, age, expression, and hairstyle). These results confirm the quality and diversity of samples generated by our model.

2) *Quantitative Evaluation:* Our qualitative observations above are confirmed by the quantitative evaluations. For the sake of demonstrating the merits of the proposed CDE-GAN over the baseline methods, we train these methods on CIFAR-

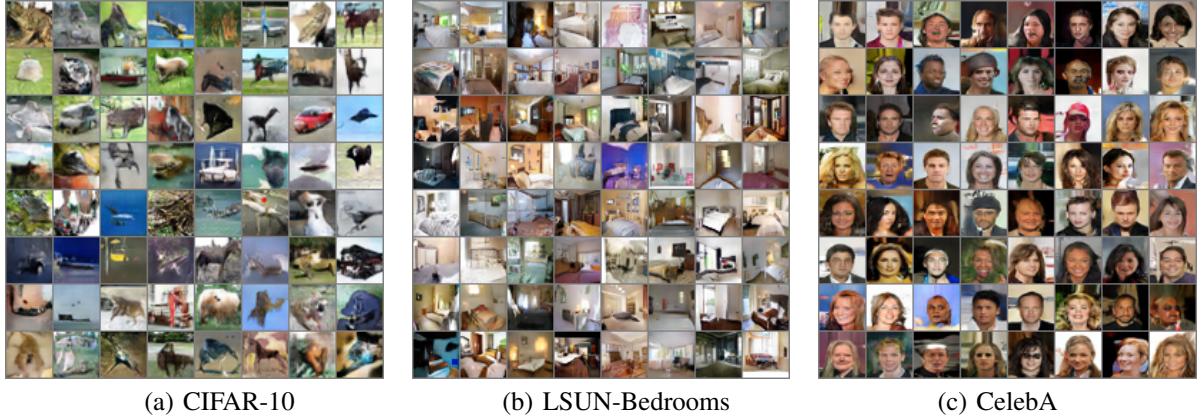


Fig. 4: Samples generated by our proposed CDE-GAN trained on various natural image datasets. Please see many more results in our [project homepage](#).

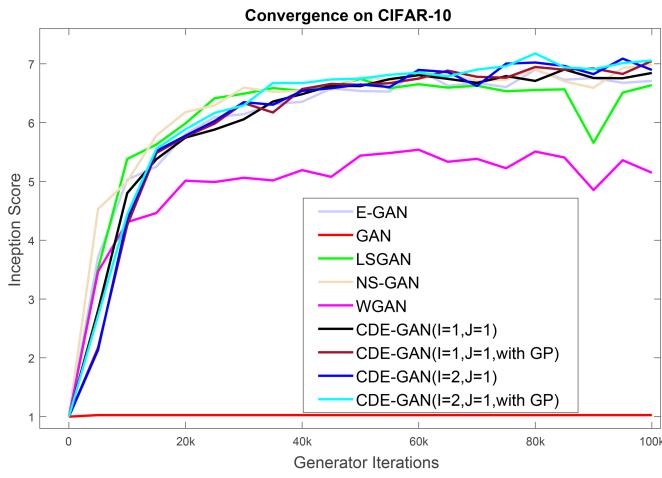


Fig. 5: Inception score of different GANs methods on CIFAR-10. I is the number of discriminators, J is the number of generators, and "with GP" denotes that gradient penalty (GP) is used.

10 and plot inception scores over the training process with same network architecture. As shown in Fig. 5, CDE-GAN can get higher inception score within 100k generator iterations. After 40k iterations, specifically, CDE-GANs with different setting consistently perform better over all the baselines. Meanwhile, the baseline methods fall in different training problems, e.g., instability at convergence (wgan and lsgan), invalid (GAN). This shows that cooperative dual evolution method is benefiting for adversarial training of GANs.

Since EGAN [25] is most similar to our method, we further take E-GAN as a baseline to compare with CDE-GAN for stability analysis. In Table II, we take various number of generators $\mu = \{1, 2, 4, 8\}$ for E-GAN and various number of discriminators $I = \{1, 2, 4, 8\}$ for CDE-GAN and train each model in 150k generator iterations to conduct experiments. Note that we implement results of E-GAN using the experimental setting of literature [25] and codes provided by authors⁵. Results show that CDE-GAN achieve better performance on CIFAR-

TABLE II: COMPARISON WITH E-GAN ON CIFAR-10 WITH OR WITHOUT GP. † [25]

