Journal of Physics:
Conference Series

PAPER • OPEN ACCESS

# Model Extraction Attack and Defense on Deep Generative Models

To cite this article: Shengyi Liu 2022 *J. Phys.: Conf. Ser.* **2189** 012024

View the article online for updates and enhancements.

You may also like

- A synchronization correction scheme based on random forest and GGD distribution for color image watermarking
Fu-ya Du, Xiu-fang Feng, Chun-peng Wang et al.
- Effect of vacuum on microwave extraction of hawthorn fruit
L Yu Ismailov, R R Safin and A V Safina
- Analytical and simulation studies for diode and triode ion beam extraction systems
M. M. Abdelrahman, N. I. Basal and S. G. Zakhary

Enabling Electrochemistry
Inside and Outside the Lab

PalmSens

Discover our diverse range of Potentiostats engineered to support your research wherever it takes you

- Intuitive software for Windows and Android
- Proven in over 4000 scientific publications
- Single-channel and multi-channel
- Benchtop, portable and wearable

Discover More




This content was downloaded from IP address 216.73.216.56 on 19/11/2026 at 09:06

# Model Extraction Attack and Defense on Deep Generative Models

**Shengyi Liu**[1]

[1] School of Mechanical Electrical and Information Engineering, Shandong University, Weihai, Shandong, 264209, PR China

201700800442@mail.sdu.edu.cn

**Abstract.** The security issues of machine learning have aroused much attention and model extraction attack is one of them. The definition of model extraction attack is that an adversary can collect data through query access to a victim model and train a substitute model with it in order to steal the functionality of the target model. At present, most of the related work has focused on the research of model extraction attack against discriminative models while this paper pays attention to deep generative models. First, considering the difference of an adversary` goals, the attacks are taxonomized into two different types: accuracy extraction attack and fidelity extraction attack and the effect is evaluated by 1-NN accuracy. Attacks among three main types of deep generative models and the influence of the number of queries are also researched. Finally, this paper studies different defensive techniques to safeguard the models according to their architectures.

## 1. Introduction

The last decade has witnessed a rapid boom in the development of machine learning and especially deep learning. In order to enable users to take advantage of the technique, companies provide interfaces to the models they have trained and deployed online for the users to query. However, there is a potential security problem. Malicious users could query these black-box models and collect the output to make a dataset. Then the adversary may choose a substitute model and train it on the data they collect to 'steal' the model that has the same function as the original one. Since a model with high performance requires a lot of resources, the attack may lead to huge losses for the providers.

Researchers have done much work on this type of attack. Tramèr et al. (2016)[1] proposed this concept of model extraction attack and their work experimented with simple machine learning models. Papernot et al.[2] studied an attack strategy using Jacobian-based dataset augmentation to get a substitute model and validated the method by generating adversarial examples with it. Shi et al.[3] utilized generative adversarial network (GAN) to generate synthetic data from a limited amount of real training data and improved the performance of extraction. Jagielski et al.[4] then taxonomized model extraction attack into accuracy extraction and fidelity extraction according to an adversary`s goal.

Protecting the victim model from model extraction attack is also very necessary and thus has been studied by some researchers. Tramèr et al.[1] proposed rounding confidence values provided is useful. Kesarwani et al.[5] proposed an extraction monitor to observes uses` queries. Lee et al.[6] devised an add-on layer for neural network in classification task to perturb the output. Juuti et al.[7] distinguished an adversary from normal users by analyzing the distribution of samples submitted by them.

Despite the deep researches and the various approaches, the researchers mainly focused on discriminative tasks. As for generative models, Hu and Pang[8] first studied model extraction between

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

GANs and proposed countermeasures. However, there are still various types of generative models totally different from GANs left to research. In this work, besides GAN, we will also explore model extraction attack and defense on other types of deep generative models that have never been studied before.

## 2. Preliminaries

In this section, we introduce the background knowledge including different types of deep generative models, how to execute extraction on them, and the metric we use to evaluate the performance of extraction.

