Abstract

This paper considers a novel application of deep AUC maximization (DAM) for multi-instance learning (MIL), in which a single class label is assigned to a bag of instances (e.g., multiple 2D slices of a CT scan for a patient). We address a neglected yet non-negligible computational challenge of MIL in the context of DAM, i.e., bag size is too large to be loaded into GPU memory for backpropagation, which is required by the standard pooling methods of MIL. To tackle this challenge, we propose variance-reduced stochastic pooling methods in the spirit of stochastic optimization by formulating the loss function over the pooled prediction as a multi-level compositional function. By synthesizing techniques from stochastic compositional optimization and non-convex min-max optimization, we propose a unified and provable multi-instance DAM (MIDAM) algorithm with stochastic smoothed-max pooling or stochastic attention-based pooling, which only samples a few instances for each bag to compute a stochastic gradient estimator and to update the model parameter. We establish a similar convergence rate of the proposed MIDAM algorithm as the state-of-the-art DAM algorithms. Our extensive experiments on conventional MIL datasets and medical datasets demonstrate the superiority of our MIDAM algorithm. The method is opensourced at https://libauc.org/.

1. Introduction

Deep AUC maximization (DAM) has recently achieved great success for many AI applications due to its capability of handling imbalanced data (Yang & Ying, 2022). For example, it earned first place at Stanford CheXpert competition (Irvin et al., 2019), and state-of-the-art performance on other datasets (Yuan et al., 2021; Wang et al., 2021b). However, a novel application of DAM for multi-instance learning (MIL) has not been studied in the literature.

MIL refers to a setting where multiple instances are observed for an object of interest and only one label is given to describe that object. Many real-life applications can be formulated as MIL. For example, the medical imaging data for diagnosing a patient usually consists of a series of 2D high-resolution images (e.g., CT scan), and only a single label (containing a tumor or not) is assigned to the patient (Quellec et al., 2017). MIL has a long history in machine learning and various methods have been proposed for traditional learning with tabular data (Babenko, 2008; Carbonneau et al., 2016) and deep learning (DL) with unstructured data (Oquab et al., 2015; Charles et al., 2017; Ilse et al., 2018). The fundamental theorem of symmetric functions (Zaheer et al., 2017; Qi et al., 2016), inspires a general three-step approach for classifying a bag of instances: (i) a transformation of individual instances, (ii) a pooling of transformed instances using a symmetric (permutation-invariant) function, (iii) a transformation of pooled representation. A key in the implementation of the three steps is the symmetric function that takes the transformations of all instances as input and produces an output, which is also known as the pooling operation. In the literature, various pooling strategies have been explored, e.g., max pooling, average pooling, and smoothed-max (i.e., log-exp-sum) pooling of predictions (Ramon et al., 2000), attention-based pooling of feature representations (Ilse et al., 2018).

However, to the best of our knowledge, none of the existing works have tackled the computational challenge of MIL in the context of DL when a bag is large due to the existence of multiple instances in the bag. The limitation of computing resources (e.g., the memory size of GPU) might prevent loading all instances of a bag at once, creating a severe computational bottleneck for training. For example, an MRI scan of the brain may produce up to hundreds of 2D slices...
We tackle this challenge of multi-instance DAM in a spirit of stochastic optimization by (i) formulating the pooled prediction as a compositional function whose inner functions are expected functions over instances of that bag, and (ii) proposing efficient and provable stochastic algorithms for solving the non-convex min-max optimization with a multi-level compositional objective function. A key feature of the proposed algorithms is replacing the deterministic pooling over all instances of a bag by a variance-reduced stochastic pooling (VRSP), whose computation only requires sampling a few instances from the bag. To ensure the optimization of the original objective, the VRSP is constructed following the principle of stochastic compositional optimization such that the variance of stochastic pooling estimators is reduced in the long term. In particular, the inner functions of the pooled prediction are tracked and estimated by moving average estimators separately for each bag. Based on VRSP, stochastic gradient estimators are computed for updating the model parameter, which can be efficiently implemented by backpropagation.

Our contributions are summarized in the following:

- We propose variance-reduced stochastic pooling estimators for both smoothed-max pooling and attention-based pooling. Building on these stochastic pooling estimators, we develop unified efficient algorithms of multi-instance DAM (MIDAM) based on a min-max objective for the two pooling operations.
- We develop novel convergence analysis of the proposed MIDAM algorithms by (i) proving the averaged error of variance-reduced stochastic pooling estimators over all iterations will converge to zero, and (ii) establishing a convergence rate showing our algorithms can successfully find an $\epsilon$-stationary solution of the min-max objective of DAM.
- We conduct extensive experiments of proposed MIDAM algorithms on conventional MIL benchmark datasets and emerging medical imaging datasets with high-resolution medical images, demonstrating the better performance of our algorithms.

2. Related Works

In this section, we introduce previous works on AUC maximization, multi-instance learning, and medical image classification, and then discuss how they are related to our work.

Deep AUC maximization (DAM). Maximizing the area under the receiver operating characteristic curve (AUC), as an effective method for dealing with imbalanced datasets, has been vigorously studied for the last two decades (Yang & Ying, 2022). Earlier studies focus on learning traditional models, e.g., SVM, decision tree (Cortes & Mohri, 2003; Joachims, 2005b; Ferri et al., 2002). Inspired by the Wilcoxon-Man-Whitney statistic, a variety of pairwise losses and optimization algorithms have been studied for AUC optimization (Gao et al., 2013; Zhao et al., 2011a; Kotlowski et al., 2011; Gao & Zhou, 2015; Calders & Jaroszewicz, 2007; Charoenphakdee et al., 2019). Inspired by the min-max objective corresponding to the pairwise square loss function (Ying et al., 2016), stochastic algorithms have been developed for DAM (Liu et al., 2020; Yuan et al., 2021). In this work, we propose efficient and scalable methods for DAM under the multi-instance learning (MIL) scenario with real big-data applications.

Multi-instance learning. Multi-instance learning (MIL) has been extensively studied and adopted for real applications since decades ago (Ramon & De Raedt, 2000; Andrews et al., 2002; Oquab et al., 2015; Kraus et al., 2016). Usually, a simple MIL pooling strategy, that is, max-pooling over a data bag has been widely utilized. This idea has been incorporated with support vector machine (SVM) and neural networks (Andrews et al., 2002; Oquab et al., 2015; Wang, 2018). Other pooling strategies have also been proposed, e.g., mean, smoothed-max (aka. log-sum-exponential), generalized mean, noisy-or, noisy-and (Wang, 2018; Ramon & De Raedt, 2000; Keeler et al., 1990; Kraus et al., 2016). Recently, attention-based pooling was proposed for deep MIL (Ilie et al., 2018). It is worth noting that almost all the pooling strategies (except max-pooling) require loading all the data from a bag to do the computation, specifically backpropagation. However, there is still no existing method that considers mitigating the computational issue when the data size is too large even for a single data bag.

Medical image classification. In MRI/CT scans, multiple slices of images are acquired at different locations of the patient’s body, which not only improves the diagnostic capabilities but also lowers doses of radiation. Hence, a patient can be represented by a series of 2D slices. A traditional approach is to concatenate these 2D slices into a 3D image and then learn a 3D convolutional neural network (CNN) (Singh et al., 2020). However, this approach suffers from several drawbacks. First, it demands more computational and memory resources as processing high-resolution 3D images is more costly than processing 2D images. As a consequence,
the mini-batch size for back-propagation in training is compromised or the resolution is reduced, which can harm the learning capability. Third, it is more difficult to interpret the prediction of a DL model based on 3D images as radiologists still use 2D slices to make diagnostic decision (Brunyé et al., 2020). To avoid these issues, we will investigate MIL and make it practical for medical image classification.

3. Preliminaries

**Notations.** Let $\mathcal{X}_i = \{x_1^i, \ldots, x_n^i\}$ denote a bag of data instances (e.g., 2D image slices of an MRI/CT scan). Let $\mathcal{D} = \{(\mathcal{X}_i, y_i), i = 1, \ldots, n\}$ denote the set of labeled data, where $y_i \in \{0, 1\}$ denotes the label associated with the bag $i$. Let $\mathcal{D}_+ \subset \mathcal{D}$ only contain $\mathcal{D}_+$ positive bags with $y_i = 1$ and $\mathcal{D}_- \subset \mathcal{D}$ only contain $\mathcal{D}_-$ negative bags with $y_i = 0$. Without loss of generality, let $w \in \mathbb{R}^d$ denote all weights to be learned, which includes the weights of the feature encoder network, the weights of the instance-level classifier, and the parameters in the attention-based pooling. Let $e(w; x) \in \mathbb{R}^d$ be the instance-level representation encoded by a neural network $w$, $\phi(w; x) \in [0, 1]$ be the instance-level prediction score (after some activation function), and $h(w; \mathcal{X}_i) \in [0, 1]$ be the pooled prediction score of the bag $i$ over all its instances. Besides, $\sigma(\cdot)$ denotes the sigmoid activation.

**Multi-instance Learning (MIL).** We work under the standard MIL assumption that (i) an instance can be associated with a label and (ii) a bag is labeled positive if at least one of its instances has a positive label, and negative if all of its instances have negative labels (Dietterich et al., 1997b). The assumption implies that a MIL model must be permutation-invariant for the prediction function $h(\mathcal{X})$. To achieve permutation invariant property, fundamental theorems of symmetric functions have been developed (Zaheer et al., 2017; Qi et al., 2016). In particular, Zaheer et al. (2017) show that a scoring function for a set of instances $\mathcal{X}$, $h(\mathcal{X}) \in \mathbb{R}$, is a symmetric function if and only if it can be decomposed as $h(\mathcal{X}) = g(\sum_{x \in \mathcal{X}} \psi(x))$, where $g$ and $\psi$ are suitable transformations. Qi et al. (2016) prove that for any $c > 0$, a Hausdorff continuous symmetric function $h(\mathcal{X}) \in \mathbb{R}$ can be arbitrarily approximated by a function in the form $g(\max_{x \in \mathcal{X}} \psi(x))$, where max is the element-wise vector maximum operator and $\psi$ and $g$ are continuous functions. These theories provide support for several widely used pooling operators used for MIL.

**Max and smoothed-max pooling of predictions.** The simplest approach is to take the maximum of predictions of all instances in the bag, i.e., $h(w; \mathcal{X}) = \max_{x \in \mathcal{X}} \phi(w; x)$. However, the max operation is non-smooth, which usually causes difficulty in optimization. In practice, a smoothed-max (aka. log-sum-exp) pooling operator is used instead:

$$h(w; \mathcal{X}) = \tau \log \left( \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \exp(\phi(w; x)/\tau) \right),$$

where $\tau > 0$ is a hyperparameter and $\phi(w; x)$ is the prediction score for instance $x$.

