Robustness Training in Four Procedures of Machine Learning

Wei Yichen

November 9, 2024

Abstract

This paper examines the importance of robustness in machine learning, particularly given the iterative models and innovative data analysis architectures that have improved the ability to exploit diverse features and uncover complex relationships within datasets. However, a significant portion of the data is often noisy or toxic, requiring robust training techniques. We analyze existing robust training techniques from four key perspectives: input training data, deep neural architecture, regularisation, and loss function. In addition, we suggest possible future research directions in this area and provide an experiment on noise prediction's feasibility to show the potential of possible future robustness training. The code can be found on https://github.com/OwenCalstroy/Robustness-Training-in-Four-Procedures-of-Machine-Learning

1 Introduction

In recent years, our ability to deal with datasets has been greatly improved with the help of the development of iterative models in various domains, as well as the emergence of advanced data analysis architectures. These advances enable researchers to discover diverse features identify the intricate relationships, and finally improve the result of machine learning. However, it is important to realize that a significant amount of data may contain high levels of noise or even toxic elements. Therefore, ensuring the robustness of the entire machine-learning process becomes quite essential. This paper aims to deepen the understanding of robustness in machine learning by analyzing existing robust training techniques, from the perspective of the input training data, the deep neural architecture, the regularisation, and the loss function, which are all parts involved in the deep learning model training process as shown in Figure 1.

2 Current robustness training methods

In the field of machine learning, even the smallest disturbance can affect the final result dramatically. The development of robustness training methods has pushed the whole field to a higher position with higher accuracy and less possibility of being wrongly trained when facing a large amount of data with noises and inaccuracies in the real world. All of the improvements are made in the following four perspectives: input training data, deep neural architecture, regularisation, and loss function, all of which play an irreplaceable part in the whole process.

2.1 Input Training Data

If we want to enhance the robustness of the model when training the deep neural network, organizing and purifying training data in the first place will be the most efficient and easy way. This will be discussed in several aspects.

Enhancing the datasets. There were some classic methods to enhance datasets: the application of techniques such as panning, rotation, zooming, resolution alteration, color modification, and others serves to expand the dataset, therefore enhancing the generalization achieved through training and conferring enhanced resilience to future data. There are also some new techniques, like CutMix[1], achieved by combining two distinct input samples to create a novel training sample, and Cutout[2], achieved by randomly erasing pixels from a single or multiple sub-regions within the image.