### *2.1. Taxonomy of generative models*

In essence, deep generative models aim to minimize the difference between the distribution of real data and that of the data generated by it. According to Goodfellow[9], these models can be classified into explicit density models and implicit density models. The former defines an explicit density to optimize while the latter does not. Further, if the density of an explicit density model is computationally tractable, it will be called tractable explicit model and the others will be regarded as explicit models requiring approximation. The present paper will work on the representative of each type of generative models which are introduced below.

*2.1.1. PixelRNN/PixelCNN.* These two models are tractable explicit models and were proposed by van den Oord et al. (2016)[10]. For an $n \times n$ image $X$, when taking pixels row by row, the pixels can be written as $x_1,\ldots,x_{n^2}$. Given all the previous pixels $x_1$, ..., $x_{i-1}$, the probability of the i-th pixel $x_i$ is $p(x_i|x_1,\ldots,x_{i-1})$. Then the joint distribution $p(X)$ is computed as: $p(X) = \prod_{i=1}^{n^2} p(x_i|x_1,\ldots,x_{i-1})$. And the goal is to maximize this likelihood. The most significant advantage of this design is that the model can output the probability of a generated image.

*2.1.2. Variational autoencoder.* Kingma and Welling[11] proposed Variational Autoencoder (VAE) which is an explicit model requiring approximation. The approach is first, inputting an image $x$ into an encoder to get a posterior distribution $q(z|x)$. Then a latent vector $z$ will be sampled from the distribution and input into a decoder to receive the reconstruction of the original image. The loss function is defined as: $Loss = KL(q(z|x)||p(z)) - E_{q(z|x)}[\log p(x|z)]$. The first part is the KL-divergence between $q(z|x)$ and $p(z)$, the prior distribution of latent vector $z$. The second item is the reconstruction loss between the generated image and the original one.

*2.1.3. Generative adversarial nets.* GAN was proposed by Goodfellow et al.(2014)[12]. GAN consists of two parts which are trained alternately: a generator $G$ aiming to generate an image given a latent vector $z$, and a discriminator $D$ that can distinguish between generated data and real data. In the game process, the generator can produce more authentic images while the discriminator will also be more discriminative. Ideally, there will be a balance between the two networks. The value function is $\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[logD(x)] + E_{z\sim P_z(z)}\left[\log\left(1 - D(G(z))\right)\right]$. $x$ is real data and sampled from distribution $P_{data}$.

### *2.2. Model extraction against deep generative models*

Model extraction against generative models is to steal the function of a victim model $G_v$ by training a substitute neural network $G_a$ with the data an adversary collects after querying $G_v$. In this paper, the adversary has no prior knowledge on the architecture or dataset of the victim model. For the simplicity of illustration, all summarized notations in this paper are presented in table 1.

Table 1. Notations.

| Notation | Description |
|---|---|
| $G_v$ | victim model (target model) |
| $G_a$ | attacker model (substitute model) |
| $P_r$ | distribution of real training data of a victim model |
| $P_v$ | distribution of real training data of a victim model |
| $P_a$ | distribution of data generated by an attacker model |

This work follows the way Hu and Pang[8] used to taxonomize the attack according to adversary`s goals. Accuracy extraction aims to minimize the difference between $P_a$ and $P_v$. Fidelity extraction means reducing the gap between $P_a$ and $P_r$. The whole process is shown in figure1.

