Large-scale Robust Deep AUC Maximization: A New Surrogate Loss and Empirical Studies on Medical Image Classification

Zhuoning Yuan† ZHUONING-YUAN@UIOWA.EDU
Yan Yan‡ YANYAN.TJU@GMAIL.COM
Milan Sonka† MILAN-SONKA@UIOWA.EDU
Tianbao Yang† TIANBAO-YANG@UIOWA.EDU
†*Department of Computer Science, The University of Iowa, IA 52242*
‡ *School of Electrical Engineering& Computer Science, Washington State University, WA 99163*

Abstract

Deep AUC Maximization (DAM) is a new paradigm for learning a deep neural network by maximizing the AUC score of the model on a dataset. Most previous works of AUC maximization focus on the perspective of optimization by designing efficient stochastic algorithms, and studies on generalization performance of large-scale DAM on difficult tasks are missing. In this work, we aim to make DAM more practical for interesting real-world applications (e.g., medical image classification). First, we propose a new **margin-based min-max surrogate loss** function for the AUC score (named as the AUC min-max-margin loss or simply AUC margin loss for short). It is **more robust** than the commonly used AUC square loss, while enjoying the same advantage in terms of large-scale stochastic optimization. Second, we conduct extensive empirical studies of our DAM method on four difficult medical image classification tasks, namely (i) classification of chest x-ray images for identifying many threatening diseases, (ii) classification of images of skin lesions for identifying melanoma, (iii) classification of mammogram for breast cancer screening, and (iv) classification of microscopic images for identifying tumor tissue. Our studies demonstrate that the proposed DAM method improves the performance of optimizing cross-entropy loss by a large margin, and also achieves better performance than optimizing the existing AUC square loss on these medical image classification tasks. Specifically, our DAM method has achieved **the 1st place** on Stanford **CheXpert** competition on Aug. 31, 2020. To the best of our knowledge, this is the first work that makes DAM succeed on large-scale medical image datasets. We also conduct extensive ablation studies to demonstrate the advantages of the new AUC margin loss over the AUC square loss on benchmark datasets. The proposed method is implemented in our open-sourced library LibAUC (`www.libauc.org`) whose github address is `https://github.com/Optimization-AI/LibAUC`.

1. Introduction

In the last decade, we have seen great progress in deep learning (DL) techniques for medical image classification driven by **large-scale medical datasets**. For example, Stanford machine learning group led by Andrew Ng has collected and released a high-quality large-scale Chest X-Ray dataset for detecting chest and lung diseases, which contains 224,316 high-quality X-rays images from 65,240 patients [22]. Various deep learning methods have been designed and evaluated on this dataset by participating the **CheXpert** competition organized by Stanford ML group [22], and many of them have achieved radiologist-level performance on detecting certain related diseases. Esteva et al. [10] have trained a CNN

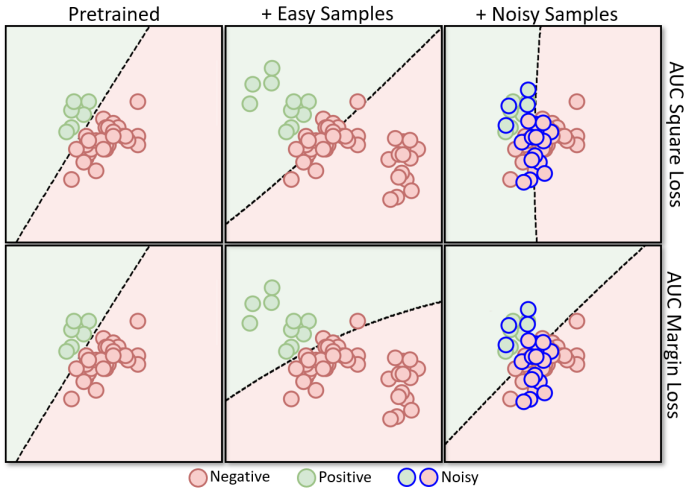

Figure 1: An illustrative example for optimizing different AUC losses on a toy data for learning a two-layer neural network with ELU activation. The top row is optimizing the AUC square loss and the bottom row is optimizing the new AUC margin loss. The first column depicts the initial decision boundary (dashed line) pre-trained on a set of examples. In the middle column, we add some easy examples to the training set and retrain the model by optimizing the AUC loss. In the last column, we add some noisily labeled data (blue circled data) to the training set and retrain the model by optimizing the AUC loss. The results demonstrate the new AUC margin loss is more robust than the AUC square loss.

using a dataset of 129,450 clinical images consisting of 2,032 different diseases, and achieved dermatologist-level performance for classification of skin lesions. Wu et al. [39] have trained a deep neural network for breast cancer screening on a large-scale medical dataset, which includes 229,426 digital screening mammography exams (1,001,093 images) from 141,473 patients. Their model is as accurate as an experienced radiologist. Despite these great efforts, an important question remains:

"Can we design a generic method that can further improve the performance of DL on these medical datasets without relying on domain knowledge"?

In this paper, we provide an affirmative answer to this question. Our solution is to optimize a novel loss for DL instead of optimizing the standard cross-entropy loss in the previous works. In particular, we choose to maximize the AUC score (a.k.a **the area under the ROC curve**) for DL. There are several benefits of maximizing AUC score over minimizing the cross-entropy loss. First, in medical classification tasks the AUC score is the default metric for evaluating and comparing different methods. Directly maximizing AUC score can potentially lead to the largest improvement in the model's performance. Second, the datasets in medical image classification tasks are usually imbalanced (e.g., the number of malignant cases is usually much less than benign cases). AUC is more suitable for handling imbalanced data distribution since maximizing AUC aims to rank the predication score of any positive data higher than any negative data. However, AUC maximization is

much more challenging than minimizing mis-classifcation error since AUC is much more sensitive to model change. A simple example in Appendix F shows that by changing the prediction scores of a few examples, the mis-classification error rate keep unchanged but the AUC score drops significantly.

AUC maximization has been studied in the community of machine learning [11, 41, 28, 23, 12]. However, existing methods for AUC maximization are still not satisfactory for practical use. The foremost challenge for AUC maximization is to determine a surrogate loss for the AUC score. A naive way is to use a pairwise surrogate loss based on the definition of the AUC score. However, optimizing a generic pairwise loss on training data suffers from a severe scalability issue, which makes it not practical for DL on large-scale datasets. Several studies have made attempts to address the scalability issue [23, 43, 41, 28]. One promising solution is to maximize the pairwise square loss for AUC by utilizing its special form [41, 28]. However, our study reveals that the AUC square loss has adverse effect when trained with easy data and is sensitive to the noisy data.

To address these issues, we propose a new margin-based surrogate loss in the min-max form for AUC (referred to as the AUC min-max-margin loss and the AUC margin loss for short), which is inspired by addressing the two issues of the AUC square loss. In particular, the AUC margin loss has two features that can alleviate the two issues, making it more robust to noisy data and not adversely affected by easy data. We will explain it with more details in the technical section and use a toy example in Figure 1 to illustrate the robustness of AUC margin loss over AUC square loss. Moreover, the min-max form of the AUC margin loss make it enjoy the same benefit as the AUC square loss in terms of scalability, making it more attractive than conventional pairwise margin-based surrogate loss for AUC maximization. In particular, we are able to directly employ existing large-scale optimization algorithms [15] designed for maximizing the AUC square loss to maximize our AUC margin loss with one line change of the code.

To demonstrate the effectiveness of our deep AUC maximization method, we conduct empirical studies on four difficult medical image classification tasks, namely classification of X-ray images for detecting chest diseases, classification of images of skin lesions, classification of mammograms for breast cancer screening and classification of microscopic images of tumor tissue. Our deep AUC maximization method has achieved great success on these difficult tasks. Specifically, we achieved **the 1st place** on Stanford **CheXpert** competition on Aug. 31, 2020, and **Top 1%** rank on Kaggle 2020 **Melanoma** classification competition. In CheXpert competition, our method is ranked 1 out of 150+ submissions, with a 2%+ improvement over Stanford baseline on a private testing data. In Kaggle competition, our ensembled model is ranked 33 out of 3314 teams. However, our best single model is better than the winning team's best model by more than 2%. Besides these medical tasks, we also conduct extensive ablation studies on benchmark datasets to compare the proposed AUC margin loss with the AUC square loss and traditional classification losses including cross-entropy and focal loss. Before ending this section, we summarize **our contributions** below:

- We proposed a new robust surrogate loss for AUC maximization, which is more robust than the AUC square loss but enjoys the same benefit of large-scale optimization.

- We conducted extensive empirical studies of the DAM method on a broad range of medical image classification data, and demonstrated its superb performance compared with standard DL methods.

To the best of our knowledge, this is the first comprehensive study of DAM on large-scale medical image classification datasets.

2. Related Work

Optimizing Pairwise Surrogate loss. Based on the definition of AUC, many studies consider to optimize a pairwise surrogate loss for AUC [11, 41, 28]. Joachims et al [23] proposed a SVM method for optimizing the AUC measure, which has a complexity of $O(n^2)$ for a dataset with n examples. Many later studies tried to improve the efficiency of optimizing a pairwise surrogate loss of AUC. Herschtal et al. [20] proposed an approximate objective for empirical pairwise loss of AUC by using partial pairs. In particular, for each negative data they only constructed a pairwise loss with only one positive data. However, the quality of such approximation highly depends on the properties of the dataset. When the examples have large intra-variance, their objective could yield poor performance. Zhao et al. [43] proposed an online method for AUC maximization by maintaining a data buffer for storing some historical positive and negative data, and constructed an approximate AUC score by pairing a newly received data with all data in the buffer. However, analysis shows that such data buffer needs to be very large in order to make the algorithm has a small regret.

Optimizing Pairwise Square loss. Pairwise square loss is an exception, which has a unique property to enable one to design efficient stochastic algorithms for large-scale data [12, 41, 27, 30]. In particular, Ying et al. [41] formulated the minimization of the pairwise square loss into an equivalent min-max optimization problem, which allows them to develop efficient stochastic algorithms without explicitly constructing and handling pairs of positive and negative data. Several papers tried to improve the convergence rate for solving the min-max optimization problems [27, 30].

Deep AUC Maximization (DAM). Most of the studies mentioned above are for learning a linear model. Recently, there are some emerging studies on DAM. In [35], the authors considered DAM for learning a deep neural network based on an online buffered gradient method proposed by [43], and applied it to classification of breast cancer based on imbalanced mammogram images. Nevertheless, the issue of this approach is that it cannot scale to large datasets as it requires a large buffer to store positive and negative samples at each iteration for computing an approximate AUC score. Hence, they only consider datasets with few thousand medical images. Recently, [28, 15] proposed efficient stochastic non-convex min-max optimization algorithms for DAM by solving the corresponding min-max objective of the AUC square loss. Their algorithms can scale up to hundreds of thousands of training examples. [14, 42] proposed federated learning algorithms for distributed DAM. However, all of these studies have neglected the deficiencies of the square loss for AUC maximization. To the best of our knowledge, this is the first work that analyzes the deficiencies of AUC square loss and proposes a better solution.