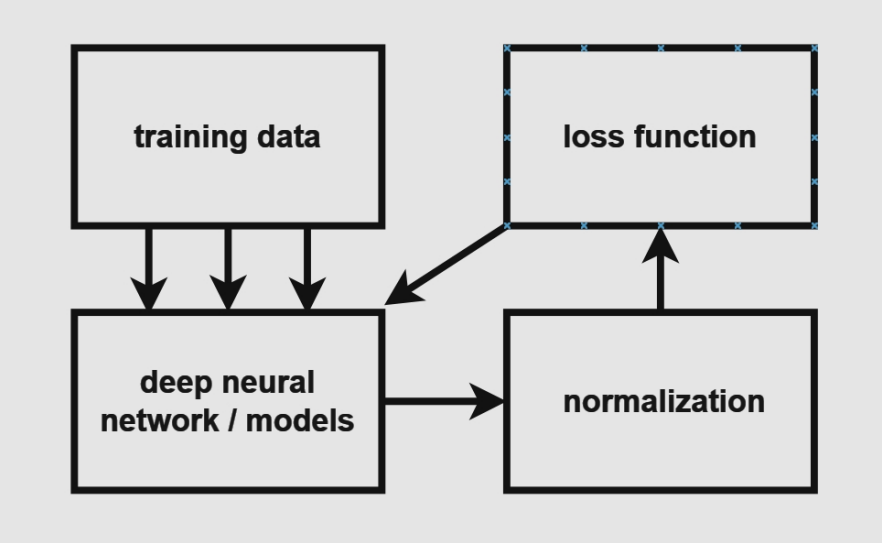

Figure 1: The deep learning model training process

Adding noise into the dataset. The inclusion of random noise in the training data enables the model to become more adaptable to the types of noise that may be present in real-world data[3]. [4]. There are also more recent studies like Structured noise injection[5], which introduces a spatial noise coding approach, whereby the input tensor is divided into distinct regions, each assigned a unique noise code. Additionally, a separate mapping is established for each code, corresponding to its respective region. This enables more detailed results to be generated. There is also an approach that implements additional optimizable weights to inject uncorrelated noise into the existing feed-forward network architecture through the use of Noise Injection Nodes (NINs) [6].

Data and category balancing. This includes Batch normalization[7], which serves to reduce internal covariate bias by introducing normalization operations in each layer of the neural network. It also includes SMOTE[8], achieved through the combination of oversampling a select number of classes and undersampling the majority class, and many other methods.

2.2 Deep Neural Architecture

The primary advancement in deep neural network architecture for improving resilience is adversarial training. In 2014, Christian Szegedy demonstrated that the perturbation of the dataset will lead to a completely different training trajectory[9], demonstrating the profound impact of the data purification process. In the same year, Goodfellow introduced the Fast Gradient Sign method[10] , which employs an adversarial process to generate new models with the help of the backpropagation method. Subsequently, Moosavi-Dezfooli (2016), Carlini and Wagner (2017), Zhang (2019), and Wang (2024) all proposed new models to enhance the model's robustness from large datasets.

Consider that most of the development is oriented from loss function or normalization, the ones that mainly focus on the model architecture itself are picked in this section.

BIM (basic iterative method) [11] generates adversarial samples by repeatedly adding perturbations to the gradient direction, with each step of the perturbation based on the current input gradient information. MIM (momentum-based basic iterative method) [12] is an improvement on BIM. The method draws on the analog momentum approach, which has been previously introduced in the parameter updating area in neural networks. In each epoch of perturbation, the current gradient direction is related to both the gradient direction computed in the previous iteration and the current input gradient.

DeepFool[13], introduced by Moosavi-Dezfooli, employs the 'fast gradient sign' approach to identify the minimal perturbation that can change the classifier's decision. Consequently, the performance of the entire structure has been evaluated in responding to potential perturbation datasets and found to be significantly improved. However, this approach relies on large computing resources and therefore is inadequate for situations that require targeted processing. This is fatal when facing large-scale datasets that the model nowadays receives. It also needs to achieve higher quality and accuracy.

AdvXL[14], introduced by Wang, was employed and operated on a large-scale dataset, for over 1 billion samples. The framework depends on a light pre-training stage and an intensive fine-tuning stage. The main idea is to first deploy a long training with weaker attacks and then deploy a short training with harsh attacks, to maintain the efficiency while reducing the calculating costs. Moreover, they employ a CLIP text encoder to extract classifier weights when dealing with large-scale datasets (specifically in their research, web-crawled ones). This technique provides a solution for large-scale data, allowing the model to enter training with a relatively accurate initial state, which finally produces training results with less deviation and better robustness. This idea partly solves the large-scale datasets problem, providing a new perspective when confronted with possible poisonous and noisy data. However, recognizing and separating the data for the pre-training stage is still a demanding job, which will lead to a more dominant position of poisonous and noisy data in the pre-training stage if not done well, because recognizing the poisonous and noisy data as data with weak attacks will definitely hurt the model at the beginning. If the characteristic of the chosen data is homogeneous and extreme, the pre-training stage will also provide a biased model, harming the fine-tuning stage.

2.3 Normalization

Regularisation is a key step in the machine learning process. It prevents overfitting results and enhances the model's generalization performance by applying penalties that prevent the parameters from being incorrectly influenced by the data. In addition, if the appropriate regularisation method is chosen, the computational complexity can be reduced, resulting in better model training results. The development of regularisation techniques in machine learning demonstrates the importance that researchers place on the robustness of the trained models.

