# COMPOSITIONAL TRAINING FOR END-TO-END DEEP AUC MAXIMIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Recently, deep AUC maximization (DAM) has achieved great success in different domains (e.g., medical image classification). However, the end-to-end training for deep AUC maximization still remains a challenging problem. Previous studies employ an ad-hoc two-stage approach that first trains the network by optimizing a traditional loss (e.g., cross-entropy loss) and then finetunes the network by optimizing an AUC loss. This is because that training a deep neural network from scratch by maximizing an AUC loss usually does not yield a satisfactory performance. This phenomenon can be attributed to the degraded feature representations learned by maximizing the AUC loss from scratch. To address this issue, we propose a novel compositional training framework for end-to-end DAM, namely **compositional DAM**. The key idea of compositional training is to minimize a compositional objective function, where the outer function corresponds to an AUC loss and the inner function represents a gradient descent step for minimizing a traditional loss, e.g., the cross-entropy (CE) loss. To optimize the non-standard compositional objective, we propose an efficient and provable stochastic optimization algorithm. The proposed algorithm enhances the capabilities of both robust feature learning and robust classifier learning by alternatively taking a gradient descent step for the CE loss and for the AUC loss in a systematic way. We conduct extensive empirical studies on imbalanced benchmark and medical image datasets, which unanimously verify the effectiveness of the proposed method. Our results show that the compositional training approach dramatically improves both the feature representations and the testing AUC score compared with traditional deep learning approaches, and yields better performance than the two-stage approaches for DAM as well.

## 1 INTRODUCTION

Deep AUC maximization (DAM) represents a new learning paradigm for deep learning, which maximizes the area under ROC curve (AUC) on a training dataset for learning a deep neural network. It has received increasing attention recently due to the advancement in large-scale non-convex optimization algorithms for AUC maximization (Liu et al., 2019a; Yuan et al., 2021; Guo et al., 2020a;b).

Recently, DAM has been successfully applied to different domains with imbalanced data (Yuan et al., 2020; Wang et al., 2021b). For example, Yuan et al. (2020) has employed DAM for a variety of medical image classification tasks, e.g., classification of X-ray images, skin lesion images, mammograms, and microscopic images, and they observed great improvements with about 1%$\sim$5% AUC increase over traditional deep learning approaches by optimizing a standard loss function, e.g., the cross-entropy (CE) loss. These pioneering studies on DAM open a new direction for deep learning in the presence of imbalanced data but also raise many questions yet to be solved. A particular question relates to how the network is trained by DAM. Existing studies employ a two-stage approach for DAM, in which the first stage pretrains the network on the training data by optimizing a traditional loss function (e.g. the CE loss) and the second stage finetunes the network by optimizing an AUC loss. It was conjectured in (Yuan et al., 2020) the feature extraction layers learned by directly optimizing AUC from scratch are not as good as optimizing the standard CE loss, similarly as optimizing a class-weighted loss for deep learning with imbalanced data (Cao et al., 2019).

Although the ad-hoc two-stage method of DAM has achieved some success, this approach leads to several undesirable consequences increasing the engineering costs in practice: (i) which layers

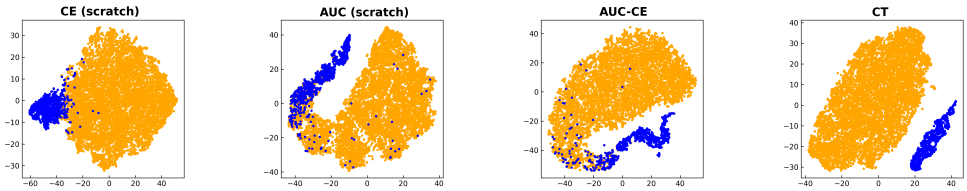


Figure 1: t-SNE visualization of feature representations of an imbalanced training set for the Catvs-Dog visualized by tSNE learned by different methods (from left to right): optimizing CE loss, an AUC loss, a linear combination of CE and AUC loss, and a compositional objective by our method. Please refer to Section 4 for details of experiment setting.

should we finetune in the second stage? Fine-tuning all layers increases the training costs but not necessarily improves the final performance (Jamal et al., 2020; Qi et al., 2020); (ii) when do we stop the training for the first-stage? A long training time for the first-stage increases the overall training costs but not necessarily increases the final prediction performance, while a short training time for the first-stage could harm the prediction performance (Kang et al., 2019). Hence, the literature has suggested different tricks for the two-stage approach, including the decoupling method that simply optimizes the classifier layer in the second-stage (Kang et al., 2019) and deferred re-weighting that only dedicates the iterations with the largest step size to the CE loss (Cao et al., 2019), which could be borrowed to DAM as well. However, an important question remains open regarding DAM:

*How can we conduct end-to-end training for deep AUC maximization?*

To answer this question, we have examined the learned feature representations by optimizing an AUC loss directly from scratch and confirmed the conjecture in (Yuan et al., 2020) that the learned feature presentations exhibit no advantage over optimizing the CE loss directly. In Figure 1 we visualize the feature representations on an imbalanced training set for the CatvsDog classification learned by different methods and visualized by t-SNE (van der Maaten & Hinton, 2008). We can see that optimizing an AUC loss from scratch (2nd column) does not yield a cleaner feature representations for the two classes of data than optimizing the CE loss. What makes end-to-end deep learning successful is its superb feature learning capability, i.e., the lower layers capture the low-level features and higher layers capture the high-level features. In terms of feature learning, different examples roughly have equal weights regardless which classes they belong to. From this perspective, we could understand why optimizing AUC loss alone bears worse feature learning capability. The AUC loss assigns different weights to different examples from different classes for more robust classifier learning. In particular, the data in the positive class has a higher weight than data in the negative class. These non-equal weights are important for learning a robust classifier given that feature representations have been learned well but are not readily helpful for learning feature representations in an end-to-end fashion. Can we achieve both effects in a unified and end-to-end learning framework, i.e., optimizing the CE loss with equal weights for robust feature learning and optimizing an AUC loss with uneven weights for robust classifier learning? A naive approach is to simply optimize a linear combination of the CE loss and an AUC loss. However, this approach has a trade-off, meaning that AUC is not necessarily maximized due to the presence of the CE loss in the objective and the learned feature representations could be degraded by the AUC loss (Figure 1, 3rd column).

In this paper, we propose a better and novel end-to-end training method that not only achieves both benefits of minimizing the CE loss for robust feature learning and minimizing an AUC loss for robust classifier learning, but also achieves the effect of "1+1>2", i.e., achieves better performance than the naive linear combination approach. The novel synthesis lies in how we composite the two training steps corresponding to the CE loss and the AUC loss. The central idea is to minimize a two-level compositional objective, where the outer function is an AUC loss, and the inner function is a gradient descent step towards minimizing the CE loss, which represents a quick adaptation to the solution to minimizing the CE loss. We propose a novel efficient stochastic algorithm with provable convergence for minimizing the compositional objective, which performs alternating gradient-based updates that are first based on the gradient of CE loss and then based on the gradient of an AUC loss at the point obtained in the first step. We summarize our contributions below.

- We propose a novel training framework for end-to-end deep AUC maximization, namely compositional DAM. The novel compositional objective enables not only the robust feature learning of

lower layers from minimizing the standard loss function but also the robust learning of a classifier from minimizing an AUC loss.

