

PAPER • OPEN ACCESS

Challenges and Corresponding Solutions of Generative Adversarial Networks (GANs): A Survey Study

To cite this article: Haiyang Chen 2021 *J. Phys.: Conf. Ser.* **1827** 012066

View the [article online](#) for updates and enhancements.

You may also like

- [Generating artificial displacement data of cracked specimen using physics-guided adversarial networks](#)

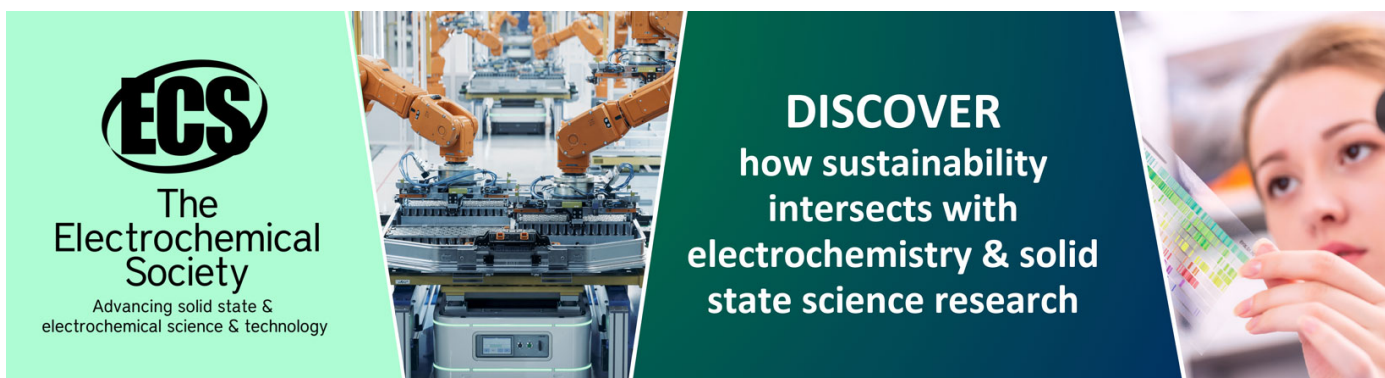
David Melching, Erik Schultheis and Eric Breitbarth

- [A review: the application of generative adversarial network for mechanical fault diagnosis](#)

Weiqing Liao, Ke Yang, Wenlong Fu et al.

- [Deblending Galaxies with Generative Adversarial Networks](#)

Shoubaneh Hemmati, Eric Huff, Hooshang Nayyeri et al.



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Challenges and Corresponding Solutions of Generative Adversarial Networks (GANs): A Survey Study

Haiyang Chen^{1, a}

¹School of Statistic and Management, Shanghai University of Finance and Economy, Shanghai 200433, China.

^ahaiyangchen@ mail.shufe.edu.cn

Abstract. Generative Adversarial Networks (GANs) are an innovative class of deep learning generative model that has been popular among academics recently. GANs are able to learn distributions on complex high-dimensional data which made it efficient in images and audio processing. Nevertheless, in the training of GANs, some major challenges exist namely mode collapse, non-convergence, and instability. In recent years, in order to overcome these challenges, researchers have proposed many variants of GANs by redesigning network architecture, changing the form of objective functions, and altering optimization algorithms. In this research, we conducted a comprehensive investigation on the progress of GANs design and optimization solutions. Finally, according to the classification method, we provided a problem-solving structure to solve conquer the GANs training challenges.

1. Introduction

Generative adversarial networks (GANs) are an innovative class of deep generative models that have been developed continuously over the past several years. It was first proposed in 2014 by Goodfellow as an alternative training methodology to generative model [1]. Since its birth, GANs have been used in a broad range of applications for their great performance in dealing with complex and high-dimensional data, for instance, computer vision, natural language, or other academic domains such as music generation or security.

The design of Generative Adversarial Networks is inspired by the theory of game. Generally, GAN is composed of two neural networks, a generator, and a discriminator. During adversarial training, generator and discriminator compete with one another to approach the Nash equilibrium. The core task of generator is to generate fake data through learning the potential probability distribution of the real data as much as possible. To be more specific, the discriminator gives the probability that the sample comes from the training set. Thus, a “perfect” discriminator should output the sample from the training set with probability 1 and output probability 0 for the generated sample from the generator. As for generator, it tries to capture the essential pattern of the training set, generates the sample, and sends the sample to the discriminator D ; it is attempted to “deceive” the discriminator so as to make the discriminator mistakenly believe that the sample comes from the training set and outputs probability 1. Through this adversarial training, GAN can generate samples that fit the real distribution without assuming a certain form of probability distribution explicitly.