Based on Dropout, Dropblock[15], introduced by Golnaz Ghiasi, made a further approach for structured data. It removes entire contiguous regions from the feature map of a given layer, rather than deleting individual units at random. The whole process was completed by randomly choosing and then adopting masks onto the input sample before the final normalization process. The method is designed to facilitate the acquisition of spatially distributed representations. This demonstrates that the specification of normalization in response to different situations may yield superior outcomes. Given that each situation and case may possess different value propositions and that the emphasis on varying aspects may result in disparate performance outcomes, this is an important consideration for future researchers to bear in mind.

Entropy Regularization[16], introduced by Ligong Han, established a method using the idea of Renyi entropy. It enables the model to achieve higher uncertainty when facing ambiguous input, which has higher entropy, while lower when the input is trustworthy, therefore exploring a broader range of predictions. The entropy value is typically obtained by deep neural networks, which generate discrete distributions of potential classes based on the input data. This allows the uncertainty of the prediction to be quantified. This method decreases the complexity and data volume and is also suitable for a variety of situations. However, it is complex in entropy computation and may be sensitive to noise, therefore failing to provide a more robust model.

The Information Bottleneck, brought out by [17] and introduced into the deep learning field by [18], is a method of compressing read data by refining relevant information, thereby reducing the workload of training while retaining core information. The core objective is "to minimize the complexity of the representation (i.e. compress the data) while maximizing information relevant to a specific target variable" [19]. This approach reduces the impact of noise and erroneous data in multidimensional datasets by maintaining essential information. Nevertheless, some valid information will inevitably be excluded due to the information bottleneck, causing the suboptimal utilization of data. However, the reduced computational burden and superior outcomes make it a dependable option in actual work.

The Information Bottleneck is part of the Mutual Information Maximization process. The goal of Mutual Information Maximization is to achieve the greatest possible mutual information between two random variables. There are other similar works like Deep InfoMax[20], which "perform unsupervised learning of representations by maximizing mutual information between an input and the output of a deep neural network encoder", and Contrastive Predictive Coding[21], which predict the future in the latent space through robust autoregressive modeling, using probabilistic contrastive loss to capture the information most useful for predicting future samples, etc.

2.4 Loss Function

The loss function is an important part of machine learning. It offers a method to show the distance between the current trained model and the ideal target model. Designing and choosing the proper loss function in different situations will affect greatly how the model performs, therefore enhancing the robustness of the loss function will improve the accuracy and efficiency of the whole model. Here are some popular optimizations in loss function robustness.

Huber loss function[22], introduced by Huber P.J., is a useful tool for dealing with data sets that contain outliers. Huber's loss function combines MSE and MAE. It sets a bar and uses MSE for values below the bar and MAE for values above. This makes the machine less sensitive to large errors and easier to optimize for small errors. This makes the model more robust.

Label smoothing [23], introduced by Christian Szegedy, achieved better robustness by changing the final target from 1 to 1 - LS, where LS is the label smoothing constant. It overcomes the fact that ordinary loss only considers the loss of correct label positions and does not focus on reducing the probability of predicting incorrect labels, reducing the severity of judgemental targets for correct outcomes leads to better prosperity. This is an easy but efficient way to strengthen the robustness of the model during training, but it is incapable when facing high-precision tasks because it sacrifices detailed accuracy to gain high robustness.