- Theoretically, we propose an efficient stochastic optimization algorithm for solving compositional DAM, and establish the same convergence rate as standard SGD for optimizing a standard averaged loss. Empirically, we conduct extensive studies on benchmark and medical image datasets, and observe that the proposed method not only improves the baseline methods including the naive linear combination approach but also improves ad-hoc two-stage approaches for DAM. The learned feature representations of compositional DAM (e.g., Figure 1 right) are much better than those learned by minimizing the CE loss or an AUC loss alone and their combination.

## 2 Related Work

**Deep AUC Maximization.** AUC maximization has a history of two decades. Most of the existing studies revolve around the design of efficient optimization algorithms. Earlier papers have focused on full batch methods (Herbrich et al., 1999; Yan et al., 2003; Ferri et al., 2002; Freund et al., 2003; Joachims, 2005; Herschtal & Raskutti, 2004; Rakotomamonjy, 2004; Zhang et al., 2012) and online optimization methods (Zhao et al., 2011; Kar et al., 2014; 2013; Gao et al., 2013). Recently, stochastic optimization for AUC maximization has become the dominating approach (Ying et al., 2016; Liu et al., 2018; Natole et al., 2018; 2019). Ying et al. (2016) propose a milestone work for stochastic optimization of AUC. They consider optimizing the pairwise square loss and propose an equivalent min-max formulation that transforms the original non-decomposable objective into a decomposable one, which enables the design of efficient stochastic methods based on mini-batch of data without explicitly constructing the pairs. The min-max formulation also serves as the basis for recent works on DAM (Liu et al., 2019a; Yuan et al., 2021; Guo et al., 2020a;b). (Liu et al., 2019a) is the first work that explicitly considers DAM and develops the first practical and provable stochastic algorithms for DAM based on the min-max formulation of the pairwise square loss function. However, this work only focuses on optimization and experiments are done on simple benchmark datasets. Later, Yuan et al. (2020) propose a new robust loss in the min-max form for DAM and evaluates the performance of DAM on various medical image classification tasks, which demonstrates great success of DAM. However, none of these works have addressed the problem of end-to-end training for DAM.

**Deep Learning with Imbalanced data.** Deep learning in the presence of imbalanced data has recently attracted tremendous attention (Cui et al., 2019; Johnson & Khoshgoftaar, 2019; Masko & Hensman, 2015; Lee et al., 2016; Khan et al., 2017; Dablain et al., 2021; Ren et al., 2018; Jamal et al., 2020; Qi et al., 2020; Lin et al., 2017; Cao et al., 2019; Kang et al., 2019; Liu et al., 2019b; Zhu & Yang, 2020; Zhou et al., 2020; Xiang et al., 2020; Wang et al., 2021a; 2020; Menon et al., 2021). Among these studies that are closely related to our work include (Lin et al., 2017; Cui et al., 2019; Cao et al., 2019; Qi et al., 2020; Kang et al., 2019; Jamal et al., 2020), which focus on optimizing different objectives from the standard CE loss, including class-weighted loss, focal loss, individually weighted loss functions, etc. Nevertheless, these works are not directly comparable to our method for maximizing AUC.

**Two-stage Approaches.** However, directly optimizing a weighted loss for training a deep neural network from scratch does not work well (Cao et al., 2019; Kang et al., 2019; Jamal et al., 2020; Qi et al., 2020; Yuan et al., 2020). This phenomenon was first observed in (Cao et al., 2019), which is attributed to the degraded feature representations. To tackle this issue, Cao et al. (2019) propose a deferred re-weighting/re-sampling trick. It minimizes the standard average loss for the first stage and switches to minimizing the class-weighted loss or re-sampling method in the second stage. In their paper, the first stage is defined as the training period from the beginning to the iteration that the step size was reduced for the first time in SGD or momentum methods. Kang et al. (2019) investigate a decoupling approach, where the first stage learns the feature representations (i.e., a feature extraction network) by optimizing a standard loss with a large number of iterations, and the second stage learns a robust classifier (i.e. the classifier layer). The authors show that the decoupling approach can achieve better performance than the deferred re-weighting trick in (Cao et al., 2019). However, the decoupling approaches do not necessarily yield the best performance. In particular, some studies have found that fine-tuning some higher layers besides the classifier layer in the second stage is beneficial (Jamal et al., 2020; Qi et al., 2020). Recently, Yang & Xu (2020) propose to use self-supervised learning for learning the feature representations for the first stage and to switch to re-weighting method in the second stage. Different from these studies, our work is to design an elegant end-to-end training framework for deep learning with an AUC loss.

# 3 COMPOSITIONAL TRAINING FOR DEEP AUC MAXIMIZATION

**Notations.** We use $(\mathbf{x}, y)$ to denote an example, where $\mathbf{x} \in \mathbb{R}^{d_0}$ denotes the input and $\mathbf{y} \in \mathcal{Y}$ denotes its corresponding label. Let $\|\cdot\|$ denote the Euclidean norm of a vector, and let $\mathbf{w} \in \mathbb{R}^d$ denote the weight parameters of a deep neural network. A function $F(\mathbf{w})$ is called $L$-smooth if its gradient is $L$-Lipschitz continuous, i.e., $\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')\| \leq L\|\mathbf{w} - \mathbf{w}'\|$. Let $f(\mathbf{w}; \mathbf{x})$ denote the prediction scores of a deep neural network parameterized by $\mathbf{w}$ on an input $\mathbf{x}$, where $f(\mathbf{w}; \mathbf{x}) \in \mathbb{R}$ for binary classification with $\mathcal{Y} = \{1, -1\}$. Denote by $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ a set of $n$ training examples. Let $\ell(\mathbf{w}; \mathbf{x}, y)$ denote a loss function on an individual data, e.g., cross-entropy loss. We let $L(\mathbf{w}; \mathcal{S})$ denote an aggregate loss function defined on a set of samples $\mathcal{S} \subseteq \mathcal{D}$. When $\mathcal{S} = \mathcal{D}$, we simply use the notation $L(\mathbf{w}) = L(\mathbf{w}; \mathcal{D})$. Let $\Pi_\Omega[\theta]$ denotes an Euclidean projection on the set $\Omega$. Denote by $n_+(n_-)$ the number of positive (negative) examples.