**Mean pooling of predictions.** The mean pooling operator just takes the average of predictions of individual instances, i.e., $h(w; \mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \phi(w; x)$. Indeed, smoothed-max pooling interpolates between the max pooling (with $\tau = 0$) and the mean pooling (with $\tau = \infty$).

**Attention-based Pooling.** Attention-based pooling was recently introduced for deep MIL (Ilse et al., 2018), which aggregates the feature representations using attention, i.e.,

$$E(w; \mathcal{X}) = \sum_{x \in \mathcal{X}} \frac{\exp(g(w; x))}{\sum_{x' \in \mathcal{X}} \exp(g(w; x'))} e(w_c; x)$$

where $g(w; x)$ is a parametric function, e.g., $g(w; x) = w_c^\top \tanh(V e(w_c; x))$, where $V \in \mathbb{R}^{m \times d_0}$ and $w_c \in \mathbb{R}^m$. Based on the aggregated feature representation, the bag level prediction can be computed by

$$h(w; \mathcal{X}) = \sigma(\sum_{x \in \mathcal{X}} \frac{\exp(g(w; x))}{\sum_{x' \in \mathcal{X}} \exp(g(w; x'))} \delta(w; x))$$

where $\delta(w; x) = w_c^\top e(w_c; x)$. In this paper, we will focus on smoothed-max pooling and attention-based pooling due to their generality and the challenge of handling them.

**Deep AUC Maximization (DAM).** AUC score can be interpreted as the probability of a positive sample ranking higher than a negative sample (Hanley & McNeil, 1982), i.e., $AUC(h) = \mathbb{E}_{\mathcal{X}, \mathcal{X}'}[\mathbb{I}(h(w; \mathcal{X}) - h(w; \mathcal{X}') \geq 0) | y = 1, y' = 0]$. In practice, one often replaces the indicator function in the above definition of AUC by a convex surrogate loss $\ell : \mathbb{R} \to \mathbb{R}^+$ which satisfies $\ell(h(w; \mathcal{X}) - h(w; \mathcal{X}')) \leq \ell(h(w; \mathcal{X}) - h(w; \mathcal{X}))$ (Joachims, 2005a; Herschtal & Raskutti, 2004; Zhang et al., 2012; Kar et al., 2013; Wang et al., 2012; Zhao et al., 2011b; Ying et al., 2016; Liu et al., 2018; Natoel et al., 2018). Hence, empirical AUC maximization can be formulated as $\min_{w \in \mathbb{R}^d} \tilde{E}[\ell(h(w; \mathcal{X}) - h(w; \mathcal{X}')) | y = 1, y' = 0]$, where $\tilde{E}$ is the empirical average over data in the training set $\mathcal{D}$.

Since optimizing the pairwise formulation is not suitable in some learning scenarios (e.g., online learning, federated learning) (Ying et al., 2016; Guo et al., 2020), recent works of DAM have followed the line of min-max optimization (Yuan et al., 2021; Liu et al., 2020). Denote $c$ as a margin parameter and $\tilde{E}_{i \in \mathcal{D}}$ as the empirical average over $i \in \mathcal{D}$. The objective is:
\[
\min_{w \in \mathbb{R}^d, (a,b) \in \mathbb{R}^2} \max_{\alpha \in \Omega} F(w, a, b, \alpha) := \\
\frac{\mathbb{E}_{i \in \mathcal{D}_+} \left[ (h(w; x_i) - a)^2 \right] + \mathbb{E}_{i \in \mathcal{D}_-} \left[ (h(w; x_i') - b)^2 \right]}{F_1(w, a)} + \frac{\alpha (c + \mathbb{E}_{i \in \mathcal{D}_+} h(w; x_i') - \mathbb{E}_{i \in \mathcal{D}_-} h(w; x_i')) - \alpha^2}{2 F_2(w, b)}
\]

where the first term is the variance of prediction scores of positive data, the second term is the variance of prediction scores of negative data. The maximization over \(\alpha \in \Omega\) yields a term that aims to push the mean score of positive data to be far away from the mean score of negative data. When \(\Omega = \mathbb{R}\), the above min-max objective was shown to be equivalent to the pairwise loss formulation (Ying et al., 2016), and when \(\Omega = \mathbb{R}^+\), the above objective is the min-max margin objective proposed in (Yuan et al., 2021). It is notable that we use conditional expectation given positive or negative labels instead of joint expectation over \((x_i, y_i)\) as in (Ying et al., 2016; Yuan et al., 2021; Liu et al., 2020). The reason is that we consider the batch learning setting and it was found in (Zhu et al., 2022) sampling positive and negative data separately at each iteration is helpful for improving the performance.

4. Multi-instance DAM

Although efficient stochastic algorithms have been developed for DAM, a unique challenge exists in multi-instance DAM due to the computing of the pooled prediction \(h(w; x')\). For example, in smoothed-max pooling computing \(h(w; x') = \tau \log \left( \frac{1}{|x'|} \sum_{x \in x'} \exp(\phi(w; x)/\tau) \right)\) requires processing all instances in the bag \(x'\) to calculate their prediction scores \(\phi(w; x), \forall x \in x'\). Hence, one may need to load all instances of a bag into the GPU memory for forward propagation and backpropagation. This is prohibited if the size of each bag (i.e., the total sizes of all instances in each bag) is large.

A naive approach to address this challenge is to replace the pooling over all instances with mini-batch pooling over randomly sampled instances of a bag. The mini-batch smoothed-max pooling can be computed as \(h(w; B_i) = \tau \log \left( \frac{1}{|B_i|} \sum_{x \in B_i} \exp(\phi(w; x)/\tau) \right)\), where \(B_i \subset x'\) only contains a few sampled instances from the bag of all instances. However, this approach does not work since \(h(w; B_i)\) is not an unbiased estimator, i.e., \(\mathbb{E}_{B_i} h(w; B_i) \neq h(w; x')\). As a result, the mini-batch pooled prediction would incur a large estimation error that depends on the number of sampled instances, i.e., \(\mathbb{E}_{B_i} [(h(w; B_i) - h(w; x'))^2] \leq O\left(\frac{1}{|B_i|}\right)\), which would lead to non-negligible optimization error (Hu et al., 2020).

We propose a solid approach to deal with this challenge. Below, we first describe the high-level idea. Then, we present more details of variance-reduced stochastic pooling estimators and the corresponding stochastic gradient estimators of the min-max objective. Finally, we present a unified algorithm for using both stochastic pooling methods.

We regard the pooled prediction as two-level compositional functions \(h(w; x_i') = f_2(f_1(w; x_i'))\), where \(f_2\) is a simple function that will be exhibited shortly for the two pooling operations, and \(f_1(w; x_i') = \mathbb{E}_{x \sim x_i'} [f_1(w; x)]\) involves averaging over the set of instances \(x \in x_i'\). As a result, we cast the terms of objective into three-level compositional functions \(f(f_2(f_1(w; x_i'))), a\), where \(f(\cdot, a) = (\cdot - a)^2\). The second term can be cast as \(\frac{1}{|D^-|} \sum_{i \in D^-} f_3(f_2(f_1(w; x_i')); b)\). As a result, the three terms of the objective can be written as

\[
F_1(w, a) = \frac{1}{|D^+|} \sum_{y_i = 1} f_2(f_1(w; x_i'), a) \\
F_2(w, b) = \frac{1}{|D^-|} \sum_{y_i = 0} f_2(f_1(w; x_i'), b) \\
F_3(w, \alpha) = \alpha \left( c + \frac{1}{|D^+|} \sum_{y_i = 1} f_2(f_1(w; x_i')) \\
- \frac{1}{|D^-|} \sum_{y_i = 0} f_2(f_1(w; x_i')) \right) - \frac{\alpha^2}{2}.
\]

To optimize the above objective, we need to compute a stochastic gradient estimator. Let us consider the gradient of the first term in terms of \(w\), i.e.,

\[
\nabla_w F_1(w, a) = \frac{1}{|D^+|} \sum_{y_i = 1} \nabla f_1(w; x_i') \nabla f_2(f_1(w; x_i')); \nabla f_2(f_1(w; x_i'), a),
\]

where \(\nabla\) denotes the partial gradient in terms of the first argument. The key challenge lies in computing the innermost function \(f_1(w; x_i')\) and its gradient \(\nabla f_1(w; x_i')\). Due to that the functional value \(f_1(w; x_i')\) is inside non-linear functions \(f_2, f\), one needs to compute an estimator of \(f_1(w; x_i')\) to ensure the convergence for solving the min-max problem. To this end, we will follow stochastic compositional optimization techniques to track and estimate \(f_1(w; x_i')\) for each bag \(x_i'\) separately such that their variance is reduced in the long term (Wang & Yang, 2022).


We write the smoothed-max pooling in (1) as \(h(w; x_i') = f_2(f_1(w; x_i'))\), where \(f_1, f_2\) are defined as:
We express the attention-based pooling in (3) as

$$f_1(w; X_i) = \frac{1}{|X_i|} \sum_{x_i' \in X_i} \exp(\phi(w; x_i')/\tau),$$

$$f_2(s_i) = \tau \log(s_i).$$

We express the attention-based pooling in (3) as $h(w; X_i) = f_2(f_1(w; X_i))$, where $f_1$, $f_2$ are defined as:

$$f_1(w; X_i) = \left[ \frac{1}{|X_i|} \sum_{x_i' \in X_i} \exp(g(w; x_i')) w_e(e(w; x_i')) \right],$$

$$f_2(s_i) = \sigma \left( \frac{s_i}{s_{\tau_i}} \right).$$

One difference between the two pooling operators is that the inner function $f_1$ for attention-based pooling is a vector-valued function with two components. For both pooling operators, the costs lie at the calculation of $f_1(w; X_i)$. To estimate $f_1(w; X_i)$, we maintain a dynamic estimator denoted by $s_i$. At the $t$-th iteration, we sample some positive bags $S^{+}_i \subset D^+$ and some negative bags $S^{-}_i \subset D^-$. For those sampled bags $i \in S^{+}_i \cup S^{-}_i$, we update $s_i$ by:

$$s_i = (1 - \gamma_0)s_i - \gamma_0 f_1(w; B_i), i \in S^{+}_i \cup S^{-}_i, (6)$$

where $B_i \subset X_i$ refers to a mini-batch of instances sampled from $X_i$ and $\gamma_0 \in [0, 1]$ is a hyperparameter. For smoothed-max pooling, $s_i$ is computed by

$$s_i = (1 - \gamma_0)s_i - \gamma_0 \sum_{x_{i'} \in B_i} \exp(\phi(w; x_{i'}))/\tau, (7)$$

and for attention-based pooling, $s_i$ is computed by

$$s_i = (1 - \gamma_0)s_i - \gamma_0 \sum_{x_{i'} \in B_i} \exp(g(w; x_{i'})) \delta(w; x_{i'}). (8)$$

With $s_i$, we refer to $f_2(s_i)$ as the variance-reduced stochastic pooling (VRSP) estimator. We will prove in the next section that the moving average estimators $s_i$ will ensure the averaged error $\frac{1}{T} \sum_{i=0}^{T-1} \|s_i - f_1(w; X_i)\|^2$ for all bags across all iterations will converge to zero as $T \to \infty$ by properly updating the model parameter and setting the hyper-parameters. As a result, the following lemma will guarantee that the stochastic pooling estimator $f_2(s_i)$ will have a diminishing error in the long term.

