Increasing-Margin Adversarial (IMA) Training to Improve Adversarial Robustness of Neural Networks: Supplementary Information

September 11, 2022

This document contains supplementary information for the manuscript 'Increasing-Margin Adversarial (IMA) Training to Improve Adversarial Robustness of Neural Networks'. Appendix A further discusses the necessity of adversarial robustness. Appendix B discusses the effect of parameter β in the IMA. Appendix C discusses the effect of maximum margin in the IMA. Appendix D discusses the trade-off between robustness and accuracy. Appendix E furthers discusses the equilibrium state. Appendix F shows some examples of the clean and noisy images used in the experiment of the main paper. Appendix G shows the tables of results that correspond to the plots in the main paper.

Appendix A. Adversarial Robustness is Essential

It seems that adversarial noises are created by algorithms (e.g. PGD) and therefore it is only a security issue caused by hackers. In fact, random imaging noises could also be "adversarial" noises leading to wrong classifications. For the COVID-19 application, we did an additional test and found out that 2.75% of the noisy samples with uniform white noises on the level of 0.05, can cause the model "ce" to make wrong classifications. 2.75% is not a negligible number for this application. We note that CT imaging noises can be better described by Poisson distribution (Wang et al., 2008). However, without the hardware parameters of the CT machine, it is impossible to simulate Poisson noises. Nevertheless, adversarial robustness should be the built-in property of a model for this application, and all of the DNN models in the previous COVID-19 studies (Shi et al., 2020) should be checked and enhanced for adversarial robustness before deploying those models in clinics and hospitals.

Appendix B. the Effect of β in the IMA

We evaluated the effect of β in the IMA method on the Fashion-MNIST dataset, and the other settings are the same as those in the corresponding Section in the main paper (e.g. white-box 100-PGD attack on the test set). The results are reported in the Table 1 and Table 2.

Table 1: Results on Fashion-MNIST (L2 norm-defined noise level)

noise	0	0.5	1	2	3	4	5
β =0.1	0.8963	0.7981	0.6678	0.4203	0.2597	0.1288	0.0335
β =0.3	0.8957	0.8001	0.6804	0.4469	0.2838	0.1494	0.0470
β =0.5	0.8881	0.8069	0.6914	0.4711	0.3046	0.1666	0.0567
β =0.7	0.8800	0.8075	0.7055	0.4861	0.3195	0.1829	0.0711
β =0.9	0.8691	0.8048	0.7128	0.5001	0.3340	0.2006	0.0852

Table 2: Results on Fashion-MNIST (Linf norm-defined noise level)

noise	0	0.05	0.1	0.15	0.2	0.25	0.3
β =0.1	0.8998	0.7638	0.6122	0.4787	0.3266	0.1919	0.0738
β =0.3	0.8945	0.7742	0.6384	0.5093	0.3596	0.2240	0.1054
β =0.5	0.8900	0.7890	0.6688	0.5457	0.4121	0.2715	0.1521
β =0.7	0.8826	0.7955	0.6864	0.5720	0.4346	0.2951	0.1745
β =0.9	0.8709	0.8029	0.7107	0.6003	0.4682	0.3265	0.1992

It can be clearly seen that smaller β leads to higher accuracy on clean data (noise level = 0) and larger β leads to higher accuracy on noisy data.

The trade-off between robustness and accuracy is highly nonlinear, as shown in the Table 3: a small decrease in accuracy on clean data can result in a large increase in accuracy on noisy data.

Table 3: Accuracy differences caused by different values of β

noise	0	0.05	0.1	0.15	0.2	0.25	0.3
β =0.1	0.8998	0.7638	0.6122	0.4787	0.3266	0.1919	0.0738
β =0.5	0.8900	0.7890	0.6688	0.5457	0.4121	0.2715	0.1521
difference	0.0098	0.0252	0.0566	0.0670	0.0855	0.0796	0.0783

The nonlinear trade-off between robustness and accuracy makes it difficult to directly compare two methods, as different methods make different trade-offs between robustness and accuracy. The average accuracy is clearly not a good measure of performance on robustness.

Adjusting the parameter β of IMA is not a computationally efficient approach to make a trade-off between robustness and accuracy, because for every possible value of β in the range of 0 to 1, the user has to train a model from scratch through many epochs. In Appendix C, we show that the user of IMA can make such a trade-off much more efficiently by "adjusting" the parameter ε_{max} of IMA.

Appendix C. the Effect of ε_{max} in the IMA

In this appendix, we show how the allowed maximum margin (i.e., ε_{max} in Algorithm 2) affects the performance of IMA, and how to choose its value.

Noisy samples with adversarial noises can be correctly classified by humans but may be incorrectly classified by neural networks, which is basically the definition of adversarial noises. By following this definition, we choose the allowed maximum margin by looking at the noisy samples: if the noise magnitude is too large such that we barely can recognize the objects in the images, then this magnitude is chosen as the allowed maximum margin. This is how we choose the allowed maximum margin in the experiments on the datasets.

Next, we provide a strategy for the user of IMA to refine the choice of ε_{max} by making a trade-off between robustness and accuracy. As shown in Fig. 1, the trade-off between clean-accuracy (i.e., accuracy on clean data) and robustness (i.e., accuracy on noisy data) can be observed during training by using our IMA method.