A standard approach of deep learning is to minimize an averaged loss on training examples, i.e.,

$$\min_{\mathbf{w} \in \mathbb{R}^d} L_{\text{AVG}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}; \mathbf{x}_i, y_i). \tag{1}$$

**AUC losses.** AUC (area under the ROC curve) is a commonly used measure for evaluating classifiers for binary classification with imbalanced data. Recently, there emerge voluminous studies on optimizing AUC score for learning a predictive model (e.g., a deep neural network). The idea is to optimize a surrogate loss for the AUC score. A special surrogate loss is the AUC square loss (Gao & Zhou, 2015), which is defined as:

$$\min_{\mathbf{w}} \frac{1}{n_+ n_-} \sum_{y_i=1} \sum_{y_j=-1} (c - (f(\mathbf{w}; \mathbf{x}_i) - f(\mathbf{w}; \mathbf{x}_j)))^2,$$

where $c$ is a margin parameter (e.g., 1). Since directly optimizing the above pairwise loss is computationally expensive, existing works transform the above problem into an equivalent min-max optimization (Liu et al., 2019a), which is decomposable over individual examples:

$$\min_{\mathbf{w}, a, b} \max_{\theta \in \Omega} \Phi(\mathbf{w}, a, b, \theta) := \frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{w}, a, b, \theta; \mathbf{x}_i, y_i), \quad \text{where} \tag{2}$$

$$\begin{aligned} \phi(\mathbf{w}, a, b, \theta; \mathbf{x}_i, y_i) &= (1-p)(f(\mathbf{w}; \mathbf{x}_i) - a)^2 \mathbb{I}_{[y_i=1]} + p(f(\mathbf{w}; \mathbf{x}_i) - b)^2 \mathbb{I}_{[y_i=-1]} \\ &- p(1-p)\theta^2 + 2\theta \left(p(1-p)c + pf(\mathbf{w}; \mathbf{x}_i)\mathbb{I}_{[y_i=-1]} - (1-p)f(\mathbf{w}; \mathbf{x}_i)\mathbb{I}_{[y_i=1]}\right), \end{aligned} \tag{3}$$

$\Omega = \mathbb{R}$ and $p = n_+/n$. From the above objective function $\phi$, we can see that each example $\mathbf{x}_i$ also has a class-level weight for their contributed loss, i.e., the data from the positive class is weighted by $1 - p$ and the data from the negative class is weighted by $p$.

It was recently shown that the AUC square loss is sensitive to noisy data and also has adverse effect when trained with easy data. Hence, Yuan et al. (2020) proposed the min-max AUC margin (AUCM) loss, whose optimization problem is (2) with $\Omega = \{0 \leq \theta \leq u\}$ for some $u$. Let us define $L_{\text{AUC}}(\mathbf{w}) = \min_{a,b} \max_{\theta \in \Omega} \Phi(\mathbf{w}, a, b, \theta)$ as the AUC loss function.

## 3.1 COMPOSITIONAL DAM: A COMPOSITIONAL TRAINING METHOD FOR DAM

In this section, we present the proposed compositional training for DAM. Our proposed objective for end-to-end deep learning is given by

$$\min_{\mathbf{w} \in \mathbb{R}^d} L_{\text{AUC}}(\mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w})), \tag{4}$$

where $\alpha$ is hyper-parameter. Different from $L_{\text{AUC}}(\mathbf{w})$, the above objective is a compositional function, where the inner component $\mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w})$ is another function of $\mathbf{w}$. We refer to the above objective as the compositional objective and a method for minimizing the above compositional objective as compositional training.

To understand the compositional objective (4), we take the second-order Taylor expansion of the compositional objective, which include three terms:

$$L_{\text{AUC}}(\mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w})) \approx L_{\text{AUC}}(\mathbf{w}) - \alpha \nabla L_{\text{AUC}}(\mathbf{w})^\top \nabla L_{\text{AVG}}(\mathbf{w}) + C\alpha^2/2 \|\nabla L_{\text{AVG}}(\mathbf{w})\|^2, \tag{5}$$

where $C$ represents the Lipchitz continuity constant of $\nabla L_{\text{AUC}}(\cdot)$. In order to understand how the three terms play their roles and evolve in the optimization process by our proposed algorithm presented in next subsection (Algorithm 1), we conduct some empirical studies on several benchmark

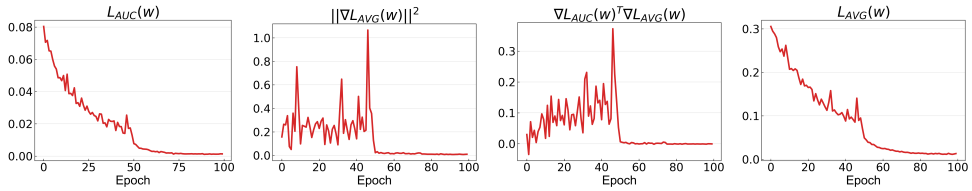


Figure 2: Evolution of different terms in (5) computed in the process of our optimization algorithm (Algorithm 1) on the CatvsDog data. $L_{\text{AVG}}$ is the averaged CE loss. Please refer to Appendix A.5 for more details of the calculations.

datasets reported in Appendix A.5. Here, we explain the result of the CatvsDog classification shown in Figure 2. Initially, the first term $L_{\text{AUC}}(\mathbf{w})$ dominates the objective and the algorithm will focus on pushing this term to be smaller (1st column), once it reaches the same level of the third term the algorithm will shift its focus to push $\|\nabla L_{\text{AVG}}(\mathbf{w})\|$ smaller (2nd column) while keeping $\nabla L_{\text{AUC}}(\mathbf{w})^\top \nabla L_{\text{AVG}}(\mathbf{w})$ to be positive (3rd column). This process will continue by alternating between the efforts of pushing $L_{\text{AUC}}(\mathbf{w})$ smaller and of pushing $\|\nabla L_{\text{AVG}}(\mathbf{w})\|^2$ to be smaller while keeping $\nabla L_{\text{AUC}}(\mathbf{w})^\top \nabla L_{\text{AVG}}(\mathbf{w})$ to be non-negative. We also notice that the final AUC loss is close to zero. The same phenomena are also observed on other datasets.

To further understand the compositional training intuitively, let us take a thought experiment by using a gradient descent method to optimize the compositional objective. First, we evaluate the inner function by $\mathbf{u} = \mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w})$. We can see that $\mathbf{u}$ is computed by a gradient descent step for minimizing the averaged loss $L_{\text{AVG}}(\mathbf{w})$, which facilitates the learning of lower layers for feature extraction due to equal weights of all examples. Then, we take a gradient descent step to update $\mathbf{w}$ for minimizing the outer function $L_{\text{AUC}}(\cdot)$ by using the gradient $\nabla L_{\text{AUC}}(\mathbf{u})$ instead of $\nabla L_{\text{AUC}}(\mathbf{w})$. Because $\mathbf{u}$ is better than $\mathbf{w}$ in terms of feature extraction layers, taking a gradient descent step using $\nabla L_{\text{AUC}}(\mathbf{u})$ would be better than using $\nabla L_{\text{AUC}}(\mathbf{w})$. In addition, taking a gradient descent step for the outer function $L_{\text{AUC}}(\cdot)$ will make the classifier more robust to the minority class due to the higher weights of examples from the minority class. Overall, we have **two alternating conceptual steps**, i.e., the inner gradient descent step $\mathbf{u} = \mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w})$ acts as a **feature purification step**, and the outer gradient descent step $\mathbf{w} - \eta(I - \alpha \nabla^2 L_{\text{AVG}}(\mathbf{w}))\nabla L_{\text{AUC}}(\mathbf{u})$ acts as a **classifier robustification** step, where $\eta$ is a step size.