Despite wide applications of GANs and continuous developments, the training of GANs is suffered from problems like mode collapse, instability, and non-convergence. Generally, researchers will innovate in the design of network architecture, the choice of loss function, and the use of optimization algorithms to achieve a better GANs architecture. To solve these problems, solutions have been



proposed. In this survey, we focus on two major problems in training, model collapse and non-convergence, and the latest solutions to these issues.

The remainder of this paper is organized as follows: The architecture, principle, and algorithms of Basic GANs are introduced in Section 2; sections 3 discusses how GANs have been improved; section 4 presents and describes two typical GAN variants; the applications where GANs have been widely employed are discussed in Section 5; and Section 6 concludes the survey.

2. Architecture

In the basic GAN architecture, the generator G takes the role of producing real-like fake samples from the noise vector z . The generated samples would be merged with real data as the input of Discriminator. The Discriminator calculates the probability that each sample comes from real data and determine whether it comes from G or real data. The result of D would be learned by G to decide the direction to optimize its parameters in the next iteration. G and D compete with each other in every round to attain their individual goals. This is exactly how adversarial training is performed in GANs. This adversarial training can be formulated as Equation (1) shown below,

$$\min_G \max_D V(G; D) = \min_G \max_D E_{\text{exp}_{\text{real}}} [\log D(x)] + E_{z \sim p_z} [\log (1 - D(G(z)))] \quad (1)$$

$V(G; D)$ is a double cross-entropy work that is regularly utilized for binary classification problems. From D 's perspective, thus the log part of Equation (1), if a sample is identified to be from real data, D will maximize its output; whereas, if a sample comes from G , D will minimize its output. Each generator and the discriminator have their own loss functions, as two individual players in a game theory, which was denoted as $J(G)$ and $J(D)$ respectively. In the first version's GANs [2], the discriminator D is defined as a binary classifier, and the loss function is represented by the cross entropy.

In Equation (2), x denotes the true data, z denotes the random noise vector, the fabricates data generated by the generator are $G(z)$, and E indicates the expectation. $D(x)$ is the probability calculated by D on whether x belonged to real data, and correspondingly $D(G(z))$ expresses the probability on x comes from generated data. The aim of D is to correctly discriminate where data come from, so it expects the value of $D(G(z))$ to be as close to 0 as possible. While the aim of G is to make $D(G(z))$ approach 1. Based on this adversarial concept, there exists a zero-sum game between these two networks. Through simplification, the loss function of the generator can be transformed to a form that derived form of the discriminator as the Equation (2) shows below.

$$J(G) = -J(D) \quad (2)$$

Through the training process, the parameter of G and D are updated simultaneously. Theoretically, this updating process ends when $G(z) = 0$, which means the generator can deceive the discriminator.

In this status, the model achieves the global optimal solution.

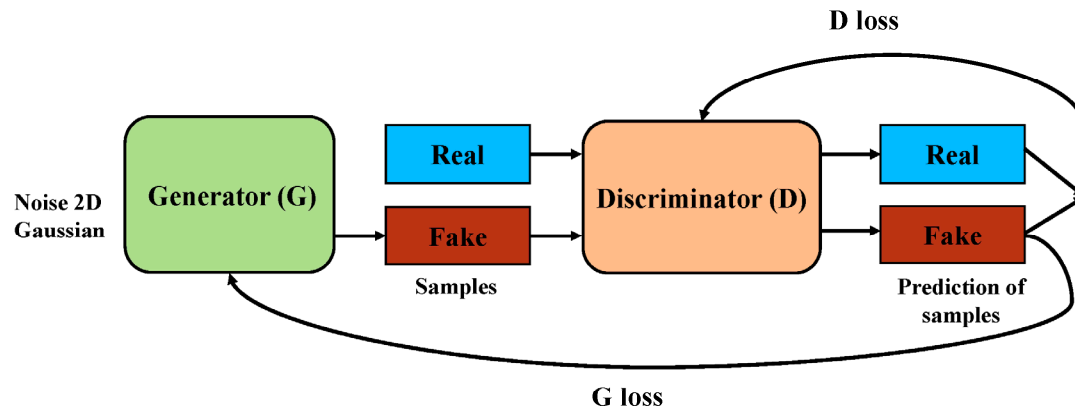


Figure 1. Basic GANs Architecture

3. The Variant GANs Models

In addition to basic GANs, researchers have proposed various variants GANs models and they are created to serve certain problems recognized by researchers with regard to basic GANs.