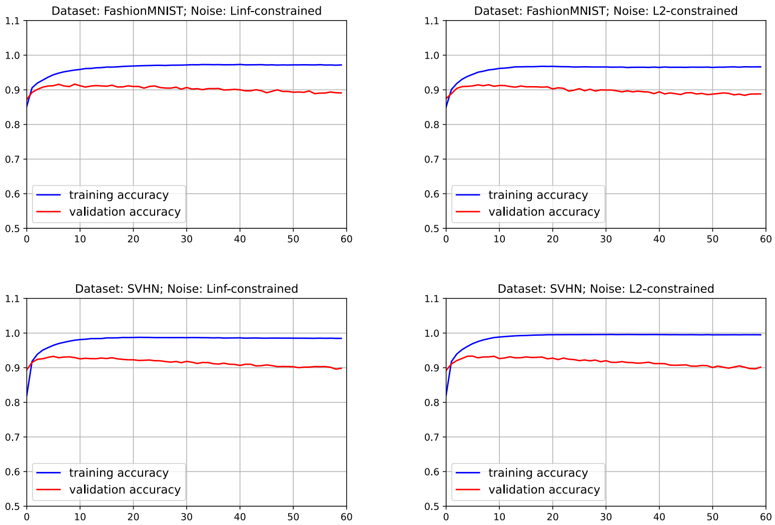


Figure 1: The training and validation curves (accuracy vs epoch) on different datasets obtained by using our IMA method. The accuracy scores are measured on clean data.

During IMA training, the (estimated) margin of a sample $x1$ in class-1 is initialized to be close to 0 and it keeps increasing as if the radius of a ball increases during the training process. The sample $x1$ is at the center of the ball, and the radius of the ball is the current margin of the sample. When the ball of $x1$ collides with the ball of another sample $x2$ from a different class-2, then a local decision boundary is formed, and the two balls stop expanding.

If there are enough training data samples, a sample x in some class will eventually meet with its counterpart in another class somewhere in the middle of the two classes, which forms a robust decision boundary. In practice, the amount of training data is never enough to cover the input space, and therefore the margin of a sample could be overestimated because of missing the counterparts that are needed to stop the expansion of the margin. If the margins of many samples are overestimated, then the balls of these samples may penetrate the optimal decision boundary and cause lower classification accuracy on the validation set of clean samples, **which explains the existence of the trade-off between robustness and clean-accuracy from the perspective of sample margins.**

The above analysis is confirmed by the trend of the training and validation accuracy curves in the Fig. 1: after some epochs, the training accuracy curve becomes stable, and the validation accuracy curve starts to decrease, which indicates margin overestimation occurs. **For our IMA method, the cause of margin overestimation is the lack of data in high dimensional space.** In the 2D Moons dataset, there are enough training samples, and therefore, the decision boundary of our IMA method is almost in the "middle" between the two classes, i.e., no margin overestimation.

Using the validation accuracy curve, the user of IMA can choose the allowed maximum margin such that validation accuracy is above a pre-defined threshold that is set by the user to meet some application requirements. Next, we demonstrate this approach on the Fashion-MNIST dataset. We note that during IMA training, the margins of the samples are gradually increasing. Immediately after M epochs, the maximum of the training sample margins is $M \times \Delta\varepsilon$ where $\Delta\varepsilon$ is the margin expansion step size in Algorithm 2. Thus, the maximum margin is $\varepsilon_{max} = M \times \Delta\varepsilon$ for the model trained for M epochs. Since we do not know the threshold on validation accuracy, which a user may be interested in on the dataset, we selected six models trained by IMA after 10, 20, 30, 40, 50, and 60 epochs, and we evaluated the robustness of the six models on the test set by using the 100-PGD attack. Basically, we re-analyzed the models trained through 60 epochs in Section 3.2. The results are reported in Table 4 and Table 5, which reveal the effect of ε_{max} on robustness.

Table 4: Effect of ε_{max} on Fashion-MNIST test set (L2 norm-defined noise level, $\Delta\varepsilon = 5/60$). The last column (named 0 (val)) shows the accuracies on the validation set (clean data).

noise	0	0.5	1	2	3	4	5	0 (val)
$\varepsilon_{max} = 10\Delta\varepsilon$	0.9106	0.7760	0.5590	0.1275	0.0037	0	0	0.9102
$\varepsilon_{max} = 20\Delta\varepsilon$	0.9058	0.8103	0.6758	0.3448	0.0831	0.0023	0	0.9087
$\varepsilon_{max} = 30\Delta\varepsilon$	0.8976	0.8122	0.6975	0.4468	0.2070	0.0441	0.0019	0.8998
$\varepsilon_{max} = 40\Delta\varepsilon$	0.8888	0.8116	0.7061	0.4805	0.2915	0.1267	0.0289	0.8890
$\varepsilon_{max} = 50\Delta\varepsilon$	0.8887	0.8058	0.6968	0.4772	0.3153	0.1685	0.0538	0.8862
$\varepsilon_{max} = 60\Delta\varepsilon$	0.8881	0.8069	0.6914	0.4711	0.3046	0.1666	0.0567	0.8882