**Lemma 1.** If $f_2$ is continuously differentiable on a compact domain and there exists $c > 0$ such that $f_2$ is $c$-Lipschitz continuous on that domain, then $(f_2(s_i') - f_2(f_1(w; X_i)))^2 \leq c^2\|s_i' - f_1(w; X_i)\|^2$ for $s_i', f_1(w; X_i) \in \text{dom} f_2$.

Building on the VRSP estimators, a stochastic gradient estimator of the objective can be easily computed. In particular, the gradient of $f(f_2(f_1(w; X_i)))$ in terms of $w$ can be estimated by $\nabla f_2(f_1(w; X_i)) \nabla f_2(s_i) \nabla f_1(w; X_i)$, and a stochastic gradient estimator of $f_2(f_1(w; X_i))$ can be computed by $\nabla f_2(f_1(w; X_i)) \nabla f_2(s_i)$. As a result, the stochastic gradient estimators in terms of $w$, $a$, $b$, $\alpha$ of the three terms $F_1(w, a)$, $F_2(w, b)$ and $F_3(w, a)$ of the objective are

**Algorithm 1 The Unified MIDAM Algorithm**

1: Initialize $w^0, s^0, v^0, \eta, \eta', \beta_1, \gamma_0$
2: for $t = 1, \ldots, T$ do
3: Sample a batch of positive bags $S^+_t \subset D^+$ and a batch of negative bags $S^-_t \subset D^-$
4: for each $i \in S^+_t \cup S^-_t$ do
5: Sample a mini-batch of instances $B_i \subset X_i$ and update $s_i = (1 - \gamma_0)s_i - \gamma_0 f_1(w; B_i)$
6: end for
7: Update stochastic gradient estimator of $(w, a, b)$:
8: $v_1 = \beta_1 v_1^{t-1} + (1 - \beta_1)(G_{1,w}^t + G_{2,w}^t + G_{3,w}^t)$
9: $v_2 = \beta_1 v_2^{t-1} + (1 - \beta_1)G_{1,a}^t$
10: $v_3 = \beta_1 v_3^{t-1} + (1 - \beta_1)G_{2,b}^t$
11: Update $(w^{t+1}, a^{t+1}, b^{t+1}) = (w^t, a^t, b^t) - \eta v^t$ (or the Adam-style update)
12: end for
13: computed as following, respectively:
14: $G_{1,w}^t = \hat{E}_{i \in S^+_t} \nabla f_1(w; B_i) \nabla f_2(s_i) \nabla f_1(w; X_i)$
15: $G_{2,w}^t = \hat{E}_{i \in S^+_t} \nabla f_1(w; B_i) \nabla f_2(s_i) \nabla f_1(w; X_i)$
16: $G_{3,w}^t = G_{1,a}^t + G_{2,b}^t + G_{3,a}^t$
17: $G_{1,a}^t = \hat{E}_{i \in S^+_t} \nabla f_1(w; B_i) \nabla f_2(s_i)$
18: $G_{2,b}^t = \hat{E}_{i \in S^+_t} \nabla f_1(w; B_i) \nabla f_2(s_i)$
19: $G_{3,a}^t = \hat{E}_{i \in S^+_t} \nabla f_1(w; B_i) \nabla f_2(s_i)$

4.2. The Unified Algorithm

Finally, we present the unified algorithm of MIDAM for using the two stochastic pooling estimators shown in Algorithm 1. The algorithm design is inspired by momentum-based methods for non-convex-strongly-concave min-max optimization (Guo et al., 2021). With stochastic gradient estimators in terms of the primal variable $(w, a, b)$, we compute a moving average of their gradient estimators denoted by $v^{t+1}$ in Step 7. Then we update the primal variable following the negative direction of $v^{t+1} = (v_1^{t+1}, v_2^{t+1}, v_3^{t+1})$, which is equivalent to a momentum update. The step size $\eta$ can be also replaced by the adaptive step size of Adam. For updating the dual variable $\alpha$, the algorithm simply uses the
stochastic gradient ascent update followed by a projection onto a feasible domain.

Computational Costs: Before ending this section, we discuss the per-iteration computational costs of the proposed MIDAM algorithm. The sampled instances include $B^t_i = \bigcup_{i \in S^t_i} \{B^t_i\}$, where $S^t = S^t_1 \cup S^t_2$ denotes the sampled bags, and $B^t_i$ denotes the sampled instances for the sampled bag $X^t_i$. For updating the estimators $s^t_i + 1, i \in S^t_i$, we need to conduct the forward propagations on these sampled instances for computing their prediction scores $\phi(w^t_i; x^t_i)$ (in smoothed-max pooling and attention-based pooling) and for computing their attention factor $\exp(\phi_a(w^t_i; x^t_i))$. For computing the gradient estimators, the main cost lies at the backpropagation for computing $\nabla f_i(w^t_i; B^t_i)$ of $i \in S^t_i$, which are required in computing $G^t_{1,w}, G^t_{2,w}, G^t_{3,w}$. Hence, with $S_+ = |S^t_1|$ and $S_- = |S^t_2|$ and $B = |B^t_i|$, the total costs of forward propagations and backpropagations are $O(S_+ + S_)Bd)$, where $(S_+ + S_-)B$ is the number of instances of each mini-batch. Hence this cost is independent of the total size of each bag $N_i = |X^t_i|$.

5. Convergence Analysis

Approach of Analysis. We first would like to point out the considered non-convex min-max multi-level compositional optimization problem is a new problem that has not been studied in the literature. To the best of our knowledge, the two related works are (Yuan et al., 2022; Gao et al., 2022). However, these two works only involve one inner functions to be estimated. In contrast, our problem involves many inner functions $f_1(w; X^t_i)$ to be estimated, while only a few of them are sampled for estimating their stochastic values. To tackle this challenge, we borrow a technique from (Wang & Yang, 2022) which was developed for a minimization problem with two-level compositional functions and multiple inner functions. We leverage their error bound analysis for two-level stochastic pooling estimators and combine with that of momentum-based methods for min-max optimization (Guo et al., 2021) to derive our final convergence.

Since the objective $F(w, a, b, \alpha)$ in (4) is 1-strongly concave w.r.t. $\alpha$, $\max_{a \in \Omega} F(w, a, b, \alpha)$ has unique solution and $\nabla \Phi(w, a, b)$ is Lipschitz continuous if $\nabla F$ is Lipschitz continuous. Following (Lin et al., 2019; Rafique et al., 2020), we define $\Phi(w, a, b) := \max_{a \in \Omega} F(w, a, b, \alpha)$ and use $\|\nabla \Phi(w, a, b)\|_2$ as an optimality measure.

Definition 1. $(w, a, b)$ is called an $\epsilon$-stationary point ($\epsilon \geq 0$) of a differentiable function $\Phi$ if $\|\nabla \Phi(w, a, b)\|_2 \leq \epsilon$.

Our theory is established based on the following assumption.

Assumption 1. (Smoothed-max Pooling) We assume that $\phi(w; x)$ is bounded, Lipschitz continuous, and has Lipschitz continuous gradient, i.e. there exist $B_\phi, C_\phi, L_\phi \geq 0$ such that $\|\phi(w; x)\|_2 \leq B_\phi, \|\nabla \phi(w; x)\|_2 \leq C_\phi, \|\nabla^2 \phi(w; x)\|_2 \leq L_\phi$ for each $x$.

Table 1. Data statistics for the benchmark datasets

<table>
<thead>
<tr>
<th>Data Format</th>
<th>Dataset</th>
<th>$D_\phi$</th>
<th>$D_\phi$</th>
<th>average bag size</th>
<th>#features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tabular</td>
<td>Elephant</td>
<td>100</td>
<td>100</td>
<td>6.6</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>Fox</td>
<td>100</td>
<td>100</td>
<td>6.6</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>Tiger</td>
<td>100</td>
<td>100</td>
<td>6.96</td>
<td>230</td>
</tr>
<tr>
<td>Histopathological</td>
<td>Breast Cancer</td>
<td>26</td>
<td>32</td>
<td>672</td>
<td>32x32x3</td>
</tr>
<tr>
<td></td>
<td>Colon Ade.</td>
<td>100</td>
<td>1000</td>
<td>256</td>
<td>32x32x3</td>
</tr>
<tr>
<td>MRI/CT Scans</td>
<td>OCT</td>
<td>747</td>
<td>1935</td>
<td>31</td>
<td>256x256x1</td>
</tr>
</tbody>
</table>

(Attention-based Pooling) We assume that $g(w; x)$ is bounded, Lipschitz continuous, and has Lipschitz continuous gradient and $\delta(w; x)$ is bounded, Lipschitz continuous, and has Lipschitz continuous gradient, i.e., there exist $B_g, C_g, L_g, \delta \geq 0$ such that $\|g(w; x)\|_2 \leq B_g, \|\nabla g(w; x)\|_2 \leq L_g, \|\nabla^2 g(w; x)\|_2 \leq L_g, \|\nabla^2 \delta(w; x)\|_2 \leq L_g$.

We provide some examples in which the assumption above holds: First, objective (4) with smoothed-max pooling, $\phi(w; x) = \sigma(w^\top e(w; x))$, and pre-trained, fixed $w_\gamma$. Second, objective (4) with bounded weight norms (e.g., $\|w_\alpha\|, \|w_\eta\|, \|V\|$) during the training process. Some prior works indicate that the weight norm may be bounded when weight decay regularization is used (Hao & Chen, 2022).

Theorem 1. Algorithm 1 with stepsizes $\beta_1 = O(\epsilon^2), \gamma_0 = O(\epsilon^2), \eta_0 = O\left(\min\left\{\frac{\delta_+}{B_\phi}, \frac{\delta_-}{B_\phi}\right\} \epsilon^2\right), \eta' = O(\epsilon^2)$ can find an $\epsilon$-stationary point in $T = O\left(\max\left\{\frac{D_\phi}{\delta_+}, \frac{D_\phi}{\delta_-}\right\} \epsilon^{-4} B_\phi^{-1} \right)$ iterations, where $S_+ = |S^t_1|$ and $S_- = |S^t_2|$ and $B = |B^t_i|$. Besides, the average estimation error $\frac{1}{T} \sum_{t=1}^T \mathbb{E}\left[\|s^t_i - f_1(w^t_i; X^t_i)\|_2^2\right] \leq O(\epsilon^2), \forall i$.

Remark: This theorem states that more bags and larger bag sizes lead to faster convergence of our algorithm, at the cost of more computational resources. The order of complexity $O(1/\epsilon^4)$ is the same as that of non-convex min-max optimization in (Guo et al., 2021). Proofs are in appendix.