GANetic loss [24], introduced by Shakhnaz Akhmedova, takes a different perspective on loss function optimization. The method makes use of genetic programming with the aim of identifying the optimal loss function. This is conceptualised as the objective of an optimisation process. In the process of loss function optimization, parameters are randomly selected from $\{y_{pred}, y_{real}, real\ numbers\}$, and the operator from {plus,minus,multiply,divide,square root,log,exp,sin,cos} , to form a loss function format. Subsequently, genetic programming is employed to iteratively refine the relatively effective loss functions, ultimately identifying a robust function with broad applicability across diverse datasets. This approach effectively addresses the challenge of designing effective loss functions and has yielded promising outcomes. Moreover, it has paved the way for the development of a universal, robust loss function that enhances model performance across different data types, offering insights into the design of loss functions in general. Nevertheless, the quality of the initial training datasets is of the utmost importance, requiring minimal error data and the most efficient type to achieve a more general loss function, which is a costly process.

Another interesting study is MCGAN [25], which introduced a new regression loss into GAN, integrated the Monte Carlo method into data sampling, and successfully solved the training instability. Therefore, it is suggested that integrating existing algorithms and methods in the classic computer science field, or simulating and analogizing studies and theories (like physics) in deep learning methods, may receive terrific results.

Free adversarial training, introduced by Ali Shafahi, reuses the backward pass for the next training process for the same mini-batch, creating a closer relationship between the input and optimization. Through repeatedly using the same mini-batch, this method reduces the requirement of high computational power and can achieve higher robustness while demanding little additional cost. This work was efficient on small-scale datasets and saving computational power, but did not solve the fundamental robustness problem in situations with large-scale datasets and enough computational power (ideally). This means that such a method requires further optimization or complete change.

3 Predictions and Outlooks towards Robustness Training and Enhancement

The field of model robustness is experiencing a surge in research activity, with an increasing number of researchers engaged in efforts to mitigate the impact of noise and enhance model generalization and robustness. In light of the aforementioned findings, I present the following predictions and outlook for future research in the field of robustness training.

Firstly, it is essential to consider the integration of established methodologies. At present, the majority of methodologies lack robustness in diverse scenarios. This is evident in specific domains such as image processing and large language modeling, where they are either not applicable or unable to focus on the distinct aspects emphasized in different domains (For instance, proximity interrelationships in image processing and word and paragraph order correlations in large language modeling). The

novel approaches may potentially facilitate the extraction of similarities from the disparate emphases inherent to disparate domains (e.g., when dealing with a single sample, to generalize the order of elements within the single sample, the relative distance relationship, etc.). Furthermore, it may enable the identification of a more general approach that can simultaneously address the core concerns of disparate domains.

Secondly, the introduction of additional models and concepts from subject areas such as physics, chemistry, biology, traditional computer science, and so forth, will facilitate more innovative modeling of noisy, harmful, and normal data, thereby achieving new results. As previously demonstrated with genetic algorithms, simulated annealing algorithms, and so forth, the combination of analogs of existing models from disciplines such as physics and chemistry with robustness training may yield superior and less arithmetic-demanding results in certain instances.

Thirdly, the prediction of noise may also prove to be a significant advancement in the field of machine learning, particularly in terms of enhancing the robustness of such models. This may rely especially on machine learning to recognize the 'noise pattern' and then produce the most suitable model. It can be postulated that noise in reality will also exhibit certain patterns. Researchers can identify and model noise in data with the assistance of machine learning, thereby forming reasonable noise estimates for specific datasets. This enables the identification of potential outlier noise points as well as malicious data. Learning noise as a normally retained part of the data rather than removing it may yield enhanced robustness and generalization. It would be beneficial to explore whether this direction can be developed with mutual reference to chaos in physics research.

Fourthly, the combination of robust training and computer security represents a potential avenue for future development. Robust training can be conceptualized as a form of data security, serving to safeguard against the detrimental impact of malicious data as well as the overfitting outcomes associated with an excessively low probability of 'perfect' data. Consequently, it would be fruitful to draw upon pertinent experiences and models from the domain of computer security and adapt them to the context of machine learning.