3.1 Model Collapse

Model collapse is one of the key reasons for instability of GANs training. Theoretically, it happens when the max-min solution to the GANs work couldn't coordinate with the min-max solution.

Figure 1 shows GAN training on a playground dataset where G generates only single mode instead of simulating multi-modes real data distribution. The generator continuously cycles between several single modes while D keeps refusing samples from G . In this process, GANs inflating from a single mode to another single mode never reach Nash equilibrium.

Generally, model collapse is the result of poor generalization of model. Mode collapse can be divided into two types: (1) most of the modes from the real data are absent from the generated data, (2) only a single mode is learned by G . The main reason for model collapse problem is the wrong selection of objective function.

To handle this issue, recent studies have introduced several variants with improved network architecture with new objective functions or alternative optimization algorithms.

3.1.1. Conditional Generation

An unconditional generative model cannot control the modes generation. To control the generation process, a generative model can be conditioned on additional information [3]. In this way, conditional GANs learn conditional probability distribution where a condition can be any auxiliary information about the data.

Because the generator input is a random noise vector z , an unconstrained input can cause a collapse of the learning mode. Therefore, Mirza and Osindero [4] introduced a conditional variable c (variable c can be any type of data like text or audio) in both the generator and the discriminator to conditions to the model using additional information to influence the process of data generation, that's how Conditional Generative Adversarial Networks (CGANs) Generative adversarial networks (CGANs) are proposed. In Figure 2, the conditional variable c and the noise vector z are the two inputs of the generator, and the inputs to the discriminator are $G(z|c)$ of the generator and real samples under the control of the same conditional variable c .

Moreover, InfoGAN, a variant of CGANs is proposed by Chen et al. [5]. The InfoGAN enable the whole process of generation to be more stable and make interpretability to be possible by introducing mutual information. To strengthen the connection between x and c , the network should solve the maximum value of mutual. Although it is similar to CGANs, but InfoGAN doesn't contain an initial known latent code that will be found in the training process. As an upgrade to the original GANs, InfoGAN designed an extra network Q for the output of conditional variables $Q(c|x)$.

On the basis of CGANs, Auxiliary Classifier GAN(ACGAN) was developed by Odena et al. [6]. In the perspective of discriminator, a conditional variable is replaced by a new classifier to display the probability over the class labels. Modification employed to the loss function is intended to increase the precision of class prediction.

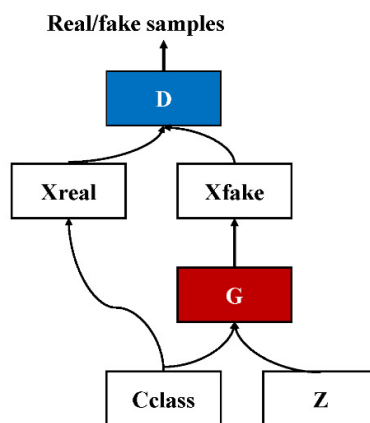


Figure 2. Architecture of Conditional GANs

4. Variant GANs Models

4.1. Non-convergence

In the traditional GANs, G uses two loss function as already introduced.

$$E_z[\log(D(G(z)))] \quad (3)$$

$$E_z[\log(1-D(G(z)))] \quad (4)$$

However, G 's loss can, unfortunately, prompt some potential issues in GANs training. The former loss function $E_z[\log(D(G(z)))]$ can bring about gradient vanishing problems. Especially when D gets a large learning rate that it can easily distinguish between real and fake samples. Under this circumstance, the learning ends when the generator is still weak, in other words, the discriminator learns too fast. For an optimal D , the optimization of G loss is similar to the minimization of the Jensen-Shannon Divergence (JSD) between a real distribution and generated distribution. The JSD will be $2\log 2$ in this case, which allows optimal D to give probability 1 to real samples, and 0 to fake ones and leads the gradient of G loss towards 0.

In GANs, D tries to minimize a cross-entropy while G tries to maximize the same cross-entropy. When D has a strong performance, D only accepts the real samples, and then G 's gradient vanishes. One available solution to eliminate this problem is to reverse the target label employed for the cross-entropy cost.

The second one is considered as the *logD* trick [7][8]. The minimization of the G 's loss function $Ez[\log(D(G(z)))]$ is equal to the minimization of equation (1), which results in unstable gradients as it minimizes the KL divergence and maximizes JSD simultaneously. This situation is called instability of G 's gradient updates. This figure also shows the growing variance of the gradients; such gradients updates will lead to a generation of low sample quality.