Table 5: Effect of ε_{max} on Fashion-MNIST test set (Linf norm-defined noise level, $\Delta\varepsilon = 0.3/60$). The last column (named 0 (val)) shows the accuracies on the validation set (clean data)

noise	0	0.05	0.1	0.15	0.2	0.25	0.3	0 (val)
$\varepsilon_{max} = 10\Delta\varepsilon$	0.9159	0.6479	0.2598	0.0402	0.0009	0	0	0.9165
$\varepsilon_{max} = 20\Delta\varepsilon$	0.9095	0.7459	0.5165	0.2449	0.0647	0.0035	0	0.9115
$\varepsilon_{max} = 30\Delta\varepsilon$	0.9015	0.7754	0.6021	0.4022	0.2050	0.0644	0.0053	0.9018
$\varepsilon_{max} = 40\Delta\varepsilon$	0.8995	0.7895	0.6452	0.4877	0.3141	0.1588	0.0446	0.9013
$\varepsilon_{max} = 50\Delta\varepsilon$	0.8957	0.7887	0.6564	0.5217	0.3657	0.2245	0.0989	0.8953
$\varepsilon_{max} = 60\Delta\varepsilon$	0.8900	0.7890	0.6688	0.5457	0.4121	0.2715	0.1521	0.8910

From Table 4 and Table 5, it can be clearly seen that smaller ε_{max} leads to higher accuracy on clean data (noise level = 0), and larger ε_{max} leads to higher accuracy on noisy data. Since the maximum of the sample margins gradually increases during IMA training, the validation accuracy changes gradually (increasing and then decreasing), which makes it easy for the user of our method to choose the trained model with validation accuracy above the user-defined threshold.

We have done similar analyses on the models trained by IMA through 60 epochs on the SVHN dataset (Section 3.2). We selected six models trained by IMA after 10, 20, 30, 40, 50, and 60 epochs, and we evaluated the robustness of the six models on the test set by using the 100-PGD attack. The results are reported in Table 6 and Table 7, which reveal the effect of ε_{max} on robustness. By monitoring the validation accuracy curve during training, the user of IMA can choose a trained model such that validation accuracy of the chosen model is above a threshold defined by the user to meet some application requirements.

Table 6: Effect of ε_{max} on SVHN test set (L2 norm-defined noise level, $\Delta\varepsilon = 2/60$). The last column (named 0 (val)) shows the accuracies on the validation set (clean data).

noise	0	0.05	0.1	0.25	0.5	1	1.5	2	0 (val)
$\varepsilon_{max} = 10\Delta\varepsilon$	0.9382	0.9173	0.8888	0.7539	0.4602	0.1131	0.0231	0.0040	0.933
$\varepsilon_{max} = 20\Delta\varepsilon$	0.9303	0.9126	0.8893	0.7918	0.5658	0.2002	0.0570	0.0147	0.9257
$\varepsilon_{max} = 30\Delta\varepsilon$	0.9204	0.9012	0.8797	0.7944	0.6020	0.2588	0.0918	0.0290	0.9173
$\varepsilon_{max} = 40\Delta\varepsilon$	0.9139	0.8952	0.8721	0.7885	0.6081	0.2820	0.1066	0.0371	0.9120
$\varepsilon_{max} = 50\Delta\varepsilon$	0.9014	0.8824	0.8616	0.7769	0.6083	0.3002	0.1258	0.0484	0.9058
$\varepsilon_{max} = 60\Delta\varepsilon$	0.8995	0.8797	0.8576	0.7731	0.6012	0.2982	0.1280	0.0500	0.9016

Table 7: Effect of ε_{max} on SVHN test set (Linf norm-defined noise level, $\Delta\varepsilon = 0.1/60$). The last column (named 0 (val)) shows the accuracies on the validation set (clean data).

noise	0	0.005	0.01	0.02	0.04	0.06	0.08	0.1	0 (val)
$\varepsilon_{max} = 10\Delta\varepsilon$	0.9362	0.8674	0.7497	0.4637	0.1258	0.0284	0.0060	0.0012	0.9293
$\varepsilon_{max} = 20\Delta\varepsilon$	0.9294	0.8708	0.7869	0.5756	0.2409	0.0850	0.0291	0.0091	0.9229
$\varepsilon_{max} = 30\Delta\varepsilon$	0.9171	0.8617	0.7878	0.6095	0.3035	0.1317	0.0543	0.0208	0.9146
$\varepsilon_{max} = 40\Delta\varepsilon$	0.9110	0.8566	0.7862	0.6137	0.3259	0.1543	0.0702	0.0320	0.9095
$\varepsilon_{max} = 50\Delta\varepsilon$	0.9049	0.8518	0.7836	0.6180	0.3418	0.1724	0.0850	0.0417	0.9034
$\varepsilon_{max} = 60\Delta\varepsilon$	0.8919	0.8387	0.7731	0.6123	0.3461	0.1818	0.0938	0.0481	0.8983

Next, we plot the curves from the COVID-19 dataset in Fig. 2. The validation accuracy decreased only slightly after 20 epochs, which means the value of the allowed maximum margin is reasonable and the risk of margin overestimation is low.