3. Method

Notations. Let $\mathbb{I}(\cdot)$ be an indicator function of a predicate, $[s]_+ = \max(s, 0)$. Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ denote a set of training data, where $\mathbf{x}_i$ represents an input training example (e.g., an image), and $\mathbf{y}_i \in \{1, -1\}$ denotes its corresponding label (e.g., the indicator of a certain disease). For notational simplicity, we use $\mathbf{z} = (\mathbf{x}, y)$. Let $\mathbf{w} \in \mathbb{R}^d$ denote the parameters of the deep neural network to be learned, and let $h_{\mathbf{w}}(\mathbf{x}) = h(\mathbf{w}, \mathbf{x})$ denote the prediction of the neural network on an input data $\mathbf{x}$. The standard approach of deep learning is to define a loss function on individual data by $L(\mathbf{w}; \mathbf{x}, y) = \ell(h_{\mathbf{w}}(\mathbf{x}), y)$, where $\ell(\hat{y}, y)$ is a surrogate loss function of the misclassification error (e.g., cross-entropy loss), and to minimize the empirical loss $\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n L(\mathbf{w}; \mathbf{x}_i, y_i)$. However, this standard approach is easily misled by the imbalanced distribution of training images in medical datasets. In medical applications, a more favorable metric for comparing and evaluating different classifiers is **AUC**. It has been shown that the algorithms designed to minimize the misclassification error rate may not lead to maximization of AUC [7].

3.1 Background on Scalable AUC Maximization

Existing works of AUC maximization consider the following definition of AUC that is equivalent to the Wilcoxon-Mann-Whitney statistic [17, 5]:

$$\begin{aligned} \mathrm{AUC}(\mathbf{w}) &= \Pr(h_{\mathbf{w}}(\mathbf{x}) \geq h_{\mathbf{w}}(\mathbf{x}') | y = 1, y' = -1) \\ &= \mathbb{E}\big[\mathbb{I}(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}') \geq 0) \big| y = 1, y' = -1\big]. \end{aligned} \tag{1}$$

It is interpreted that the AUC score is the probability of a positive sample ranking higher than a negative sample.

For optimization purpose, the indicator function in the above definition of AUC is usually replaced by a *convex surrogate loss* $\ell : \mathbb{R} \to \mathbb{R}^+$ which satisfies $\mathbb{I}(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}') < 0) \leq \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}'))$. As a result, many existing works formulate the AUC maximization on a training data $\mathcal{S}$ as

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{N_+ N_-} \sum_{\mathbf{x}\in\mathcal{S}_+} \sum_{\mathbf{x}'\in\mathcal{S}_-} \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}')), \tag{2}$$

where $\mathcal{S}_+, \mathcal{S}_-$ denote the set of positive and negative examples, and N_+, N_- denote their size, respectively. Nonetheless, directly optimizing the above formulation is not scalable to large datasets as the complexity could be as worse as $O(n^2)$ due to there are $O(n^2)$ pairs, where n is the total number of examples.

To address the scalability issue, existing studies have proposed some promising solutions. One solution that attracts great attention is to optimize the square loss due to its algorithmic simplicity. With a square loss $\ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}')) = (1 - h_{\mathbf{w}}(\mathbf{x}) + h_{\mathbf{w}}(\mathbf{x}'))^2$ as the surrogate loss of AUC, it was shown that the objective is equivalent to the following min-max problem [41]:

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^d \\ (a,b)\in\mathbb{R}^2}} \max_{\alpha\in\mathbb{R}} f(\mathbf{w}, a, b, \alpha) := \mathbb{E}_{\mathbf{z}}\left[F(\mathbf{w}, a, b, \alpha; \mathbf{z})\right], \tag{3}$$

where $\mathbf{z} = (\mathbf{x}, y) \in \mathcal{S}$ is a random sample, and

$$\begin{aligned} F(\mathbf{w}, a, b, \alpha; \mathbf{z}) &= (1-p)\left(h_{\mathbf{w}}(\mathbf{x}) - a\right)^2 \mathbb{I}_{[y=1]} \\ &+ p(h_{\mathbf{w}}(\mathbf{x}) - b)^2 \mathbb{I}_{[y=-1]} - p(1-p)\alpha^2 \\ &+ 2\alpha\left(p(1-p) + p h_{\mathbf{w}}(\mathbf{x})\mathbb{I}_{[y=-1]} - (1-p)h_{\mathbf{w}}(\mathbf{x})\mathbb{I}_{[y=1]}\right), \end{aligned} \tag{4}$$

and $p = \Pr(y = 1)$. Since the objective function in the above formulation is decomposable over individual examples, hence it enables one to develop efficient primal-dual stochastic algorithms for updating the model parameter $\mathbf{w}$ without explicitly constructing positive-negative pairs. Several studies have developed efficient stochastic algorithms for solving the above min-max formulation, which are able to scale to hundreds of thousands of examples [41, 27, 28].

3.2 Drawbacks of the AUC Square Loss

Although the AUC square loss makes AUC maximization scalable, it has two issues that have been ignored by existing studies. In particular, it has adverse effect when trained with well-classified data (i.e., easy data), and is sensitive to noisily labeled data (i.e., noisy data). Below, we will elaborate these two issues by considering a linear model $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ for illustration and understand these issues from the viewpoint of stochastic gradient update. We give a one-dimensional data in Appendix E.2 to support our arguments. When we use the min-max formulation (3) to explain these issues, we will make some simplification. In particular, we will use the optimal value of a, b, α given $\mathbf{w}$, i.e., $a = a(\mathbf{w}) := \mathrm{E}[h_{\mathbf{w}}(\mathbf{x})|y = 1], b = b(\mathbf{w}) := \mathrm{E}[h_{\mathbf{w}}(\mathbf{x})|y = -1], \alpha = 1 + b - a$, where a, b can be interpreted as the mean prediction score on positive data and negative data, respectively (please refer to Appendix A for a derivation). The same trick will be used to illustrate the benefit of the AUC Margin loss.

Adverse Effect on Easy Data. To illustrate this, let us consider a scenario: the current model parameter is given by $\mathbf{w}$ and there comes a positive and negative data pair $(\mathbf{x}, y = 1), (\mathbf{x}', y' = -1)$. Suppose these data are easy examples meaning that the prediction $h_{\mathbf{w}}(\mathbf{x})$ is large and $h_{\mathbf{w}}(\mathbf{x}')$ is small such that $h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}') > 1$. By taking the stochastic gradient descent update of the square loss $\ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}')) = (1 - h_{\mathbf{w}}(\mathbf{x}) + h_{\mathbf{w}}(\mathbf{x}'))^2$, we have the updated model given by $\mathbf{w}_+ = \mathbf{w} - \eta 2(1 - h_{\mathbf{w}}(\mathbf{x}) + h_{\mathbf{w}}(\mathbf{x}'))(-\mathbf{x} + \mathbf{x}')$, where $\eta > 0$ is a step size. Since $1 - h_{\mathbf{w}}(\mathbf{x}) + h_{\mathbf{w}}(\mathbf{x}') < 0$, the model parameter $\mathbf{w}$ will move towards the negative direction of the positive data $\mathbf{x}$ and the positive direction of the negative data $\mathbf{x}'$. As a result, the new model $\mathbf{w}_+$ tends to push the score $h_{\mathbf{w}_+}(\mathbf{x})$ on the positive data smaller and the score $h_{\mathbf{w}_+}(\mathbf{x}')$ on the negative data larger, which makes its classification capability worse. A similar effect happens when we use the min-max objective (3) to conduct the update. We include the analysis in Appendix D.

Sensitivity to Noisy Data. Next, we elaborate the issue of sensitivity to noisily labeled examples. To this end, we consider a scenario: the current model parameter is given by $\mathbf{w}$ and there comes a positive and negative data pair $(\mathbf{x}, y = 1, \hat{y} = -1), (\mathbf{x}', y' = -1, \hat{y}' = 1)$, where y, y' denote the true labels of $\mathbf{x}, \mathbf{x}'$ that are not revealed, respectively, and $\hat{y} = -1, \hat{y}' = 1$ denote the noisy labels. Again, assume the prediction $h_{\mathbf{w}}(\mathbf{x})$ is large and $h_{\mathbf{w}}(\mathbf{x}')$ is small. The SGD update of the model parameter $\mathbf{w}$ based on the min-max objective is given by

$$\mathbf{w}_+ = \mathbf{w} - 2\eta\{(1-p)(h_{\mathbf{w}}(\mathbf{x}') - a - \alpha)\mathbf{x}' + p(h_{\mathbf{w}}(\mathbf{x}) - b + \alpha)\mathbf{x}\}.$$

By plugging the optimal values of a, b, α given $\mathbf{w}$, i.e., $\alpha = 1 + b - a$ and $a = \mathbb{E}[h_{\mathbf{w}}(\mathbf{x})|y = 1], b = \mathbb{E}[h_{\mathbf{w}}(\mathbf{x}')|y' = -1]$, we can see that the term in the update of $\mathbf{w}$ that involves $\mathbf{x}$ is $-2\eta p(h_{\mathbf{w}}(\mathbf{x}) + 1 - \mathbb{E}[h_{\mathbf{w}}(\mathbf{x})|y = 1])\mathbf{x}$, and that involves $\mathbf{x}'$ is $-2\eta p(h_{\mathbf{w}}(\mathbf{x}') - 1 - \mathbb{E}[h_{\mathbf{w}}(\mathbf{x}')|y' = 1])\mathbf{x}'$. Then it is clear to see that when $h_{\mathbf{w}}(\mathbf{x})$ is large enough such that $h_{\mathbf{w}}(\mathbf{x}) + 1 - \mathbb{E}[h_{\mathbf{w}}(\mathbf{x})|y = 1] > 0$, the update of $\mathbf{w}$ will move to the negative direction of the truly positive data $\mathbf{x}$, and similarly it will move to the positive direction of the truly negative data $\mathbf{x}'$ when $h_{\mathbf{w}}(\mathbf{x}')$ is small enough.

3.3 The Proposed AUC Margin Loss

To alleviate the two issues of the AUC square loss, we propose a new margin-based surrogate loss. The new surrogate loss is a direct modification of the square loss to alleviate the two issues. To motivate the new AUC margin loss, we reformulate the AUC square loss as following (please refer to Appendix B for a derivation):

$$\begin{aligned} A_{\mathrm{S}}(\mathbf{w}) &= \mathbb{E}[(1 - h_{\mathbf{w}}(\mathbf{x}) + h_{\mathbf{w}}(\mathbf{x}'))^2 | y = 1, y' = -1] \\ &= \underbrace{\mathbb{E}[(h_{\mathbf{w}}(\mathbf{x}) - a(\mathbf{w}))^2 | y = 1]}_{A_1(\mathbf{w})} \\ &+ \underbrace{\mathbb{E}[(h_{\mathbf{w}}(\mathbf{x}') - b(\mathbf{w}))^2 | y' = 1]}_{A_2(\mathbf{w})} + \underbrace{(1 - a(\mathbf{w}) + b(\mathbf{w}))^2}_{A_3(\mathbf{w})} \\ &= A_1(\mathbf{w}) + A_2(\mathbf{w}) + \max_{\alpha} \{2\alpha(1 - a(\mathbf{w}) + b(\mathbf{w})) - \alpha^2\}, \end{aligned} \tag{5}$$

where $a(\mathbf{w}) = \mathbb{E}[h_{\mathbf{w}}(\mathbf{x})|y = 1], b(\mathbf{w}) = \mathbb{E}[h_{\mathbf{w}}(\mathbf{x}')|y' = 1]$, and in the second equality we use the fact $s^2 = \max_{\alpha} 2\alpha s - \alpha^2$. The three terms $A_1(\mathbf{w}), A_2(\mathbf{w}), A_3(\mathbf{w})$ have meaningful interpretations. In particular, minimizing $A_1(\mathbf{w}), A_2(\mathbf{w})$ aim to minimize the variance of prediction scores on positive data and negative data, respectively; minimizing the $A_3(\mathbf{w})$ aims to push the mean prediction scores of positive and negative examples to be far away. However, the square function in the last term makes it suffer from the two aforementioned issues. Our solution is to use a squared hinge function to replace $A_3(\mathbf{w})$, which is widely used in margin-based SVM classifiers. In particular, we replace $A_3(\mathbf{w})$ by $\max_{\alpha \geq 0}\{2\alpha(m - a(\mathbf{w}) + b(\mathbf{w})) - \alpha^2\} = (m - a(\mathbf{w}) + b(\mathbf{w}))_+^2$, where m is a hyper-parameter that specifies desired margin between $a(\mathbf{w})$ and $b(\mathbf{w})$. Hence, **our new AUC margin loss is defined by**

$$\begin{aligned} A_{\mathrm{M}}(\mathbf{w}) &= A_1(\mathbf{w}) + A_2(\mathbf{w}) \\ &+ \max_{\alpha \geq 0} 2\alpha(m - a(\mathbf{w}) + b(\mathbf{w})) - \alpha^2. \end{aligned} \tag{6}$$

Without the non-negative constraint on α, the loss becomes the square loss with a tunable margin parameter m.