Methods	Inception Score
E-GAN ($\mu = 1$, without GP)	6.88 ± 0.10
E-GAN ($\mu = 2$, without GP)	6.71 ± 0.06
E-GAN ($\mu = 4$, without GP)	6.96 ± 0.09
E-GAN ($\mu = 8$, without GP)	6.72 ± 0.09
E-GAN ($\mu = 1$, with GP)†	7.13 ± 0.07
E-GAN ($\mu = 2$, with GP)†	7.23 ± 0.08
E-GAN ($\mu = 4$, with GP)†	7.32 ± 0.09
E-GAN ($\mu = 8$, with GP)†	7.34 ± 0.07
(Ours) CDE-GAN ($I = 1$, without GP)	6.85 ± 0.05
(Ours) CDE-GAN ($I = 2$, without GP)	6.93 ± 0.09
(Ours) CDE-GAN ($I = 4$, without GP)	7.06 ± 0.09
(Ours) CDE-GAN ($I = 8$, without GP)	7.35 ± 0.06
(Ours) CDE-GAN ($I = 1$, with GP)	7.05 ± 0.05
(Ours) CDE-GAN ($I = 2$, with GP)	7.18 ± 0.05
(Ours) CDE-GAN ($I = 4$, with GP)	7.48 ± 0.10
(Ours) CDE-GAN ($I = 8$, with GP)	7.51 ± 0.05

10 than E-GAN totally. Furthermore, if E-GAN uses gradient penalty (GP) term during training, its generative performance is greatly improved compared to the results of its original version; while our CDE-GAN just achieves a little of improvement if GP is used. This reveals that E-GAN is unstable during training due to its single evolution, and thus GP term is effective for regularizing the discriminator to provide significant gradients for updating the generators. Benefiting from cooperative dual evolution, CDE-GAN injects diversity into training and thus it can covers different data modes. Furthermore, with the number of discriminator increasing, the balance of generator and discriminators of CDE-GAN is well adjusted and thus it continuously achieves obvious improvement. This shows that multi-objective optimization is effective for stabilizing the training process of CDE-GAN. To this end, the discriminators can continuously provide informative gradient for generator updation, which performs the function of GP.

D. Experiment 3: Comparisons With State-of-the-Art Methods

In this section, we compare CDE-GAN with 16 state-of-the-art GAN methods, including modifying training objective based GANs (GAN [1], DCGAN [43], WGAN [12], and SN-GAN [16]), multi-discriminator based GANs(D2GAN [17], GMAN