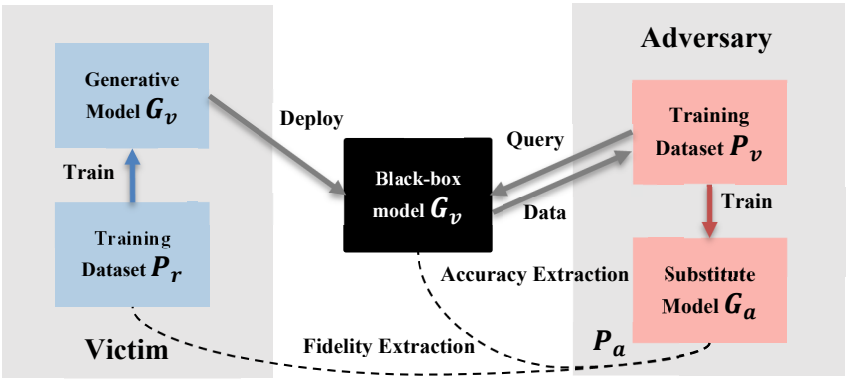



Figure 1. Structure of model extraction on generative models.

*2.3. Metrics*

In order to evaluate the performance of models above and extraction attacks against them, this paper uses the 1-Nearest Neighbor classifier (1-NN classifier) as a metric. Lopez-Paz and Oquab (2016)[13] first regarded 1-NN accuracy as a statistic for testing the similarity between two distributions of samples and conducted experiments on GAN evaluation. In this work, we just care about the application of this metric and implementation will not be introduced. As for the 1-NN accuracy result, the closer it is to 0.5, the more effective the generative model. In other words, if all the results are over 0.5, which is the case in this work, the lower the result, the better the effect. Moreover, this metric can also apply to evaluation the performance of extraction here because it could represent the difference between $P_a$ and $P_v$ in accuracy extraction or that between $P_r$ and $P_a$ in fidelity extraction. Xu et al.[14] showed that 1-NN accuracy has almost all the merits (high discriminability and efficiency) of other metrics. Actually, images are input into a 34-layer ResNet to obtain features of them and the features, not the images themselves, are computed in this work.

## 3. Experiment on extracting deep generative models

This section introduces the setting of the experiments in detail, and analysis of results.

*3.1. Experimental setting*

PixelCNN, VAE and GAN will be experimented with in this work as representative of difference types of deep generative models. The architectures of these models mainly follow the information provided in original papers[10,11,12]. MNIST[15] dataset is used in the experiments. When computing the 1-NN accuracy, 1,000 images generated by a model will be sampled. As reference, the performance of the three models is provided (PixelCNN:0.61, VAE:0.86, and GAN:0.65). Moreover,

these results do not mean that the design of PixelCNN is more effective than VAE and GAN but only provided for comparison in this work.

*3.2. Accuracy extraction*

In order to examine the influence that the difference of model type has on the attack, this work combines the three models above in pairs and 9 different victim-adversary combinations are obtained. The number of queries is as many as 30% of the number of training data of MNIST. The result is shown in table 2.

Table 2. Performances of extraction when the number of queries is 30% of the number of training data of MNIST.

| Victim-Adversary | 1-NN accuracy(accuracy) | 1-NN accuracy(fidelity) |
|---|---|---|
| PixelCNN-PixelCNN | 0.61 | 0.77 |
| VAE-PixelCNN | 0.67 | 0.93 |
| GAN-PixelCNN | 0.71 | 0.86 |
| PixelCNN-VAE | 0.95 | 0.97 |
| VAE-VAE | 0.67 | 0.97 |
| GAN-VAE | 0.89 | 0.94 |
| PixelCNN-GAN | 0.93 | 0.93 |
| VAE-GAN | 0.96 | 0.97 |
| GAN-GAN | 0.89 | 0.89 |

Considering the number of queries may affect the attack, this work also simulates the case when an adversary collects different numbers of images to train the substitute model. The number of queries can be 10%, 30%, 50%, 70% or 100% of the size of training dataset used by a victim. The results are shown in figure 2. There are some facts. On one hand, when the attacker model is of the same type as the victim model, the attack performance is better, even though in many cases the difference is not obvious when the number of queries is big. Plus, considering the performance of models used in this work, PixelCNN does achieve a better result compared with GAN and VAE. On the other hand, the more queries an adversary makes, the lower the 1-NN accuracy. This is because the aim of generative models is to learn the distribution of training data no matter which family it belongs to in taxonomy. The more data an adversary can receive, the better the extraction performs since it makes attacker model`s learning the distribution of data more accurate.