Benefits of the AUC Margin Loss. We first show that the above objective is equivalent to a min-max objective.

Theorem 1 *Minimizing the AUC margin loss (6) is equivalent to the following min-max optimization:*

$$\min_{\substack{\mathbf{w} \in \mathbb{R}^d \\ (a,b) \in \mathbb{R}^2}} \max_{\alpha \geq 0} \mathbb{E}_{\mathbf{z}} \left[F_M(\mathbf{w}, a, b, \alpha; \mathbf{z}) \right], \quad \textit{where} \tag{7}$$

$$
\begin{aligned}
F_M(\mathbf{w}, a, b, \alpha; \mathbf{z}) &= (1-p)\left(h_{\mathbf{w}}(\mathbf{x}) - a\right)^2 \mathbb{I}_{[y=1]} \\
&+ p(h_{\mathbf{w}}(\mathbf{x}) - b)^2 \mathbb{I}_{[y=-1]} - p(1-p)\alpha^2 \\
&+ 2\alpha\left(p(1-p)m + p h_{\mathbf{w}}(\mathbf{x})\mathbb{I}_{[y=-1]} - (1-p)h_{\mathbf{w}}(\mathbf{x})\mathbb{I}_{[y=1]}\right).
\end{aligned} \tag{8}
$$

We highlight that $\min_{a,b}\max_{\alpha\geq 0}\mathbb{E}_{\mathbf{z}}\left[F_{\mathrm{M}}\left(\mathbf{w}, a, b, \alpha; \mathbf{z}\right)\right] = p(1-p)A_{\mathrm{M}}(\mathbf{w})$. Please see proof in Appendix C.

Robust to Easy Data. Based on the above min-max formulation, let us first elaborate the benefits of the new loss that alleviate the two issues of the AUC square loss. First, let us consider how the non-negative constraint $\alpha \geq 0$ helps alleviate the adverse effect when trained with easy data. Following the same logic as before, we compute the gradient of $F_{\mathrm{M}}(\mathbf{w}, a, b, \alpha)$ by

$$
\begin{aligned}
\nabla_{\mathbf{w}} F_{\mathrm{M}}(\mathbf{w}, a, b, \alpha; \mathbf{z}) =& 2(1-p)\mathbf{x}\mathbb{I}_{[y=1]} \cdot (h_{\mathbf{w}}(\mathbf{x}) - a - \alpha) \\
&+ 2p\mathbf{x}\mathbb{I}_{[y=-1]} \cdot (h_{\mathbf{w}}(\mathbf{x}) - b + \alpha).
\end{aligned}
$$

Different from the square loss, the optimal α given $\mathbf{w}$ is $\alpha = m + b(\mathbf{w}) - a(\mathbf{w})$ if $m + b(\mathbf{w}) - a(\mathbf{w}) \geq 0$, and $\alpha = 0$ if $m + b(\mathbf{w}) - a(\mathbf{w}) < 0$, where $a(\mathbf{w}) = \mathrm{E}[h_{\mathbf{w}}(\mathbf{x})|y = 1], b(\mathbf{w}) = \mathrm{E}[h_{\mathbf{w}}(\mathbf{x})|y = -1]$. When the model is good enough, i.e., $m + b(\mathbf{w}) - a(\mathbf{w}) < 0$ meaning that the mean prediction scores of positive data is larger than the mean prediction scores of negative data by a margin $m > 0$, then the gradient becomes $\nabla_{\mathbf{w}} F_{\mathrm{M}}(\mathbf{w}, a, b, \alpha; \mathbf{z}) = 2(1-p)\mathbf{x}\mathbb{I}_{[y=1]} \cdot (h_{\mathbf{w}}(\mathbf{x}) - a) + 2p\mathbf{x}\mathbb{I}_{[y=-1]} \cdot (h_{\mathbf{w}}(\mathbf{x}) - b)$. Taking a stochastic gradient decent update for $\mathbf{w}$ will only push the prediction score of the sampled data to be close to their mean score. When the model is poor, i.e., $m + b(\mathbf{w}) - a(\mathbf{w}) > 0$, the gradient becomes $\nabla_{\mathbf{w}} F_{\mathrm{M}}(\mathbf{w}, a, b, \alpha; \mathbf{z}) = 2(1-p)\mathbf{x}\mathbb{I}_{[y=1]} \cdot (h_{\mathbf{w}}(\mathbf{x}) - m - b(\mathbf{w})) + 2p\mathbf{x}\mathbb{I}_{[y=-1]} \cdot (h_{\mathbf{w}}(\mathbf{x}) + m - a(\mathbf{w}))$. Since the model is poor in this case, it is likely that $h_{\mathbf{w}}(\mathbf{x}) - m - b(\mathbf{w}) < 0$ for a positive data $\mathbf{x}$, and $h_{\mathbf{w}}(\mathbf{x}) + m - a(\mathbf{w}) > 0$ for a negative data $\mathbf{x}$. As a result, taking a stochastic gradient decent update for $\mathbf{w}_+ = \mathbf{w} - \eta\nabla_{\mathbf{w}} F_{\mathrm{M}}(\mathbf{w}, a, b, \alpha; \mathbf{z})$ will likely move the model in the right direction pushing the prediction score of positive data larger, and that of negative data smaller.

Robust to Noisy Data. Next, let us elaborate how adding a tunable margin parameter m can help alleviate the sensitivity to noisy data. Similar to the AUC square loss, the update in the noisy data case is given by

$$
\mathbf{w}_+ = \mathbf{w} - 2\eta\{(1-p)(h_{\mathbf{w}}(\mathbf{x}') - a - \alpha)\mathbf{x}' + p(h_{\mathbf{w}}(\mathbf{x}) - b + \alpha)\mathbf{x}\},
$$

where $\mathbf{x}'$ is a true negative data but labeled as positive and $\mathbf{x}$ is a true positive data but labeled as negative. Let us consider the case that model is not good enough such that the optimal value of $\alpha = m + b(\mathbf{w}) - a(\mathbf{w})$. Then the term in the update of $\mathbf{w}$ that involves the true positive data $\mathbf{x}$ is $-2\eta p(h_{\mathbf{w}}(\mathbf{x}) + m - \mathbb{E}[h_{\mathbf{w}}(\mathbf{x})|y = 1])\mathbf{x}$, and that involves the true negative data $\mathbf{x}'$ is $2\eta p(m + \mathbb{E}[h_{\mathbf{w}}(\mathbf{x}')|y' = 1] - h_{\mathbf{w}}(\mathbf{x}'))\mathbf{x}'$. Note that even when $h_{\mathbf{w}}(\mathbf{x})$ is large and $h_{\mathbf{w}}(\mathbf{x}')$ is small such that the model $\mathbf{w}_+$ is moving in the wrong direction, by tuning m to a smaller value, we can ensure that the movement into the wrong direction is much reduced. Hence, adding the tunable margin parameter m can alleviate the sensitivity to the noisy data.

3.4 DAM with the AUC Margin Loss

As seen from Theorem 1, the AUC margin loss is equivalent to a min-max optimization problem, that is similar to that of the AUC square loss. Hence, any stochastic algorithms

Algorithm 1 PESG for optimizing the AUC margin loss

Require: η, γ, λ, T

1: Initialize $\mathbf{v}_1, \alpha_1 \geq 0$
2: **for** $t = 1, \ldots, T$ **do**
3: Compute $\nabla_{\mathbf{v}} F_{\mathrm{M}}(\mathbf{v}_t, \alpha_t; \mathbf{z}_t)$ and $\nabla_{\alpha} F_{\mathrm{M}}(\mathbf{v}_t, \alpha_t; \mathbf{z}_t)$.
4: Update primal variables

$$\mathbf{v}_{t+1} = \mathbf{v}_t - \eta(\nabla_{\mathbf{v}} F_{\mathrm{M}}(\mathbf{v}_t, \alpha_t; \mathbf{z}_t) + \gamma(\mathbf{v}_t - \mathbf{v}_{\mathrm{ref}})) - \lambda\eta\mathbf{v}_t$$

5: Update $\alpha_{t+1} = [\alpha_t + \eta\nabla_{\alpha} F_{\mathrm{M}}(\mathbf{v}_t, \alpha_t; \mathbf{z}_t)]_+$.
6: Decrease η by a factor and update $\mathbf{v}_{\mathrm{ref}}$ periodically
7: **end for**

proposed for solving the min-max objective of the AUC square loss can be easily adapted to solving the min-max objective of the AUC margin loss. In particular, for any update on the dual variable α, we follow by a projection step that projects α into non-negative orthant. In this paper, we employ the proximal epoch stochastic method (named PESG) proposed in [15] to update variables $\mathbf{w}, a, b, \alpha$. To present the algorithm, we use a notation $\mathbf{v} = (\mathbf{w}, a, b)$ to denote all primal variables. The key steps are presented in Algorithm 1. In the algorithm, λ denotes the standard regularization parameter (i.e, weight decay parameter), $\gamma > 0$ is an algorithmic regularization parameter that can help improve the generalization, $\mathbf{v}_{\mathrm{ref}}$ is a reference solution that is updated periodically by using the accumulated average of $\mathbf{v}_t$ in the previous stage (before decaying learning rate). We refer the readers to [28, 15] for more discussion and convergence analysis of this algorithm.

A Two-stage Framework for DAM. From our preliminary studies on deep AUC maximization, we observe that directly optimizing the AUC margin loss can easily handle the recognition tasks on simple datasets, e.g., CIFAR. However, it shows some difficulties on complex tasks, e.g., CheXpert, Melanoma. We conjecture that the feature extraction layers learned by directly optimizing AUC from scratch are not as good as optimizing the standard cross-entropy loss on these difficult data. Inspired by recent works on two-stage methods, e.g., [24], we also employ a two-stage framework **on difficult medical image classification tasks** that includes a *pre-training* step that minimizes the standard cross-entropy loss, and an *AUC maximization* step that maximizes an AUC surrogate loss of the pre-trained CNN for learning all layers with the last classifier layer randomly initialized.

4. Empirical Studies

In this section, we present extensive empirical studies on the proposed robust DAM method with the AUC margin loss. First, we present results on some benchmark datasets and then we present the results on four medical image classification tasks. The code for reproducing the results of our method in this paper can be found here [1].