⁵<https://github.com/WANG-Chaoyue/EvolutionaryGAN-pytorch>

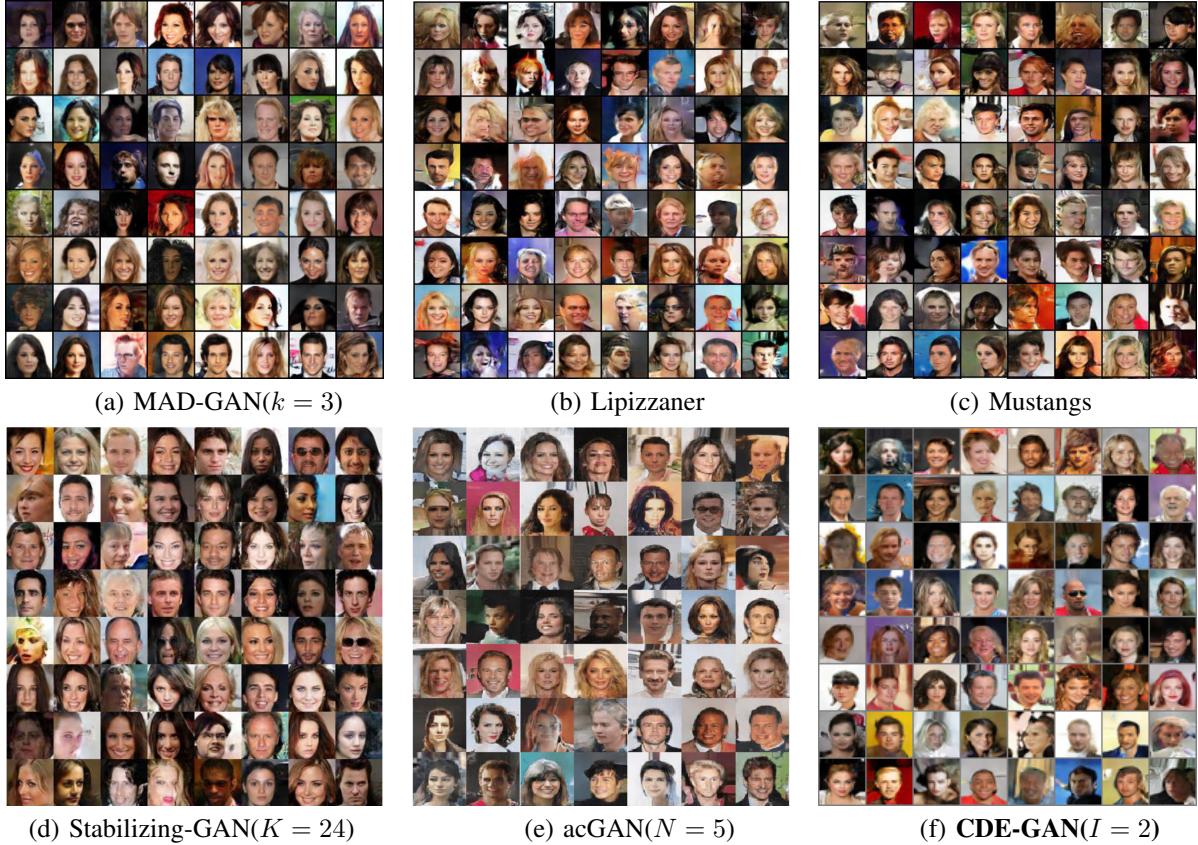


Fig. 6: Samples generated by different methods on various natural image datasets. The samples generated by different methods are provided by the original literatures, i.e., MAD-GAN [23], Lipizzaner [24], Mustangs [27], Stabilizing-GAN [10], and acGAN [19].

TABLE III: COMPARISON WITH STATE-OF-THE-ART METHODS ON CIFAR-10. THE N , K , T , μ , AND I REPRESENT THE NUMBERS OF DISCRIMINATORS OR GENERATORS FOR DIFFERENT METHODS, *Addit.* *Superv.* *Info.* DENOTES THAT ADDITIONAL SUPERVISED INFORMATION IS USED BY MEHTOD. THE BEST TWO RESULTS ARE MARKED WITH **BOLD**, AND UNDERLINE.

Methods	Addit. Superv. Info.	Inception Score
Real data		11.24 ± 0.12
GAN-GP [1]	\times	6.93 ± 0.08
DCGAN [43]	\times	6.64 ± 0.14
WGAN-GP [12]	\times	6.68 ± 0.06
SN-GAN [16]	\times	7.42 ± 0.08
GMAN ($N = 5$) [18]	\times	6.40 ± 0.19
D2GAN [17]	\times	7.15 ± 0.07
HV($K = 24$) [20]	\times	7.32 ± 0.26
MicroGAN ($K = 2$) [55]	\times	6.77 ± 0.00
MIX+WGAN ($T = 5$) [9]	\times	4.36 ± 0.04
MGAN ($K = 10$) [22]	\checkmark	8.33 ± 0.10
E-GAN ($\mu = 8$) [25]	\times	7.34 ± 0.07
(Ours) CDE-GAN ($I = 8$)	\times	7.51 ± 0.05

[18], Stabilizing-GAN [10], acGAN [19], HV [20], and MicroGAN [55]), multi-generator based GANs (MIX+WGAN [9], MGAN [22], MAD-GAN [23]), and evolutionary computation based GANs (Lipizzaner [24], E-GAN [25], and Mustangs

[27]).

Specifically, we first report the inception scores obtained by our CDE-GAN and baselines on CIFAR-10 in Table III. Overall, experimental results show that CDE-GAN outperforms almost all baselines on the optimal settings (except for MGAN). The reason is that MGAN takes *additional supervised information* (generators' labels) for supporting learning, while CDE-GAN is a completely unsupervised manner. Indeed, this technique is compatible with our method, thus integrating it could be a promising avenue for our future work.

It is worthy to note that CDE-GAN terms adversarial training as a multi-objective optimization problem, when its number of discriminator more than two, i.e., $I \geq 2$. Thus we further discuss the advantage of CDE-GAN compared with multi-discriminator based GANs. In the second group of Table III, all multi-discriminator based GANs using various numbers of discriminators are inferior to our CDE-GAN, i.e., GMAN ($K = 5$), D2GAN (2 discriminators), HV($K = 24$), MicroGAN ($K = 2$). This further demonstrates that cooperative dual evolution is effective for multi-objective optimization and adversarial training of GANs.