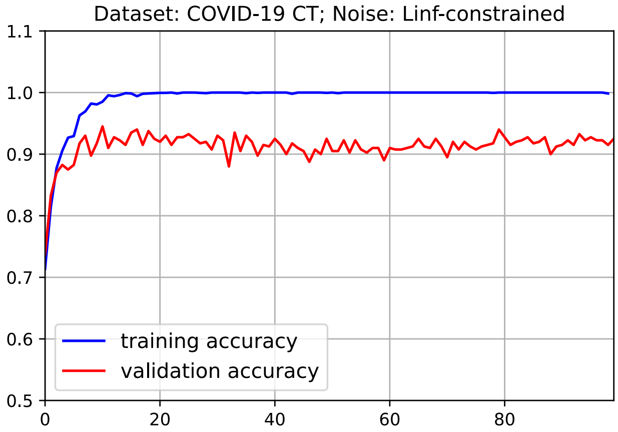


Figure 2: The training and validation curves (accuracy vs epoch) on the COVID-19 CT dataset obtained by using our IMA method. The accuracy scores are measured on clean data.

We note that **using the margin distribution estimated by IMA, we can obtain a good perturbation magnitude ε (0.1 or 0.15) for the vanilla adversarial training. When ε=0.15, vanilla adversarial training achieved the best performance on the COVID-19 CT dataset, compared to other methods.** Please see Fig. 4 which contains the margin distributions estimated by IMA, and Table 4 which contains accuracy scores in a large range of noise levels. In practice, we only need robustness against noises within a certain level, because significantly noisy images can only be produced from a malfunctioning CT machine. Thus, vanilla adversarial training (denoted by adv ε) should be good enough (much less computation cost and time compared to advanced adversarial training methods) as long as its parameter ε is appropriate. In this application, good values of ε are 0.1 and 0.15, as revealed by the margin distribution estimated by IMA. For the convenience of the reader, we re-plot the results in the new Fig. 3 and Fig. 4.

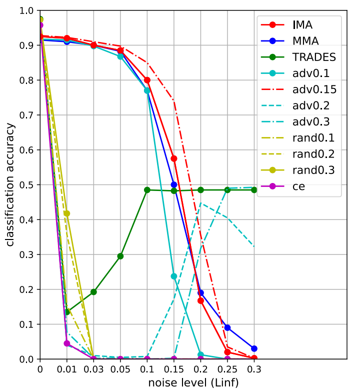


Figure 3: The performance of the methods on COVID-19 test set

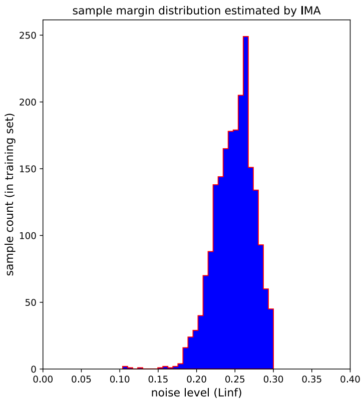


Figure 4: Margin distribution estimated by IMA on training set

For vanilla adversarial training, a straightforward way to find a good ε would be running grid research and evaluating the performance on the validation set. However, the grid-research in a large range (e.g. 0 to 0.3) is impractical because adversarial training is computationally expensive and time-consuming, compared to standard training with cross-entropy loss and clean data. The margin distribution estimated by IMA can reveal a good range of the ε, and then grid-research in this small range can be performed to find the optimal ε. After the optimal ε is found for vanilla adversarial training, we could combine other techniques to further improve robustness.

Appendix D. the Trade-off Between Robustness and Accuracy: Sample Margins' Perspective

In this appendix, we discuss the trade-off between robustness and accuracy from the perspective of sample margins. Let x denote a clean sample (i.e., unaltered by any adversarial attacks). Let $x_{(\delta)}$ denote the noisy sample generated by adding an adversarial noise δ to x, i.e., $x_{(\delta)} = x + \delta$. The vector norm of δ is ε. Let y denote the true class label of x.

For vanilla adversarial training, the class label of the noisy sample $x_{(\delta)}$ is assumed to be the same as the class label y of the clean sample x, no matter how large the noise level ε is. This label assignment can be wrong if ε is very large: large enough such that $x_{(\delta)}$ may "look like" a sample in a different class (not y), which has been shown in numerous studies. Here, "look like" refers to the judgment of a human. However, it is impractical to let a person actually look at every noisy sample to assign the right label to it.

If there are unlimited training samples that cover the input space, the samples can reveal the true class distributions $p(x|y)$. Then, the optimal decision boundary can be obtained by Bayesian classification using $p(x|y)$ and $p(y)$ of each of the classes. For example, $p(x|y)$ can be simply modeled by a Gaussian mixture model, and $p(y)$ may be assumed to be the same for every class. If we assume Bayesian classification with unlimited data is as good as the judgment of human experts, then Bayesian decision boundary is optimal and robust against noises. Let $g(x)$ denote the Bayesian classifier (g means ground truth), and it outputs the true label of the input: $y = g(x)$ and $y_{(\delta)} = g(x_{(\delta)})$. Using the Bayesian-optimal decision boundary, the true margin of a clean sample x can be obtained, i.e., the (minimum) distance between x and the decision boundary, which is denoted by $m(x)$.