It can be stated that robustness training for machine learning still has significant room for improvement. Future research will concentrate on enhancing the adaptability and stability of models in complex, dynamic environments with large data volumes, particularly in domains such as combating attacks, data noise, and responding to real-world changes. It seems probable that the future development of robust training methods will be informed by the introduction of deep learning (as exemplified by [24]). This approach is likely to complement the development of machine learning models, to achieve even better results.

4 Experiment

For the third point, which is learning the noise pattern with machine learning, and then predicting the noise, I give out such an experiment to justify the huge advantage it will bring to the whole training process, enhancing the robustness of the model and boosting the efficiency of the learning process.

Let us suppose that the natural data exhibit a pattern of normal distribution concerning the noise. If we can successfully learn that:

1. The noise pattern of the input data exhibits characteristics similar to a normal distribution.
2. The approximate mean and standard deviation can be calculated.

Then we can greatly increase the efficiency of the training process and the robustness of the model.

In this simulation, it is hypothesized that the prediction will be realized through the application of machine learning. A normal distribution was added to the idealized dataset to simulate the natural noise. Two training processes were then conducted: one with the dataset before the predicted noise pattern was removed, and the other with the dataset after the predicted noise pattern was removed. The results demonstrate that the latter process exhibits a consistently diminishing loss number in every epoch, thereby progressing towards the target function (represented by the equation $y = 2.1 + 3.2x + 1.6x^2 + 0.9x^3$ in this experimental simulation) at an accelerated rate. Such result remains the same for different means and standard deviations, as is shown in Figure 2, Figure 3, Figure 4, and Figure 5. The code website can be found in the abstract.

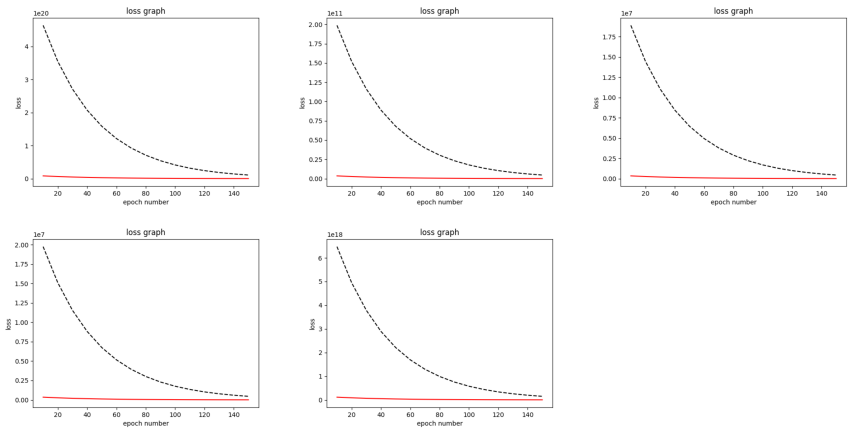

Figure 2: $(mean, \sigma) = (0.0, 0.1)$ and simulating learned parameters $(mean, \sigma) = (0.01, 0.08)$

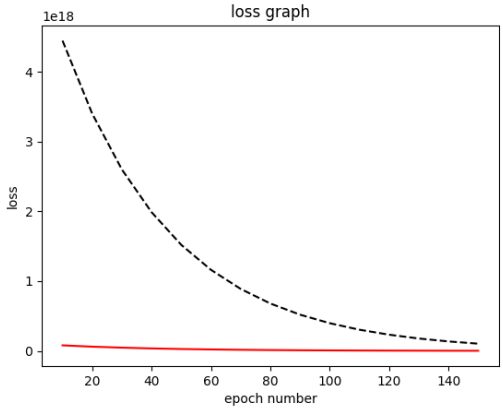

Figure 3: $(mean, \sigma) = (0.0, 0.5)$ and simulating learned parameters $(mean, \sigma) = (0.2, 0.48)$