Finally, we display visual quality comparisons between several state-of-the-art methods (i.e., Stabilizing-GAN [10], acGAN [19], MAD-GAN [23], Lipizzaner [24], and Mustangs [27]) and our method on CelebA. See from Fig. 6, the most of face images generated by Lipizzaner and Mustangs are

incompleted or blurred, while our CDE-GAN generates high-fidelity face images. This demonstrates that the proposed cooperative dual evolutionary strategy performs better on adversarial training compared to other evolutionary computation based GANs. Furthermore, CDE-GAN also performs advantages over other GAN methods, i.e., MAD-GAN with 3 generators, Stabilizing-GAN with 24 discriminators, acGAN with 5 discriminators. Note that MAD-GAN trained its model using *additional supervised information* (generators' labels) and Stabilizing-GAN used the cropped version of the images of CelebA.

V. DISCUSSION

We would like to have more discussion here about the advantages and limitation of the proposed CDE-GAN method. First, we analyze the mechanism of how and why our method can circumvent mode collapse and instability problem of GANs.

- Compared to single evolution strategy, CDE-GAN injects dual diversity into training benefiting from the cooperative dual evolution. It decompose the complex adversarial optimization problem into two subproblems (i.e., generation and discrimination), and each subproblem is solved by a separated subpopulation (i.e., E-Generators and E-Discriminators), evolved by an individual evolutionary algorithm. In this way, the dual evolutionary population injects dual diversity into the training, and thus it can effectively covers different data modes. This significantly mitigates training pathologies of mode collapse for GANs. Experiments in Section IV-C and IV-D intuitively verified this claims.
- CDE-GAN terms adversarial training as a multi-objective dynamic optimization problem when $I \geq 2$, thus the multiple discriminators provide informative feedback gradient to generator for stabilizing training process. In ideal, we prefer that generator always has strong gradients from the the discriminator during training. Since the discriminator quickly learns to distinguish real and fake samples, however, the single-objective optimization based GANs make this difficult to ensure. To this end, they cannot provide meaningful error signal to improve the generator thereafter. In contrast, multi-objective optimization simultaneously minimizes the losses provided by different models to favor worse discriminators, thus providing more useful gradients to the generator. The experiments in Section IV-B2 and IV-C2 supported this conclusion.
- The complementary mutations in E-Generators and E-Discriminators support the model place fair distribution of probability mass across the modes of the data generating distribution. The complementary mutations are helpful for CDE-GAN to evolve in different possible directions during various training stages, which guides CDE-GAN to fairly cover the data distribution. As shown in Fig. 3(a), CDE-GAN effectively covers the 8 modes of gaussian distribution, while the compared baselines suffer from mode collapse to a greater or lesser degree. Section III-B and III-C provided more theoretical analysis for this merit.

Indeed, CDE-GAN limits in costing more time in each iteration. Theoretically, if we take updating a generator iteration

as an training step, CDE-GAN will cost $\mathcal{O}(K \cdot I \cdot N) + \mathcal{O}(J \cdot M)$ operations for one step (all notations are defined in Algorithm 1, and J is set as 1 in this paper). In practice, time consuming of CDE-GANs at each iteration are reported in Table IV. Specifically, to train a CDE-GAN model for 32×32 images using DCGAN architecture with different numbers of discriminators ($I = \{1, 2, 4, 8\}$), it will cost around 0.079 ± 0.003 , 0.079 ± 0.003 , 0.158 ± 0.021 , and 0.303 ± 0.002 seconds respectively for one generator iteration (excluding generating images for score test) on a single GPU. Since the cooperative dual evolutionary strategy is effective for adversarial training, however, CDE-GAN performed significantly fewer training steps to achieve the same generative performance compared to other baselines (see Fig. 5).

TABLE IV: TIME CONSUMING OF CDE-GANs AT EACH GENERATOR ITERATION.

Methods	Time/Iteration (Seconds)
CDE-GAN($I = 1$)	0.039 ± 0.000
CDE-GAN($I = 2$)	0.079 ± 0.003
CDE-GAN($I = 4$)	0.158 ± 0.021
CDE-GAN($I = 8$)	0.303 ± 0.002

VI. CONCLUSION

In this paper, we proposed a novel GAN (CDE-GAN), incorporating cooperative dual evolution with respect to generator(s) and discriminators into a unified evolutionary adversarial frameworks, to circumvent adversarial optimization difficulties of GANs, i.e., mode collapse and instability. Notably, the dual evolution provides a dynamic strategy to generator(s) and discriminators, exploits the complementary properties, and injects dual mutation diversity into learning, which diversify the estimated density in capturing multi-modes and improve generative performance of CDE-GAN. Additionally, we introduced a Soft Mechanism to balance E-Generators and E-Discriminators for conducting effective and stable training. The competitive results on one synthetic dataset (i.e., 2D mixture of 8 Gaussian dataset) and three real-world image datasets (i.e., CIFAR-10, LSUN-Bedrooms, and CelebA), demonstrate the superiority and great potentials of cooperative dual evolution for GANs. Extensive experiments also show that CDE-GAN performs obvious advantages over all the compared state-of-the-art methods.