**VS. linear combination approach.** It is notable that minimizing the compositional objective $L_{\text{AUC}}(\mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w}))$ is different from minimizing a linear combination of an AUC loss and the averaged loss, i.e., $L_{\text{AUC}}(\mathbf{w}) + cL_{\text{AVG}}(\mathbf{w})$, where $c > 0$ is a combination weight. First, minimizing the latter objective will push $\nabla L_{\text{AUC}}(\mathbf{w}) + c\nabla L_{\text{AVG}}(\mathbf{w}) = 0$. This makes $\nabla L_{\text{AUC}}(\mathbf{w})$ to have an opposite direction from $\nabla L_{\text{AVG}}(\mathbf{w})$ at the optimal solution, which is different from minimizing the compositional objective in light of the three terms in (5). Second, if we take a gradient descent method for minimizing this objective, the update of $\mathbf{w}$ is given by $\mathbf{w} - \eta(\nabla L_{\text{AUC}}(\mathbf{w}) + c\nabla L_{\text{AVG}}(\mathbf{w}))$. This update is fundamentally different from the two alternating steps of compositional training that use gradients of the AUC loss and the average loss at different points. Third, minimizing the linear combination has a trade-off, meaning that the AUC score is not necessarily maximized due to the presence of the CE loss in the objective and the learned feature representations are degraded due to the presence of AUC loss.

**More discussions.** Finally, we note that the inner gradient descent step $\mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w})$ is similar to the idea used in model-agnostic meta learning (MAML) (Finn et al., 2017). However, our compositional objective works fundamentally different from that for MAML. In MAML, the outer loss function and the loss function for the inner gradient descent step is the same. In contrast, the two loss functions in our objective are different. But similar to MAML approaches, we can also run multiple gradient descent steps for the inner function, i.e, the inner function $\mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w})$ can be replaced by multiple gradient descent steps. In our experiments, we also found this trick to be helpful for improving the performance.

### 3.2 Stochastic Optimization Algorithms

In this subsection, we develop efficient stochastic optimization algorithms for optimizing the compositional objective (4) for DAM. First, we argue the necessity for such development. (i) the problem is a min-max form and the objective is a compositional function, which makes computing an unbiased stochastic gradient of the objective $L_{\text{AUC}}(\mathbf{w} - \alpha \nabla L_{\text{AVG}}(\mathbf{w}))$ impossible. Existing algorithms of

**Algorithm 1** Primal-Dual Stochastic Compositional Adaptive (PDSCA) method for solving (6)

1: Require Parameters: $\beta_0, \beta_1, \beta_2, \alpha, G_0, \eta_1, \eta_2, \tau$
2: Initialization: $\bar{\mathbf{w}}_0 = (\mathbf{w}_0; a_0; b_0) \in \mathbb{R}^{d+2}, \theta_0, \mathbf{u}_0 \in \mathbb{R}^{d+2}, v_0 \in \mathbb{R}$
3: **for** $t = 0, 1, ..., T$ **do**
4: Sample two sets of examples denoted by $\mathcal{S}_1, \mathcal{S}_2$
5: $\mathbf{u}_{t+1} = (1-\beta_0)\mathbf{u}_t + \beta_0 h(\bar{\mathbf{w}}_t; \mathcal{S}_1)$
6: $\mathcal{O}_t = \nabla_{\bar{\mathbf{w}}} h(\bar{\mathbf{w}}_t; \mathcal{S}_1)^\top [\nabla_{\mathbf{u}} g_1(\mathbf{u}_{t+1}; \mathcal{S}_2) + \theta_t \nabla_{\mathbf{u}} g_2(\mathbf{u}_{t+1}; \mathcal{S}_2)]$
7: $\mathbf{z}_{t+1} = (1-\beta_1)\mathbf{z}_t + \beta_1 \mathcal{O}_t$
8: $\mathbf{z}_{2,t+1} = h_t(\{\mathcal{O}_j, j = 0, \ldots, t)\}$ $\diamond h_t$ can be implemented by that in Appendix B
9: $\bar{\mathbf{w}}_{t+1} = \bar{\mathbf{w}}_t - \eta_1 \frac{\mathbf{z}_{t+1}}{\sqrt{\mathbf{z}_{2,t+1}} + G_0}$ $\diamond$with the simplest form $h_t = 1$
10: $v_{t+1} = (1-\beta_2)v_t + \beta_2 g_2(\mathbf{u}_{t+1}; \mathcal{S}_1 \cup \mathcal{S}_2)$
11: $\theta_{t+1} = (1-\eta_1)\theta_t + \eta_2 \Pi_\Omega[\theta_t + \tau(v_{t+1} - \nabla g_3(\theta_t))]$
12: **end for**

non-convex min-max optimization for DAM that focus on minimizing $L_{\text{AUC}}(\mathbf{w})$ (Liu et al., 2019a; Yuan et al., 2021; Guo et al., 2020a;b) are not applicable due to the presence of inner function $\mathbf{w} - \alpha\nabla L_{\text{AVG}}(\mathbf{w})$. (ii) Our objective is also different from that of MAML due to that $L_{\text{AUC}}(\cdot)$ is a min-max form, which renders existing algorithms for MAML (Finn et al., 2017; Fallah et al., 2020) not applicable. Hence, below we propose an efficient stochastic algorithm for solving the compositional training for DAM whose objective is of the min-max compositional form, and establish its convergence rate similar to that of standard SGD for minimizing the standard averaged loss.

In particular, for the considered AUC loss, the compositional objective becomes:

$$\min_{\mathbf{w},a,b} \max_{\theta\in\Omega} \Phi\left(\mathbf{w} - \alpha\nabla L_{\text{AVG}}(\mathbf{w}), a, b, \theta\right) = \frac{1}{n}\sum_{i=1}^{n} \phi\left(\mathbf{w} - \alpha\nabla L_{\text{AVG}}(\mathbf{w}), a, b, \theta; \mathbf{x}_i, y_i\right). \quad (6)$$

We denote by a tuple $\bar{\mathbf{w}} = (\mathbf{w}; a; b)$. For simplicity of presentation, we write $\phi(\mathbf{w}, a, b, \theta, \mathbf{x}_i, y_i)$ as

$$\phi(\bar{\mathbf{w}}, \theta; \mathbf{x}_i, y_i) = g_1(\bar{\mathbf{w}}; \mathbf{x}_i, y_i) + \theta g_2(\bar{\mathbf{w}}; \mathbf{x}_i, y_i) - g_3(\theta),$$

where

$$\begin{aligned} g_1(\bar{\mathbf{w}}; \mathbf{x}_i, y_i) =& (1-p)\left(f(\mathbf{w}; \mathbf{x}_i) - a\right)^2 \mathbb{I}_{[y_i=1]} + p(f(\mathbf{w}; \mathbf{x}_i) - b)^2 \mathbb{I}_{[y_i=-1]} \\ &+ 2pf(\mathbf{w}; \mathbf{x}_i)\mathbb{I}_{[y_i=1]} - 2(1-p)f(\mathbf{w}; \mathbf{x}_i)\mathbb{I}_{[y_i=-1]}, \end{aligned} \quad (7)$$

and $g_2(\bar{\mathbf{w}}; \mathbf{x}_i, y_i) = 2\left(pf(\mathbf{w}; \mathbf{x}_i)\mathbb{I}_{[y_i=-1]} - (1-p)f(\mathbf{w}; \mathbf{x}_i)\mathbb{I}_{[y_i=1]}\right)$ and $g_3(\theta) = p(1-p)\theta^2$.