During adversarial training (vanilla or IMA), a noisy sample is generated, $x_{(\delta)} = x + \delta$ with $\varepsilon = ||\delta||$, and the class label y is assigned to it. If the noise/perturbation magnitude ε is larger than the true margin $m(x)$, then the assigned class label is wrong for $x_{(\delta)}$, and training the model with $x_{(\delta)}$ may cause its decision boundary to deviate from the Bayesian-optimal decision boundary even if the model $f(x)$ is initialized to be very close to $g(x)$. As a result, the classification accuracy on the test/validation set (clean data) will become lower, but the classification accuracy on the training set (clean data) may not change because the decision boundary has been pushed far away from the clean training sample x. **This phenomenon has been revealed by the training and validation accuracy curves of IMA (see Fig. 11 in Appendix F), which explains the well-known trade-off between robustness and accuracy.**

Thus, the success of adversarial training depends on the accurate estimation of the true margin $m(x)$: if $\mathcal{E}(x) = m(x)$, then IMA could be perfect. For the 2D Moons dataset, IMA can indeed find a nearly perfect decision boundary, significantly better than the other methods (see Fig. 6).

However, for a high dimensional dataset (e.g. Fashion-MNIST), there are not enough training samples to cover the input space. As a result, perfect Bayesian classification is not achievable because of inaccurate estimation of $p(x|y)$, and IMA cannot obtain perfect estimations of the true margins because there may not exist the counterparts in other classes (not y) that can stop the expansion of ε-ball (margin estimation) of x in class y. Margin overestimation is revealed by the gradually-decreasing trend of the validation accuracy curve (see Fig. 1). However, the good news for the user of IMA is that the user can choose a good value of ε_{max} such that the validation accuracy is above a pre-defined threshold (e.g. 90% for the COVID-19 application). Also, the user does not need to pre-define the value of ε_{max}: the allowed maximum margin will increase with the number of training epochs (see Appendix D), and the user only needs to monitor the training and validation accuracy curves. Although IMA is not perfect, it brings such a significant **benefit to the user: the ease of "adjusting" ε_{max} to make a trade-off between robustness and accuracy (simply set β=0.5).**

From IMA, when an equilibrium state is reached, the distributions (i.e., local densities) of noisy samples in different classes are the same along the decision boundary (note: it does not necessarily mean the clean samples in different classes are equally spaced from the decision boundary). This is somewhat analog to Bayesian classification: at the optimal decision boundary, the distributions (densities) of samples in two classes are the same, assuming the classes have equal prior probabilities. From this perspective, the noisy samples, which are generated by IMA, serve as the surrogates of the real samples. Obviously, we cannot claim it is Bayesian classification because noisy samples may not reveal the true distributions. From this perspective, more advanced adversarial training methods may be developed such that the generated samples may reveal the true distributions (i.e., $p(x|y)$); if so, then the resulting decision boundary could be optimal and robust.

Additional Note: The validation accuracy curves of IMA in Fig. 1 in Appendix C increase and then gradually decrease. Please do not confuse this with the common concept of overfitting on clean data. Actually, when the models were trained with cross-entropy loss and clean data, there is no gradually-decreasing trend in validation accuracy curves, as shown in Fig. 5. Thus, the only explanation of the gradually-decreasing trend in Fig. 1 in Appendix C is margin overestimation.

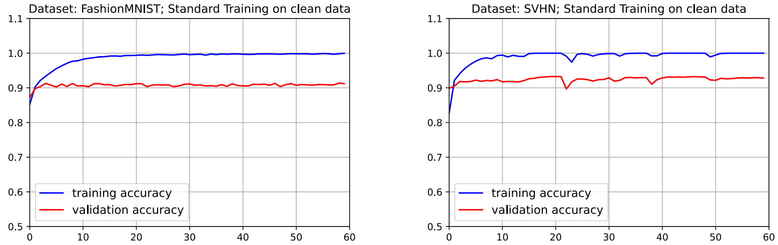


Figure 5: Training and validation accuracy curves (accuracy vs epoch) on the datasets, using standard training with cross-entropy loss and clean data. The accuracy scores are measured on clean data.

Appendix E. Further Discussion of the Equilibrium State

The basic idea of our IMA method

If there are only two classes and the data samples are linearly separable, then linear SVM (support vector machine) will produce a linear decision boundary in the “middle” between the two classes. The decision boundary of SVM is robust against noises: the classification output will not change if a small amount of noise δ is added to x as long as the vector norm of δ is smaller than the margin of x. Here, the margin of x is the (minimum) distance between x and the decision boundary.

In general, the data samples in multiple classes are nonlinearly separable, and the robust decision boundary should be somewhere in the middle between classes, which is the goal that IMA pursues. The result on the Moons dataset shows that IMA can indeed produce a nonlinear decision boundary in the “middle” between classes. We use 2D Moons dataset because it is impossible to directly visualize a nonlinear decision boundary in a high dimensional space. As shown on this dataset, other methods are not trying to find a decision boundary in the "middle" of the two classes.