6. Experiments

In this section, we present some experimental results. We choose datasets from three domains, namely traditional tabular datasets, histopathological image datasets, and MRI/CT datasets. Statistics of these datasets are described in Tabel 1. Details of these datasets will be presented later.

Baselines. We mainly compare with two categories of approaches for MIL with different pooling operators. The first category is optimizing the CE loss by Adam optimizer with mean, smoothed-max (smx), max, attention-based (att) poolings, denoted by CE (XX), where XX is the name of a pooling. The second category is optimizing the min-max...
We adopt a simple 2-layer feed-forward neural network (Yuan et al., 2021) with the same set of poolings, denoted as DAM (XX). We note that for large-resolution medical image datasets, deterministic pooling is unrealistic due to limits of GPU memory. For example, the CE (att) method could consume about 22 Giga-Bytes GPU memory for PDGM dataset even with a single bag of data. Medical researchers also have raised concern for the GPU constraint of large size histopathological images (Tizhoosh & Pantanowitz, 2018). Hence, we implement the naive mini-AUC loss, which are denoted as CE (3D) and DAM (3D). Our methods are denoted as MIDAM (smx) and MIDAM (att) for using two stochastic pooling operations, respectively. We fix the margin parameter as 0.1 for DAM and MIDAM. For attention-based pooling, we use the one defined in (2) with an attentional factor \( \exp(w_x \tanh(V_e(w_x; x))) \) according to (Ilse et al., 2018).

### 6.1. Results on Tabular Benchmarks

Five benchmark datasets, namely, MUSK1, MUSK2, Fox, Tiger, Elephant (Dietterich et al., 1997a; Andrews et al., 2002), are commonly used for evaluating MIL methods. For the MUSK1 and MUSK2 datasets, they contain drug molecules that will (or not) bind strongly to a target protein. Each molecule (a bag) may adopt a wide range of shapes or conformations (instances). A positive molecule has at least one shape that can bind well (although it is not known which one) and a negative molecule does not have any shapes that can make the molecule bind well (Dietterich et al., 1997a). For Fox, Tiger, and Elephant datasets, each object contains one shape that can bind well (although it is not known which one) and a negative molecule does not have any shapes that can make the molecule bind well (Dietterich et al., 1997a). Each molecule (a bag) may adopt a wide range of shapes or conformations (instances). A positive molecule has at least one shape that can bind well (although it is not known which one) and a negative molecule does not have any shapes that can make the molecule bind well (Dietterich et al., 1997a).

We adopt a simple 2-layer feed-forward neural network (FFNN) as the backbone model, whose neuron number equals data dimension. We apply tanh as the activation function for the middle layer and sigmoid as a normalization function for prediction score for computing AUC loss function. We uniformly randomly split the data with 0.9/0.1 train/test ratio and run 5-fold-cross-validation experiments with 3 different random seeds (totally 15 different trials). The initial learning rate is tuned in \{1e-1,1e-2,1e-3\}, and is decreased by 10 fold at the end of the 50-th epoch and 75-th epoch over the 100-epoch-training period. For all experiments in this work, the weight decay is fixed as \( 1e-4 \), and we fix \( n^\phi = 1, (1 - \beta_1) = 0.9 \) in our proposed algorithm decreasing by 2 fold at the same time with learning rate. We report the testing AUC based on a model with the largest validation AUC value. For each iteration, we sample 8 positive bags and 8 negative bags (\( S_+ = S_- = 8 \)), and for each bag sample at most 4 instances for our methods but use all instances for baselines, given that the dataset is small and bag size is not identical across all bags. The mean and standard deviation of testing AUC are presented in Table 2.

From the results, we observe that MIDAM (att) or MIDAM (smx) method achieves the best performance on these classical tabular benchmark datasets. This might sound surprising given that the DAM baselines use all instances for each bag for computing the pooling. To understand this, we plot the training and testing convergence curves (shown in Figure 4 in Appendix C due to limit of space). We find that the better testing performance of our MIDAM methods is probably due to that the stochastic sampling over instances prevents overfitting (since training performance is worse) and hence improves the generalization (testing performance is better). In addition, DAM is better than CE except for DAM (att).

### 6.2. Experiments on Medical Image Datasets

We choose two histopathological image datasets, namely Breast Cancer and Colon Adenocarcinoma (Gelasca et al., 2008; Borkowski et al., 2019a). These have been used in previous deep MIL works (Ilse et al., 2018) for evaluation. Histopathological images are microscopic images of the tissue for disease examination, which are prevalent for cancer diagnosis (Borkowski et al., 2019b). Since histopathological images have a high resolution, it is difficult to process the whole image. Hence MIL approaches are appealing that treat each image as a bag of local small batches. For more detailed information, please refer to (Ilse et al., 2018).

#### 6.2.1. Results on Histopathological Image Datasets

<table>
<thead>
<tr>
<th>Methods</th>
<th>Breast Cancer</th>
<th>Colon Adc.</th>
<th>PDGM</th>
<th>OCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE (3D)</td>
<td>0.925(0.061)</td>
<td>0.724(0.165)</td>
<td>0.582(0.118)</td>
<td>0.789(0.032)</td>
</tr>
<tr>
<td>DAM (3D)</td>
<td>0.725(0.2)</td>
<td>0.846(0.075)</td>
<td>0.545(0.122)</td>
<td>0.807(0.027)</td>
</tr>
<tr>
<td>CE (MB-mean)</td>
<td>0.850(0.242)</td>
<td>0.888(0.042)</td>
<td>0.616(0.023)</td>
<td>0.799(0.019)</td>
</tr>
<tr>
<td>DAM (MB-mean)</td>
<td>0.875(0.137)</td>
<td>0.877(0.017)</td>
<td>0.635(0.113)</td>
<td>0.839(0.029)</td>
</tr>
<tr>
<td>CE (MB-max)</td>
<td>0.325(0.232)</td>
<td>0.856(0.032)</td>
<td>0.462(0.108)</td>
<td>0.793(0.047)</td>
</tr>
<tr>
<td>DAM (MB-max)</td>
<td>0.475(0.215)</td>
<td>0.825(0.044)</td>
<td>0.624(0.112)</td>
<td>0.841(0.01)</td>
</tr>
<tr>
<td>CE (MB-smx)</td>
<td>0.575(0.127)</td>
<td>0.863(0.031)</td>
<td>0.491(0.111)</td>
<td>0.826(0.018)</td>
</tr>
<tr>
<td>DAM (MB-smx)</td>
<td>0.725(0.184)</td>
<td>0.905(0.01)</td>
<td>0.659(0.058)</td>
<td>0.829(0.008)</td>
</tr>
<tr>
<td>CE (MB-att)</td>
<td>0.90(0.146)</td>
<td>0.960(0.042)</td>
<td>0.546(0.072)</td>
<td>0.823(0.017)</td>
</tr>
<tr>
<td>DAM (MB-att)</td>
<td>0.875(0.112)</td>
<td>0.882(0.029)</td>
<td>0.624(0.112)</td>
<td>0.842(0.013)</td>
</tr>
<tr>
<td>MIDAM-smx</td>
<td>0.875(0.137)</td>
<td>0.91(0.02)</td>
<td>0.669(0.032)</td>
<td>0.848(0.01)</td>
</tr>
<tr>
<td>MIDAM-att</td>
<td>0.95(0.1)</td>
<td>0.893(0.08)</td>
<td>0.635(0.052)</td>
<td>0.843(0.012)</td>
</tr>
</tbody>
</table>

#### 6.2.2. Results on Medical Image Datasets

For Fox, Tiger, and Elephant datasets, each object contains features extracted from an image. Each positive bag is a bag that contains the animal of interest (Andrews et al., 2002). The testing AUC on benchmark datasets.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Breast Cancer</th>
<th>Colon Adc.</th>
<th>PDGM</th>
<th>OCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE (mean)</td>
<td>0.900(0.113)</td>
<td>0.701(0.049)</td>
<td>0.855(0.094)</td>
<td>0.885(0.044)</td>
</tr>
<tr>
<td>DAM (mean)</td>
<td>0.920(0.076)</td>
<td>0.759(0.094)</td>
<td>0.838(0.060)</td>
<td>0.902(0.037)</td>
</tr>
<tr>
<td>CE (max)</td>
<td>0.678(0.121)</td>
<td>0.657(0.147)</td>
<td>0.855(0.094)</td>
<td>0.885(0.044)</td>
</tr>
<tr>
<td>DAM (max)</td>
<td>0.739(0.126)</td>
<td>0.859(0.099)</td>
<td>0.859(0.060)</td>
<td>0.902(0.037)</td>
</tr>
<tr>
<td>CE (smx)</td>
<td>0.769(0.121)</td>
<td>0.851(0.111)</td>
<td>0.865(0.078)</td>
<td>0.902(0.068)</td>
</tr>
<tr>
<td>DAM (smx)</td>
<td>0.806(0.118)</td>
<td>0.854(0.108)</td>
<td>0.867(0.070)</td>
<td>0.902(0.052)</td>
</tr>
<tr>
<td>CE (att)</td>
<td>0.808(0.112)</td>
<td>0.760(0.122)</td>
<td>0.870(0.073)</td>
<td>0.883(0.069)</td>
</tr>
<tr>
<td>DAM (att)</td>
<td>0.760(0.139)</td>
<td>0.757(0.154)</td>
<td>0.848(0.067)</td>
<td>0.872(0.074)</td>
</tr>
<tr>
<td>MIDAM-smx</td>
<td>0.834(0.112)</td>
<td>0.905(0.088)</td>
<td>0.862(0.071)</td>
<td>0.879(0.010)</td>
</tr>
<tr>
<td>MIDAM-att</td>
<td>0.826(0.107)</td>
<td>0.843(0.107)</td>
<td>0.733(0.097)</td>
<td>0.876(0.066)</td>
</tr>
</tbody>
</table>

8 positive bags and 8 negative bags (\( S_+ = S_- = 8 \)), and for each bag sample at most 4 instances for our methods but use all instances for baselines, given that the dataset is small and bag size is not identical across all bags. The mean and standard deviation of testing AUC are presented in Table 2.

The code is available at https://github.com/DixianZhu/MIDAM
Breast Cancer, there are 58 weakly labeled $896 \times 768$ hema-
toxylin and eosin (H&E) stained whole-slide images. An
image is labeled malignant if it contains breast cancer cells,
otherwise it is benign (examples shown in Figure 3). We
divide every image into $32 \times 32$ patches. This results in
672 patches per bag. For Colon Adenocarcinoma dataset\(^2\),
there are originally 5000 (H&E) images for benign colon
tissue and 5000 for Colon Adenocarcinoma. We uniformly
randomly sample 1000 benign images and 100 Adenocar-
cinoma images to form the new Colon Ade. dataset for
our study. We divide every $512 \times 512$ image into $32 \times 32$
patches and get 256 patches for each image. We also use
two real-world MRI/OCT image datasets. The first data set
is from the University of California San Francisco Preopera-
tive Diffuse Glioma MRI (UCSF-PDGM) (Calabrese et al.,
2022), short as PDGM in this work. The problem is to pre-
dict patients with grade II or grade IV diffuse gliomas. The
second dataset contains multiple OCT images for a large
number of patients (Xie et al., 2022). The goal is to predict
hypertension from OCT images, which is useful for physi-
cians to understand the relationship between eye-diseases
and Hypertension. Exemplar images of two datasets are
shown in Figure 3 in the Appendix C.

