Towards Nonlinear Sparse AUC Maximization via Compositional Stochastic Hard Thresholding

Wenkang Wang[1], **Dongxu Liu**[1], **Bin Gu**[1*]

[1]School of Artificial Intelligence, Jilin University, Changchun, Jilin, China
{wangwk25, liudx9924}@mails.jlu.edu.cn, jsgubin@gmail.com

Abstract

The Area Under the ROC Curve (AUC) is an important evaluation metric for both linear and, in particular, nonlinear classification models, owing to its robustness against class imbalance. Sparse learning with an ℓ_0 constraint can enhance model interpretability and generalization. Prior work has shown that, in the linear setting, the pairwise formulation of AUC maximization can be reformulated as a standard pointwise empirical risk minimization problem, which enables efficient optimization using hard-thresholding gradient descent for ℓ_0-constrained AUC maximization. Extending this approach to the nonlinear setting remains largely unexplored, even though we establish that pairwise AUC maximization in this setting is equivalent to a pointwise compositional optimization problem; however, designing a compositional optimization algorithm compatible with hard-thresholding operators remains an open challenge. To address this challenge, in this paper, we propose a novel algorithm—Compositional Stochastic Hard Thresholding (CSHT)—for nonlinear sparse AUC maximization. Specifically, CSHT integrates stochastic variance-reduced gradient techniques with hard-thresholding projections to effectively reduce gradient estimation variance while enforcing sparsity. Notably, we provide a rigorous convergence analysis and prove that CSHT achieves linear convergence up to a tolerance bound. To the best of our knowledge, this is the first stochastic hard-thresholding algorithm tailored for nonlinear sparse AUC maximization. Extensive experiments on (a) nonlinear sparse AUC maximization using Random Fourier Feature-based kernel approximation and (b) universal adversarial attack scenarios demonstrate the superior performance of CSHT over existing methods, attributed to its unified treatment of nonlinearity and sparsity.

1 Introduction

In imbalanced classification tasks, traditional accuracy metrics often suffer from poor generalization due to their inherent bias towards the majority class (Gultekin et al. 2020). For instance, in medical diagnosis of a rare disease where less than 1% of patients are infected, a trivial classifier that always predicts "healthy" would achieve over 99% accuracy, yet would fail to identify any true positive cases—rendering it clinically useless. In contrast, the Area Under the ROC Curve (AUC) measures the probability that a randomly selected positive instance receives a higher decision score than a randomly selected negative instance, thereby exhibiting robustness to class distribution imbalances (Hanley and McNeil 1982). As a result, AUC has become a widely adopted evaluation metric for imbalanced learning problems in critical domains such as medical image diagnosis (Yuan et al. 2021), molecular property prediction (Wu et al. 2018), and financial risk assessment (Zhou, Lai, and Yen 2009). However, in high-dimensional settings with thousands of potential features, models risk overfitting and obscured decision-making. Therefore, akin to an experienced doctor focusing on critical symptoms, enforcing sparsity to identify the most predictive features becomes essential.

Model sparsity is a fundamental objective in machine learning and is typically achieved by imposing an ℓ_0 constraint. Beyond improving the memory, computational, and environmental footprint of models, sparse constraints also help mitigate overfitting and obtain consistent statistical estimation (Yuan and Li 2021; Negahban et al. 2012). Hard-thresholding gradient algorithm (Nguyen, Needell, and Woolf 2017; Yuan, Li, and Zhang 2018) is a key technique for efficiently solving ℓ_0-constrained optimization problems. Its core mechanism is an alternation between gradient update steps and hard-thresholding projections, ensuring strict adherence to the ℓ_0 constraint during parameter updates. Compared with convex relaxation approaches based on ℓ_1 regularization (Van de Geer 2008), hard-thresholding can often attain similar precision while being more computationally efficient, as it can directly ensure a desired sparsity level instead of tuning an ℓ_1 penalty or constraint.

In the batch learning setting, AUC maximization can be formulated as a convex empirical optimization problem using a surrogate loss. However, the pairwise nature of its objective makes it challenging for large-scale datasets. To overcome this challenge, (Ying, Wen, and Lyu 2016) pioneered a reformulation of AUC maximization as a stochastic min-max saddle point problem, and designed a primal-dual style stochastic gradient algorithm with an $\widetilde{O}(1/\sqrt{t})$ convergence rate. Subsequently, in order to obtain sparse solutions, (Liu et al. 2018; Zhou, Ying, and Skiena 2020) followed this saddle point formulation for AUC maximization with ℓ_1 constraints. However, as many researchers observed (Duchi and Singer 2009; Langford, Li, and Zhang 2009; Xiao 2009), ℓ_1-

*Corresponding author

based stochastic algorithms, although attractive as a convex approach, may struggle to preserve truly sparse solutions.

To achieve exact sparsity, (Yang et al. 2020) reformulated the pairwise formulation of AUC maximization in the linear setting as a standard pointwise empirical risk minimization problem, which enables efficient optimization using hard-thresholding gradient descent for ℓ_0-constrained AUC maximization. Under the restricted strong convexity and restricted strong smoothness (RSC/RSS) assumptions, they proved that the proposed algorithm achieves linear convergence up to a tolerance bound. This paper extends this approach to the nonlinear setting, establishing that pairwise AUC maximization in this setting is equivalent to a pointwise compositional optimization problem. However, designing a compositional optimization algorithm compatible with hard-thresholding operators remains an open challenge.

To bridge this gap, we propose a novel Compositional Stochastic Hard Thresholding (CSHT) algorithm specifically designed for nonlinear sparse AUC maximization. Specifically, we draw on the ideas of (Lian, Wang, and Liu 2017) and adopt the SVRG technique for efficient optimization. Simultaneously, following the approach of (Shen and Li 2018), we apply a hard-thresholding projection operator after each parameter update to satisfy the ℓ_0 constraint.

The main contributions of this paper are summarized as follows:

- We establish that pairwise AUC maximization in the nonlinear setting is equivalent to a pointwise compositional optimization problem and propose a Compositional Stochastic Hard Thresholding (CSHT) algorithm. To the best of our knowledge, this is the first stochastic hard-thresholding algorithm tailored for nonlinear sparse AUC maximization.
- We integrate stochastic variance-reduced gradient techniques with hard-thresholding projections to effectively reduce gradient estimation variance while enforcing sparsity. Furthermore, we rigorously establish the algorithm's linear convergence up to a tolerance bound under standard assumptions.
- We conduct extensive experiments on nonlinear sparse AUC maximization using Random Fourier Feature-based kernel approximation, as well as on universal adversarial attack scenarios. Results demonstrate the superior performance of CSHT over existing methods, attributed to its unified treatment of nonlinearity and sparsity.