To cope with the non-convergence and instability problems, several GANs design and optimization solutions have been proposed. We shall discuss key solutions in the subsequent sections.

4.1.1. New probability distance

JS divergence (JSD) causes Original GANs training to unstable due to its discontinuity and absence of a usable gradient [9]. New probabilistic distances and divergence are needed to obtain usable gradients everywhere. To solve the model collapse problem, researchers often introduce a new definition of probability distances and divergence.

In this part, we are going to talk about several commonly used probability distance and divergence being used in learning distribution to increase the stability of training and solve the original GAN's mode collapse problem.

Arjovsky et al. [10] proposed a variant called Wasserstein Generative Adversarial Networks (WGAN) which made loss function have another role as a measure of convergence. WGAN has non-zero gradients everywhere and the implementation includes removing the sigmoid function in the objective and increasing the weight of D 's network. To improve the efficiency of GANs training and keep the training process settled, WGAN was developed to optimize the JSD with an outstanding approach of Earth-Mover (EM) distance [11].

The EM distance is continuous and differentiable. Therefore, it can effectively solve mode collapse problem through stabilizing GAN training. WGAN can train critics until it reaches the optimality that keeps the balance between G and D . Well-trained critics provide high-quality gradients for training G . However, when the gradient of the loss function is large, WGAN may be unstable. Therefore, too much weight will be clipped after each SGD update. Although WGAN supports stability and better mode coverage, its training speed is very slow. Moreover, adjusting weight clipping and hyperparameters is a tedious task.

WGAN [12] discussed the assumption that the model should have infinite capabilities which causes training problems in the basic GAN because it limits the model to be located in the Lipschitz space. For WGAN, the Lipschitz condition comes from the Kantorovich-Rubinstein duality, and only critics are bound. Loss-sensitive GAN (LS-GAN) [13] uses weight decay regularization technology to limit the weight of the model in a bounded area to meet the condition of the Lipschitz function.

Guo et al. [14] proposed that a new statistical divergence could be implemented into GANs. In large-scale calculations, Relaxed Wasserstein (RW) divergence is a fair choice whose parameters are determined by a series of strictly convex and differentiable functions containing different curvature information. RWGAN is more powerful and faster than WGAN, introduced f-GAN to minimize the change estimation of f-divergence [15], in which training D is formulated as a density ratio estimation. The purpose of f-GAN is to find the minimum of the difference between the distribution of the real data and the generated data.

5. Application

As a generative model, generating data is GANs' fundamental use, that is, to generate fake samples with the distribution learned from real samples. This section will discuss the main applications of GANs, including the application in computer vision, natural language processing, and other domains.

5.1. Computer Vision

Currently, the best application area of GANs is computer vision, involving image enhancement, image transformation, image compounding, and video production. These applications will be talked about specifically as follows.

5.1.1. Image Resolution Enhancement

For the purpose to enhance the resolution of image, Ledig [16] developed a Super-Resolution Generative Adversarial Network (SRGAN). It reads a low-resolution image and generates as input data to produce an image with 4 times high-resolution image. A year after, Wang proposed the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) [17] which uplifts the realness of the texture information generated by SRGAN and reducing noise by redesigning the structure of the network and formats of loss function.

5.1.2. Image Translation

In order to transform image content between multiple fields, Isola proposed an image-to-image conversion method using Conditional GAN called pix2pix[18]. Experiments show that pix2pix is effective in graphics tasks as well as visual tasks. In the following versions, pix2pixHD keeps improving the quality and clarity of the generated images. Using an innovative anti-loss term, this method produces a high-quality image with a resolution of 2048×1024. Despite Pix2pix's strong performance in image conversion problems, its training space is strictly paired in X and Y space. Nevertheless, such pairing data is rare in real life. Under such circumstances, CycleGAN [19], DiscoGAN [20], and DualGAN [21] are three encoder-decoder frameworks established on the idea of cyclic consistency, which made train the mapping from X space to Y space on unpaired data to be feasible.

5.2. Natural Language Processing

At this moment, GAN has also been an important technology in the field of language and speech processing. SeqGAN, a strategy gradient-based network outperforming most traditional methods in lecture, literature, and music generation was proposed by Yu et al. [22]. Lin et al. [23] used a ranker as a substitute for a distinguisher and developed RandkGAN which accomplish great performance. Li et al. [24] also apply adversarial training methods to the generation of open-domain conversation. Regarded as a reinforcement learning problem, this task train the generator and discriminator at the same time. The output of the discriminator is used as the reward part of reinforcement learning to reward the generator, and the conversation generated by the generator is so good that it's like two real people having a conversation.