For all the medical images, we adopt ResNet20 as the back-
bone model. For AUC loss function, we apply sigmoid as
normalization for the output. The weight decay is fixed as
1e-4. For all the experiments, we run 100 epochs for
each trial and decrease learning rate by 10 fold at the end
of the 50-th epoch and 75-th epoch. For the Breast Cancer
dataset, we generate data train/test (0.9/0.1) splitting 2 times
with different random seeds and conduct five-fold cross
validation (10 trials). For the other datasets, we do single

---

random train/test (0.9/0.1) splitting and conduct five-fold cross validation (5 trials). The margin parameter is tuned in $\{0, 0.1, 0.5, 1.0\}$ for AUC loss function. The initial learning rate is tuned in $\{1e-1, 1e-2, 1e-3\}$ for histopathological image datasets, and is fixed as $1e-2$ for PDGM, $1e-1$ for OCT.

The results are shown in Table 3. From these results, we make the following observations. (1) Our MIDAM still performs the best. Although CE (3D) on Breast Cancer, and CE (MB-att) on the two histopathological image datasets are competitive, most of the CE based approaches are less competitive with DAM based approaches. (2) By comparing MIDAM (att) with DAM (MB-att) and MIDAM (smx) with DAM (MB-smx), our methods perform consistently better. This confirms the importance of using variance-reduced stochastic pooling operations instead of the naive mini-batch based stochastic poolings. This can be also verified by comparing their training/testing convergence in Figure 1(a,b) and Figure 5 in the Appendix. (3) For Breast Cancer data, our method MIDAM (att) performs better than all baseline methods. It is notable that CE (3D) and CE (MB-att) are competitive approaches but still have worse performance than MIDAM (att). On Colon Ade. dataset, our method MIDAM (smx) performs the best and CE (MB-att) is still competitive. Finally, we see that there is no clear winner between MIDAM (att) and MIDAM (smx). (4) In general, MIL pooling based methods can achieve better performance than the traditional baseline using 3D data input. Hence, our MIDAM algorithms are a good fit for 3D medical images.

### 6.3. Ablation Studies

First, we conduct an experiment to study the influence for different instance-batch sizes ($B$) on four tabular datasets. The results on MUSK2 are shown in Figure 1 (c,d) with more plotted in Figure 6, which demonstrate our methods converge faster with a larger $B$ with fixed bag-batch size $S_+$ and $S_-$. In addition, we observe that with $B = 4$ MIDAM converges to almost same level as using all instances in 100 epochs, even for the MUSK2 dataset with average bag size as 64.69. Second, we show an ablation study on the two histopathological image datasets that fixes the total budget for bag-batch-size $\times$ instance-batch-size. Exemplar results are plotted in Figure 1 (e,f,g,h) with more results plotted in Figure 7. We can see that due to sampling of instances per-bag, we have more flexibility to choose the bag-batch size $S_+ = S_- = \hat{B}/2$ and instance-batch size $\hat{B}$ to have faster training, e.g., with $\hat{B} = 4, B = 64$ MIDAM converges the fastest, which demonstrates the superiority of our design. Third, we demonstrate the effectiveness of stochastic attention pooling based MIDAM on a Breast Cancer example by attention weights and prediction scores for each instance (image patch). The results are presented in Figure 2, where we can observe the lesion parts for the histopathology tissue have larger prediction scores and attention weights (the brighter patches). More demonstration on a negative examples are included in Figure 8 in Appendix C, where the attention module focus on a blank patch to generate low overall prediction score.

### 7. Conclusions

We have proposed efficient algorithms for multi-instance deep AUC maximization. Our algorithms are based on variance-reduced stochastic poolings in a spirit of compositional optimization to enjoy a provable convergence. We have demonstrated the effectiveness and superiority of our algorithms on benchmark datasets and real-world high-resolution medical image datasets.

### 8. Acknowledgement

This work is partially supported by NSF Career Award 2246753, NSF Grant 2246757 and NSF Grant 2246756.

### References


Calders, T. and Jaroszewicz, S. Efficient auc optimization for classification. In *European Conference on Principles


Yuan, Z., Yan, Y., Sonka, M., and Yang, T. Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. In
Provable Multi-instance Deep AUC Maximization with Stochastic Pooling


A. Technical Lemmas

Lemma 2. Based on Assumption 1, we have that $h(w; x)$ is bounded, Lipschitz continuous, and has Lipschitz continuous gradient, i.e., there exists $B_h, C_h, L_h \geq 0$ such that $|h(w; x)| \leq B_h, \|\nabla h(w; x)\|_2 \leq C_h,$ and $\nabla^2 h(w; x) \leq L_h I.$

Proof. **Smoothed-max pooling:** Property of the LogSumExp (LSE) function implies that

$$|h(w; x)| \leq \tau \max_{x \in \mathcal{X}} \frac{\phi(w(x))}{\tau} + (1 - \tau) \log |\mathcal{X}|.$$ 

The norm of $\nabla h(w; x)$ can be bounded as

$$\|\nabla h(w; x)\|_2 = \left\| \sum_{x \in \mathcal{X}} \frac{\exp(\phi(w(x))/\tau)}{\sum_{x' \in \mathcal{X}} \exp(\phi(w(x'))/\tau)} \nabla \phi(w; x) \right\|_2 \leq C_\phi.$$ 

The norm of $\nabla^2 h(w; x)$ can be bounded as

$$\|\nabla^2 h(w; x)\|_2 \leq \left\| \sum_{x \in \mathcal{X}} \frac{\exp(\phi(w(x))/\tau)}{\sum_{x' \in \mathcal{X}} \exp(\phi(w(x'))/\tau)} \nabla \phi(w; x) \right\|_2 + \left\| \sum_{x \in \mathcal{X}} \frac{\exp(\phi(w(x))/\tau)}{\sum_{x' \in \mathcal{X}} \exp(\phi(w(x'))/\tau)} \nabla^2 \phi(w; x) \right\|_2 \leq 2C_\phi/\tau + L_\phi.$$ 

**Attention-based pooling:** According to (3), it is clear that $|h(w; x)| \leq 1.$ The norm of $\nabla h(w; x)$ can be bounded as

$$\|\nabla h(w; x)\|_2 \leq 0.25 \left\| \sum_{x \in \mathcal{X}} \frac{\exp(\phi(w(x))/\tau)}{\sum_{x' \in \mathcal{X}} \exp(g(w(x')))} \nabla g(w; x) \right\|_2 + 0.25 \left\| \sum_{x \in \mathcal{X}} \frac{\exp(g(w(x))}{\sum_{x' \in \mathcal{X}} \exp(g(w(x')))} \nabla \delta(w; x) \right\|_2 + 0.25 \left\| \sum_{x \in \mathcal{X}} \frac{\exp(g(w(x))}{\sum_{x' \in \mathcal{X}} \exp(g(w(x')))} \nabla^2 \delta(w; x) \right\|_2 \leq 0.5C_\delta B_\delta + 0.25C_\delta.$$

For brevity, we denote the softmax function $s(w; x) := \frac{\exp(g(w(x))}{\sum_{x' \in \mathcal{X}} \exp(g(w(x'))$. The norm of $\nabla^2 h(w; x)$ can be bounded as

$$\|\nabla^2 h(w; x)\|_2 \leq 0.1 \|\nabla h(w; x)\|_2 + 0.25 \left\| \sum_{x \in \mathcal{X}} s(w; x) (\delta(w; x) \nabla g(w; x) \nabla \delta(w; x) \nabla^2 g(w; x) \nabla^2 \delta(w; x) \nabla g(w; x) \right\|_2 + 0.25 \left\| \sum_{x \in \mathcal{X}} s(w; x) \nabla g(w; x) \nabla \delta(w; x) \nabla^2 g(w; x) \nabla^2 \delta(w; x) \nabla g(w; x) \right\|_2 + 0.25 \left\| \sum_{x \in \mathcal{X}} s(w; x) \nabla g(w; x) \nabla \delta(w; x) \nabla^2 g(w; x) \nabla^2 \delta(w; x) \nabla g(w; x) \right\|_2 + 0.25 \left\| \sum_{x \in \mathcal{X}} s(w; x) \nabla g(w; x) \nabla \delta(w; x) \nabla^2 g(w; x) \nabla^2 \delta(w; x) \nabla g(w; x) \right\|_2 + 0.25 \left\| \sum_{x \in \mathcal{X}} s(w; x) \nabla g(w; x) \nabla \delta(w; x) \nabla^2 g(w; x) \nabla^2 \delta(w; x) \nabla g(w; x) \right\|_2 \leq 0.1C_\delta^2 + 0.5(B_\delta C_\delta + C_\delta) C_\delta + 0.25(C_\delta C_\delta + B_\delta L_\delta + L_\delta) + 0.25(B_\delta C_\delta + C_\delta)^2 C_\delta + 0.25C_\delta^2 (B_\delta + 1) + 0.25B_\delta(C_\delta^2 + L_\delta).$$
Lemma 3. Under Assumption 1, MIDAM with $\gamma \in (0, 1)$, $s_t^0 = 0$, we have $|s_t^i| \leq B_s$ for all $t > 0$.

Proof. This lemma follows from Assumption 1 and the facts that $f_1$ is continuously differentiable on its domain and the update formula of $s_i$ is a convex combination. \hfill \Box

Lemma 4. If $\eta \in (0, 0.5)$ and $a^0 = 0, b^0 = 0$, there exist $B_a, B_b > 0 |a^t| \leq B_a, |b^t| < B_b$ for all $t > 0$.

Proof. Note that $G_{1,a}^t = -\frac{2}{|S_i^t|} \sum_{i \in S_i^t} (f_2(s_i^t) - a^t)$ and $G_{2,b}^t = -\frac{2}{|S_i^t|} \sum_{i \in S_i^t} (f_2(s_i^t) - b^t)$. Thus, the update formulae of $a$ and $b$ can be re-written as

$$a^{t+1} = a^t - \eta G_{1,a}^t = (1 - 2\eta)a^t + 2\eta \frac{1}{|S_i^t|} \sum_{i \in S_i^t} f_2(s_i^t),$$

$$b^{t+1} = b^t - \eta G_{2,b}^t = (1 - 2\eta)b^t + 2\eta \frac{1}{|S_i^t|} \sum_{i \in S_i^t} f_2(s_i^t).$$

Due to Lemma 3 and the fact that $f_2$ is continuously differentiable on its domain, $a^t$ and $b^t$ are bounded in all iterations as long as $\eta \in (0, 0.5)$ such that the update formulae of $a^t$ and $b^t$ are convex combinations. \hfill \Box

Lemma 5. Under Assumption 1, there exists $L_F > 0$ such that $\nabla F$ is $L_F$-Lipschitz continuous.