2 Related Work

AUC Maximization. Stochastic AUC maximization in the classical online setting is challenging due to its pairwise nature. As aforementioned, to address this issue, (Ying, Wen, and Lyu 2016) adopted the primal-dual stochastic gradient and obtained $\widetilde{O}(1/\sqrt{t})$ convergence. (Natole, Ying, and Lyu 2018) added a strongly convex regularizer, employed a stochastic proximal gradient method, and achieved $\widetilde{O}(1/t)$ convergence rate. (Yang et al. 2020) directly applied hard-thresholding gradient descent to the reformulated pointwise empirical risk minimization problem, achieving ℓ_0-constrained AUC maximization. However, all of them only consider learning a linear model. To improve the model's capacity to capture nonlinearity, (Dang et al. 2020) approximated the kernel with Random Fourier Features and used triply stochastic gradients to update the solution, achieving an $O(1/t)$ convergence rate after t iterations. In addition, (Liu et al. 2019) adopted a min-max saddle point approach for AUC maximization with neural networks, achieving an $O(1/\epsilon)$ convergence rate under the Polyak-Łojasiewicz (PL) condition.

Compositional optimization. Problems formulated as $\min_{\boldsymbol{x}} \mathbb{E}_i F_i(\mathbb{E}_j G_j(\boldsymbol{x}))$, which is referred to as the composition problem, have received lots of attention in machine learning. (Wang, Fang, and Liu 2017) pioneered two solution algorithms, Basic SCGD and accelerated SCGD to solve it, achieving a sublinear convergence rate for convex and strongly convex cases. Later, to deal with the nonsmooth regularization penalty, (Wang, Liu, and Fang 2017) proposed the first proximal gradient method called ASC-PG, which has a sublinear convergence rate. (Lian, Wang, and Liu 2017) first solved the finite sample case of stochastic composition optimization and obtained two linear-convergent algorithms based on the SVRG technique. However, the algorithms do not handle the regularizer either. Subsequently, (Huo et al. 2018) adopted the SVRG variance reduction scheme, while (Zhang and Xiao 2019) used the SAGA scheme (Defazio, Bach, and Lacoste-Julien 2014) to solve the composition problem with nonsmooth regularization penalty, both achieving linear convergence rates under certain assumptions.

Hard Thresholding. As previously noted, the ℓ_0-sparse problem holds significant importance in machine learning. Among the early approaches proposed to address this problem, the Iterative Hard Thresholding (IHT) algorithm (Blumensath and Davies 2008) stands out as an iterative greedy selection approach. It retains the top s largest entries after each gradient step using hard-thresholding, yielding a solution with only s nonzero elements. Subsequently, (Foucart 2011) proposed the Hard Thresholding Pursuit (HTP) algorithm by incorporating orthogonal projection into the IHT framework. (Nguyen, Needell, and Woolf 2017) developed a stochastic gradient descent (SGD) version of hard-thresholding (StoIHT), and further, (Khanna and Kyrillidis 2018) investigated integrating Nesterov's acceleration technique into the hard-thresholding operator, while (Shen and Li 2018; Zhou, Yuan, and Feng 2018) incorporated the SVRG technique, with both achieving linear convergence rates. Moreover, (Shen and Li 2018) improved the previously widely used bound on the deviation induced by hard-thresholding, resulting in a tighter estimate that plays a critical role in the convergence analysis.

3 Method

3.1 Preliminaries and Notations

Given an integer $n \geq 1$, define the index set $[n] = \{1, 2, \ldots, n\}$. For a vector $\boldsymbol{v} = (v_1, \ldots, v_d)^\top \in \mathbb{R}^d$, its cardinality (i.e., the number of non-zero elements) is denoted by $\|\boldsymbol{v}\|_0$. For any positive integer d, suppose that Ω is a subset of $[d]$. Then for any vector $\boldsymbol{v} \in \mathbb{R}^d$, we define $\mathcal{P}_\Omega(\cdot)$ as

the orthogonal projection operator onto the support set Ω, i.e., $(\mathcal{P}_\Omega(\boldsymbol{v}))_i = v_i$ if $i \in \Omega$, and 0 otherwise. In particular, let Γ be the set indexing the k largest absolute components of vector $\boldsymbol{v}$. In this way, the hard-thresholding operator is defined as:

$$\mathcal{H}_k(\boldsymbol{v}) = \mathcal{P}_\Gamma(\boldsymbol{v}),$$

The orthogonal projection of a vector $\boldsymbol{v}$ onto the ℓ_2-ball of radius ω is defined as:

$$\Pi_\omega(\boldsymbol{v}) = \frac{\boldsymbol{v}}{\max\{1, \|\boldsymbol{v}\|_2/\omega\}},$$

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the input space and $\mathcal{Y} = \{\pm 1\}$ be the label set. Assume that the training dataset $\mathcal{S} = \{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i \in [n]\}$ is independently and identically sampled from an unknown distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. For each $1 \le i \le n$, if $y_i = 1$, then $\boldsymbol{z}_i$ is referred to as a positive instance; otherwise, it is a negative instance. Let n_+ denote the number of positive instances, n_- the number of negative instances, and define $r = n_+/n$ as the imbalance ratio.