5.3. Other Domains

GANs are also widely used in many fields. In medical treatment, Schlegl et al. [25] proposed an AnoGAN for detecting abnormality in scanning image abnormality, and summarize the features of lesions by training on health data sets. Killoran et al. [26] used GAN to optimize protein binding in DNA sequence generation. In the cybersecurity field, Hu and Tan [27] applied GAN to identify malware.

6. Conclusion

In recent years, GANs have attracted widespread attention in producing lifelike realistic images and have become very popular in the scientific world. Applications of GANs have seen astounding growth, for instance, in speech generation, cybersecurity. Nevertheless, GANs are difficult to train, and training faces two major problems, namely mode collapse, and non-convergence. One feasible method to make GAN solve these two challenges is to redesign the network architecture to get a more powerful model. Changing a suitable objective function, or choosing appropriate optimization algorithms. In recent years, many different GAN variants with different characteristics have been proposed according to these solutions.

The research on GAN is very extensive, and some GAN designs and training solutions for these challenges are also been introduced in previous findings. In this article, we describe the original GAN framework and investigate the development of variants to better design and optimize GANs. Moreover, we summarize a novel taxonomy of GAN variants in these years and classify them by optimization techniques and discuss how existing work addresses these challenges. Our work attempts to provide a panoramic view of current progress and a comprehensive survey of the methods. Based on the new

taxonomy, a problem-solving structure is introduced that researchers can continue to improve in the future.

References

- [1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- [2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- [3] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.
- [4] Qi, G.-J. (2020). Loss-sensitive generative adversarial networks on lipschitz densities. *International Journal of Computer Vision*, 128(5):1118–1140.
- [5] (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180.
- [6] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier GANs,” in Proc. 34th Int. Conf. Mach. Learn., vol. 70, Aug. 2017, pp. 2642–2651. [Online]. Available: <http://proceedings.mlr.press/v70/odena17a.html>
- [7] Ghosh, A., Kulharia, V., and Namboodiri, V. (2016). Message passing multi-agent gans. *arXiv preprint arXiv:1612.01294*.
- [8] Ghosh, A., Kulharia, V., Namboodiri, V. P., Torr, P. H., and Dokania, P. K. (2018). Multi-agent diverse generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8513–8521.
- [9] Jaiswal, A., AbdAlmageed, W., Wu, Y., and Natarajan, P. (2018). Bidirectional conditional generative adversarial networks. In *Asian Conference on Computer Vision*, pages 216–232. Springer.
- [10] Martin Arjovsky, S. and Bottou, L. (2017). Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia*.
- [11] Hoang, Q., Nguyen, T. D., Le, T., and Phung, D. (2017). Multi-generator generative adversarial nets. *arXiv preprint arXiv:1708.02556*.
- [12] Mátyus, G. and Urtasun, R. (2018). Matching adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8024–8032.
- [13] Mescheder, L., Nowozin, S., and Geiger, A. (2017). Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*.
- [14] Guo, X., Hong, J., Lin, T., and Yang, N. (2017). Relaxed wasserstein with applications to gans. *arXiv preprint arXiv:1705.07164*.
- [15] Nguyen, T., Le, T., Vu, H., and Phung, D. (2017). Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2670–2680.
- [16] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690.
- [17] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, page(0-0)
- [18] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).

- [19] Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279.
- [20] Perarnau, G., Van De Weijer, J., Raducanu, B., and Álvarez, J. M. (2016). Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*.
- [21] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- [22] Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017, February). Seqgan: Sequence generative adversarial nets with policy gradient. In Thirty-first AAAI conference on artificial intelligence.
- [23] Lin, Z., Khetan, A., Fanti, G., and Oh, S. (2020). Pacgan: The power of two samples in generative adversarial networks. *IEEE Journal on Selected Areas in Information Theory*, 1(1):324–335.
- [24] Arora, S., Ge, R., Liang, Y., Ma, T., and Zhang, Y. (2017). Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*.
- [25] Rubner, Y. (1998). The earth mover’s distance as a metric for image retrieval. Technical report, Tech Rep. Stanford Univ.
- [26] Killoran, N., Lee, L. J., DeLong, A., Duvenaud, D., & Frey, B. J. (2017). Generating and designing DNA with deep generative models. *arXiv preprint arXiv:1712.06148*.
- [27] Hu, W., & Tan, Y. (2017). Generating adversarial malware examples for black-box attacks based on gan. *arXiv preprint arXiv:1702.05*