Proof. Note that $h(w; X_i) = f_2(f_1(w; X_i))$, $\nabla h(w; X_i) = \nabla f_1(w; X_i) \nabla f_2(f_1(w; X_i))$. For distinct $(w, a, b, \alpha)$ and $(w', a', b', \alpha')$, we have

$$\|\nabla_{(w,a,b)} F(w, a, b, \alpha) - \nabla_{(w',a',b',\alpha')} F(w', a', b', \alpha')\|_2 \leq 2 \left( \frac{2}{|D_+|} \sum_{i \in D_+} (h(w; X_i) - a) - \nabla h(w', X_i)(h(w'; X_i) - a) \right) + 2 \left( \frac{2}{|D_-|} \sum_{i \in D_-} (h(w; X_i) - b) - \nabla h(w', X_i)(h(w'; X_i) - b) \right)$$

$$+ \alpha \frac{1}{|D_+|} \sum_{i \in D_+} \nabla h(w; X_i) - \alpha \frac{1}{|D_-|} \sum_{i \in D_-} \nabla h(w; X_i) \right) \leq 2 \left( \frac{2}{|D_+|} \sum_{i \in D_+} (h(w; X_i) - a) - \frac{2}{|D_+|} \sum_{i \in D_+} (h(w'; X_i) - a') \right) + 2 \left( \frac{2}{|D_-|} \sum_{i \in D_-} (h(w; X_i) - b) - \frac{2}{|D_-|} \sum_{i \in D_-} (h(w'; X_i) - b') \right)$$

$$+ |\alpha - \alpha'| \leq 2(2L_B + 2C_2^2 + (B_a + B_b)L_h + B_0L_h + 3C_h) \|w - w'\|_2 + 2(C_h + 1)|a - a'| + 2(C_h + 1)|b - b'| + 2(C_h + 1)|\alpha - \alpha'|.$$ \hfill \Box

Lemma 6 (Lemma 4.3 in Lin et al. (2019)). For an $F$ defined in (4) that has Lipschitz continuous gradient and $\Phi(w, a, b) := \max_{x \in \Omega} F(w, a, b, \alpha)$ with a convex and bounded $\Omega$, we have that $\Phi(w, a, b)$ is $L_\Phi$-smooth and $\nabla \Phi(w, a, b) = \nabla_{(w,a,b)} F(w, a, b, \alpha^*(w, a, b))$. Besides, $\alpha^*(w, a, b)$ is 1-Lipschitz continuous.
where \( E \cdot t \cdot = (1 + 1, v_1, a, b, G_t^W) \), and \( G_t^W \) are, where

\[
G_{t,1}^W = \hat{E}_{i \in S_t^W} f_1(w_t; B_i) \nabla f_2(f_1(w_t; X_i)) \nabla f_1(f_2(f_1(w_t; X_i)), a'),
\]

\[
G_{t,2}^W = \hat{E}_{i \in S_t^W} f_2(f_1(w_t; X_i))) - \hat{E}_{i \in S_t^W} f_1(w_t; B_j) \nabla f_2(f_1(w_t^i; X_i)), b'),
\]

\[
G_{t,3}^W = \hat{E}_{i \in S_t^W} f_2(f_1(w_t; X_i)) a',
\]

\[
G_{t,4}^W = \hat{E}_{i \in S_t^W} \nabla f_2(f_2(f_1(w_t; X_i), a'), b').
\]

**Lemma 7** (Lemma 11 in Wang et al. (2021a)). Suppose that \( X = \frac{1}{n} \sum_{i=1}^n X_i \). If we sample a size-\( B \) minibatch \( B \) from \( \{1, \ldots, n\} \) uniformly at random, we have \( E \left[ \frac{1}{B} \sum_{i \in B} (X_i - X) \right] = 0 \) and

\[
E \left[ \frac{1}{B} \sum_{i \in B} (X_i - X) \right]^2 \leq \frac{n - B}{B(n - 1)} \sum_{i=1}^n \|X_i - X\|^2 \leq \frac{n - B}{B(n - 1)} \sum_{i=1}^n \|X_i\|^2.
\]

**Lemma 8.** Under Assumption 1, there exists \( C_G, C_T > 0 \) for MIDAM such that

\[
\sum_{t=0}^{T-1} E \left[ \Delta^t \right] \leq \frac{\Delta_0}{\beta_1} + 2T \beta_1 C_G + 5L^2 \sum_{t=0}^{T-1} E \left[ \Psi^{t+1} \right] + \frac{3n^2 \beta_2^2}{T^2} \sum_{t=0}^{T-1} E \left[ \|v_t\|^2 \right] + 5C_T \sum_{t=0}^{T-1} E \left[ T^{t+1} \right] + 5C_T \sum_{t=0}^{T-1} E \left[ T^{t+1} \right]
\]

\[
+ 5C_T \sum_{t=0}^{T-1} \frac{1}{D_x} \sum_{i \in S_t^W} \|s_i^{t+1} - s_i^t\|^2 \left[ \sum_{i \in S_t^W} \|s_i^{t+1} - s_i^t\|^2 \right].
\]

where \( \Delta^t := \|v_t - \nabla \Phi(W_t)\|_2^2, \ Y_t^{+} := \frac{1}{T^2} \sum_{i \in D_x} \|s_i^t - f_1(w_t^i; X_i)\|_2^2, \ Y_t^{-} := \frac{1}{T^2} \sum_{i \in D_x} \|s_i^t - f_1(w_t^i; X_i)\|_2^2, \ \Psi^t := \|\alpha^t - \alpha^* (W_t)\|_2^2.\)

**Proof.** Based on the update rule of \( v_t \), we have

\[
E_{t+1} [\Delta^{t+1}] = E_t \left[ \|v^{t+1} - \nabla \Phi(W^{t+1})\|_2^2 \right] = E_t \left[ \|v^{t+1} - \nabla W F(W^{t+1}, \alpha^* (W^{t+1}))\|_2^2 \right]
\]

\[
= E_{t+1} \left[ \| (1 - \beta_1) v_t + \beta_1 G_{W_t}^{t+1} - \nabla W F(W^{t+1}, \alpha^* (W^{t+1})) \|_2^2 \right]
\]

\[
= E_{t+1} \left[ \left\| (1 - \beta_1) (v_t - \nabla \Phi(W_t)) + (1 - \beta_1) (\nabla \Phi(W_t) - \nabla \Phi(W^{t+1})) + \beta_1 (G_{W_t}^{t+1} - G_{W_t}^{t+1}) \right\|^2 \right].
\]

Note that \( E_{t+1}[\bullet \cdot \bullet] = 0, E_{t+1}[\bullet \cdot \bullet] = 0, E_{t+1}[\bullet \cdot \bullet] = 0. \) Then,

\[
E_{t+1} [\Delta^{t+1}]
\]

\[
= (1 - \beta_1)^2 \Delta_t + (1 - \beta_1)^2 \| \nabla \Phi(W_t) - \nabla \Phi(W^{t+1}) \|_2^2 + \beta_1^2 E_{t+1} [\|G_{W_t}^{t+1} - G_{W_t}^{t+1}\|_2^2]
\]

\[
+ \beta_1^2 E_{t+1} [\|G_{W_t}^{t+1} - \nabla W F(W^{t+1}, \alpha^{t+1})\|_2^2] + \beta_1^2 \| \nabla W F(W^{t+1}, \alpha^{t+1}) - \nabla \Phi(W^{t+1}) \|_2^2
\]

\[
+ 2(1 - \beta_1)^2 \| v_t - \nabla \Phi(W_t), \nabla \Phi(W_t) - \nabla \Phi(W^{t+1}) \|_2^2 + 2\beta_1 (1 - \beta_1) E_{t+1} [\|v_t - \nabla \Phi(W_t), \nabla W F(W^{t+1}, \alpha^{t+1}) - \nabla \Phi(W^{t+1})\|_2^2]
\]

\[
+ 2\beta_1 (1 - \beta_1) E_{t+1} [\| \nabla \Phi(W_t) - \nabla \Phi(W^{t+1}), G_{W_t}^{t+1} - G_{W_t}^{t+1} \|_2^2]
\]

\[
+ 2\beta_1^2 E_{t+1} [\|G_{W_t}^{t+1} - G_{W_t}^{t+1}, \nabla W F(W^{t+1}, \alpha^{t+1}) - \nabla \Phi(W^{t+1})\|_2^2]
\]

\[
+ 2\beta_1^2 E_{t+1} [\|G_{W_t}^{t+1} - G_{W_t}^{t+1}, \nabla W F(W^{t+1}, \alpha^{t+1}) - \nabla \Phi(W^{t+1})\|_2^2].
\]

21
Use Young’s inequality for products.
\[
\begin{align*}
E_{t+1} \left[ \Delta^{t+1} \right] & \leq (1 - \beta_1)^2 (1 + \beta_1) \Delta^t + \frac{3(1 - \beta_1)^2 (1 + \beta_1)}{\beta_1} \left\| \nabla \Phi(W^t) - \nabla \Phi(W^{t+1}) \right\|^2 + 2 \beta_1^2 E_{t+1} \left[ \| \tilde{G}_W^{t+1} - \nabla W F(W^{t+1}, \alpha^{t+1}) \|^2 \right] \\
& \quad + (3 \beta_1 + 5 \beta_1^2/3) \left\| G_W^{t+1} - \tilde{G}_W^{t+1} \right\|^2 + (3 \beta_1 + 5 \beta_1^2/3) E_{t+1} \left[ \| \nabla W F(W^{t+1}, \alpha^{t+1}) - \nabla \Phi(W^{t+1}) \|^2 \right] \\
& \leq (1 - \beta_1)^2 \Delta^t + \frac{3L_2 \eta^2}{\beta_1} \left\| \nu \nu \right\|^2 + 2 \beta_1^2 E_{t+1} \left[ \| \tilde{G}_W^{t+1} - \nabla W F(W^{t+1}, \alpha^{t+1}) \|^2 \right] \\
& \quad + 5\beta_1 E_{t+1} \left[ \| G_W^{t+1} - \tilde{G}_W^{t+1} \|^2 \right] + 5\beta_1 || \nabla W F(W^{t+1}, \alpha^{t+1}) - \nabla \Phi(W^{t+1}) ||^2 ,
\end{align*}
\]