The AUC at the population level for a scoring function $h : \mathcal{X} \to \mathbb{R}$ is defined as:

$$\mathrm{AUC}(h) = \Pr(h(\boldsymbol{x}) \ge h(\boldsymbol{x}')|y = 1, y' = -1),$$

where $\boldsymbol{z} = (\boldsymbol{x}, y)$ and $\boldsymbol{z}' = (\boldsymbol{x}', y')$ are independent instance pairs randomly sampled. Following prior work (Ying, Wen, and Lyu 2016; Liu et al. 2019), we adopt the squared loss as a surrogate for the indicator function and consider learning a generic nonlinear model $h(\boldsymbol{w}; \boldsymbol{x})$ parameterized by $\boldsymbol{w}$. The nonlinear sparse AUC maximization problem on the dataset $\mathcal{S}$ is formulated as:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & f(\boldsymbol{w}) = \frac{1}{n_+ n_-} \sum_{i=1}^{n} \sum_{j=1}^{n} (1 - h(\boldsymbol{w}; \boldsymbol{x}_i) \\ & \qquad + h(\boldsymbol{w}; \boldsymbol{x}_j))^2 \mathbb{I}_{[y_i=1]} \mathbb{I}_{[y_j=-1]}. \\ \text{s.t.} \quad & \|\boldsymbol{w}\|_0 \le k \end{aligned} \tag{1}$$

The objective function $f(\boldsymbol{w})$ is the average of pairwise losses and takes the form of a U-statistic (Clémençon, Lugosi, and Vayatis 2008).

3.2 Equivalent Reformulation

Since the objective function $f(\boldsymbol{w})$ involves pairs of positive and negative instances simultaneously, directly applying stochastic hard-thresholding algorithms becomes challenging in large-scale, high-dimensional scenarios (Yang et al. 2020). To address this challenge, inspired by (Yang et al. 2020; Lei and Ying 2021), we reformulate the U-statistic objective defined in Eq. (1) as a compositional optimization problem. To this end, we define the means of positive and negative instances as:

$$\begin{aligned} \mu_+(\boldsymbol{w}) &= \frac{1}{n_+} \sum_{i=1}^{n} h(\boldsymbol{w}; \boldsymbol{x}_i) \mathbb{I}_{[y_i=1]}, \\ \mu_-(\boldsymbol{w}) &= \frac{1}{n_-} \sum_{i=1}^{n} h(\boldsymbol{w}; \boldsymbol{x}_i) \mathbb{I}_{[y_i=-1]}, \end{aligned} \tag{2}$$

Next, define the vector-valued function $G_j(\boldsymbol{w})$:

$$G_j(\boldsymbol{w}) = \begin{pmatrix} \boldsymbol{w} \\ \frac{1}{r} h(\boldsymbol{w}; \boldsymbol{x}_j) \mathbb{I}_{[y_j=1]} \\ \frac{1}{1-r} h(\boldsymbol{w}; \boldsymbol{x}_j) \mathbb{I}_{[y_j=-1]} \end{pmatrix}, \quad j = 1, \ldots, n$$

and its mean function $G(\boldsymbol{w})$:

$$G(\boldsymbol{w}) = \frac{1}{n} \sum_{j=1}^{n} G_j(\boldsymbol{w}) = \begin{pmatrix} \boldsymbol{w} \\ \mu_+(\boldsymbol{w}) \\ \mu_-(\boldsymbol{w}) \end{pmatrix}.$$

Then, we have the following proposition.

Proposition 1. *(Proof in Appendix A) The AUC maximization objective function $f(\boldsymbol{w})$, defined in Eq.* (1)*, can be equivalently reformulated as:*

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & f(\boldsymbol{w}) = F(G(\boldsymbol{w})) = \frac{1}{n} \sum_{i=1}^{n} F_i \left(\frac{1}{n} \sum_{j=1}^{n} G_j(\boldsymbol{w}) \right), \\ \textit{s.t.} \quad & \|\boldsymbol{w}\|_0 \le k \end{aligned} \tag{3}$$

where the outer component function F_i is defined as:

$$\begin{aligned} F_i(G(\boldsymbol{w})) = & (1 + \mu_-(\boldsymbol{w}) - \mu_+(\boldsymbol{w}))^2 \\ & + \frac{1}{r} (h(\boldsymbol{w}; \boldsymbol{x}_i) - \mu_+(\boldsymbol{w}))^2 \mathbb{I}_{y_i=1} \\ & + \frac{1}{1-r} (h(\boldsymbol{w}; \boldsymbol{x}_i) - \mu_-(\boldsymbol{w}))^2 \mathbb{I}_{y_i=-1}. \end{aligned} \tag{4}$$

3.3 Algorithm

This section presents the complete algorithm for solving the finite-sum compositional optimization problem (3), referred to as Compositional Stochastic Hard Thresholding (CSHT). Following the approach of (Lian, Wang, and Liu 2017), given a reference point $\widetilde{\boldsymbol{w}}$, we first compute and store the corresponding full gradient $\tilde{f} = \nabla f(\widetilde{\boldsymbol{w}})$ and inner function value $\widetilde{G} = G(\widetilde{\boldsymbol{w}})$. In the inner loop of the algorithm, to estimate the gradient at the current iterate $\boldsymbol{w}_t$, we first estimate $G(\boldsymbol{w}_t)$ by sampling a mini-batch multiset $\mathcal{A}_t$ of size A:

$$\widehat{G}_t = \widetilde{G} - \frac{1}{A} \sum_{j \in \mathcal{A}_t} \left(G_j(\widetilde{\boldsymbol{w}}) - G_j(\boldsymbol{w}_t) \right), \tag{5}$$

Based on $\widehat{G}_t$, the gradient $\nabla f(\boldsymbol{w}_t)$ is estimated by

$$\hat{f}'_t = (\partial G_{j_t}(\boldsymbol{w}_t))^\top \nabla F_{i_t}(\widehat{G}_t) - (\partial G_{j_t}(\widetilde{\boldsymbol{w}}))^\top \nabla F_{i_t}(\widetilde{G}) + \tilde{f} \tag{6}$$

where i_t and j_t are sampled independently and uniformly from $\{1, 2, \ldots, n\}$. Note that unlike SVRG, the estimator $\hat{f}'_t$ is generally biased, i.e., $\mathbb{E}_{i_t, j_t, \mathcal{A}_t}(\hat{f}'_t) \neq \nabla f(\boldsymbol{w}_t)$. This bias arises because replacing the exact inner function value $G(\boldsymbol{w}_t)$ with its stochastic approximation $\widehat{G}_t$ leads to $\nabla F_{i_t}(\widehat{G}_t) \neq \nabla F_{i_t}(G(\boldsymbol{w}_t))$, which is the key challenge addressed in our convergence analysis. Finally, the hard-thresholding operator $\mathcal{H}_k(\cdot)$ and the orthogonal projection $\Pi_\omega(\cdot)$ are applied to the parameter update, effectively enforcing the ℓ_0 constraint while ensuring parameter boundedness. The CSHT algorithm is outlined in Algorithm 1.