In the future, we will futher improve CDE-GAN for speeding up learning. Meanwhile, we will also apply evolutionary computation based GANs to other generative tasks, e.g., text synthesis, video prediction.

VII. ACKNOWLEDGMENT

The authors would like to thank Dr. Chaoyue Wang at the School of Computer Science, University of Sydney, for his assistance with coding and theoretical suggestions.

REFERENCES

- [1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings Advances in Neural Information Processing Systems (NIPS)*, 2014.

- [2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251.
- [3] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing Motion and Content for Video Generation," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1526–1535.
- [4] J. Xie, Y. Lu, R. Gao, S. Zhu, and Y. Wu, "Cooperative training of descriptor and generator networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 27–45, 2020.
- [5] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings AAAI Conference on Artificial Intelligence*, 2017.
- [6] Z. Liu, J. Wang, and Z. Liang, "Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation," in *Proceedings AAAI Conference on Artificial Intelligence*, 2020.
- [7] C. Ledig, L. Theis, F. Huszár, J. A. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114.
- [8] I. J. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.
- [9] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," in *Proceedings International Conference on Machine Learning (ICML)*, 2017, pp. 224–232.
- [10] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti, "Stabilizing gan training with multiple random projections," *arXiv preprint arXiv:1705.07831*, 2017.
- [11] D. Berthelot, T. Schumm, and L. Metz, "Began: Boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:abs/1703.10717*, 2017.
- [12] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings International Conference on Machine Learning (ICML)*, 2017, pp. 214–223.
- [13] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," in *Proceedings International Conference on Learning Representations (ICLR)*, 2017, pp. 1–25.
- [14] D. Warde-Farley and Y. Bengio, "Improving generative adversarial networks with denoising feature matching," in *Proceedings International Conference on Learning Representations (ICLR)*, 2017, pp. 1–11.
- [15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proceedings Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5767–5777.
- [16] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proceedings International Conference on Learning Representations (ICLR)*, 2018, pp. 1–26.
- [17] T. D. Nguyen, T. Le, H. Vu, and D. Q. Phung, "Dual discriminator generative adversarial nets," in *Proceedings Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 2670–2680.
- [18] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," in *Proceedings International Conference on Learning Representations (ICLR)*, 2017, pp. 1–14.
- [19] T. Doan, J. Monteiro, I. Albuquerque, B. Mazoure, A. Durand, J. Pineau, and R. D. Hjelm, "Online adaptative curriculum learning for gans," in *Proceedings AAAI Conference on Artificial Intelligence*, 2019.
- [20] I. Albuquerque, J. Monteiro, T. Doan, B. Considine, T. H. Falk, and I. Mitliagkas, "Multi-objective training of generative adversarial networks with multiple discriminators," in *Proceedings International Conference on Machine Learning (ICML)*, 2019, pp. 202–211.
- [21] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "Adagan: Boosting generative models," in *Proceedings Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5424–5433.
- [22] Q. Hoang, T. D. Nguyen, T. Le, and D. Q. Phung, "Mgan: Training generative adversarial nets with multiple generators," in *Proceedings International Conference on Learning Representations (ICLR)*, 2018, pp. 1–24.
- [23] A. Ghosh, V. Kulharia, V. P. Namboodiri, P. H. S. Torr, and P. K. Dokania, "Multi-agent diverse generative adversarial networks," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8513–8521.
- [24] T. Schmiedlechner, A. Al-Dujaili, E. Hemberg, and U.-M. O'Reilly, "Towards distributed coevolutionary gans," in *Proceedings AAAI 2018 Fall Symposium*, 2018, pp. 1–6.
- [25] C. Wang, C. Xu, X. Yao, and D. Tao, "Evolutionary generative adversarial networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 921–934, 2019.
- [26] V. Costa, N. Lourenço, J. Correia, and P. Machado, "Coegan: Evaluating the coevolution effect in generative adversarial networks," in *Proceedings The Genetic and Evolutionary Computation Conference (GECCO)*, 2019, pp. 374–382.
- [27] J. Toutouh, E. Hemberg, and U.-M. O'Reilly, "Spatial evolutionary generative adversarial networks," in *Proceedings The Genetic and Evolutionary Computation Conference (GECCO)*, 2019, pp. 472–480.