Note that \( s_i^t, a^t, b^t, \alpha^t \) are bounded due to Lemma 3, Lemma 4 and the projection step of updating \( \alpha \). Besides, there exist \( B_{f_1}, C_{f_1}, B_{f_2}, C_{f_2}, L_{f_2} > 0 \) such that \( \| f_1 \|_2 \leq B_{f_1}, \| \nabla f_1 \|_2 \leq C_{f_1}, \| f_2 \|_2 \leq B_{f_2}, \| \nabla f_2 \|_2 \leq C_{f_2}, \| \nabla^2 f_2 \|_2 \leq L_{f_2} \) due to Assumption 1. Then, the definition of \( \tilde{G}_1^{t+1}, \tilde{G}_2^{t+1}, \tilde{G}_3^{t+1}, \tilde{G}_4^{t+1} \) leads to
\[
\begin{align*}
E_{t+1} \left[ \| G_W^{t+1} - \nabla W F(W^{t+1}, \alpha^{t+1}) \|^2 \right] &= \sum_{i \in S_{t+1}} \nabla f_1(w^{t+1}, \alpha^{t+1}) \left( \nabla f_2(s_i^t) f_2(s_i^t) - \nabla f_2(f_1(w^{t+1}, \alpha^t)) f_2(f_1(w^{t+1}, \alpha^t)) \right) \\
& \leq \frac{2}{|S_{t+1}|} \sum_{i \in S_{t+1}} \nabla f_1(w^{t+1}, \alpha^{t+1}) \left( \nabla f_2(s_i^t) f_2(s_i^t) - \nabla f_2(f_1(w^{t+1}, \alpha^t)) f_2(f_1(w^{t+1}, \alpha^t)) \right) \\
& \quad + \frac{2}{|S_{t+1}|} \sum_{i \in S_{t+1}} \nabla f_1(w^{t+1}, \alpha^{t+1}) \left( \nabla f_2(s_i^t) f_2(s_i^t) - \nabla f_2(f_1(w^{t+1}, \alpha^t)) f_2(f_1(w^{t+1}, \alpha^t)) \right) \\
& \quad + \frac{2}{|S_{t+1}|} \sum_{i \in S_{t+1}} (f_2(s_i^t) - f_2(f_1(w^{t+1}, \alpha^t)))^2 + \frac{2}{|S_{t+1}|} \sum_{i \in S_{t+1}} (f_2(s_i^t) - f_2(f_1(w^{t+1}, \alpha^t)))^2 ,
\end{align*}
\]

Next, we turn to bound the \( \Delta \) term.
\[
\begin{align*}
E_{t+1} \left[ \| G_W^{t+1} - \tilde{G}_W^{t+1} \|^2 \right] & \leq 3E_{t+1} \left[ \left( \sum_{i \in S_{t+1}} \nabla f_1(w^{t+1}, \alpha^{t+1}) \left( \nabla f_2(s_i^t) f_2(s_i^t) - \nabla f_2(f_1(w^{t+1}, \alpha^t)) f_2(f_1(w^{t+1}, \alpha^t)) \right) \right)^2 \right] \\
& \quad + 3E_{t+1} \left[ \left( \sum_{i \in S_{t+1}} \nabla f_1(w^{t+1}, \alpha^{t+1}) \left( \nabla f_2(s_i^t) f_2(s_i^t) - \nabla f_2(f_1(w^{t+1}, \alpha^t)) f_2(f_1(w^{t+1}, \alpha^t)) \right) \right)^2 \right] \\
& \quad + 6E_{t+1} \left[ \left( \frac{1}{|S_{t+1}|} \sum_{i \in S_{t+1}} \nabla f_1(w^{t+1}, \alpha^{t+1}) \left( \nabla f_2(s_i^t) - f_2(f_1(w^{t+1}, \alpha^t)) \right) \right)^2 \right] \\
& \quad + 6E_{t+1} \left[ \left( \frac{1}{|S_{t+1}|} \sum_{i \in S_{t+1}} \nabla f_1(w^{t+1}, \alpha^{t+1}) \left( \nabla f_2(s_i^t) - f_2(f_1(w^{t+1}, \alpha^t)) \right) \right)^2 \right] \\
& \quad + E_{t+1} \left[ \left( \sum_{i \in S_{t+1}} (f_2(s_i^t) - f_2(f_1(w^{t+1}, \alpha^t)))^2 \right)^2 \right] + E_{t+1} \left[ \left( \sum_{i \in S_{t+1}} (f_2(s_i^t) - f_2(f_1(w^{t+1}, \alpha^t)))^2 \right)^2 \right] .
\end{align*}
\]

Provable Multi-instance Deep AUC Maximization with Stochastic Pooling
Note that \( s_i^t \) and \( w_{t+1} \) are independent of \( S_+^{t+1} \) and \( S_-^{t+1} \).

\[
E_{t+1} \left[ \left\| G_{t+1}^{W} - G_{t+1}^{W} \right\|_2^2 \right] \\
\leq 12C^2 f_1 \frac{1}{D_+} \sum_{i \in D_+} \left\| \nabla f_2(s_i^t) f_2(s_i^t) - \nabla f_2(f_1(w_{t+1}; \chi_i)) f_2(f_1(w_{t+1}; \chi_i)) \right\|_2^2 \\
+ 12C^2 f_1 \frac{1}{D_-} \sum_{i \in D_-} \left\| \nabla f_2(s_i^t) f_2(s_i^t) - \nabla f_2(f_1(w_{t+1}; \chi_i)) f_2(f_1(w_{t+1}; \chi_i)) \right\|_2^2 \\
+ 6B^2 f_1 L_2 f_2 \frac{1}{D_+} \sum_{i \in D_+} \left\| s_i^t - f_1(w_{t+1}; \chi_i) \right\|_2^2 + 6B^2 f_1 L_2 f_2 \frac{1}{D_-} \sum_{i \in D_-} \left\| s_i^t - f_1(w_{t+1}; \chi_i) \right\|_2^2 \\
+ 4C^2 f_2 \frac{1}{D_+} \sum_{i \in D_+} \left\| s_i^t - f_1(w_{t+1}; \chi_i) \right\|_2^2 + 4C^2 f_2 \frac{1}{D_-} \sum_{i \in D_-} \left\| s_i^t - f_1(w_{t+1}; \chi_i) \right\|_2^2 \\
\leq C_T \left( \frac{1}{D_+} E_{t+1} \left[ \sum_{i \in D_+} \left\| s_i^{t+1} - f_1(w_{t+1}; \chi_i) \right\|_2^2 \right] + \frac{1}{D_-} E_{t+1} \left[ \sum_{i \in D_-} \left\| s_i^{t+1} - f_1(w_{t+1}; \chi_i) \right\|_2^2 \right] \right) \\
+ C_T \left( \frac{1}{D_+} E_{t+1} \left[ \sum_{i \in D_+} \left\| s_i^{t+1} - s_i^t \right\|_2^2 \right] + \frac{1}{D_-} E_{t+1} \left[ \sum_{i \in D_-} \left\| s_i^{t+1} - s_i^t \right\|_2^2 \right] \right),
\]

where we define \( C_T := 48C^2 f_1 (C^2 f_2 + B^2 f_2^2) + 12C^2 f_1 B^2 f_2 + 8C^2 f_2 \). Note that \( s_i^{t+1} = s_i^t \) for those \( i \not\in S_+^t \cup S_-^t \). Then,

\[
\sum_{i \in D_+} \left\| s_i^{t+1} - s_i^t \right\|_2^2 = \sum_{i \in S_+^t} \left\| s_i^{t+1} - s_i^t \right\|_2^2, \\
\sum_{i \in D_-} \left\| s_i^{t+1} - s_i^t \right\|_2^2 = \sum_{i \in S_-^t} \left\| s_i^{t+1} - s_i^t \right\|_2^2.
\]

We define

\[
\Upsilon_+^t := \frac{1}{D_+} \sum_{i \in D_+} \left\| s_i^t - f_1(w_{t}; \chi_i) \right\|_2^2, \quad \Upsilon_-^t := \frac{1}{D_-} \sum_{i \in D_-} \left\| s_i^t - f_1(w_{t}; \chi_i) \right\|_2^2, \quad \Psi^t = \left\| \alpha - \alpha^*(W^t) \right\|_2^2,
\]

such that

\[
E_{t+1}[\Delta^{t+1}] \leq (1 - \beta_1) \Delta^t + \frac{3L_2^2 \eta^2}{\beta_1} \left\| \nabla \psi^t \right\|_2^2 + 2\beta_1^2 C_G + 5\beta_1 C_T \left( \Upsilon_+^t + C_T \Upsilon_-^t \right) + 5\beta_1^2 L_2^2 \Psi^t \\
+ 5\beta_1 C_T \left( \frac{1}{D_+} E_{t+1} \left[ \sum_{i \in S_+^t} \left\| s_i^{t+1} - s_i^t \right\|_2^2 \right] + \frac{1}{D_-} E_{t+1} \left[ \sum_{i \in S_-^t} \left\| s_i^{t+1} - s_i^t \right\|_2^2 \right] \right).
\]

Sum over \( t = 0, \ldots, T - 1 \),

\[
\sum_{t=0}^{T-1} E_{t+1}[\Delta^t] \leq \frac{\Delta^0}{\beta_1} + 2T\beta_1 C_G + 5L_2^2 \sum_{i=0}^{T-1} E_{t+1}[\Psi^t] + \frac{3L_2^2 \eta^2}{\beta_1} \sum_{t=0}^{T-1} E_{t+1}[\left\| \nabla \psi^t \right\|_2^2] + 5C_T \sum_{t=0}^{T-1} E_{t+1}[\Upsilon_+^t] + 5C_T \sum_{t=0}^{T-1} E_{t+1}[\Upsilon_-^t] \\
+ 5C_T \sum_{t=0}^{T-1} \frac{1}{D_+} E_{t+1}[\sum_{i \in S_+^t} \left\| s_i^{t+1} - s_i^t \right\|_2^2] + 5C_T \sum_{t=0}^{T-1} \frac{1}{D_-} E_{t+1}[\sum_{i \in S_-^t} \left\| s_i^{t+1} - s_i^t \right\|_2^2].
\]

\[\square\]

**Lemma 9 (Lemma 1 in Wang & Yang (2022)).** Suppose that \( |S_+^t| \equiv S_+ \), \( |S_-^t| \equiv S_- \) and we define \( D_+ = |D_+| \), \( D_- = |D_-| \). Under Assumption 1, MIDAM satisfies that

\[
\sum_{t=0}^{T-1} E_{t+1}[\Upsilon_+^t] \leq \frac{4D_+ \Upsilon_0^0}{\gamma_0 S_+} + \frac{8T \gamma_0 B^2 f_2 (N - B)}{B(N-1)} + \frac{20D_2^2 \eta^2 C_2^2}{\gamma_0^2 S_+^2} \sum_{t=0}^{T-1} E_{t+1}[\left\| \nabla \psi^t \right\|_2^2] - \frac{1}{\gamma_0 S_+} \sum_{t=0}^{T-1} E_{t+1}[\sum_{i \in S_+^t} \left\| s_i^{t+1} - s_i^t \right\|_2^2],
\]

\[
\sum_{t=0}^{T-1} E_{t+1}[\Upsilon_-^t] \leq \frac{4D_- \Upsilon_0^0}{\gamma_0 S_-} + \frac{8T \gamma_0 B^2 f_2 (N - B)}{B(N-1)} + \frac{20D_2^2 \eta^2 C_2^2}{\gamma_0^2 S_-^2} \sum_{t=0}^{T-1} E_{t+1}[\left\| \nabla \psi^t \right\|_2^2] - \frac{1}{\gamma_0 S_-} \sum_{t=0}^{T-1} E_{t+1}[\sum_{i \in S_-^t} \left\| s_i^{t+1} - s_i^t \right\|_2^2].
\]
Lemma 10. Under Assumption 1, MIDAM satisfies that
\[
\sum_{t=0}^{T-1} E \left[ \Psi^t \right] \leq \frac{4\psi_0^2}{\eta} + 64\eta^2(T B_j^2 + B_k^2) + 32 \sum_{t=0}^{T-1} E \left[ \Upsilon_t^+ \right] + 32 \sum_{t=0}^{T-1} E \left[ \Upsilon_t^- \right] + \frac{20\eta^2}{(\eta')^2} \sum_{t=0}^{T-1} E \left[ \left\| v^t \right\|_2^2 \right],
\]
where \( \Psi^t := \left\| \alpha^t - \alpha^*(v^t) \right\|_2^2 \).