Algorithm 1: Compositional Stochastic Hard-Thresholding (CSHT)

Input: Maximum number of stages S, number of inner loop iterations m, initial solution $\widetilde{\boldsymbol{w}}_0$, learning rate η, sparsity level k, size of mini-batch multiset A, projection radius ω
Output: $\widetilde{\boldsymbol{w}}_S$

for $s = 1, 2, \ldots, S$ **do**
  Update the reference point: $\widetilde{\boldsymbol{w}} \leftarrow \widetilde{\boldsymbol{w}}_{s-1}$
  $\widetilde{G} \leftarrow G(\widetilde{\boldsymbol{w}})$
  $\widetilde{f} \leftarrow \nabla f(\widetilde{\boldsymbol{w}})$
  $\boldsymbol{w}_0 \leftarrow \widetilde{\boldsymbol{w}}$
  for $t = 0, 1, \ldots, m-1$ **do**
    Uniformly sample from $\{1, 2, \ldots, n\}$ for A times with replacement to form a mini-batch multiset $\mathcal{A}_t$
    Estimate $G(\boldsymbol{w}_t)$ by $\widehat{G}_t$ using (5)
    Independently and uniformly sample from {1, 2, . . ., n} to obtain i_t and j_t
    Estimate $\nabla f(\boldsymbol{w}_t)$ by $\hat{f}'_t$ using (6)
    Update $\boldsymbol{w}_{t+1}$ by
      $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \hat{f}'_t$
      $\boldsymbol{r}_{t+1} \leftarrow \mathcal{H}_k(\boldsymbol{b}_{t+1})$
      $\boldsymbol{w}_{t+1} \leftarrow \Pi_\omega(\boldsymbol{r}_{t+1})$
  end for
  $\widetilde{\boldsymbol{w}}_s \leftarrow \boldsymbol{w}_{r_s}$ for randomly chosen $r_s \in \{0, ..., m-1\}$
end for

4 Convergence Analysis

This section presents the convergence results for Algorithm 1. Due to space limitations, all proof details are provided in the appendix. Before presenting the main results, we introduce several global assumptions commonly adopted in sparse learning and stochastic compositional optimization, which are standard in the field, facilitating direct comparison with prior work.

First, we introduce two standard assumptions in sparse learning, namely restricted strong convexity (RSC) and restricted strong smoothness (RSS) (Nguyen, Needell, and Woolf 2017; Zhou, Yuan, and Feng 2018; Shen and Li 2018), defined as follows.

Definition 1. ***(Restricted Strong Convexity)*** *A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is said to satisfy the property of restricted strong convexity with parameter $\alpha_r > 0$, if for all vectors $\boldsymbol{w}$, $\boldsymbol{w}' \in \mathbb{R}^d$ with $\|\boldsymbol{w} - \boldsymbol{w}'\|_0 \leq r$, it holds that*

$$f(\boldsymbol{w}') - f(\boldsymbol{w}) - \langle \nabla f(\boldsymbol{w}), \boldsymbol{w}' - \boldsymbol{w} \rangle \geq \frac{\alpha_r}{2} \|\boldsymbol{w}' - \boldsymbol{w}\|_2^2.$$

Definition 2. ***(Restricted Strong Smoothness)*** *A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is said to satisfy the property of restricted strong smoothness with parameter $L_r > 0$, if for all vectors $\boldsymbol{w}$, $\boldsymbol{w}' \in \mathbb{R}^d$ with $\|\boldsymbol{w} - \boldsymbol{w}'\|_0 \leq r$, it holds that*

$$\|\nabla f(\boldsymbol{w}') - \nabla f(\boldsymbol{w})\|_2 \leq L_r \|\boldsymbol{w}' - \boldsymbol{w}\|_2.$$

4.1 Basic Assumptions

Based on the above definitions, we make the following assumptions.

Assumption 1. ***Boundedness***

1. *Assume the Jacobian matrices $\partial G_j(\boldsymbol{w})$ of the inner component functions and their projections have bounded norms. Specifically, for any $\boldsymbol{w}$ and $j \in \{1, \cdots, n\}$:*

$$\|\partial G_j(\boldsymbol{w})\| \leq B_G, \tag{7}$$

$$\left\| \mathcal{P}_\Omega \left[(\partial G_j(\boldsymbol{w}))^\top \right] \right\| \leqslant B_{G,\Omega}, \tag{8}$$

2. *Assume the ℓ_2-norm of the parameter $\boldsymbol{w}$ is bounded (enforced via projection $\Pi_\omega(\boldsymbol{w})$):*

$$\|\boldsymbol{w}\|_2 \leq \omega, \forall \boldsymbol{w}, \tag{9}$$

3. *Assume the ℓ_2-norm of the gradient of the objective function $f(\boldsymbol{w})$ at the optimal solution $\boldsymbol{w}^*$, restricted to a support set of size $(3k + k^*)$, is bounded:*

$$Q := \max_{\Omega:|\Omega| \leq 3k+k^*} \|\mathcal{P}_\Omega(\nabla f(\boldsymbol{w}^*))\|_2 \tag{10}$$

Remark 1. *$\mathcal{P}_\Omega(A)$ denotes projecting matrix A onto the index set Ω, i.e., retaining elements corresponding to the rows in Ω and setting others to zero.*

Remark 2. *If f is differentiable and unconstrained at $\boldsymbol{w}^*$, then $\nabla f(\boldsymbol{w}^*) = \mathbf{0}$, and consequently $Q = 0$. However, under the ℓ_0 constraint, $\boldsymbol{w}^*$ is generally not an unconstrained critical point, hence $Q > 0$ is the common case in practical optimization scenarios.*

Assumption 2. ***Objective Function Properties***

1. *F_i* ***Smoothness.*** *Assume there exists a constant $L_F > 0$ such that for all $\boldsymbol{w}, \boldsymbol{w}'$ and $i \in \{1, \cdots, n\}$:*

$$\|\nabla F_i(\boldsymbol{w}) - \nabla F_i(\boldsymbol{w}')\| \leqslant L_F \|\boldsymbol{w} - \boldsymbol{w}'\|, \tag{11}$$