4.1 Performance on Benchmark datasets

For benchmark datasets, we construct imbalanced Cat&Dog (C2), CIFAR-10 (C10), CIFAR-100 (C100), STL-10 (S10) [9, 25, 6] following instructions by [28]. Specifically, we first randomly split the training data by class ID into two even portions as the positive and negative classes, and then we randomly remove some samples from the positive class to make it imbalanced. We keep the testing set untouched. We refer to imbalance ratio (**imratio**) as the ratio of # of positive examples to # of all examples. Statistics of these datasets are presented in Appendix G.

We experiment with two network structures, i.e., DenseNet121 (**D**) ([21]) and ResNet20 (**R**) ([18]) with ELU activation functions. We explore the imbalance ratio = 1%, 10%, and use a 9:1 train/val split to conduct cross-valuation for tuning parameters. We compare DAM using our AUC margin loss (AUC-M) with three baselines, DAM using AUC square loss (AUC-S), and DL with two other popular loss functions i.e., cross-entropy loss (CE) and focal loss (Focal) trained by SGD. We use the $\hat{\alpha}$-balanced Focal loss $-\hat{\alpha}(1-p_t)^{\hat{\gamma}}\log(p_t)$, and tune its parameter $\hat{\alpha}, \hat{\gamma}$ from [0.25, 0.5, 0.75] and [1,2,5] on the validation set, respectively. For DAM, we tune γ in [1/100, 1/300, 1/500, 1/700, 1/1000]. For AUC-M loss, we tune margin parameter m in [0.1, 0.3, 0.5, 0.7, 1.0]. For optimization, we run 100 epochs with a stagewise learning rate: initial value of 0.1 and decaying at 50% and 75% of the total number of training epochs for all experiments. We use a weight decay, i.e., λ, as $1e$-4 for all methods. The batch size is set to 128 on all datasets except for S10, which is set to 32 due to smaller data size. For each method, we run the experiment with five different random training sets (by randomly removing some positive examples with different random seeds), and evaluate on the same testing set by comparing the averaged testing AUC scores. We also found that using a L2 normalization of the predication scores in a mini-batch is helpful. We refer to this normalization as **Batch Score Normalization** (BSN). Hence, in the following experiments we use the BSN before computing both the AUC-S and AUC-M losses. Please refer to section 5.1 for an ablation study on comparing with and without BSN.

The results for DenseNet121/ResNet20 with imratio=1% are reported in Table 1. We include the results for imratio=10% to the Appendix H. Overall, we observe that the AUC-M and AUC-S perform much better than non-AUC-based losses in most cases. Comparing AUC-M with AUC-S, we can see that AUC-M performs better in most cases, especially in the extremely imbalanced setting with imratio=1%.

We also conduct some ablation studies on the benchmark datasets to demonstrate the robustness of the proposed AUC-M loss in comparison with AUC-S loss for DAM with added easy and noisy data, and the effectiveness of non-negative constraint on α. The results are included in Section 5.2.

4.2 Medical Image Classification Tasks

Below, we present results on four difficult medical image classification tasks, namely classification of X-ray images for detecting chest diseases, classification of images of skin lesions for detecting melanoma, classification of mammograms for breast cancer screening, and classification of microscopic images for identifying tumor tissue. A summary of these tasks and their data is reported in Table 2.

Table 1: Testing AUC on benchmark datasets with imratio=1%.

Dataset	CE	Focal	AUC-S	AUC-M
C2 (D)	0.718±0.018	0.713±0.009	0.803±0.018	**0.809±0.016**
C10 (D)	0.698±0.017	0.700±0.007	0.745±0.010	**0.760±0.006**
S10 (D)	0.641±0.032	0.660±0.027	0.669±0.070	**0.703±0.030**
C100 (D)	0.588±0.011	0.591±0.017	0.607±0.010	**0.614±0.016**
C2 (R)	0.730±0.028	0.724±0.020	0.748±0.007	**0.756±0.017**
C10 (R)	0.690±0.011	0.681±0.011	0.702±0.015	**0.715±0.008**
S10 (R)	0.641±0.021	0.634±0.024	0.645±0.029	**0.659±0.020**
C100 (R)	0.563±0.015	0.565±0.022	0.587±0.017	**0.596±0.016**

Table 2: Summary of Medical Classification Tasks.

Dataset	Image Domain	Imratio	# Training
CheXpert	Chest X-ray	20.21%	224,316
Melanoma	Skin Lesion	7.1%	46,131
DDSM+	Mammogram	13%	55,000
PatchCamelyon	Microscopic	1%	148,960

4.2.1 CheXpert Competition

CheXpert competition is a medical AI competition organized by Stanford ML group [22], which released a large-scale Chest X-Ray dataset for detecting chest and lung diseases [22]. The training data consists of 224,316 high-quality X-ray images from 65,240 patients. The validation dataset consists of 234 images from 200 patients. The testing data has images for 500 patients, which is not released to the public and is maintained by the organizer for final evaluation. The training images were annotated by a labeler to automatically detect the presence of 14 observations in radiology reports, capturing uncertainties inherent in radiography interpretation. The validation images were manually annotated by 3 board-certified radiologists. The testing images were annotated by a consensus of 5 board-certified radiologists. The average resolution of CheXpert images is 2828x2320 pixels, which is about 6 times larger than ImageNet. The competition requires participants to submit the trained models for evaluation of the AUC score on predicting 5 selected diseases, i.e., Cardiomegaly, Edema, Consolidation, Atelectasis, Pleural Effusion. These tasks have an average imratio of 20.21%. They also reported another metric that compares the model's performance with 3 radiologists' predictions for reference.

Model Pre-training. To tackle the uncertain data in CheXpert, we adopt a label smoothing method similar to that in works [31]. We choose five networks: DenseNet121, DenseNet161, DensNet169, DensNet201 and Inception-renset-v2[21, 36]. With limited resources, we scale the resolution of all raw images to 320x320. For data augmentation, we use random rotation, random translation and random scaling. For *pre-training* step, we optimize CE loss by Adam on the 5 classification tasks with weight decay parameter of 1e-5. The total training time is 2 epochs with a batch size of 32 and initial learning rate of 1e-5. In the second step of AUC maximization, we replace the last classifier layer trained in the first step by random weights and use our DAM method to optimize the last classifier layer and all previous layers. We tune γ in $\{1/300, 1/500, 1/800\}$, set weight decay λ to 0,

Table 3: Averaged Testing AUC Scores on CheXpert. NBRC means the # of radiologists out of 3 are beaten by AI algorithms.

Model	AUC	NRBC	Rank
Stanford Baseline [22]	0.9065	1.8	85
YWW [40]	0.9289	2.8	5
Hierarchical Learning [31]	0.9299	2.6	2
DAM (Ours)	**0.9305**	**2.8**	**1**

set the initial learning rate to 0.1 and decrease the learning rate at 2000, 8000 iterations by 3 times, run a total of 2 epochs for Algorithm 1.

Competition Results. Our final submission is the ensemble of five models trained by DAM with the AUC-M loss for each disease. On Aug 31, 2020, we submitted our models to CheXpert and we achieved a mean testing AUC score of **0.9305**, which is currently ranked at **1st place** over all submissions. The leaderboard is shown in [13], where our submission is named as DeepAUC-v1 (ensemble). We also compare our results with other methods in Table 3, where Hierarchical Learning [31] utilizes domain knowledge to pre-define a disease hierarchy used for conditional training, YWW [40] utilizes weakly-supervised lesion localization technique through a novel Probabilistic-CAM (PCAM) pooling operator to improve the model training. All these solutions are trained by CE loss. Our AUC-based solution surpasses these solutions and it is also better than 2.8 out of 3 radiologists (NRBC) for 5 selected diseases on average as in Table 3. Finally, we noticed that a recent work that optimizes AUC square loss for DAM on CheXpert only achieves a mean testing AUC score of 0.922 [15].

4.2.2 Melanoma Classification

Melanoma is a skin cancer, which is the major cause for skin cancer death [29]. We conduct empirical studies on the Kaggle Melanoma dataset [32], which is released through a Kaggle competition. The data is split into 33,126 training images with 584 malignant melanoma images (imbalance ratio=1.76%) and 10,892 testing images with an unknown number of melanoma images. Further, the testing set is split into public testing set and private testing set at 30%/70% ratio by patient ID. The public testing set (noting that their ground-truth labels are not revealed) is used to rank participating teams at the early stage. The private testing set is used to evaluate the participating teams for the final ranking. The public AUC score is updated daily but private AUC score is released after the end of competition.

Data preparations. The raw dataset has various sizes of images, e.g., 6000x4000, 1920x1080. We resize all images to lower resolutions due to limited computational resources. To evaluate the model locally, we follow [8] to construct a 5-fold Stratified Leak-Free version cross-validation by 8:2 train/valid split. The data split follows two rules: 1) images from same patients are either put in train set or in validation set. 2) train and validation set have same imbalance ratio 1.76%. In addition, we also utilize two external data sources to complement the provided data in train set: 1) 12,859 images from previous competitions, e.g., ISIC2017 and ISIC2018, and 2) 580 malignant melanoma images parsed from the website of The International Skin Imaging Collaboration [2]. We merge all data sources and finally obtain a training set of 46,131 images with an imbalance ratio of 7.1%.

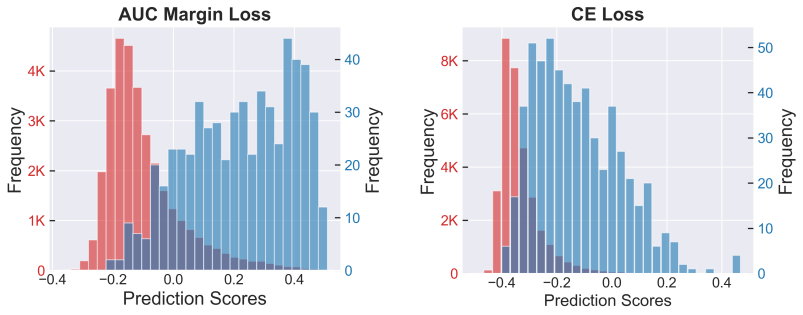

Figure 2: Prediction histogram of positive (blue) and negative (red) samples for the models trained by AUC-M loss and CE loss on Melanoma training dataset.

Comparison with Baselines. We first compare with three baselines as above, i.e., optimizing CE, Focal and AUC-S losses. We choose the family of EfficientNet [37] as the main network. Data augmentation is very crucial in this competition, and we use a set of augmentations, e.g., horizontal flipping, rotating, scaling, shearing, coarse dropout following a public notebook [8]. In addition, we use the cyclical learning rate with a base learning rate [34] of 3e-5 and a maximum learning rate of 2.4e-4 and with 8 epochs for a full cycle. We use a weight decay of 1e-5. For focal loss [26], we tune $\hat{\gamma}$={1,2,5}, $\hat{\alpha}$={0.25,0.5,0.75} and report the best result. For non-AUC losses, we train a total of 16 epochs with batch size of 256. For DAM, we start optimization from the pretrained backbone trained by optimizing the CE loss. For AUC losses, we set γ to 1/500 which is tuned by cross validation. For AUC Margin loss, we also tune $m = \{0.3, 0.5, 0.7, 1.0\}$. For experiments, we train 35 epochs in total with same batch size and initial learning rate of 0.01 decreasing by 2 times every 10 epochs using Algorithm 1. In addition, we find patient-level information (metadata) useful, e.g., age, sex, and location of imaged site. To utilize metadata, after training EfficientNet, we merge it with a 2-layer neural network (256x128) with a 0.5:0.5 weighted ratio, which is trained independently. The network structure is illustrated in Figure 9 in Appendix K.