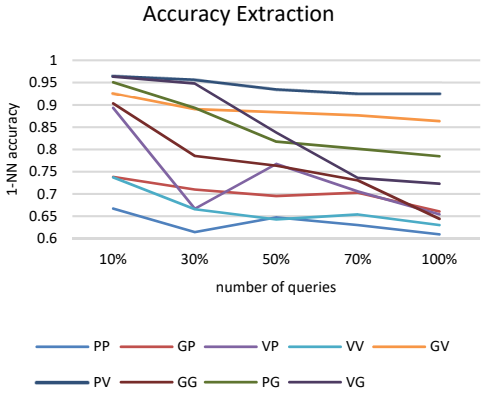



Figure 2. Accuracy extraction performance on different numbers of queries.

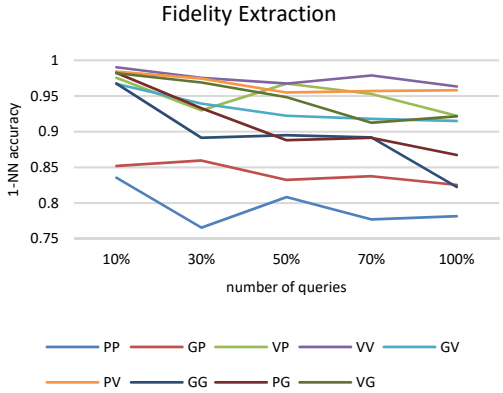



Figure 3. Fidelity extraction performance on different numbers of queries.

### *3.3. Fidelity extraction*

In the process of extraction, there is no distinction between fidelity extraction and accuracy extraction. However, in the evaluation of fidelity extraction, 1,000 images generated by an attacker model and 1,000 samples from the training data of MNIST will be calculated to obtain 1-NN accuracy. The results are shown in table 2 when the number of queries is as many as 30% of the number of training data of MNIST. The influence of number of queries can be found in figure 3. The 1-NN accuracy is higher in fidelity extraction than in accuracy extraction. The explanation is that there is always a difference between the distribution of data generated and the training data for a generative model to some extent. A more effective model may meet a less serious problem and vice versa. In fidelity extraction, there is already a gap between $P_r$ and $P_v$ for the victim model and also one between $P_v$ and $P_a$ for the attacker model. In this way, the difference between $P_r$ and $P_v$ may be more obvious.

## 4. Defense

This section introduces some countermeasures to defend model extraction attack against deep generative models mentioned above.

### *4.1. Generic strategy*

It is easy to conclude that limitation on the number of queries is a general approach to protect the victim model from both accuracy extraction and fidelity extraction. As it is presented in figure 2 and figure 3, the less data an adversary gets, the harder to perform an attack. However, there could hardly be any other method to defend accuracy extraction because whatever change made to the output of a victim model, the evaluation is still based on the data an adversary obtains and the data an attacker model generated with it. Therefore, the countermeasures below are for fidelity extraction and the number of queries by an adversary will be the same as 30% of the number of training data. Plus, the victim model in this section is of the same architecture as the attacker model.

### *4.2. PixelRNN/PixelCNN*

The most significant merit of explicit density models is that they can output the probability of a generated image. The higher the value, the better the quality of the image. In other words, the image should fit in distribution of training data when the probability is high. Considering this, if a victim sets a threshold and only outputs data with a probability lower than the threshold, $P_v$ will be different from $P_r$. Then $P_a$, the distribution of data produced by a substitute model trained on data from $P_v$ will also be dissimilar to $P_r$. The probability might be very low because we need to multiply small decimals hundreds of times. Among 1,000 samples, half of them are generated with a probability lower than $7.18\times10^{-222}$. The effect will be more obvious if the threshold is lower but it may lead to lower quality images. Therefore, the experiment here takes $7.18\times10^{-222}$ as a threshold. Finally, the 1-NN accuracy is 0.83 after filtering out some of the generated images while the 1-NN accuracy is 0.77 when there is no defense.