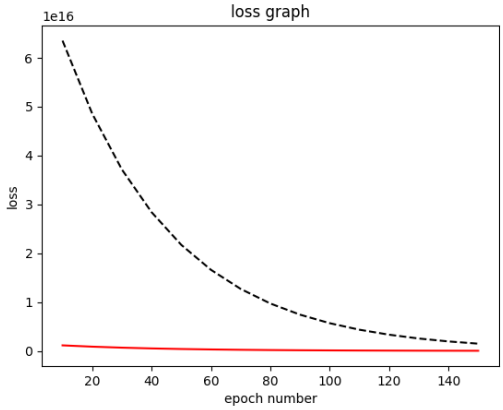

Figure 4: $(mean, \sigma) = (0.0, 1.0)$ and simulating learned parameters $(mean, \sigma) = (0.1, 0.9)$

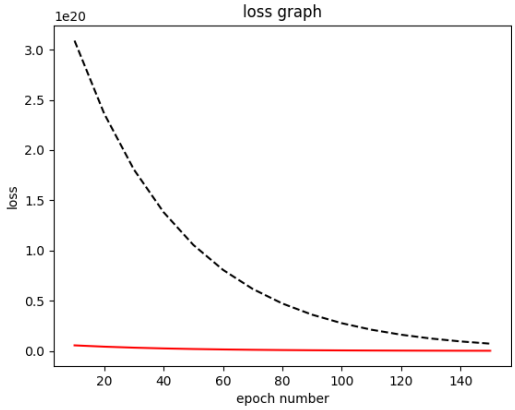

Figure 5: $(mean, \sigma) = (3.0, 1.0)$ and simulating learned parameters $(mean, \sigma) = (2.8, 0.9)$

This experiment shows how powerful machine learning can be when it learns patterns and removes noise from data. There is still a lot to learn about robustness training in machine learning. It is definitely a promising area for future research.

4.1 Conclusion

In this paper, we analyze existing robust training techniques from four key perspectives: input training data, deep neural architecture, regularisation, and loss function, realizing that there have been plenty of useful and profound measures from different perspectives. Moreover, we suggest possible future development directions in this area and provide an experiment on noise prediction's feasibility to show the potential of possible future robustness training. This experiment shows the enormous potential of noise prediction, which demonstrates that robustness training is still in its infancy and a considerable amount of time will elapse before it reaches a point of maturity.

References

[1] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019.

[2] Terrance DeVries. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[3] Xiangtao Meng, Chao Liu, Zhiyong Zhang, and Dong Wang. Noisy training for deep neural networks, 2014.

[4] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 588–597, 2019.

[5] Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5134–5142, 2020.

[6] Noam Levi, Itay Bloch, Marat Freytsis, and Tomer Volansky. Noise injection as a probe of deep learning dynamics. *arXiv preprint arXiv:2210.13599*, 2022.

[7] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

[9] C Szegedy. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[11] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2017.

[12] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[13] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[14] Zeyu Wang, Xianhang Li, Hongru Zhu, and Cihang Xie. Revisiting adversarial training at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24675–24685, 2024.

[15] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31, 2018.

[16] Ligong Han, Yang Zou, Ruijiang Gao, Lezi Wang, and Dimitris Metaxas. Unsupervised domain adaptation via calibrating uncertainties. In *CVPR Workshops*, volume 9, 2019.

[17] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000.

[18] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information, 2017.

[19] Ravid Shwartz Ziv and Yann LeCun. To compress or not to compress—self-supervised learning and information theory: A review. *Entropy*, 26(3):252, 2024.

[20] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization, 2019.

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[22] Peter J Huber and Elvezio M Ronchetti. *Robust statistics*. John Wiley & Sons, 2011.

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[24] Shakhnaz Akhmedova and Nils Körber. Ganetic loss for generative adversarial networks with a focus on medical applications. *arXiv preprint arXiv:2406.05023*, 2024.

[25] Baoren Xiao, Hao Ni, and Weixin Yang. Mcgan: Enhancing gan training with regression-based generator loss, 2024.