2. ***Lipschitzian Gradients.*** *Assume there exists a constant $L_s > 0$ ($s = 3k + k^*$) such that for all $\boldsymbol{w}, \boldsymbol{w}'$ satisfying $\|\boldsymbol{w} - \boldsymbol{w}'\|_0 \leq s$ and for all $i, j \in \{1, \cdots, n\}$:*

$$\left\| (\partial G_j(\boldsymbol{w}))^\top \nabla F_i(G(\boldsymbol{w})) - (\partial G_j(\boldsymbol{w}'))^\top \nabla F_i(G(\boldsymbol{w}')) \right\| \leqslant L_s \|\boldsymbol{w} - \boldsymbol{w}'\|, \tag{12}$$

Note that from Eq. (12)*, it directly follows that the objective function $f(\boldsymbol{w})$ satisfies the RSS condition with parameter L_s for all $\boldsymbol{w}, \boldsymbol{w}'$ satisfying $\|\boldsymbol{w} - \boldsymbol{w}'\|_0 \leq s$:*

$$\begin{aligned} &\|\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{w}')\| \\ \leq &\frac{1}{n^2} \sum_{i,j} \left\| \partial G_j(\boldsymbol{w}) \nabla F_i(G(\boldsymbol{w})) - \partial G_j(\boldsymbol{w}')^\top \nabla F_i(G(\boldsymbol{w}')) \right\| \\ \leq &L_s \|\boldsymbol{w} - \boldsymbol{w}'\|. \end{aligned} \tag{13}$$

3. ***Restricted Strong Convexity.*** *Assume the objective function $f(\boldsymbol{w})$ satisfies the RSC condition with parameter α_s ($s = 3k + k^*$).*

For convenience, we define

$$\alpha := \alpha_s, L := L_s, \kappa = L/\alpha.$$

Here, κ is referred to as the condition number of the problem.

4.2 Main Theorem

Next, we present the main convergence theorem for the CSHT algorithm. The following theorem and corollary are based on the above assumptions.

Theorem 1. *(Proof in Appendix B) For Algorithm 1, if Assumptions 1 and 2 hold, then*

$$\mathbb{E}\left[f(\tilde{\boldsymbol{w}}_s) - f(\boldsymbol{w}^*)\right] \leq \frac{\gamma_1}{\gamma_2}\mathbb{E}\left[f(\tilde{\boldsymbol{w}}_{s-1}) - f(\boldsymbol{w}^*)\right] + C, \tag{14}$$

where

$$\begin{aligned}
\gamma_1 &= m\nu\left(3\eta^2L^2 + \beta_1\right) + 1,\\
\gamma_2 &= m\nu\alpha\left(\eta - 4\eta^2L\right) + m\left(1 - \nu\beta_2\right),\\
C &= \frac{1}{\gamma_2}\Bigg(2m\left(\nu\beta_2 - 1\right)Q\omega + 2m\nu\left(3\eta^2L^2 + \beta_1\right)Q\omega\\
&\quad + 2Q\omega + m\nu\alpha\left(8L\eta^2Q\omega + 2\eta^2Q^2\right)\Bigg),\\
\beta_1 &= \frac{1}{A}\left(\frac{72\eta}{\alpha} + 10\eta^2\right)B_{G,\Omega}^2B_G^2L_F^2,\\
\beta_2 &= 1 - \frac{8\alpha\eta}{9} + 3\eta^2L^2 + \beta_1,\\
\nu &= 1 + \frac{\rho + \sqrt{(4+\rho)\rho}}{2},\ \rho = \frac{\min\{k^*, d-k\}}{k - k^* + \min\{k^*, d-k\}}.
\end{aligned}$$

Remark 3. *Theorem 1 demonstrates that Algorithm 1 achieves a linear convergence within a tolerance error C. The error C primarily stems from Q, which denotes an upper bound on the ℓ_2-norm of the gradient of f in sparse directions at the optimal point $\boldsymbol{w}^*$. As previously discussed, under ℓ_0-norm constraint, $\nabla f(\boldsymbol{w}^*)$ is typically non-zero, and thus $Q > 0$ leads to convergence within an error bound. In the special case where $\nabla f(\boldsymbol{w}^*) = \mathbf{0}$ (i.e., $Q = 0$), we have $C = 0$ and the algorithm achieves exact linear convergence.*

To ensure convergence, the parameters η, A, k, and m must be chosen such that the ratio $\gamma_1/\gamma_2 < 1$. The following corollary provides a specification for these parameters.

Corollary 1. *(Proof in Appendix C). Choose parameters in Algorithm 1 as follows:*

$$\begin{aligned}
\eta &= \frac{\alpha}{9L^2},\\
A &\geq \left(\frac{864}{\alpha^2} + \frac{40}{3L^2}\right)B_{G,\Omega}^2B_G^2L_F^2,\\
k &\geq \left(9\kappa^2 + 8\kappa\right)\left(9\kappa^2 + 8\kappa - 1\right)k^*,\\
m &\geq 324\kappa^2.
\end{aligned}$$

we obtain the linear convergence rate with coefficient $\gamma_1/\gamma_2 = 16/17$ for Algorithm 1.

5 Experiments

We evaluate the effectiveness of the proposed algorithm on two tasks: (a) nonlinear sparse AUC maximization using Random Fourier Features-based kernel approximation; and (b) universal adversarial attacks with sparsity constraint.

ID	Datasets	Instances	Features	Imbalance Rate
1	climate	540	18	0.085
2	german	1,000	24	0.302
3	spambase	4,601	57	0.394
4	a9a	32,561	123	0.240
5	ijcnn1	49,990	22	0.097
6	covtype	58,102	54	0.486
7	connect	67,557	126	0.095
8	acoustic	78,823	50	0.231

Table 1: Statistics of Experimental Datasets

5.1 Nonlinear Sparse AUC Maximization

Compared Algorithms Following the approach of (Gu et al. 2018), we construct a nonlinear scoring function by approximating the kernel using Random Fourier Features:

$$h(\boldsymbol{w}; \boldsymbol{x}_i) = \sum_{l \in A} \boldsymbol{w}(l)\phi_l(\boldsymbol{x}_i),$$

where the feature mapping $\phi_l(\boldsymbol{x}_i)$ is defined as:

$$\phi_l(\boldsymbol{x}_i) = \sqrt{\frac{2}{D}}\cos\left(\boldsymbol{\omega}_l^T\boldsymbol{x}_i\right).$$

- $\boldsymbol{\omega}_l$: Sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^{-2}I)$, which determines the nonlinear feature transformation.
- D: Size of the random features set A, which controls the number of features.