The equilibrium state

Our IMA method is a heuristic-based method that is not derived from any theory. We use the equilibrium state analysis to provide a theoretical explanation of the method. We have shown that an equilibrium state can be achieved when the noisy samples have the same spatial distribution on the decision boundary. Here, we will analyze what will happen if the spatial distributions of the noisy samples in different classes are not the same on the current decision boundary. We note that our IMA method will actively generate and put noisy samples on ("close to" due to numerical precision) the current decision boundary of the neural network model, and the training is a dynamic process to adjust the decision boundary of the model. Let's focus on the following two terms:

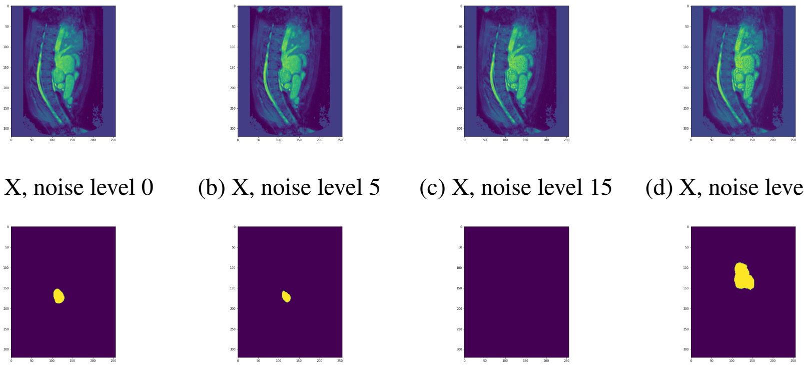
$$F_i \triangleq E_{X_n \in \ c_i \ and \ X_n \in \ B_{ij}} = -\int q_i(x) log(P_i(x))dx \qquad (1)$$

$$F_j \triangleq E_{X_n \in \ c_j \ and \ X_n \in \ B_{ij}} = -\int q_j(x) log(P_j(x))dx \qquad (2)$$

where $q_i(x)$ and $q_j(x)$ are the distributions (i.e., densities) of the noisy samples on the current decision boundary between the two classes, and $q_i(x)$ and $q_j(x)$ may not be equal to each other. In fact, F_i and F_j can be interpreted as two forces that try to expand the margins of the samples in the two classes against each other. By dividing the decision boundary into small regions (i.e., linear segments), the two integrals can be evaluated in the individual regions. In a region, if $q_i\ (x) > q_j(x)$ (i.e., more samples in class-i) then the current state is not in equilibrium: after updating the model using these noisy samples, the noisy samples in class-i will be correctly classified and the noisy samples in class-j will be incorrectly-classified (this is a simple result of classification with imbalanced data in the region), which means the decision boundary will shift towards the samples in class-j, and therefore the margins of the corresponding samples in class-i will expand and the margins of the corresponding samples in class-j will shrink. Thus, the decision boundary may shift locally towards the samples in one of the classes. Obviously, the decision boundary will stop shifting when the local densities of noisy samples in different classes are the same along the decision boundary, i.e., $q_i(x)$ becomes equal to $q_j(x)$, which means an equilibrium state is reached.

Appendix F. Examples of Sample Images

D2 Heart

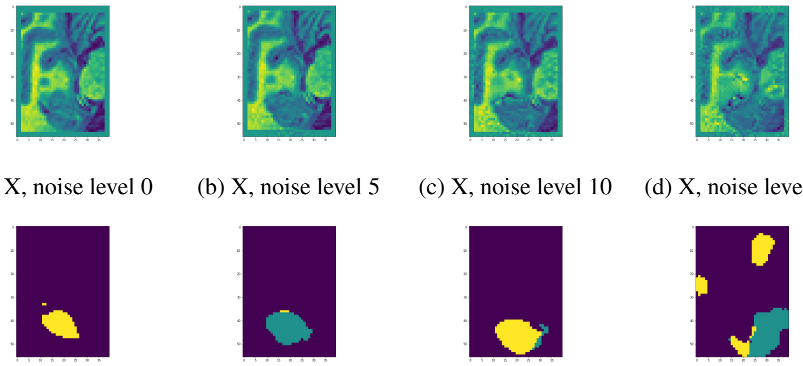


(a) X, noise level 0 (b) X, noise level 5 (c) X, noise level 15 (d) X, noise level 25

(e) Y, noise level 0 (f) Y, noise level 5 (g) Y, noise level 15 (h) Y, noise level 25

Figure 6: D2 Heart: The first row shows the input of nnUnet under different levels of 100-PGD adversarial noises. The second row shows the corresponding prediction of nnUnet under different levels of 100-PGD adversarial noises.

D4 Hippocampus

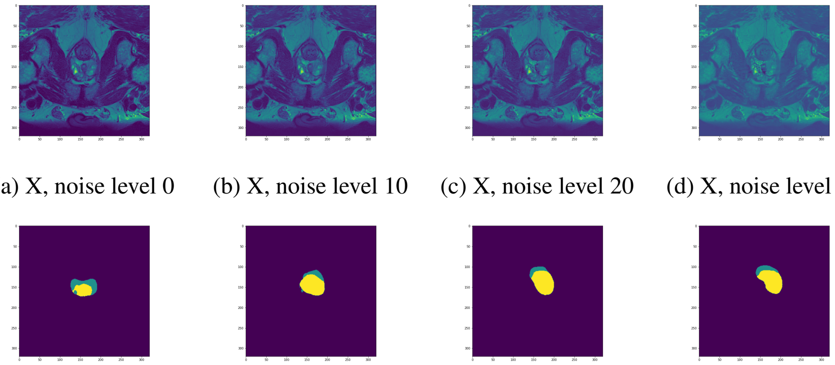


(a) X, noise level 0 (b) X, noise level 5 (c) X, noise level 10 (d) X, noise level 15

(e) Y, noise level 0 (f) Y, noise level 5 (g) Y, noise level 10 (h) Y, noise level 15

Figure 7: D4 Hippocampus: The first row shows the input of nnUnet under different levels of 100-PGD adversarial noises. The second row shows the corresponding prediction of nnUnet under different levels of 100-PGD adversarial noises.