Denote by $g_1(\bar{\mathbf{w}}; \mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} g_1(\bar{\mathbf{w}}; \mathbf{x}_i, y_i)$, $g_2(\bar{\mathbf{w}}; \mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} g_2(\bar{\mathbf{w}}; \mathbf{x}_i, y_i)$. Let $h(\bar{\mathbf{w}}) = (\mathbf{w} - \alpha\nabla L_{\text{AVG}}(\mathbf{w}); a; b)$, $\nabla_{\bar{\mathbf{w}}} h(\bar{\mathbf{w}}) = (I - \alpha\nabla^2_{\mathbf{w}} L_{\text{AVG}}(\mathbf{w}); 1; 1)$, and $h(\bar{\mathbf{w}}; \mathcal{S}) = (\mathbf{w} - \alpha\nabla L_{\text{AVG}}(\mathbf{w}; \mathcal{S}); a; b)$.

We propose a primal-dual stochastic algorithm shown in Algorithm 1, which is referred to as PDSCA. We provide some explanations of our algorithmic design. First, the step 5 of updating $\mathbf{u}_{t+1}$ corresponds to feature purification step. We use a moving average technique to update $\mathbf{u}_{t+1}$ that takes all historical updates into account, which is inspired by existing stochastic algorithms for optimizing compositional functions (Wang et al., 2017). This is important for us to prove the convergence rate of $O(1/\sqrt{T})$ without using a large batch size at each iteration. If we simply using $\mathbf{u}_{t+1} = h(\bar{\mathbf{w}}_t; \mathcal{S}_1)$ (i.e., setting $\beta_0 = 1$) to estimate $h(\bar{\mathbf{w}}_t)$, there will be a large error in estimating the gradient $\nabla_{\mathbf{u}} g_1(\mathbf{u}_{t+1}; \mathcal{S}_2)$ and $\nabla_{\mathbf{u}} g_2(\mathbf{u}_{t+1}; \mathcal{S}_2)$ in step 6. Second, the step 6 is to estimate the gradient of the outer function. We use two independent mini-batches $\mathcal{S}_1, \mathcal{S}_2$ to ensure that $\mathcal{O}_t$ is an unbiased estimator of $\nabla h(\bar{\mathbf{w}}_t)^\top \nabla_{\bar{\mathbf{w}}} \phi(\mathbf{u}_{t+1}, \theta)$. Using two independent mini-batches is also helpful for improving generalization as demonstrated in experiments. Third, the steps 7 - 9 are similar to the momentum and adaptive methods for updating the model parameter. The step 8 is used for computing the adaptive step size $1/\sqrt{\mathbf{z}_{2,t+1}} + \epsilon_0$, which is similar to adaptive methods used for deep learning, such as Adam, AMSGrad, AdaBound (Kingma & Ba, 2015; Reddi et al., 2018; Luo et al., 2019). We use a general function $h_t$ in the algorithm, which can be implemented by different methods corresponding to different adaptive step size choices. We present different $h_t$ in the Appendix B. The simplest one $h_t = 1$ corresponds to that we do not use adaptive step size and only use the momentum update. Fourth, the steps 10 - 11 are for updating the dual variable $\theta$ using a momentum method. Finally, we point out that PDSCA is similar to some existing non-convex strongly-concave min-max optimization algorithms (Guo et al., 2021) but with additional care on the inner gradient descent step $\mathbf{w} - \alpha\nabla L_{\text{AVG}}(\mathbf{w})$. We present an informal convergence of PDSCA below.

**Theorem 1.** *(Informal) Under appropriate conditions on the functions $L_{AVG}$, $g_1, g_2$ and a boundness condition on $\theta_t, \mathbf{w}_t, \nabla\ell(\mathbf{w}_t; \mathbf{x}, y), \nabla^2\ell(\mathbf{w}_t; \mathbf{x}, y)$, with $\beta_0, \beta_1, \beta_2 = O(1/\sqrt{T}), \eta_1, \eta_2 = O(1/\sqrt{T})$ and a small constant $\tau$, Algorithm 1 ensures that $\mathbb{E}\left[\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla F(\bar{\mathbf{w}}_t)\|^2\right] \leq O(\frac{1}{\sqrt{T}})$. where $F(\bar{\mathbf{w}}) = \max_{\theta\in\Omega}\Phi\left(\mathbf{w} - \alpha\nabla L_{AVG}(\mathbf{w}), a, b, \theta\right)$.*

**Remark:** We will present the detailed conditions in the supplement when proving the above theorem due to limit of space. The above theorem indicates that we can optimize the compositional objective (6) with the same convergence rate as optimizing the averaged loss (1) for deep learning.

**Practical Implementations.** It is notable that $\nabla h(\bar{\mathbf{w}}, \mathcal{S}_1) = (I - \alpha\nabla^2 L(\mathbf{w}; \mathcal{S}_1); 1, 1)$ (step 6) involves the Hessian matrix $\nabla^2 L(\mathbf{w}; \mathcal{S}_1)$. Indeed, we only need to compute the Hessian vector product involving in step 6. Similar computation occurs in the meta learning algorithms (Finn et al., 2017; Fallah et al., 2020). Inspired by practical implementations of MAML (Finn et al., 2017) that simply ignore the second-order term, we use the same trick in our experiments. An additional useful trick inspired by MAML is that we can take $k \geq 1$ gradient descent steps for the inner function, correspondingly we maintain and update several $\mathbf{u}$ variables similar to step 5, i.e., using $h(\bar{\mathbf{w}}_t, \mathcal{S}_1)$ for updating the first $\mathbf{u}_{t+1}^{(1)}$, and using $h(\mathbf{u}_{t+1}^{(1)}; \mathcal{S}_1)$ for updating second $\mathbf{u}_{t+1}^{(2)}$, and so on so forth. In our experiments, we found that tuning $k \in \{1, 2, 3\}$ is useful.

Finally, it is notable that although we focus on optimizing AUC loss for binary classification in this work, our compositional training method can be also extended to optimize other weighted losses in an end-to-end fashion, and we include some discussion and results in the Appendix D.

## 4 EXPERIMENTS

In this section, we present some experimental results. We choose five baselines: optimizing the AUC loss from scratch (AUC$^{sc}$), optimizing the CE loss (CE), optimizing a linear combination of the AUC loss and the CE loss with a tuned weight (AUC-CE), the two-stage method with deferred re-weighting trick (Cao et al., 2019) (TS-DRW), the two-stage method by decoupling the learning of feature network by minimizing CE loss and the learning of a classifier by minimizing the AUC loss (TS-DEC) (Kang et al., 2019). We denote our method by CT (AUC). For AUC loss, we use AUCM loss with the margin parameter fixed to be 1 (Yuan et al., 2020). We conduct experiments on four benchmark datasetes and four medical image datasets. The statistics of these datasets are included in the Appendix A.1. More training configurations can be found in Appendix A.2.

**Benchmark datasets.** We choose four benchmark image classification datasets, namely CatvsDog, CIFAR10 (C10), CIFAR100 (C100), and STL10 (S10). For AUC maximization, we construct imbalanced binary versions of these datasets by varying the imbalanced ratios (the ratio of positive examples to the total number of training examples) similar to (Yuan et al., 2020). We use ResNet20 as the prediction network. The weight decay is set to 1e-4 for all experiments. For algorithms to maximize AUC, we use a batch size = 128 and train a total of 100 epochs, and we use step size 0.1 and decrease it by 10 times at 50% and 75% of total training time. We tune the beta parameters of our method in a range $[0.1, 0.99]$ with a grid search and find that good values are around 0.9. For linear combination methods, we tune the weight $c$ of two losses in $\{0.25, 0.5, 0.75\}$. We tune the number of inner gradient steps for CT in $k \in \{1, 2, 3\}$ with $\alpha = 0.1$. For all benchmark data, we run three times for different random seeds and compute the mean and standard deviations.