To verify the effectiveness of CSHT in optimizing AUC, we will compare it with two batch learning methods and four online AUC optimization methods.

- SOLAM: a stochastic online algorithm for AUC maximization based on a saddle point problem formulation, as proposed in (Ying, Wen, and Lyu 2016).
- SPAM: a stochastic proximal algorithm for AUC maximization proposed in (Natole, Ying, and Lyu 2018). Based on different regularizations, we refer to SPAM using ℓ^1 and ℓ^2 as SPAM-ℓ^1, SPAM-ℓ^2 respectively.
- FTRL-AUC: an online AUC optimization algorithm based on the FTRL framework, as proposed in (Zhou, Ying, and Skiena 2020).
- SHT-AUC: a stochastic hard-thresholding algorithm for ℓ_0-constrained AUC maximization, as proposed in (Yang et al. 2020).
- HT-SVRG: a Hard Thresholded Stochastic Variance Reduced Gradient method that optimizes the logistic loss with ℓ_0 constraint, as proposed in (Shen and Li 2018).

Experimental Setup We select eight benchmark datasets from LIBSVM (Chang and Lin 2011) and the UCI Machine Learning Repository (Asuncion, Newman et al. 2007), with detailed statistics provided in Table 1. Consistent with prior work (Dang et al. 2020; Gao et al. 2013), all features are scaled to the range $[-1, 1]$, and multi-class datasets are converted into imbalanced binary classification tasks. Each dataset is partitioned into training (80%) and testing (20%)

Dataset	SOLAM	SPAM-ℓ^1	SPAM-ℓ^2	FTRL-AUC	SHT-AUC	HT-SVRG	CSHT
climate	0.940±0.019	0.882±0.081	0.908±0.047	0.935±0.024	0.935±0.054	0.948±0.017	**0.969±0.029**
german	0.778±0.031	0.783±0.036	0.792±0.035	0.764±0.052	0.791±0.028	0.780±0.012	**0.803±0.010**
spambase	0.907±0.003	0.897±0.011	0.903±0.011	0.913±0.007	0.928±0.003	0.932±0.002	**0.955±0.009**
a9a	0.893±0.003	0.885±0.004	0.899±0.004	0.886±0.005	0.891±0.004	0.881±0.002	**0.901±0.004**
ijcnn1	0.930±0.003	0.925±0.004	0.914±0.005	0.929±0.001	0.932±0.002	0.713±0.008	**0.978±0.003**
covtype	0.818±0.001	0.768±0.002	0.783±0.002	0.813±0.001	0.823±0.001	0.811±0.003	**0.898±0.002**
connect	0.690±0.009	0.697±0.016	0.695±0.016	0.697±0.007	0.692±0.006	0.745±0.007	**0.791±0.005**
acoustic	0.886±0.001	0.870±0.003	0.871±0.004	0.890±0.001	0.892±0.001	0.913±0.001	**0.922±0.001**

Table 2: Comparison of the testing AUC values (mean±std).

sets. The randomness of both data splitting and parameter updates is controlled by fixing the random seed.

For the number of Random Fourier Features D, we set it to values proportional to the training sample size n and adopt $D = n/4$ to maintain performance while significantly reducing memory footprint. Similarly, we set k to values proportional to D and choose $k = D/4$, which achieves an optimal balance between nonlinear expressiveness and generalization by mitigating redundancy. Detailed experimental information is provided in Appendix D.

Furthermore, we detail the hyperparameter tuning process and provide a systematic analysis of key parameter sensitivity in Appendix E.

Results and Discussion Using 5-fold cross-validation on training sets, we determine optimal parameters for each algorithm. Under these parameters, we run three experiments with different random seeds to report mean AUC and standard deviation. The results are shown in Table 2. The key observations are summarized as follows:

- **Superior Performance.** CSHT achieved the best AUC performance across all datasets, benefiting from its combination of nonlinear modeling to capture complex nonlinear structures and sparsity to mitigate redundancy.
- **Importance of the Optimization Objective.** HT-SVRG, despite employing a nonlinear scoring function, performed poorly on some highly imbalanced datasets such as IJCNN1 with an imbalance ratio of 0.097, achieving an AUC of only 0.713. This is because HT-SVRG optimizes the logistic loss instead of directly optimizing the AUC objective, which demonstrates the necessity of direct AUC maximization for imbalanced classification.
- **Effectiveness of Sparsity Constraint.** SHT-AUC outperformed other linear baselines in terms of overall AUC performance due to the incorporation of sparsity constraint. This validates the effectiveness and necessity of sparsity constraint.

To further evaluate the algorithm's efficiency, we employ ASC-PG (Wang, Liu, and Fang 2017), VRSC-PG (Huo et al. 2018), and C-SAGA (Zhang and Xiao 2019) to optimize the objective function defined in Eq. (3), all of which are designed for the composition problem with nonsmooth regularization penalty. We set the ℓ_1 regularization parameter λ to 0.001, and the results are shown in Figure 1. The y-axis represents the AUC score, while the x-axis denotes the number of oracle calls. Each oracle call corresponds to a single access of an individual component function value or gradient—namely, querying $G_j(x)$, $\partial G_j(x)$, $F_i(y)$, or $\nabla F_i(y)$.

Notably, ASC-PG is the slowest due to its lack of variance reduction schemes. In contrast, VRSC-PG and C-SAGA outperform ASC-PG in terms of convergence speed. Our proposed method, CSHT, also benefits from SVRG techniques and achieves a convergence rate comparable to VRSC-PG and C-SAGA. However, CSHT incorporates an additional ℓ_0 constraint, which slightly delays convergence during the initial training phase. As the number of oracle calls increases, CSHT gradually surpasses all other methods in AUC performance. This trend highlights the advantage of sparsity constraints: by suppressing redundant features, CSHT promotes model generalization and robustness. Detailed experimental information is provided in Appendix E.