### *4.3. VAE and GAN*

Compared with PixelRNN/PixelCNN, one need to input latent vectors to obtain data generated by VAE or GAN. This feature enables a victim to leverage the latent vector to control the output of these generative models. Both VAE and GAN paper[11,12] did experiments by walking in latent space. Essentially, this strategy could work because perturbation on latent vector $z$ leads to the change of $P_v$. This work takes linearly interpolating with two latent vectors as defense. Given 100 queries each time, we randomly select two latent vectors and interpolate 98 vectors between them. For GAN, the 1-NN accuracy rises from 0.89 to 0.96 and for VAE, there is also an increase from 0.97 to 0.99.

## 5. Conclusion

This paper discussed model extraction attacks on deep generative models. According to an adversary`s goal, there are two types of extraction: accuracy extraction and fidelity extraction. For each type of

extraction, this work conducted experiments with MNIST dataset, and PixelCNN, VAE and GAN were studied as victim models or attacker models. To evaluate the effect of extraction, 1-NN accuracy was selected as a metric. The influence of numbers on the attack was also researched. We found that a stronger generative model performed better as an attacker model and with the increase in the number of queries, the performance could also be better.

We also discussed corresponding countermeasures to protect a victim model according to their design feature. For future work, we plan to devise other strategies to defend model extraction attack against generative models including leveraging conditional versions of those models.

**References**

[1] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., Ristenpart, T. (2016) Stealing machine learning models via prediction apis. In: 25th USENIX Security Symposium. Austin. pp. 601-618.

[2] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., Swami, A. (2017) Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. Dubai. pp. 506-519.

[3] Shi, Y., Sagduyu, Y. E., Davaslioglu, K., Li, J. H. (2018) Generative adversarial networks for black-box API attacks with limited training data. In: 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). Louisville. pp. 453-458.

[4] Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., Papernot, N. (2020) High accuracy and high fidelity extraction of neural networks. In: 29th USENIX Security Symposium. Boston. pp. 1345-1362.

[5] Kesarwani, M., Mukhoty, B., Arya, V., Mehta, S. (2018) Model extraction warning in mlaas paradigm. In: Proceedings of the 34th Annual Computer Security Applications Conference. San Juan. pp. 371-380.

[6] Lee, T., Edwards, B., Molloy, I., Su, D. (2018) Defending against machine learning model stealing attacks using deceptive perturbations. arXiv preprint arXiv:1806.00054.

[7] Juuti, M., Szyller, S., Marchal, S., Asokan, N. (2019) PRADA: protecting against DNN model stealing attacks. In: 2019 IEEE European Symposium on Security and Privacy (EuroS&P). Stockholm. pp. 512-527.

[8] Hu, H., Pang, J. (2021) Model Extraction and Defenses on Generative Adversarial Networks. arXiv preprint arXiv:2101.02069.

[9] Goodfellow, I. (2016) Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160.

[10] Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K. (2016) Pixel recurrent neural networks. In: 33rd International Conference on Machine Learning. New York. pp. 1747-1756.

[11] Kingma, D. P., Welling, M. (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[12] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014) Generative adversarial nets. Advances in neural information processing systems 27, 3: 2672–2680.

[13] Lopez-Paz, D., Oquab, M. (2016) Revisiting classifier two-sample tests. arXiv preprint arXiv:1610.06545.

[14] Xu, Q., Huang, G., Yuan, Y., Guo, C., Sun, Y., Wu, F., Weinberger, K. (2018) An empirical study on evaluation metrics of generative adversarial networks. arXiv preprint arXiv:1806.07755.

[15] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11): 2278-2324.