**Medical image datasets.** We also conduct experiments on naturally imbalanced medical datasets. We choose four medical image datasets, namely Melanoma data, CheXpert, DDSM+, and PatchCam data. The Melanoma dataset is from the Kaggle 2020 competition (Rotemberg et al., 2021), which contains 33,126 labeled images in training set, including 584 positive samples and 32,542 negative samples. We manually construct training, validation and testing datasets following 70/10/20 split. For this dataset, we use the images with 256x256 resolution in the experiments. CheXpert is a large-scale chest X-ray dataset (Irvin et al., 2019), which has 224,316 images with 224 x 224 resolution. The dataset contains 5 binary classification tasks corresponding to 5 diseases, i.e., Cardiomegaly (C0), Edema , Consolidation, Atelectasis, Pleural Effusion. We evaluate the performance on the official validation set consisting of 200 patient studies and report the averaged AUC scores of all 5 diseases. The DDSM+ data is a combination of two datasets namely DDSM and CBIS-DDSM (Bowyer et al., 1996; Heath et al., 1998), which consists of 55,000 mammographic training images (224×224) with an imratio of 13% and 13,900 images for testing with an imratio of 4%. The PatchCamelyon dataset consists of 294,912 color images (96×96) extracted from histopathologic scans

Table 1: Testing performance on benchmark datasets and medical datasets. The percentage number is the second row denotes the imbalanced ratio (cf the text).

| Datasets | For **AUC** Maximization | | | |
|---|---|---|---|---|
| | **imratio** | **1%** | **10%** | **30%** |
| CATvsDOG | CE | 0.742±0.003 | 0.917±0.006 | 0.957±0.001 |
| | $\text{AUC}^{\text{sc}}$ | 0.753±0.003 | 0.915±0.002 | 0.964±0.003 |
| | AUC-CE | 0.770±0.007 | 0.939±0.004 | 0.974±0.003 |
| | TS-DRW | 0.750±0.009 | 0.914±0.003 | 0.961±0.001 |
| | TS-DEC | 0.754±0.010 | 0.918±0.003 | 0.963±0.001 |
| | **CT (AUC)** | **0.789±0.008** | **0.946±0.002** | **0.977±0.001** |
| CIFAR10 | CE | 0.689±0.003 | 0.901±0.002 | 0.944±0.001 |
| | $\text{AUC}^{\text{sc}}$ | 0.728±0.002 | 0.905±0.002 | 0.946±0.001 |
| | AUC-CE | 0.735±0.003 | 0.928±0.001 | 0.957±0.001 |
| | TS-DRW | 0.708±0.002 | 0.896±0.002 | 0.946±0.003 |
| | TS-DEC | 0.707±0.002 | 0.897±0.002 | 0.944±0.001 |
| | **CT (AUC)** | **0.739±0.004** | **0.935±0.001** | **0.964±0.001** |
| STL10 | CE | 0.655±0.005 | 0.819±0.004 | 0.885±0.004 |
| | $\text{AUC}^{\text{sc}}$ | 0.665±0.005 | 0.805±0.017 | 0.887±0.007 |
| | AUC-CE | 0.668±0.007 | 0.836±0.006 | 0.905±0.001 |
| | TS-DRW | 0.655±0.004 | 0.803±0.013 | 0.887±0.002 |
| | TS-DEC | 0.661±0.002 | 0.816±0.007 | 0.882±0.007 |
| | **CT (AUC)** | **0.673±0.010** | **0.837±0.006** | **0.906±0.001** |
| CIFAR100 | CE | 0.586±0.001 | 0.691±0.005 | 0.758±0.004 |
| | $\text{AUC}^{\text{sc}}$ | 0.606±0.004 | 0.705±0.003 | 0.779±0.003 |
| | AUC-CE | 0.605±0.004 | 0.716±0.003 | 0.795±0.001 |
| | TS-DRW | 0.588±0.003 | 0.691±0.004 | 0.762±0.001 |
| | TS-DEC | 0.587±0.001 | 0.692±0.003 | 0.762±0.002 |
| | **CT (AUC)** | **0.609±0.002** | **0.725±0.001** | **0.809±0.002** |

| Datasets | For **AUC** Maximization | |
|---|---|---|
| | **Method** | **AUC** |
| Melanoma | CE | 0.8772 |
| | $\text{AUC}^{\text{sc}}$ | 0.8699 |
| | AUC-CE | 0.8849 |
| | TS-DRW | 0.8862 |
| | TS-DEC | 0.8812 |
| | **CT (AUC)** | **0.9050** |
| CheXpert | CE | 0.8885 |
| | $\text{AUC}^{\text{sc}}$ | 0.9006 |
| | AUC-CE | 0.9053 |
| | TS-DRW | 0.9022 |
| | TS-DEC | 0.9001 |
| | **CT (AUC)** | **0.9146** |
| DDSM+ | CE | 0.9504 |
| | $\text{AUC}^{\text{sc}}$ | 0.9276 |
| | AUC-CE | 0.9553 |
| | TS-DRW | 0.9387 |
| | TS-DEC | 0.9404 |
| | **CT (AUC)** | **0.9814** |
| PatchCam | CE | 0.8661 |
| | $\text{AUC}^{\text{sc}}$ | 0.8612 |
| | AUC-CE | 0.8625 |
| | TS-DRW | 0.8739 |
| | TS-DEC | 0.8576 |
| | **CT (AUC)** | **0.8888** |

of lymph node section for training and 32,768 images for testing with balanced class ratio (Veeling et al., 2018; Bejnordi et al., 2017). For PathCamelyon, we manually construct an imbalanced training dataset with an imratio of 1% and keep the testing set balanced. For Melanoma data, we adopt a EfficientNetV2-S (Tan & Le, 2021) as the network structure, and for CheXpert, DDSM+, PatchCam, we use DenseNet121 (Huang et al., 2017). We tune the number of inner gradient steps for CT in $k \in \{1, 2, 3\}$ and also tune $\alpha$ in $\{0.1, 0.05, 0.01\}$ for the inner steps.

**Results.** The testing AUC results are reported in Table 1. We can see that the proposed composition training method outperforms all baselines on all datasets for maximizing AUC. In addition, we have the following observations: (i) optimizing an AUC loss from scratch does not necessarily yield a better performance than minimizing the standard CE loss; (ii) the CT (AUC) method dramatically improves the performance of optimizing the AUC loss from scratch, with about 2%∼5% improvement on difficult medical classification tasks; (iii) the CT (AUC) method is generally better than the linear combination approach (AUC-CE), especially on the more difficult medical image datasets. It is notable that the reported results on some medical datasets are not comparable with that in (Yuan et al., 2020) because (i) we report the performance on official CheXpert validation data instead of the official testing data; (ii) we use a smaller resolution on Melanoma data; (ii) we do not tune the margin parameter in the AUCM loss. We also plot the learned feature presentations of training data of different datasets visualized by t-SNE in Figure 3. We can see that the proposed CT (AUC) method obtains better feature representations than the baseline approaches of optimizing CE or AUC alone and the naive linear combination approach.

### 4.1 Ablation Study

We conduct some ablation study including (i) the comparison of convergence curves and the running time analysis of different methods; (ii) the verification of our algorithmic design.