5.2 Universal Adversarial Attacks

Universal Adversarial Perturbations (UAP) (Moosavi-Dezfooli et al. 2017) are input-agnostic perturbation patterns whose core property is that a single perturbation vector can induce misclassification on most natural images by a target model. Specifically, we aim to find a vector δ such that

$$\hat{k}(x+\delta) \neq \hat{k}(x) \text{ for most } x \sim \mathcal{X} \text{ and } \|\delta\|_p \leq \xi$$

where $\hat{k}$ is a classification model.

Baselines To evaluate the effectiveness of CSHT for universal adversarial attacks, we compare it with three baseline methods:

- DF-UAP (Zhang et al. 2020): Treats the perturbation as a dominant feature and iteratively optimizes it to maximize the target class logit while suppressing other classes.
- HP-UAP (Zhang et al. 2021): Produces universal perturbations constrained by high-pass filtering to improve stealthiness while preserving attack efficacy.
- GAN (Mopuri et al. 2018): Models the adversarial perturbation distribution space using a Generative Adversarial Network (GAN) framework.

Experimental Setup We treat the perturbation vector δ to be learned as the parameter $\boldsymbol{w}$ in Equation (3), while keeping the target classifier's parameters frozen. The attack goal is to induce misclassification (i.e., label flipping) after adding the perturbation δ, which transforms the optimization objective into AUC minimization. Therefore, the gradient descent

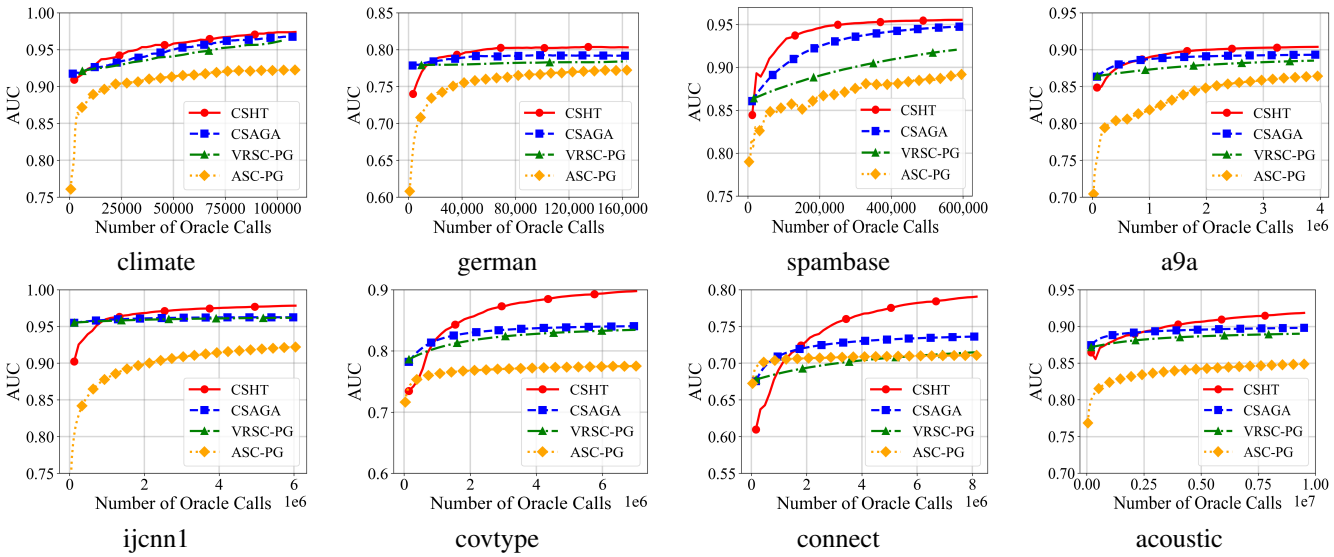

Figure 1: AUC score vs. Number of Oracle Calls.

	AlexNet	ResNet-34	VGG-16	VGG-19	GoogleNet
DF-UAP	40.04±2.33	41.46±1.15	41.76±1.58	41.71±1.59	38.18±0.25
HP-UAP	47.73±0.24	42.08±0.82	43.54±3.39	42.50±0.20	44.99±0.85
GAN	38.28±2.03	**43.81±1.56**	40.09±4.20	41.61±0.62	40.65±1.22
CSHT	**48.17±0.37**	42.45±0.44	**44.65±4.66**	**42.54±0.12**	**46.49±0.47**

Table 3: Mean and Standard Deviation of Fooling Ratio (%) for Different Algorithms Across Target Models

step in the algorithm is replaced with gradient ascent. We use pre-trained weights for the target classifiers provided by torchvision and select the Cat&Dog dataset for evaluation. To adapt to the binary classification task, we modify the final fully connected layer of the classifier by replacing nn.Linear (original_dim, 1000) with nn.Linear (original_dim, 512) $\rightarrow$ ReLU $\rightarrow$ Dropout $\rightarrow$ nn.Linear (512, 2). The model parameters are fine-tuned using cross-entropy loss and the Adam optimizer. Following the approach of (Liu et al. 2019), the output of the positive class after the final Softmax layer is used as the scoring function.

Parameter Settings. For the baseline methods, we follow the original paper settings with $p = \infty$ and $\xi = 10$. For the proposed CSHT method, following (Moosavi-Dezfooli et al. 2017), we set $p = 2$ and $\xi = 2000$. All methods use 1,000 images to compute the perturbation and report the fooling ratio on 5,000 images, that is, the proportion of images whose labels change when perturbed by our universal perturbation. After selecting the optimal parameters, each method is run three times with different random seeds to calculate the mean fooling ratio and standard deviation. The parameter settings are detailed in Appendix E.

Main Results Table 3 reports the mean fooling ratios and standard deviations of different attack methods across target models. The results show that the CSHT algorithm achieves exceptional attack performance. Specifically, it outperforms all baseline methods on AlexNet, VGG-16, VGG-19, and GoogleNet architectures, while achieving performance comparable to the best baseline method on ResNet-34. This fully validates that the sparse perturbation patterns generated by CSHT possess a stronger generalization performance. Visualizations of the perturbations and their attack results are detailed in Appendix F.