The comparison between different methods for learning EfficientNet-B5 on resized images with a fixed resolution of 384×384 is given in Table 4. For each method, we report four numbers that represent performance on the public testing data (in early stage of competition) and private testing data (for final ranking) with/without test-time data augmentation (TTA)[33]. We can see that DAM methods improve over the standard DL methods for minimizing CE and Focal losses. In addition, the AUC Margin loss is better than AUC Square loss. We also plot the histogram of predictions on training data of our best DAM method (AUC-M+Meta) compared with standard DL method with CE loss in Figure 2. We can see that the predictions by the DAM method have two well-separated patterns corresponding to positive and negative data. In contrast, the predictions by optimizing the CE loss is more mixed together.

Competition Results. For final submission towards this competition, we use an ensemble method. We train different nets including EfficientNet (B3, B5, B6) and different resolutions , i.e., $256 \times 256, 384 \times 384, 512 \times 512, 768 \times 768$. Our final result is averaged over 10 models, which is also reported in Table 4. Our method achieves AUC scores of

Table 4: Comparison of Testing AUC on Melanoma dataset for Optimizing EfficientNetB5. TTA (30) means that predictions are averaged over 30 augmented copies of each image in test set.

	wo/ TTA		**w/ TTA(30)**	
Loss	**Public**	**Private**	**Public**	**Private**
CE	0.9391	0.9285	0.9447	0.9345
Focal	0.9412	0.9266	0.9424	0.9303
AUC-S	0.9482	0.9332	0.9502	0.9364
AUC-M	**0.9497**	**0.9357**	**0.9503**	**0.9393**
AUC-S (Meta)	0.9495	0.9358	0.9501	0.9409
AUC-M (Meta)	**0.9522**	**0.9380**	**0.9520**	**0.9423**
Our Submission	-	-	**0.9685**	**0.9438**

0.9685/0.9438 on public/private sets, which rank at **42nd/33rd** out of 3314 teams. To our best knowledge, this is also the first solution to optimize AUC in the competition. The winning team has an AUC score of 0.9490 on the private testing set [16]. We would like to emphasize that the winning team has used several useful tricks to improve the final result. In particular, they used an ensemble of 18 models and also used images at higher resolution of $896*896$. We expect these tricks can be also used for improving our results. In terms of learning a single model, our DAM method has a higher AUC score of 0.9423 than their single model's AUC score of 0.9167 (e.g., model 7 under similar configurations, e.g., EfficientNetB5, 384x384, metadata [16]). After the competition, we find the ensemble of EffecientNetB5($384 * 384$, AUC-M loss, metadata) and EffecientNetB6($512 * 512$, CE loss) achieves highest private AUC of **0.9505**.

4.3 Other Two Medical Classification Tasks

Finally, we present results on two more medical classification tasks, i.e., classification of mammogram for breast cancer screening on DDSM+ data, and classification of microscopic images for identifying tumor tissue on PathCamelyon Data. The DDSM+ data is a combination of two datasets namely DDSM and CBIS-DDSM [4, 19], which consists of 55,000 mammographic images (224×224) taken at lower doses than usual X-rays for training with imratio of 13% and 13,900 images for testing with imratio of 4%. The PathCamelyon dataset consists of 294,912 color images (96×96) extracted from histopathologic scans of lymph node section for training and 32,768 images for testing with balanced class ratio [38, 3]. For second task, we manually construct an imbalanced dataset with imratio of 1% following section 4.1. For experiments, we train DenseNet121 and use batch size of 32 for DDSM+ and 64 for PatchCamelyon. For non-AUC losses, we train models using Adam with weight decay of 1e-5 for 5 epochs. We tune learning rate {1e-1 $\sim$ 1e-5} on validation set sampled from 10% training data. For focal loss, we tune $\hat{\gamma}$={1,2,5}, $\hat{\alpha}$={0.25,0.5,0.75}. For AUC losses, we start from pretrained model of last iteration by CE loss and train a total of 1 epoch. We tune learning rate {1e-1, 1e-2, 1e-3}, γ={1/300, 1/500, 1/800} and set $\lambda = 0$. For AUC-M, we tune m={0.3, 0.5, 0.7, 1.0}. We report the best results for each method in table 5. The results indicate that AUC-M performs consistently better than baseline methods on these two datasets.

Table 5: Testing AUC of two medical datasets on DenseNet121.

Data (imratio)	CE	Focal	AUC-S	AUC-M
DDSM+ (13%)	0.9392	0.9495	0.9469	**0.9544**
PatchCamelyon (1%)	0.8394	0.8556	0.8703	**0.8896**

5. Ablation Studies

5.1 Batch Score Normalization (BSN)

We run experiments with DenseNet121 on four benchmark datasets with two imbalance ratio, e.g., 1%, 10% with and without applying batch score normalization. The results are shown in Figure 3. We can see that applying the BSN can improve the performance.

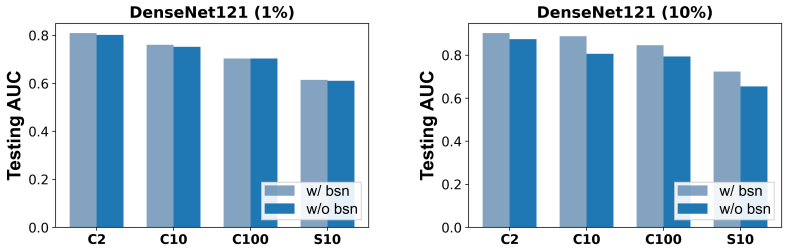

Figure 3: Ablation Study on Batch Score Normalization.

5.2 AUC-Margin Loss

Robustness to Noisy Data and Easy Data. We conduct ablation studies on the C2-IB data. To verify the robustness of our AUC-M loss to noisy data, we manually create some data with noisy labels. We construct the noisy dataset by modifying the C2 (imratio=1%). To this end, we sample 1% and 5% from negative class to flip their labels to positive, and also randomly sample 1% and 5% positive data from the deleted positive examples and flip their labels and add them to the training data. This gives us two datasets with 1% and 5% noisy ratio. To verify the robustness of our AUC loss to easy data, we first pre-train a model by minimizing CE loss on C2 (imratio=1%) and then we make predictions on the removed positive samples and sort all prediction scores in descending order. Finally, we choose top 10%, 20% of sorted samples and add them to training data. We train DenseNet121 using batch size of 128 and initial learning rate of 0.1. Other parameter settings are the same as in Section 4.1. We run experiments 5 times and plot the average testing AUC curve in Figure 4 for the setting with 1% noisy data and 10% easy data. In Figure 5, we report results on other settings. All results clearly show that AUC-M outperforms AUC-S by a large margin.

Effect of Alpha Constraint. To verify the effectiveness of non-negative constraint on α, we design an experiment to compare the performance of AUC-M with and without $\alpha \geq 0$ constraint. We start with C2-IB with imbalance ratio of 1% and add 40% easy (positive) samples and 1% noisy samples to the training set similar to that is done above. We fix margin $m = 0.1$. The curve of testing AUC and the curve of α v.s. # of epochs are plotted in Figure 4 (bottom panel). We observe that the performance with enforcing $\alpha \geq 0$

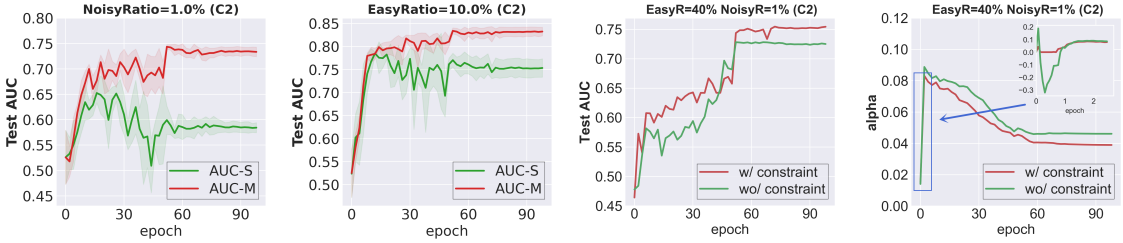

Figure 4: First two plots: comparison when adding noisy and easy samples. Last two plots: comparison between with/without $\alpha \geq 0$.

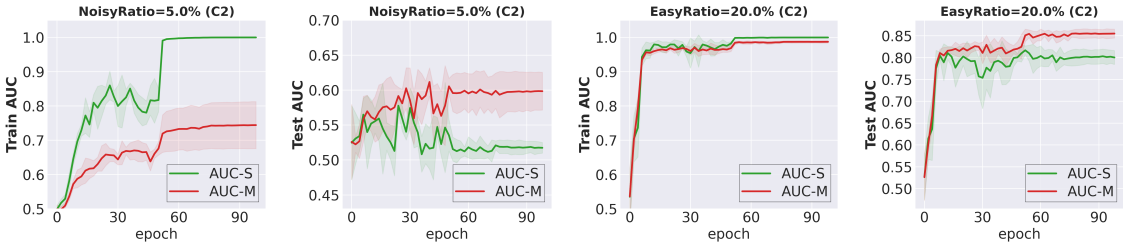

Figure 5: Comparison when adding extra 20% easy samples and 5% noisy samples.

is better than that without enforcing it. The bottom right plot in Figure 4 gives us a better illustration about the change of α during training. The plot inside it reveals the change of α in the first 2 epochs. It shows that the constraint prevents the value of α from dropping to a bad region and hence yields a faster convergence and better result.

6. Conclusion

In this paper, we have considered large-scale robust deep AUC maximization. We have proposed a new margin-based surrogate loss for AUC to address the two major issues of square loss, and demonstrated its robustness to noisy and easy data. We thoroughly evaluate our methods on four benchmark datasets and four real-world medical datasets. The results not only demonstrate the effectiveness of the new margin loss and also the success of our deep AUC maximization methods on medical image classification tasks.

Acknowledgements

We are grateful to the anonymous reviewers for their constructive comments and suggestions. This work is partially supported by TY's NSF CAREER Award 1844403.

References

[1] Deep auc maximization code. `https://github.com/Optimization-AI/ICCV2021_DeepAUC`.

[2] The international skin imaging collaboration (isic). https://www.isic-archive.com/. 2020-08.

[3] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

[4] K Bowyer, D Kopans, WP Kegelmeyer, R Moore, M Sallam, K Chang, and K Woods. The digital database for screening mammography. In *Third international workshop on digital mammography*, volume 58, page 27, 1996.

[5] Stephan Clemencon, Gabor Lugosi, and Nicolas Vayatis. Ranking and empirical minimization of u-statistics. *The Annals of Statistics*, 36(2):844–874, 2008.

[6] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[7] Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 313–320. MIT Press, 2004. URL `http://papers.nips.cc/paper/2518-auc-optimization-vs-error-rate-minimization.pdf`.

[8] Chris Deotte. Triple stratified kfold with tfrecords. In *Kaggle*, 2020.

[9] Jeremy Elson, John R Douceur, Jon Howell, and Jared Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *ACM Conference on Computer and Communications Security*, volume 7, pages 366–374, 2007.

[10] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[11] Wei Gao and Zhi-Hua Zhou. On the consistency of auc pairwise optimization. In *IJCAI*, pages 939–945. Citeseer, 2015.