**Convergence Curve.** Below, we compare the convergence speed of our CT (AUC) approach with other end-to-end learning baselines, i.e., CE, AUC, AUC-CE, with results on four benchmark datasets plotted in Figure 4. The results indicate that our algorithm enjoys even faster convergence in terms of number of epochs. The convergence curves of testing AUC are included in Appendix A.3.

**Runtime analysis.** We notice our method has larger running time per-iteration than minimizing the CE and the AUC loss alone because of several backpropagations, but with a reward of faster convergence in epochs and better testing performance. For fair comparison, we have run the baselines with the same amount of time as our method and observed they are still worse than our method (cf Appendix A.4). For example, on CIFAR10 (10%), CT (AUC) can achieve an AUC of 0.944 with a running time of 1000s, in contrast, the baselines CE, AUC, AUC-CE, TS-DRW, and TS-DEC use same running time and achieve AUC scores of 0.925, 0.915, 0.943, 0.917, 0.917, respectively.

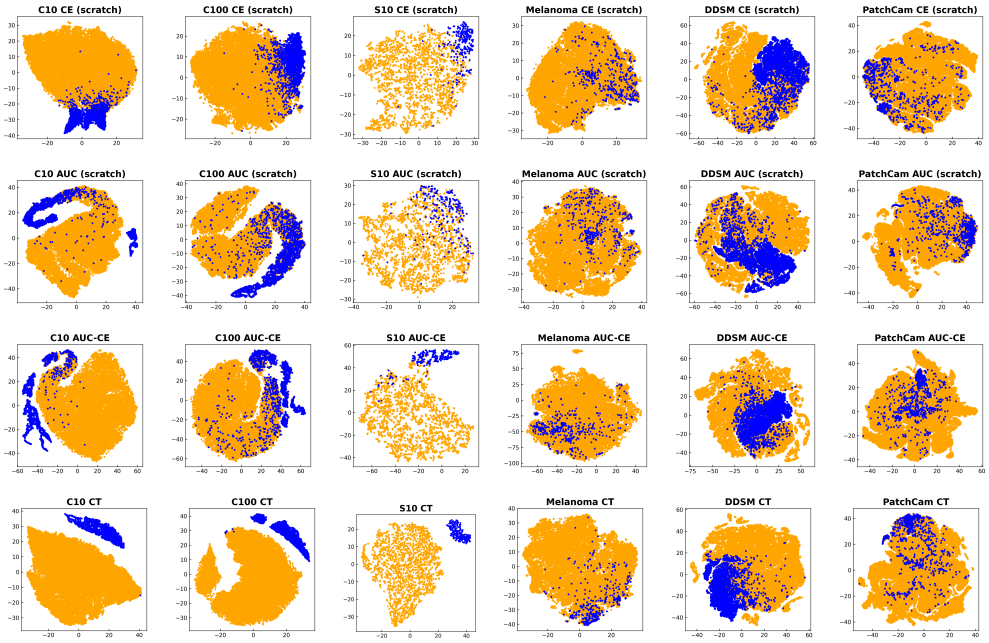


Figure 3: t-SNE visualization of training data (● is positive and ● is negative) by (from top to bottom) optimizing the CE loss, an AUC loss from scratch, a linear combined loss, and our CT method.

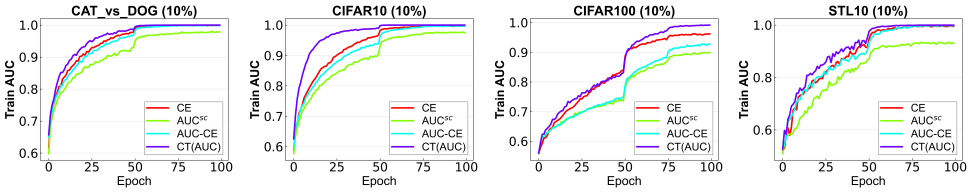


Figure 4: Convergence curves on four benchmark datasets with an imbalance ratio of 10%.

**Verification of Algorithmic Design.** We validate three algorithmic choices. (i) Using two independent mini-batches $\mathcal{S}_1 \neq \mathcal{S}_2$ is generally better than using the same mini-batch. (ii) Using the momentum update for $\mathbf{u}_{t+1}$ (i.e., $\beta_0 < 1$) is better than without using momentum update ($\beta_0 = 1$). (iii) tuning the number of inner gradient steps $k \in \{1, 2, 3\}$ is helpful for improving the performance. The results are demonstrated in Table 2, where all results are averaged over three trials.

Table 2: Left: $\mathcal{S}_1 \neq \mathcal{S}_2$ vs $\mathcal{S}_1 = \mathcal{S}_2$, right: $\beta_0 = 1$ vs $\beta_0 < 1$ in Algorithm 1. Note that we tune $k \in \{1, 2, 3\}$ for for the left table and fix $k = 1$ for the right table. left ($\mathcal{S}_1 \neq \mathcal{S}_2$) vs right ($\beta_0 \leq 1$) verifies that tuning $k$ is helpful.

| Dataset | Method | Imbalance Ratio 1% | Imbalance Ratio 10% | Imbalance Ratio 30% | Method | Imbalance Ratio 1% | Imbalance Ratio 10% | Imbalance Ratio 30% |
|---|---|---|---|---|---|---|---|---|
| CATvsDOG | $\mathcal{S}_1 = \mathcal{S}_2$ | 0.784±0.007 | 0.941±0.002 | 0.975±0.001 | $\beta_0 = 1$ | 0.765±0.005 | 0.937±0.004 | 0.971±0.002 |
| | $\mathcal{S}_1 \neq \mathcal{S}_2$ | **0.789±0.008** | **0.946±0.002** | **0.977±0.001** | $\beta_0 < 1$ | **0.769±0.007** | **0.939±0.004** | **0.975±0.006** |
| CIFAR10 | $\mathcal{S}_1 = \mathcal{S}_2$ | 0.738±0.005 | 0.931±0.002 | 0.959±0.000 | $\beta_0 = 1$ | 0.724±0.006 | 0.928±0.004 | 0.957±0.001 |
| | $\mathcal{S}_1 \neq \mathcal{S}_2$ | **0.740±0.004** | **0.935±0.001** | **0.964±0.001** | $\beta_0 < 1$ | **0.725±0.011** | **0.929±0.002** | **0.960±0.001** |
| STL10 | $\mathcal{S}_1 = \mathcal{S}_2$ | 0.671±0.007 | 0.820±0.025 | 0.902±0.003 | $\beta_0 = 1$ | 0.663±0.012 | 0.819±0.018 | 0.895±0.004 |
| | $\mathcal{S}_1 \neq \mathcal{S}_2$ | **0.681±0.005** | **0.839±0.004** | **0.907±0.001** | $\beta_0 < 1$ | **0.666±0.006** | **0.832±0.008** | **0.900±0.002** |
| CIFAR100 | $\mathcal{S}_1 = \mathcal{S}_2$ | 0.608±0.004 | 0.709±0.002 | 0.788±0.005 | $\beta_0 = 1$ | 0.590±0.014 | 0.714±0.004 | 0.791±0.004 |
| | $\mathcal{S}_1 \neq \mathcal{S}_2$ | **0.609±0.002** | **0.725±0.000** | **0.809±0.002** | $\beta_0 < 1$ | **0.598±0.002** | **0.714±0.003** | **0.801±0.001** |

## 5 Conclusions

In this paper, we have proposed a novel end-to-end compositional training framework for deep AUC maximization by optimizing a compositional objective. We also proposed an efficient stochastic optimization method for compositional training of deep AUC maximization. We demonstrated the effectiveness of compositional training on multiple benchmark datasets and medical datasets for maximizing AUC. In future work, we will investigate compositional training for other imbalanced loss functions more extensively.