- [28] Y. Sun, G. G. Yen, and Z. Yi, "Evolving unsupervised deep neural networks for learning meaningful representations," *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 89–103, 2019.
- [29] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "A particle swarm optimization-based flexible convolutional autoencoder for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 2295–2309, 2019.
- [30] Y. Sun, B. Xue, G. G. Yen, and M. Zhang, "Evolving deep convolutional neural networks for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 24, pp. 394–407, 2020.
- [31] Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang, "Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor," *IEEE Transactions on Evolutionary Computation*, vol. 24, pp. 350–364, 2020.
- [32] J. Toutouh, E. Hemberg, and U.-M. O'Reilly, "Re-purposing heterogeneous generative ensembles with evolutionary computation," in *Proceedings The Genetic and Evolutionary Computation Conference (GECCO)*, 2020, pp. 425–434.
- [33] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Completely automated cnn architecture design based on blocks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, pp. 1242–1254, 2020.
- [34] Y. Sun, B. Xue, G. G. Yen, and M. Zhang, "Automatically designing cnn architectures using genetic algorithm for image classification," *IEEE Transactions on cybernetics*, 2020.
- [35] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 103–127, 2009.
- [36] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, pp. 378–393, 2014.
- [37] S. He, G. Jia, Z. Zhu, D. A. Tennant, Q. Huang, K. Tang, J. Liu, M. Musolesi, J. K. Heath, and X. Yao, "Cooperative co-evolutionary module identification with application to cancer disease module discovery," *IEEE Transactions on Evolutionary Computation*, vol. 20, pp. 874–891, 2016.
- [38] X. Lu, S. Menzel, K. Tang, and X. Yao, "Cooperative co-evolution-based design optimization: A concurrent engineering perspective," *IEEE Transactions on Evolutionary Computation*, vol. 22, pp. 173–188, 2018.
- [39] X. Y. Zhang, Y. jiao Gong, Y. Lin, J. Zhang, S. Kwong, and J. Zhang, "Dynamic cooperative coevolution for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 935–948, 2019.
- [40] J. Zou, Q. Li, S. Yang, J. Zheng, Z. Peng, and T. Pei, "A dynamic multiobjective evolutionary algorithm based on a dynamic evolutionary environment model," *Swarm Evolution Computation*, vol. 44, pp. 247–259, 2019.
- [41] W. Ding, C.-T. Lin, and Z. Cao, "Deep neuro-cognitive co-evolution for fuzzy attribute reduction by quantum leaping pso with nearest-neighbor memeplexes," *IEEE Transactions on Cybernetics*, vol. 49, pp. 2744–2757, 2019.
- [42] Q. Zhang, S. Yang, S. Jiang, R. Wang, and X. Li, "Novel prediction strategies for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, pp. 260–274, 2020.
- [43] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings International Conference on Learning Representations (ICLR)*, 2016, pp. 1–15.
- [44] K. Deb, "Multi-objective optimization using evolutionary algorithms," vol. 16. Chichester, U.K.: Wiley, 2001.
- [45] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proceedings IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2813–2821.
- [46] X. Mao, Q. Li, H. Xie, Z. Wang, R. Y. K. Lau, and S. P. Smolley, "On the effectiveness of least squares generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2947–2960, 2019.

- [47] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proceedings International Conference on Learning Representations (ICLR)*, 2017, pp. 1–17.
- [48] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proceedings Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 2234–2242.
- [49] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6629–6640.
- [50] V. Nagarajan and J. Z. Kolter, "Gradient descent gan optimization is locally stable," in *Proceedings Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5591–5600.
- [51] O. Kramer, *Machine Learning for Evolution Strategies*. Cham, Switzerland: Springer, 2016, vol. 20.
- [52] A. Krizhevsky, "Learning multiple layers of features from tiny images." Department Computer Science, University Toronto, Toronto, ON, Canada, Report TR-2009, 2009.
- [53] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.
- [54] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," *Proceedings IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.
- [55] G. Mordido, H. Yang, and C. Meinel, "microbatchgan: Stimulating diversity with multi-adversarial discrimination," in *Proceedings Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 3050–3059.