[12] Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass auc optimization. In *International conference on machine learning*, pages 906–914, 2013.

[13] Stanford ML Group. Chexpert: A large chest x-ray dataset and competition. `https://stanfordmlgroup.github.io/competitions/chexpert/`, Jan 2019.

[14] Zhishuai Guo, Mingrui Liu, Zhuoning Yuan, Li Shen, Wei Liu, and Tianbao Yang. Communication-efficient distributed stochastic AUC maximization with deep neural networks. In *International Conference on Machine Learning*, 2020.

[15] Zhishuai Guo, Zhuoning Yuan, Yan Yan, and Tianbao Yang. Fast objective and duality gap convergence for non-convex strongly-concave min-max problems. *arXiv preprint arXiv:2006.06889*, 2020.

[16] Qishen Ha, Bo Liu, and Fuxu Liu. Identifying melanoma images using efficientnet ensemble: Winning solution to the siim-isic melanoma classification challenge. *arXiv preprint arXiv:2010.05351*, 2020.

[17] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Michael Heath, Kevin Bowyer, Daniel Kopans, P Kegelmeyer, Richard Moore, Kyong Chang, and S Munishkumaran. Current status of the digital database for screening mammography. In *Digital mammography*, pages 457–460. Springer, 1998.

[20] Alan Herschtal and Bhavani Raskutti. Optimising area under the roc curve using gradient descent. In *Proceedings of the twenty-first international conference on Machine learning*, page 49, 2004.

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[22] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 590–597, 2019.

[23] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384, 2005.

[24] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=r1gRTCVFvB`.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[27] Mingrui Liu, Xiaoxuan Zhang, Zaiyi Chen, Xiaoyu Wang, and Tianbao Yang. Fast stochastic auc maximization with $o(1/n)$-convergence rate. In *International Conference on Machine Learning*, pages 3189–3197, 2018.

[28] Mingrui Liu, Zhuoning Yuan, Yiming Ying, and Tianbao Yang. Stochastic auc maximization with deep neural networks. *arXiv preprint arXiv:1908.10831*, 2019.

[29] Arlo J Miller and Martin C Mihm Jr. Melanoma. *New England Journal of Medicine*, 355(1):51–65, 2006.

[30] Michael Natole, Yiming Ying, and Siwei Lyu. Stochastic proximal algorithms for auc maximization. In *International Conference on Machine Learning*, pages 3710–3719, 2018.

[31] Hieu H. Pham, Tung T. Le, Dat T. Ngo, Dat Q. Tran, and Ha Q. Nguyen. Interpreting chest x-rays via cnns that exploit hierarchical disease dependencies and uncertainty labels. In *Medical Imaging with Deep Learning*, 2020. URL `https://openreview.net/forum?id=4awO8EwEKe`.

[32] Veronica Rotemberg, Nicholas Kurtansky, Brigid Betz-Stablein, Liam Caffery, Emmanouil Chousakos, Noel Codella, Marc Combalia, Stephen Dusza, Pascale Guitera, David Gutman, et al. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *arXiv preprint arXiv:2008.07360*, 2020.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.

[35] Jeremias Sulam, Rami Ben-Ari, and Pavel Kisilev. Maximizing auc with deep learning for classification of imbalanced mammogram datasets. In *VCBM*, pages 131–135, 2017.

[36] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.

[37] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[38] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pages 210–218. Springer, 2018.

[39] Nan Wu, Jason Phang, Jungkyu Park, Yiqiu Shen, Zhe Huang, Masha Zorin, Stanisław Jastrzkebski, Thibault Févry, Joe Katsnelson, Eric Kim, et al. Deep neural networks improve radiologists' performance in breast cancer screening. *IEEE transactions on medical imaging*, 39(4):1184–1194, 2019.

[40] Wenwu Ye, Jin Yao, Hui Xue, and Yi Li. Weakly supervised lesion localization with probabilistic-cam pooling, 2020.

[41] Yiming Ying, Longyin Wen, and Siwei Lyu. Stochastic online auc maximization. In *Advances in neural information processing systems*, pages 451–459, 2016.

[42] Zhuoning Yuan, Zhishuai Guo, Yi Xu, Yiming Ying, and Tianbao Yang. Federated deep auc maximization for hetergeneous data with a constant communication complexity. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12219–12229. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/yuan21a.html`.

[43] Peilin Zhao, Steven C. H. Hoi, Rong Jin, and Tianbao Yang. Online auc maximization. In *ICML*, pages 233–240, 2011.

Appendix A. Optimal Values of a, b, α in AUC Square Loss

In Section 3.2, we use the optimal values of a, b, α. In this section, we show how to derive these values. We first re-present the min-max problem in (3) as follows

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^d \\ (a,b)\in\mathbb{R}^2}} \max_{\alpha\in\mathbb{R}} f(\mathbf{w}, a, b, \alpha) := \mathbb{E}_{\mathbf{z}}\left[F(\mathbf{w}, a, b, \alpha; \mathbf{z})\right],$$

where

$$\begin{aligned} F(\mathbf{w}, a, b, \alpha; \mathbf{z}) &= (1-p)\left(h_{\mathbf{w}}(\mathbf{x}) - a\right)^2 \mathbb{I}_{[y=1]} \\ &+ p(p(1-p) + h_{\mathbf{w}}(\mathbf{x}) - b)^2 \mathbb{I}_{[y=-1]} - p(1-p)\alpha^2 \\ &+ 2\alpha\left(p h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=-1]} - (1-p) h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=1]}\right). \end{aligned}$$

Given a fixed $\mathbf{w}$,the variable a is only involved in the first term in F, so we have the a-subproblem as

$$\begin{aligned} \min_a & \mathrm{E}_{\mathbf{z}}[(1-p)(h_{\mathbf{w}}(\mathbf{x}) - a)^2 \mathbb{I}_{[y=1]}] \\ =& (1-p)\mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x}) - a)^2] \cdot \mathrm{E}_z[\mathbb{I}_{[y=1]}] \\ =& (1-p)\mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x}) - a)^2 | y = 1] \cdot p. \end{aligned}$$

As can be seen, $\mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x}) - a)^2 | y = 1]$ achieves minimum value when $a = \mathrm{E}[h_{\mathbf{w}}(\mathbf{x}) | y = 1]$, which becomes the variance of $h_{\mathbf{w}}(\mathbf{x})$. The optimal value of $b = \mathrm{E}[h_{\mathbf{w}}(\mathbf{x}) | y = -1]$ can be achieved in the same way as a. The subproblem of α is

$$\begin{aligned} \max_\alpha & \mathrm{E}_{\mathbf{z}}[2\alpha(p(1-p) + p h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=-1]} - (1-p) h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=1]})] - p(1-p)\alpha^2 \\ =& 2\alpha(p(1-p) + p\mathrm{E}_z[h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=-1]}] - (1-p)\mathrm{E}_{\mathbf{z}}[h_{\mathbf{w}}(\mathbf{x})] \mathbb{I}_{[y=1]}) - p(1-p)\alpha^2 \\ =& 2\alpha(p(1-p) + p(1-p)\mathrm{E}_z[h_{\mathbf{w}}(\mathbf{x}) | y = -1] - p(1-p)\mathrm{E}_{\mathbf{z}}[h_{\mathbf{w}}(\mathbf{x}) | y = -1]) - p(1-p)\alpha^2 \\ =& p(1-p) \cdot (1 + 2\alpha(\mathrm{E}_z[h_{\mathbf{w}}(\mathbf{x}) | y = -1] - \mathrm{E}_{\mathbf{z}}[h_{\mathbf{w}}(\mathbf{x}) | y = -1]) - \alpha^2) \end{aligned}$$

where we can derive its optimal value simply setting its gradient as zero. This leads to

$$\begin{aligned} \alpha^* =& 1 + \mathrm{E}_z[h_{\mathbf{w}}(\mathbf{x}) | y = -1] - \mathrm{E}_{\mathbf{z}}[h_{\mathbf{w}}(\mathbf{x}) | y = -1] \\ =& 1 + b(\mathbf{w}) - a(\mathbf{w}). \end{aligned}$$

Appendix B. Reformulation of AUC Square Loss

In this section, we reformulate AUC square loss as follows

$$\begin{aligned} A_{\mathrm{S}}(\mathbf{w}) =& \mathrm{E}[(1 - h_{\mathbf{w}}(\mathbf{x}) + h_{\mathbf{w}}(\mathbf{x}'))^2 | y = 1, y' = -1] \\ =& \mathrm{E}[(1 - a(\mathbf{w}) + a(\mathbf{w}) - h(\mathbf{w}; \mathbf{x}) + h(\mathbf{w}; \mathbf{x}') - b(\mathbf{w}) + b(\mathbf{w}))^2 | y = 1, y' = -1] \\ =& \mathrm{E}[\big((a(\mathbf{w}) - h(\mathbf{w}; \mathbf{x}) + h(\mathbf{w}; \mathbf{x}') - b(\mathbf{w})) + (1 + b(\mathbf{w}) - a(\mathbf{w}))^2\big) | x = 1, y' = -1] \\ =& \mathrm{E}[(a(\mathbf{w}) - h(\mathbf{w}; \mathbf{x}) + h(\mathbf{w}; \mathbf{x}') - b(\mathbf{w}))^2 + (1 + b(\mathbf{w}) - a(\mathbf{w}))^2 \\ & + 2\big(a(\mathbf{w}) - h(\mathbf{w}; \mathbf{x}) + h(\mathbf{w}; \mathbf{x}') - b(\mathbf{w})\big) \cdot (1 + b(\mathbf{w}) - a(\mathbf{w})) | y = 1, y' = -1] \\ \overset{(e1)}{=}& \mathrm{E}[(h(\mathbf{w}; \mathbf{x}) - a(\mathbf{w}))^2 + (h(\mathbf{w}; \mathbf{x}') - b(\mathbf{w}))^2 - 2(h(\mathbf{w}; x) - a(\mathbf{w})) \cdot (h(\mathbf{w}; \mathbf{x}') - b(\mathbf{w})) \\ & + (1 + b(\mathbf{w}) - a(\mathbf{w}))^2 | y = 1, y' = -1] \\ \overset{(e2)}{=}& \mathrm{E}[(h(\mathbf{w}; \mathbf{x}) - a(\mathbf{w}))^2 | y = 1] + \mathrm{E}[(h(\mathbf{w}; \mathbf{x}') - b(\mathbf{w}))^2 | y' = -1] \\ & + (1 + b(\mathbf{w}) - a(\mathbf{w}))^2 \\ \overset{(e3)}{=}& \mathrm{E}[(h(\mathbf{w}; \mathbf{x}) - a(\mathbf{w}))^2 | y = 1] + \mathrm{E}[(h(\mathbf{w}; \mathbf{x}') - b(\mathbf{w}))^2 | y' = -1] \\ & + \max_\alpha 2\alpha(1 + b(\mathbf{w}) - a(\mathbf{w})) - \alpha^2, \end{aligned}$$

where equality $(e1)$ is due to the definitions $a(\mathbf{w}) = \mathrm{E}[h(\mathbf{w};\mathbf{x})|y=1]$ and $b(\mathbf{w}) = \mathrm{E}[h(\mathbf{w};\mathbf{x}')|y'=-1]$, $\mathrm{E}[a(\mathbf{w})] = a(\mathbf{w})$ and $\mathrm{E}[b(\mathbf{w})] = b(\mathbf{w})$ ($a(\mathbf{w})$ and $b(\mathbf{w})$ are expectations, so they are constants).Equality $(e2)$ is due to the independence of the positive and negative samples. Equality $(e3)$ is due to the convex conjugate of the square function:

$$x^2 = \max_y 2y \cdot x - y^2.$$

Appendix C. Proof of Theorem 1

Below, we start from the min-max problem and prove it is equivalent to the AUC margin loss in (6).