## REFERENCES

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.

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, pp. 27, 1996.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pp. 1567–1578, 2019.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Damien Dablain, Bartosz Krawczyk, and Nitesh V. Chawla. Deepsmote: Fusing deep learning and SMOTE for imbalanced data. *CoRR*, abs/2105.02340, 2021. URL https://arxiv.org/abs/2105.02340.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1082–1092. PMLR, 2020.

C Ferri, PA Flach, and J Hernández-Orallo. Learning decision trees using the area under the roc curve. In Claude Sammut and Achim Hoffmann (eds.), *Proceedings of the 19th International Conference on Machine Learning*, pp. 139 – 146. Morgan Kaufmann, 2002. ISBN 1558608737.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017. URL http://proceedings.mlr.press/v70/finn17a.html.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4(null):933–969, December 2003. ISSN 1532-4435.

Wei Gao and Zhi-Hua Zhou. On the consistency of AUC pairwise optimization. In Qiang Yang and Michael J. Wooldridge (eds.), *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 939–945. AAAI Press, 2015. URL http://ijcai.org/Abstract/15/137.

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

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

Saeed Ghadimi, Andrzej Ruszczynski, and Mengdi Wang. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM J. Optim.*, 30(1):960–979, 2020. doi: 10.1137/18M1230542. URL https://doi.org/10.1137/18M1230542.

Zhishuai Guo, Mingrui Liu, Zhuoning Yuan, Li Shen, Wei Liu, and Tianbao Yang. Communication-efficient distributed stochastic AUC maximization with deep neural networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3864–3874. PMLR, 2020a. URL http://proceedings.mlr.press/v119/guo20f.html.

Zhishuai Guo, Zhuoning Yuan, Yan Yan, and Tianbao Yang. Fast objective and duality gap convergence for non-convex strongly-concave min-max problems. *CoRR*, abs/2006.06889, 2020b. URL https://arxiv.org/abs/2006.06889.

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. On stochastic moving-average estimators for non-convex optimization. *CoRR*, abs/2104.14840, 2021. URL https://arxiv.org/abs/2104.14840.

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

Ralf Herbrich, Thore Graepel, and Klause Obermayer. Large Margin Rank Boundaries for Ordinal Regression. In *Advances in Large Margin Classifiers*, chapter 7, pp. 115–132. The MIT Press, 1999. URL http://www.herbrich.me/papers/nips98_ordinal.pdf.

Alan Herschtal and Bhavani Raskutti. Optimising area under the ROC curve using gradient descent. In Carla E. Brodley (ed.), *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004. doi: 10.1145/1015330.1015366. URL https://doi.org/10.1145/1015330.1015366.

Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order momentum methods from mini to minimax optimization. *arXiv preprint arXiv:2008.08170*, 2020.

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*, pp. 4700–4708, 2017.

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, pp. 590–597, 2019.

Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7610–7619, 2020.

Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pp. 377–384, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102399. URL https://doi.org/10.1145/1102351.1102399.

Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Purushottam Kar, Bharath Sriperumbudur, Prateek Jain, and Harish Karnick. On the generalization ability of online learning algorithms for pairwise loss functions. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 441–449, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/kar13.html.

Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pp. 694–702, Cambridge, MA, USA, 2014. MIT Press.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, pp. 3713–3717. IEEE, 2016.

Tianyi Lin, Chi Jin, and Michael I Jordan. On gradient descent ascent for nonconvex-concave minimax problems. *arXiv preprint arXiv:1906.00331*, 2019.

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*, pp. 2980–2988, 2017.

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*, pp. 3195–3203, 2018.

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

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 2537–2546. Computer Vision Foundation / IEEE, 2019b. doi: 10.1109/CVPR.2019.00264. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Liu_Large-Scale_Long-Tailed_Recognition_in_an_Open_World_CVPR_2019_paper.html.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *7th International Conference on Learning Representations (ICLR)*, 2019.

David Masko and Paulina Hensman. The impact of imbalanced training data for convolutional neural networks, 2015.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=37nvvqkCo5.

Michael Natole, Yiming Ying, and Siwei Lyu. Stochastic proximal algorithms for auc maximization. In *International Conference on Machine Learning*, pp. 3707–3716, 2018.

Michael Natole, Yiming Ying, and Siwei Lyu. Stochastic auc optimization algorithms with linear convergence. *Frontiers in Applied Mathematics and Statistics*, 5, 2019.

Qi Qi, Yi Xu, Rong Jin, Wotao Yin, and Tianbao Yang. Attentional biased stochastic gradient for imbalanced classification. *CoRR*, abs/2012.06951, 2020. URL https://arxiv.org/abs/2012.06951.

Alain Rakotomamonjy. Support vector machines and area under roc curves. Technical report, 2004.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *6th International Conference on Learning Representations (ICLR)*, 2018.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

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. *Scientific data*, 8(1):1–8, 2021.

Mingxing Tan and Quoc V Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021.

Laurens van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

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*, pp. 210–218. Springer, 2018.

Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2): 419–449, 2017.

Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X. Yu. Long-tailed recognition by routing diverse distribution-aware experts. *CoRR*, abs/2010.01809, 2020. URL https://arxiv.org/abs/2010.01809.

Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=D9I3drBz4UC.

Zhengyang Wang, Meng Liu, Youzhi Luo, Zhao Xu, Yaochen Xie, Limei Wang, Lei Cai, Qi Qi, Zhuoning Yuan, Tianbao Yang, and Shuiwang Ji. Advanced graph and sequence neural networks for molecular property prediction and drug discovery, 2021b.

Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pp. 247–263. Springer, 2020. doi: 10.1007/978-3-030-58558-7_15. URL https://doi.org/10.1007/978-3-030-58558-7_15.

Lian Yan, Robert Dodier, Michael C. Mozer, and Richard Wolniewicz. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 848–855. AAAI Press, 2003. ISBN 1577351894.

Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19290–19301. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e025b6279c1b88d3ec0eca6fcb6e6280-Paper.pdf.

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

Zhuoning Yuan, Yan Yan, Milan Sonka, and Tianbao Yang. Robust deep AUC maximization: A new surrogate loss and empirical studies on medical image classification. *CoRR*, abs/2012.03173, 2020. URL https://arxiv.org/abs/2012.03173.

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 (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12219–12229. PMLR, 2021. URL http://proceedings.mlr.press/v139/yuan21a.html.

Xinhua Zhang, Ankan Saha, and S. V. N. Vishwanathan. Smoothing multivariate performance measures. *J. Mach. Learn. Res.*, 13(1):3623–3680, December 2012. ISSN 1532-4435.

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

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9716–9725. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00974. URL https://doi.org/10.1109/CVPR42600.2020.00974.

Linchao Zhu and Yi Yang. Inflated episodic memory with region self-attention for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 4343–4352. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00440. URL https://doi.org/10.1109/CVPR42600.2020.00440.