D5 Prostate



(a) X, noise level 0 (b) X, noise level 10 (c) X, noise level 20 (d) X, noise level 40

(e) Y, noise level 0 (f) Y, noise level 10 (g) Y, noise level 20 (h) Y, noise level 40

Figure 8: D5 Prostate: The first row shows the input of nnUnet under different levels of 100-PGD adversarial noises. The second row shows the corresponding prediction of nnUnet under different levels of 100-PGD adversarial noises.

Covid-19 CT Images

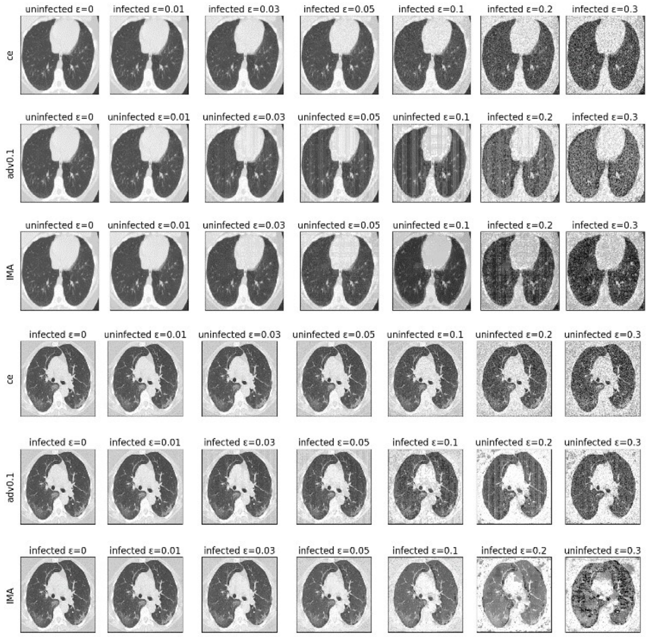


Figure 9: Each row shows a clean image and noisy images associated with a training method. The title of each image shows the predicted class label and the noise level. The clean images are correctly classified.

Appendix G. Tables of Results

Table 8: Results on Moons dataset (L-inf norm-defined noise level)

noise	0.0	0.05	0.1	0.15	0.2	0.25	0.3
IMA	1.0	0.999	0.9975	0.9915	0.9755	0.927	0.823
MMA	1.0	0.9985	0.993	0.957	0.882	0.772	0.597
TRADES	0.996	0.9805	0.9575	0.936	0.906	0.875	0.8255
adv	0.9975	0.985	0.957	0.9365	0.9095	0.883	0.822
ce	1.0	1.0	0.9975	0.9855	0.935	0.7995	0.606

Table 9: Results on Fashion-MNIST dataset (L-inf norm-defined noise level)

noise	0.0	0.05	0.1	0.15	0.2	0.25	0.3
IMA	0.89	0.789	0.6688	0.5457	0.4121	0.2715	0.1521
MMA	0.8955	0.746	0.6166	0.5051	0.3902	0.2644	0.1425
TRADES	0.9116	0.4759	0.4151	0.3722	0.3298	0.2453	0.0949
adv0.1	0.9112	0.7739	0.6613	0.4743	0.2376	0.0602	0.0046
adv0.2	0.9101	0.5879	0.5218	0.4885	0.4164	0.2199	0.0313
adv0.3	0.9132	0.3637	0.2506	0.1992	0.1671	0.1364	0.0854
ce	0.9117	0.0236	0.0	0.0	0.0	0.0	0.0

Table 10: Results on Fashion-MNIST dataset (L2 norm-defined noise level)

noise	0.0	0.5	1.0	2.0	3.0	4.0	5.0
IMA	0.8881	0.8069	0.6914	0.4711	0.3046	0.1666	0.0567
MMA	0.8926	0.807	0.6927	0.4617	0.2651	0.1361	0.0526
DDN	0.8666	0.789	0.6989	0.5066	0.3409	0.1965	0.0781
TRADES	0.9176	0.6103	0.4977	0.3425	0.1857	0.039	0.0008
adv1	0.9086	0.8096	0.6549	0.2726	0.0513	0.0015	0.0
adv3	0.9172	0.641	0.5859	0.4754	0.3325	0.1558	0.0229
adv5	0.9148	0.4851	0.4161	0.3011	0.2121	0.1419	0.075
ce	0.9117	0.2512	0.0033	0.0	0.0	0.0	0.0

Table 11: Results on SVHN dataset (L-inf norm-defined noise level)

noise	0.0	0.005	0.01	0.02	0.04	0.06	0.08	0.1
IMA	0.8919	0.8387	0.7731	0.6123	0.3461	0.1818	0.0938	0.0481
MMA	0.887	0.8357	0.771	0.6204	0.3693	0.2139	0.1216	0.0688
TRADES	0.9243	0.0995	0.0705	0.044	0.0371	0.0827	0.1748	0.1885
adv0.01	0.9152	0.8352	0.7095	0.4401	0.1174	0.0281	0.0062	0.0017
adv0.06	0.9105	0.6851	0.5737	0.5115	0.3599	0.1852	0.0784	0.0318
adv0.10	0.9209	0.2112	0.0086	0.0052	0.0048	0.0022	0.004	0.0472
ce	0.932	0.56	0.2384	0.0368	0.001	0.0	0.0	0.0