$$\begin{aligned}
&\min_{a,b}\max_{\alpha\geq 0}\mathrm{E}_{\mathbf{z}}[F_{\mathrm{M}}(\mathbf{w},a,b,\alpha;\mathbf{z})] \\
&= \min_{a,b}\max_{\alpha\geq 0}\mathrm{E}_{\mathbf{z}}\Bigg[(1-p)\left(h_{\mathbf{w}}(\mathbf{x})-a\right)^2\mathbb{I}_{[y=1]} + p(h_{\mathbf{w}}(\mathbf{x})-b)^2\mathbb{I}_{[y=-1]} - p(1-p)\alpha^2 \\
&\qquad + 2\alpha\left(p(1-p)m + ph_{\mathbf{w}}(\mathbf{x})\mathbb{I}_{[y=-1]} - (1-p)h_{\mathbf{w}}(\mathbf{x})\mathbb{I}_{[y=1]}\right)\Bigg] \\
&= \min_{a,b}\max_{\alpha\geq 0}\Bigg[(1-p)\mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x})-a)^2\mathbb{I}_{[y=1]}] + p\mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x})-b)^2\mathbb{I}_{[y=-1]}] - p(1-p)\alpha^2 \\
&\qquad + 2\alpha\left(p(1-p)m + p\mathrm{E}_{\mathbf{z}}[h_{\mathbf{w}}(\mathbf{x})\mathbb{I}_{[y=-1]}] - (1-p)\mathrm{E}_{\mathbf{z}}[h_{\mathbf{w}}(\mathbf{x})\mathbb{I}_{[y=1]}]\right)\Bigg] \\
&= \max_{\alpha\geq 0} \quad p(1-p)\Bigg[\mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x})-a(\mathbf{w}))^2|y=1] + \mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x})-b(\mathbf{w}))^2|y=-1] - \alpha^2 \\
&\qquad + 2\alpha\left(m+b(\mathbf{w})-a(\mathbf{w})\right)\Bigg] = p(1-p)A_{\mathrm{M}}(\mathbf{w}) \qquad (9) \\
&= \quad p(1-p)\Bigg[\mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x})-a(\mathbf{w}))^2|y=1] + \mathrm{E}_{\mathbf{z}}[(h_{\mathbf{w}}(\mathbf{x})-b(\mathbf{w}))^2|y=-1] + (m+b(\mathbf{w})-a(\mathbf{w}))_+^2\Bigg]
\end{aligned}$$

where (9) shows the equivalence between minimizing $A_{\mathrm{M}}(\mathbf{w})$ in (6) and $\min_{\mathbf{w},a,b}\max_{\alpha\geq 0}\mathrm{E}_{\mathbf{z}}[F_{\mathrm{M}}(\mathbf{w},a,b,\alpha;\mathbf{z})]$, i.e.,

$$\min_{\mathbf{w},a,b}\max_{\alpha\geq 0}\mathrm{E}_{\mathbf{z}}[F_{\mathrm{M}}(\mathbf{w},a,b,\alpha;\mathbf{z})] = p(1-p)A_{\mathrm{M}}(\mathbf{w}).$$

The last equality is to explicitly show the squared hinge loss.

Appendix D. Analysis of Adverse Effect on Easy Data of Square loss based on the min-max formulation

In particular, the gradient of $F(\mathbf{w},a,b,\alpha;\mathbf{z})$ is given by $\nabla_{\mathbf{w}}F(\mathbf{w},a,b,\alpha;\mathbf{z}) = 2(1-p)\mathbf{x}\mathbb{I}_{[y=1]}\cdot(h_{\mathbf{w}}(\mathbf{x})-a-\alpha) + 2p\mathbf{x}\mathbb{I}_{[y=-1]}\cdot(h_{\mathbf{w}}(\mathbf{x})-b+\alpha)$. When $\mathbf{z}$ is positive, the first term above is active, by plugging the optimal value of a,b,α given $\mathbf{w}$, the stochastic gradient descent update will yields an updated model as

$$\mathbf{w}_+ = \mathbf{w} - \eta 2(1-p)\mathbf{x}\mathbb{I}_{[y=1]}\cdot(h_{\mathbf{w}}(\mathbf{x})-1-b),$$

where b is the mean prediction score on negative data. When $\mathbf{x}$ is an easy positive data such that $h_{\mathbf{w}}(\mathbf{x}) - 1 - b > 0$, then $\mathbf{w}_+$ will move towards the negative direction of the positive data $\mathbf{x}$, as a result it will push the score $h_{\mathbf{w}_+}(\mathbf{x})$ on the positive data smaller than $h_{\mathbf{w}}(\mathbf{x})$, which is harmful for AUC maximization. Similarly, we have the same phenomenon when the sampled data $\mathbf{z}$ is negative.

Appendix E. A 1-Dim Example of Easy/Noisy Data for AUC Square and Margin Loss

Suppose we have a 1-dimensional AUC maximization problem with a linear model parameterized by a 1-dimensional model $\mathbf{w}$, i.e., $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$, so that $\nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}$. Recall the definition of F in (3), we have its gradient w.r.t. $\mathbf{w}$ as follows

$$\begin{aligned}\nabla_{\mathbf{w}} F(\mathbf{w}, a, b, \alpha; \mathbf{z}) =& 2(1-p)(h_{\mathbf{w}}(\mathbf{x}) - b) \cdot \nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=1]} + 2p(h_{\mathbf{w}}(\mathbf{x}) - b) \cdot \nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=-1]} \\ &+ 2\alpha(p \nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=-1]} - (1-p) \nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=1]}) \\ =& 2(1-p) \nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=1]} \cdot \underbrace{(h_{\mathbf{w}}(\mathbf{x}) - a - \alpha)}_{=B} + 2p \nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}) \mathbb{I}_{[y=-1]} \cdot \underbrace{(h_{\mathbf{w}}(\mathbf{x}) - b + \alpha)}_{=C},\end{aligned}$$

where our study focuses on the two terms B and C, which determines the direction of $\nabla_{\mathbf{w}} F$ for $y = 1$ and $y = -1$, respectively.

α is the key difference between AUC square loss in (3) and AUC margin loss in (6). To simplify the explanation, we let $a = a(\mathbf{w})$ and $b = b(\mathbf{w})$ achieve their optimal values. In AUC square loss (3), α is not constrained, and the optimal value is $\alpha = 1 + b - a$. In AUC margin loss (6), it has a non-negative constraint on α, so the optimal value is $\alpha = \max\{0, 1 + b - a\}$.

E.1 Easy Data for AUC Square Loss

At the t-th iteration, let $\mathbf{w}_t = 1$ and we have two easy data ($\mathbf{x}_1 = 1, y_1 = 1$) and ($\mathbf{x}_2 = -1, y_2 = -1$). We assume that $a = 0.5$ and $b = -0.5$.

For $(\mathbf{x}_1, y_1 = 1)$

$$B = h_{\mathbf{w}}(\mathbf{x}_1) - a - \alpha = h_{\mathbf{w}}(\mathbf{x}_1) - 1 - b = 1 \times 1 - 1 - (-0.5) = 0.5,$$

which indicates that $\nabla_{\mathbf{w}} F \propto \nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}_1)$ (they are in the same direction). By assuming all the constants and the step size can be merged into a constant value 0.1, the stochastic gradient descent can be

$$\mathbf{w}_{t+1} = \mathbf{w}_t - 0.1 \times \nabla_{\mathbf{w}} h_{\mathbf{w}_t}(\mathbf{x}_1) = 1 - 0.1 \times \mathbf{x}_1 = 1 - 0.1 \times 1 = 0.9.$$

Then we re-evaluate the prediction score by $\mathbf{w}_{t+1}$:

$$h_{\mathbf{w}_{t+1}}(\mathbf{x}_1) = 0.9 \times 1 = 0.9 < h_{\mathbf{w}_t}(\mathbf{x}_1) = 1.$$

In this case, the prediction score for a positive sample decreases, which is an undesirable update.

For $(\mathbf{x}_2, y_2 = -1)$

$$C = h_{\mathbf{w}}(\mathbf{x}_1) - b + \alpha = h_{\mathbf{w}}(\mathbf{x}_1) + 1 - a = -1 + 1 - 0.5 = -0.5,$$

which indicates that $\nabla_{\mathbf{w}} F \propto -\nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}_1)$ (they are in the negative direction of each other). By assuming all the constants and the step size can be merged into a constant value 0.1, the stochastic gradient descent can be

$$\mathbf{w}_{t+1} = \mathbf{w}_t - 0.1 \times (-1) \times \nabla_{\mathbf{w}} h_{\mathbf{w}_t}(\mathbf{x}_1) = 1 + 0.1 \times \mathbf{x}_2 = 1 + 0.1 \times (-1) = 0.9.$$

Then we re-evaluate the prediction score by $\mathbf{w}_{t+1}$:

$$h_{\mathbf{w}_{t+1}}(\mathbf{x}_2) = 0.9 \times (-1) = -0.9 > h_{\mathbf{w}_t}(\mathbf{x}_2) = -1.$$

In this case, the prediction score for a negative sample increases, which is an undesirable update.

E.2 Easy Data for AUC Margin Loss

Since the optimal $\alpha = \max\{0, m + b - a\}$, we consider the two cases, respectively.

Case 1: $\alpha = 0$**.** This case indicates that $m + b - a \leq 0$ or $m + b \leq a$, which is a good situation, because a (the mean prediction of positive data) and b (the mean prediction of negative data) are sufficiently far away from each other by a margin of m. Here for simplicity, we assume that at the t-th iteration, $\mathbf{w} = 1$, $m = 1$, $a = 1$ and $b = -0.5$.

For $(\mathbf{x}_1 = 0.75, y = 1)$:

$$B = h_{\mathbf{w}}(\mathbf{x}_1) - a - \alpha = h_{\mathbf{w}}(\mathbf{x}_1) - a = 0.75 - 1 = -0.25 \qquad \text{(negative direction)},$$

where $h_{\mathbf{w}}(\mathbf{x}_1) > m + b = 0.5$ means that $\mathbf{x}_1$ is well classified, but F_{M} still suffers a penalty on it and push it to be closer to $a = 1$.

For $(\mathbf{x}_1 = 1.25, y = 1)$:

$$B = h_{\mathbf{w}}(\mathbf{x}_1) - a - \alpha = h_{\mathbf{w}}(\mathbf{x}_1) - a = 1.25 - 1 = 0.25 \qquad \text{(negative direction)},$$

where $h_{\mathbf{w}}(\mathbf{x}_1) > m + b = 0.5$ means that $\mathbf{x}_1$ is well classified, but F_{M} still suffers a penalty on it and push it to be closer to $a = 1$. To sum up, when the model is good enough, i.e., $m + b < a$, F_{M} only push positive data towards a and negative data towards b.

Case 2: $\alpha = m + b - a$**.** This case indicates that $m + b - a > 0$ or $m + b > a$, which is a undesirable situation, because a (the mean prediction of positive data) and b (the mean prediction of negative data) are within a margin of m. Here for simplicity, we assume that at the t-th iteration, $\mathbf{w} = 1, m = 1, a = 0, b = -0.5$.