6 Conclusion

In this paper, we establish the equivalence between the nonlinear AUC maximization and a pointwise compositional optimization problem. Based on this reformulation, we propose a novel Compositional Stochastic Hard Thresholding (CSHT) algorithm, which integrates stochastic variance-reduced gradient techniques with hard-thresholding projections to enforce sparsity while effectively reducing gradient estimation variance. We rigorously analyze the convergence properties of CSHT and prove that it achieves linear convergence up to a tolerance bound. Furthermore, experiments on nonlinear sparse AUC maximization with kernel approximation and universal adversarial attack tasks demonstrate that CSHT achieves optimal performance or matches the best-performing methods, attributable to its unified handling of nonlinearity and sparsity. For future work, we will explore the design of efficient algorithms for sparse zeroth-order optimization in the black-box setting and establish corresponding theoretical guarantees. Moreover, extending the proposed method to multi-classification tasks is also an interesting direction worth further investigation.

References

Asuncion, A.; Newman, D.; et al. 2007. UCI machine learning repository.

Blumensath, T.; and Davies, M. E. 2008. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5): 629–654.

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3): 1–27.

Clémençon, S.; Lugosi, G.; and Vayatis, N. 2008. Ranking and empirical minimization of U-statistics.

Dang, Z.; Li, X.; Gu, B.; Deng, C.; and Huang, H. 2020. Large-scale nonlinear AUC maximization via triply stochastic gradients. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3): 1385–1398.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27.

Duchi, J.; and Singer, Y. 2009. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research*, 10: 2899–2934.

Foucart, S. 2011. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on numerical analysis*, 49(6): 2543–2563.

Gao, W.; Jin, R.; Zhu, S.; and Zhou, Z.-H. 2013. One-pass AUC optimization. In *International conference on machine learning*, 906–914. PMLR.

Gu, B.; Xin, M.; Huo, Z.; and Huang, H. 2018. Asynchronous doubly stochastic sparse kernel learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Gultekin, S.; Saha, A.; Ratnaparkhi, A.; and Paisley, J. 2020. MBA: mini-batch AUC optimization. *IEEE transactions on neural networks and learning systems*, 31(12): 5561–5574.

Hanley, J. A.; and McNeil, B. J. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1): 29–36.

Huo, Z.; Gu, B.; Liu, J.; and Huang, H. 2018. Accelerated method for stochastic composition optimization with nonsmooth regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Khanna, R.; and Kyrillidis, A. 2018. IHT dies hard: Provable accelerated iterative hard thresholding. In *International Conference on Artificial Intelligence and Statistics*, 188–198. PMLR.

Langford, J.; Li, L.; and Zhang, T. 2009. Sparse Online Learning via Truncated Gradient. *Journal of Machine Learning Research*, 10(3).

Lei, Y.; and Ying, Y. 2021. Stochastic proximal AUC maximization. *Journal of Machine Learning Research*, 22(61): 1–45.

Lian, X.; Wang, M.; and Liu, J. 2017. Finite-sum composition optimization via variance reduced gradient descent. In *Artificial Intelligence and Statistics*, 1159–1167. PMLR.

Liu, M.; Yuan, Z.; Ying, Y.; and Yang, T. 2019. Stochastic auc maximization with deep neural networks. *arXiv preprint arXiv:1908.10831*.

Liu, M.; Zhang, X.; Chen, Z.; Wang, X.; and Yang, T. 2018. Fast stochastic AUC maximization with $O(1/n)$-convergence rate. In *International Conference on Machine Learning*, 3189–3197. PMLR.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1765–1773.

Mopuri, K. R.; Ojha, U.; Garg, U.; and Babu, R. V. 2018. Nag: Network for adversary generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 742–751.

Natole, M.; Ying, Y.; and Lyu, S. 2018. Stochastic proximal algorithms for AUC maximization. In *International Conference on Machine Learning*, 3710–3719. PMLR.

Negahban, S. N.; Ravikumar, P.; Wainwright, M. J.; and Yu, B. 2012. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers.

Nguyen, N.; Needell, D.; and Woolf, T. 2017. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Transactions on Information Theory*, 63(11): 6869–6895.

Shen, J.; and Li, P. 2018. A tight bound of hard thresholding. *Journal of Machine Learning Research*, 18(208): 1–42.

Van de Geer, S. A. 2008. High-dimensional generalized linear models and the lasso.

Wang, M.; Fang, E. X.; and Liu, H. 2017. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161: 419–449.

Wang, M.; Liu, J.; and Fang, E. X. 2017. Accelerating stochastic composition optimization. *Journal of Machine Learning Research*, 18(105): 1–23.

Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; and Pande, V. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2): 513–530.

Xiao, L. 2009. Dual averaging method for regularized stochastic learning and online optimization. *Advances in Neural Information Processing Systems*, 22.

Yang, Z.; Zhou, B.; Lei, Y.; and Ying, Y. 2020. Stochastic hard thresholding algorithms for AUC maximization. In *2020 IEEE International Conference on Data Mining (ICDM)*, 741–750. IEEE.

Ying, Y.; Wen, L.; and Lyu, S. 2016. Stochastic online AUC maximization. *Advances in neural information processing systems*, 29.

Yuan, X.; and Li, P. 2021. Stability and risk bounds of iterative hard thresholding. In *International conference on artificial intelligence and statistics*, 1702–1710. PMLR.

Yuan, X.-T.; Li, P.; and Zhang, T. 2018. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18(166): 1–43.

Yuan, Z.; Yan, Y.; Sonka, M.; and Yang, T. 2021. Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3040–3049.

Zhang, C.; Benz, P.; Imtiaz, T.; and Kweon, I. S. 2020. Understanding adversarial examples from the mutual influence of images and perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14521–14530.

Zhang, C.; Benz, P.; Karjauv, A.; and Kweon, I. S. 2021. Universal adversarial perturbations through the lens of deep steganography: Towards a fourier perspective. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 3296–3304.

Zhang, J.; and Xiao, L. 2019. A composite randomized incremental gradient method. In *International Conference on Machine Learning*, 7454–7462. PMLR.

Zhou, B.; Ying, Y.; and Skiena, S. 2020. Online AUC optimization for sparse high-dimensional datasets. In *2020 IEEE International Conference on Data Mining (ICDM)*, 881–890. IEEE.

Zhou, L.; Lai, K. K.; and Yen, J. 2009. Credit scoring models with AUC maximization based on weighted SVM. *International journal of information technology & decision making*, 8(04): 677–696.

Zhou, P.; Yuan, X.; and Feng, J. 2018. Efficient stochastic gradient hard thresholding. *Advances in Neural Information Processing Systems*, 31.