Table 12: Results on SVHN dataset (L2 norm-defined noise level)

noise	0.0	0.05	0.1	0.25	0.5	1.0	1.5	2.0
IMA	0.8995	0.8797	0.8576	0.7731	0.6012	0.2982	0.128	0.05
MMA	0.8765	0.8548	0.8322	0.7518	0.5974	0.3215	0.1519	0.0654
DDN	0.8649	0.8432	0.8184	0.7396	0.5852	0.3183	0.153	0.0669
TRADES	0.8652	0.8377	0.8076	0.6903	0.4581	0.1238	0.0256	0.0047
adv0.5	0.8986	0.876	0.849	0.7508	0.539	0.2062	0.0654	0.0208
adv1.0	0.873	0.8513	0.8284	0.7504	0.5857	0.2654	0.0955	0.0326
adv2.0	0.8663	0.8436	0.8175	0.7195	0.5458	0.2883	0.1454	0.0631
ce	0.932	0.8656	0.7588	0.4125	0.1185	0.0092	0.0006	0.0

Table 13: Results on COVID-19 CT image dataset (L-inf norm-defined noise level)

noise	0.0	0.01	0.03	0.05	0.1	0.15	0.2	0.25	0.3
IMA	0.925	0.92	0.9	0.885	0.8	0.575	0.1675	0.02	0.0025
MMA	0.915	0.91	0.9	0.8825	0.77	0.5	0.19	0.09	0.03
TRADES	0.975	0.135	0.1925	0.295	0.485	0.4825	0.485	0.485	0.485
adv0.1	0.9175	0.915	0.8975	0.8675	0.77	0.2375	0.0125	0.0	0.0
adv0.15	0.9275	0.9225	0.91	0.8975	0.85	0.74	0.355	0.035	0.0025
adv0.2	0.975	0.04	0.01	0.005	0.0075	0.1725	0.4475	0.405	0.3225
adv0.3	0.975	0.0775	0.0025	0.0025	0.0	0.0025	0.3175	0.49	0.4925
rand0.1	0.9775	0.155	0.0	0.0	0.0	0.0	0.0	0.0	0.0
rand0.2	0.965	0.3575	0.0	0.0	0.0	0.0	0.0	0.0	0.0
rand0.3	0.9725	0.4175	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ce	0.9575	0.045	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 14: TVDI Results on D2 dataset (L-2 norm-defined noise level)

noise	0	5	15	25
IMA25	0.85	0.8	0.69	0.56
adv25	0.8	0.76	0.68	0.59
adv15	0.84	0.8	0.67	0.51
adv5	0.88	0.81	0.6	0.35
nnUnet	0.92	0.69	0.02	0.0

Table 15: ADI Results on D2 dataset (L-2 norm-defined noise level)

noise	0	5	15	25
IMA25	0.77	0.7	0.56	0.44
adv25	0.67	0.62	0.53	0.44
adv15	0.72	0.66	0.53	0.38
adv5	0.75	0.65	0.46	0.24
nnUnet	0.8	0.52	0.02	0.0

Table 16: TVDI Results on D4 dataset (L-2 norm-defined noise level)

noise	0	1	5	10	15
IMA15	0.82	0.8	0.72	0.56	0.32
adv15	0.8	0.78	0.69	0.51	0.27
adv5	0.83	0.82	0.71	0.51	0.2
adv1	0.85	0.82	0.61	0.21	0.02
nnUnet	0.86	0.78	0.15	0.0	0.0

Table 17: ADI Results on D4 dataset (L-2 norm-defined noise level)

noise	0	1	5	10	15
IMA15	0.74	0.72	0.64	0.49	0.27
adv15	0.69	0.67	0.58	0.41	0.22
adv5	0.71	0.69	0.58	0.38	0.16
adv1	0.74	0.69	0.49	0.15	0.01
nnUnet	0.75	0.64	0.09	0.0	0.0

Table 18: TVDI Results on D5 dataset (L-2 norm-defined noise level)

noise	0	10	20	40
IMA40	0.72	0.66	0.59	0.43
adv40	0.71	0.63	0.55	0.39
adv20	0.73	0.65	0.56	0.39
adv10	0.73	0.62	0.5	0.29
nnUnet	0.81	0.37	0.18	0.07

Table 19: ADI Results on D5 dataset (L-2 norm-defined noise level)

noise	0	10	20	40
IMA40	0.67	0.61	0.55	0.38
adv40	0.65	0.56	0.46	0.29
adv20	0.66	0.56	0.45	0.28
adv10	0.66	0.54	0.4	0.21
nnUnet	0.74	0.3	0.14	0.04

References

Shi, F.; Wang, J.; Shi, J.; Wu, Z.; Wang, Q.; Tang, Z.; He, K.; Shi, Y.; and Shen, D. 2020. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. *IEEE reviews in biomedical engineering*.

Wang, J.; Lu, H.; Liang, Z.; Eremina, D.; Zhang, G.; Wang, S.; Chen, J.; and Manzione, J. 2008. An experimental study on the noise properties of x-ray CT sinogram data in Radon space. *Physics in Medicine & Biology*, 53(12): 3327.