For $(\mathbf{x}_1 = 0.25, y_1 = 1)$:

$$B = h_{\mathbf{w}}(\mathbf{x}_1) - a - \alpha = h_{\mathbf{w}}(\mathbf{x}_1) - m - b = 0.25 - 1 + 0.5 = -0.25 \qquad \text{(negative direction)},$$

where $h_{\mathbf{w}}(\mathbf{x}_1) < m + b = 0.5$ means that $\mathbf{x}_1$ is not well classified. Thus, the stochastic gradient descent for updating $\mathbf{w}_t$ can be

$$\mathbf{w}_{t+1} = \mathbf{w}_t - 0.1 \times (-1) \times \nabla_w h_{\mathbf{w}_t}(\mathbf{x}_1) = 1 + 0.1 \times \mathbf{x}_1 = 1 + 0.1 \times 0.25 = 1.025,$$

which makes the prediction of $\mathbf{x}_1$ larger: $h_{\mathbf{w}_{t+1}}(\mathbf{x}_1) = 1.025 \times 0.25 = 0.2562 > h_{\mathbf{w}_t}(\mathbf{x}_1) = 0.25$.

Examples for negative data can be derived in a similar way, so we omit those presentation.

E.3 Noisy Data for AUC Square Loss

Assuming $\mathbf{w} = 1$, consider the case where $m + b > a$, i.e., the model is not good, e.g., $a = 0.25, b = -0.5$. For $(\mathbf{x}_1 = 0.25, y_1 = -1, y_1^{\text{true}} = 1)$, since only y_1 is revealed, we will use term C to determine $\nabla_{\mathbf{w}} F$. On the other hand, since $y_1^{\text{true}} = 1$, we know that $h_{\mathbf{w}}(\mathbf{x})$ can be large. Then we can compute its term C

$$C = h_{\mathbf{w}}(\mathbf{x}_1) - b + \alpha = h_{\mathbf{w}}(\mathbf{x}_1) + 1 - a = 0.25 \times 1 + 1 - 0.25 = 1 \qquad \text{(positive direction)},$$

which means that $\nabla_{\mathbf{w}} F$ is in the same direction of $\nabla_w h_{\mathbf{w}}(\mathbf{x}_1)$. It is exactly the same case in Section E.1 when $B > 0$, so it will give an undesirable update.

Negative sample ($\mathbf{x}_2 = -1, y_1 = 1, y_1^{\text{true}} = -1$) can be developed in the same way, which also gives an undesirable update.

E.4 Noisy Data for AUC Margin Loss

Assuming $\mathbf{w} = 1$, consider the case where $m + b > a$, i.e., the model is not good, and $\alpha = m + b - a$. We assume $a = 0.25, b = -0.5$. For $(\mathbf{x}_1 = 0.25, y_1 = -1, y_1^{\text{true}} = 1)$:
$C = h_{\mathbf{w}}(\mathbf{x}_1) - b + \alpha = h_{\mathbf{w}}(\mathbf{x}_1) - b + (m + b - a) = h_{\mathbf{w}}(\mathbf{x}_1) + m - a = 0.25 \times 1 + m - 0.25 = m$.
m is positive by dentition. However, unlike the previous AUC square loss where $m = 1$, in AUC margin loss m is a hyper-parameter. Even though we cannot completely resolve the noisy data issue by using AUC margin loss, we can still reduce the magnitude of update along with the wrong direction by changing m to a smaller value from constant 1.

The same situation happens for noisy negative data on the not-so-good model.

Appendix F. An Example of Sensitivity of AUC

Table 6: Illustrations of sensitivity of Accuracy and AUC on an imbalanced dataset of 25 samples with a positive ratio of 3/25. The accuracy threshold is 0.5. **Example 1** shows that all positive instances rank higher than negative instances and two negative instances are misclassified to positive class. **Example 2** shows that 1 positive instance ranks lower than 7 negative instances and 1 positive and 1 negative instances are missclassifed. **Example 3** shows that 2 positive instances rank lower than 7 negative instances, and 2 positive instances are also missclassifed as negative class. Overall, we can observe that AUC drops dramatically as the ranks of positive instances drop but meanwhile Accuracy remains unchanged.

Example 1		**Example 2**		**Example 3**	
Prediction	Ground Truth	Prediction	Ground Truth	Prediction	Ground Truth
0.9	1	0.9	1	0.9	1
0.8	1	**0.41**(↓)	1	**0.41**(↓)	1
0.7	1	0.7	1	**0.40**(↓)	1
0.6	0	0.6	0	**0.49**(↓)	0
0.6	0	**0.49**(↓)	0	**0.48**(↓)	0
0.47	0	0.47	0	0.47	0
0.47	0	0.47	0	0.47	0
0.45	0	0.45	0	0.45	0
0.43	0	0.43	0	0.43	0
0.42	0	0.42	0	0.42	0
⋮	⋮	⋮	⋮	⋮	⋮
0.1	0	0.1	0	0.1	0
Acc=0.92		Acc=0.92 (—)		Acc=0.92 (—)	
AUC=1.00		AUC=**0.89** (↓)		AUC=**0.78** (↓)	

Appendix G. Descriptions of Imbalanced Datasets

Table 7: Description of of Datasets. Note that "size of training set" refers to the number of samples for the original training set. Datasets with suffix "-IB" denote that we manually construct the imbalanced datasets by randomly removing some positive samples.

Datasets	Size of image	Size of training set
Cat&Dog-IB	low resolution	25,000
CIFAR10-IB	low resolution	50,000
CIFAR100-IB	low resolution	50,000
STL10-IB	medium resolution	5,000
PatchCamelyon-IB	medium resolution	294,912
Melanoma	high resolution	46,131
CheXpert	high resolution	223,416
DDSM+	high resolution	55,890

Appendix H. More Experiments on Benchmark Datasets

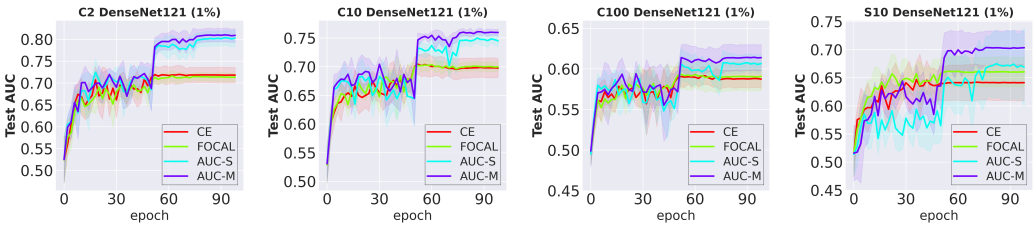

Figure 6: Testing AUC vs epochs on Benchmark Datasets for DenseNet121.

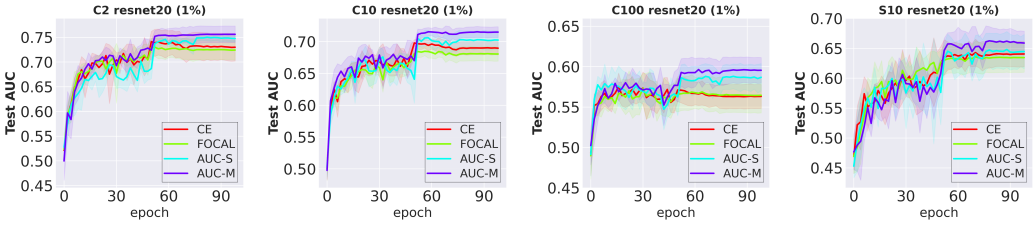

Figure 7: Testing AUC vs epochs on Benchmark Datasets for ResNet20.

Table 8: Testing AUC of benchmark datasets with DenseNet121(D) and ResNet20(R) for imratio=10%. Note that when the imbalance ratio increases e.g., from 1% to 10%, data becomes less imbalanced and the classification becomes easier.

Dataset	imratio	CE	Focal	AUC-S	AUC-M
C2 (D)	10%	0.893±0.004	0.879±0.005	0.901±0.002	**0.902±0.001**
C10 (D)	10%	**0.898±0.005**	0.879±0.005	0.889±0.002	0.887±0.005
S10 (D)	10%	0.820±0.015	0.819±0.010	0.825±0.013	**0.846±0.015**
C100 (D)	10%	0.710±0.007	0.705±0.007	0.720±0.003	**0.723±0.006**
C2 (R)	10%	0.920±0.004	0.881±0.008	0.897±0.007	**0.920±0.006**
C10 (R)	10%	**0.898**±0.004	0.851±0.018	0.872±0.007	0.898±0.005
S10 (R)	10%	**0.825**±0.013	0.813±0.009	0.819±0.013	0.821±0.011
C100(R)	10%	0.669±0.006	0.666±0.012	0.686±0.005	**0.695±0.003**

Appendix I. The Choice of Margin m for AUC-M Loss

Margin m is an important parameter for AUC-M loss. As illustrated in Section 3.3, when the model is not good enough, noisy data may produce a stochastic gradient that indicates a wrong direction. In this case, a smaller m can alleviate such sensitivity to noisy data. Tuning m parameter can trade off the margin benefit and the robustness to noisy data. That is the reason why tuning m is important in AUC-M. On benchmark datasets, the average values of m over different random trials are 0.7,0.8,0.7,0.5 on C2, C10, S10, C100, respectively. On Melanoma, the best m is 0.8. On CheXpert, the best m is 0.8 in average over 5 classes. On DDSM, the best m is 0.5. On PatchCamelyon, the best m is 0.7. For the results of ablation studies, we use $m = 0.3$ for AUC-M loss.

Appendix J. A Two-stage Training Framework for DAM

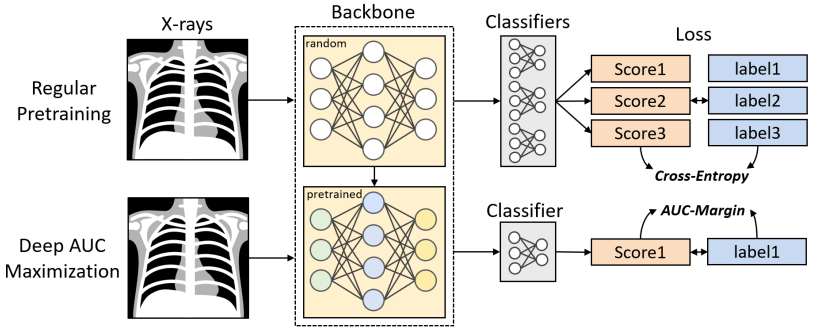

Figure 8: A Two-stage Deep AUC Maximization Framework. For the pretraining stage, we focus on learning representation by optimizing a standard CrossEntropy loss. For the AUC maximization stage, we focus on finetuning the decision boundary of classifier by optimizing AUC margin loss.

Appendix K. Network Architecture for Melanoma Classification

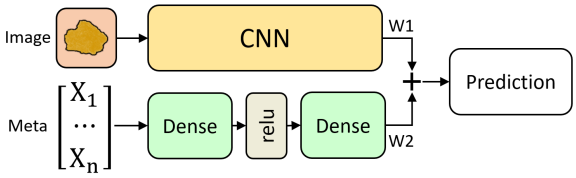

Figure 9: A mixed network architecture of a CNN (EfficientNet) and a 2-layer Neural Network for predicting Melanoma using image and patient contextual data. For training, we first train the CNN model and then train DNN model (using same configurations) but freeze the parameter updates for CNN model. The training configurations are described in main section.