Proof. We define \( G_{\alpha}^t := G_{3, \alpha}^t - \alpha^t \) and \( \bar{G}_{\alpha}^t := c + \frac{1}{D^2} \sum_{i \in D^+} f(s_i^{t-1}) - \frac{1}{D^*} \sum_{i \in D_+} f(s_i^{t-1}) - \alpha^t \). The update formula of \( \alpha \) and the 1-strong convexity of \( F(W, \cdot) \) implies that
\[
E_t \left[ \left\| \alpha^{t+1} - \alpha^*(W^t) \right\|_2^2 \right]
\]
\[
= E_t \left[ \left\| \Pi \left[ \alpha^t + \eta G_{\alpha}^t \right] - \Pi \left[ \alpha^* \left( W^t \right) + \eta \nabla \alpha F(W^t, \alpha^t) \right] \right\|_2^2 \right]
\]
\[
\leq E_t \left[ \left\| \alpha^t + \eta G_{\alpha}^t - \alpha^* \left( W^t \right) - \eta \nabla \alpha F(W^t, \alpha^t) \right\|_2^2 \right] + (\eta')^2 E_t \left[ \left\| G_{\alpha}^t - \nabla \alpha F(W^t, \alpha^t) \right\|_2^2 \right]
\]
\[
= \left\| \alpha^t + \eta \nabla \alpha F(W^t, \alpha^t) - \alpha^* \left( W^t \right) - \eta \nabla \alpha F(W^t, \alpha^* \left( W^t \right)) \right\|_2^2 + (\eta')^2 E_t \left[ \left\| G_{\alpha}^t - \nabla \alpha F(W^t, \alpha^t) \right\|_2^2 \right]
\]
\[
\leq (1 + \eta / 2) \left\| \alpha^t + \eta \nabla \alpha F(W^t, \alpha^t) - \alpha^* \left( W^t \right) - \eta \nabla \alpha F(W^t, \alpha^* \left( W^t \right)) \right\|_2^2 + (\eta')^2 E_t \left[ \left\| G_{\alpha}^t - \nabla \alpha F(W^t, \alpha^t) \right\|_2^2 \right]
\]
We have
\[
\left\| \bar{G}_{\alpha}^t - \nabla \alpha F(W^t, \alpha^t) \right\|_2^2
\]
\[
= \left\| \frac{1}{D^2} \sum_{i \in D^+} f_2(s_i^{t-1}) - \frac{1}{D^*} \sum_{i \in D_+} f_2(s_i^{t-1}) - \frac{1}{D^2} \sum_{i \in D^+} f_2(f_1(w^t; \chi)) + \frac{1}{D^*} \sum_{i \in D_+} f_2(f_1(w^t; \chi)) \right\|_2^2
\]
\[
\leq \frac{2}{D^2} \sum_{i \in D^+} \left\| s_i^{t-1} - f_1(w^t; \chi) \right\|_2^2 + \frac{2}{D^*} \sum_{i \in D_+} \left\| s_i^{t-1} - f_1(w^t; \chi) \right\|_2^2
\]
\[
\leq \frac{4}{D^2} \sum_{i \in D^+} \left\| s_i^{t-1} - f_1(w^t; \chi) \right\|_2^2 + \frac{4}{D^*} \sum_{i \in D_+} \left\| s_i^{t-1} - f_1(w^t; \chi) \right\|_2^2 + \frac{4}{D^2} \sum_{i \in D^+} \left\| s_i^{t-1} - s_i^{t-1} \right\|_2^2 + \frac{4}{D^*} \sum_{i \in D_+} \left\| s_i^{t-1} - s_i^{t-1} \right\|_2^2
\]
= \frac{4}{D^2} \sum_{i \in D^+} \left\| s_i^{t-1} - f_1(w^t; \chi) \right\|_2^2 + \frac{4}{D^*} \sum_{i \in D_+} \left\| s_i^{t-1} - f_1(w^t; \chi) \right\|_2^2
\]
\[
+ \frac{4}{D^2} \sum_{i \in S_i^{t-1}} \left\| s_i^{t-1} - s_i^{t-1} \right\|_2^2 + \frac{4}{D^*} \sum_{i \in S_i^{t-1}} \left\| s_i^{t-1} - s_i^{t-1} \right\|_2^2,
\]
where the last step is due to \( s_i^{t-1} = s_i^{t-1} \) for those \( i \notin S_i^{t-1} \cup S_i^{t-1} \). Besides, we have
\[
\left\| G_{\alpha}^t - \nabla \alpha F(W^t, \alpha^t) \right\|_2^2
\]
\[
= \left\| \frac{1}{S^t} \sum_{i \in S^t} f_2(s_i^{t-1}) - \frac{1}{S^+} \sum_{i \in S^+} f_2(s_i^{t-1}) - \frac{1}{D} \sum_{i \in D^+} h(w^t; \chi) - \frac{1}{D^*} \sum_{i \in D_+} h(w^t; \chi) \right\|_2^2 \leq 8(B_j^2 + B_k^2).
\]
Due to the 1-strong convexity of \( F(W, \cdot) \), we have
\[
E \left[ \left\| \alpha^t + \eta \nabla \alpha F(W^t, \alpha^t) - \alpha^* \left( W^t \right) - \eta \nabla \alpha F(W^t, \alpha^* \left( W^t \right)) \right\|_2^2 \right] \leq (1 - \eta') E \left[ \left\| \alpha^t - \alpha^* \left( W^t \right) \right\|_2^2 \right].
Note that $\alpha^*(\cdot)$ is 1-Lipschitz (Lemma 6) such that
\[
\mathbb{E}\left[ \left\| \alpha^{t+1} - \alpha^*(W^{t+1}) \right\|^2 \right] \\
\leq (1 + \eta' / 4) \mathbb{E}\left[ \left\| \alpha^{t+1} - \alpha^*(W^t) \right\|^2 \right] + (1 + 4 / \eta') \mathbb{E}\left[ \left\| \alpha^*(W^t) - \alpha^*(W^{t+1}) \right\|^2 \right] \\
\leq (1 - \eta' / 4) \mathbb{E}\left[ \left\| \alpha^t - \alpha^*(W^t) \right\|^2 \right] + 16(\eta')^2 (B^2_{\beta} + B^2_{\alpha}) + 8\eta' (\mathbb{E}[\Psi^t] + \mathbb{E}[\Upsilon^t]) + \frac{5\eta^2}{\eta'} \mathbb{E}\left[ \left\| \alpha^t \right\|^2 \right] \\
+ \frac{8\eta'}{D_+} \mathbb{E}\left[ \sum_{i \in S^t_+} \left\| s^t_i - s^{t-1}_i \right\|^2 \right] + \frac{8\eta'}{D_-} \mathbb{E}\left[ \sum_{i \in S^t_-} \left\| s^t_i - s^{t-1}_i \right\|^2 \right].
\]

Define $\Psi^t := \left\| \alpha^t - \alpha^*(\Psi^t) \right\|^2$. Then, we have
\[
\sum_{t=0}^{T-1} \mathbb{E}\left[ \Psi^t \right] \leq \frac{4\Psi^0}{\eta'} + 64\eta' T (B^2_{\beta} + B^2_{\alpha}) + 32 \sum_{t=0}^{T-1} \mathbb{E}\left[ \Psi^t \right] + 32 \sum_{t=0}^{T-1} \mathbb{E}\left[ \Upsilon^t \right] + \frac{20\eta^2}{(\eta')^2} \sum_{t=0}^{T-1} \mathbb{E}\left[ \left\| \alpha^t \right\|^2 \right] \\
+ \frac{32}{D_+} \sum_{t=0}^{T-1} \mathbb{E}\left[ \sum_{i \in S^t_+} \left\| s^t_i - s^{t-1}_i \right\|^2 \right] + \frac{32}{D_-} \mathbb{E}\left[ \sum_{i \in S^t_-} \left\| s^t_i - s^{t-1}_i \right\|^2 \right].
\]

**B. Proof of Theorem 1**

According to Lemma 6, we have
\[
\Phi(W^{t+1}) - \Phi(W^t) = \left\langle \nabla \Phi(W^t), W^{t+1} - W^t \right\rangle + \frac{L_{\Phi}}{2} \left\| W^{t+1} - W^t \right\|_2^2 = -\left\langle \nabla \Phi(W^t), \eta \Psi^t \right\rangle + \frac{L_{\Phi}}{2} \eta \left\| \Psi^t \right\|_2^2 \\
\leq \frac{\eta}{2} \left\| \Psi^t \right\|_2^2 - \frac{\eta}{2} \left\| \nabla \Phi(W^t) \right\|_2^2 - \frac{\eta(1 - \eta L_{\Phi})}{2} \left\| \Psi^t \right\|_2^2.
\]

If $\eta \leq \frac{1}{2 L_{\Phi}}$, Lemma 8 implies that
\[
\sum_{t=0}^{T-1} \mathbb{E}\left[ \left\| \nabla \Phi(W^t) \right\|_2^2 \right] \leq \frac{2(\Phi(W^0) - \inf \Phi)}{\eta} + \sum_{t=0}^{T-1} \mathbb{E}\left[ \Delta^t \right] - \frac{1}{2} \sum_{t=0}^{T-1} \mathbb{E}\left[ \left\| \Psi^t \right\|_2^2 \right] \\
\leq \frac{2(\Phi(W^0) - \inf \Phi)}{\eta} + \frac{\Delta^0}{\beta_1} + 2T \beta_1 C_2 + 5L_{\Phi}^2 \sum_{t=0}^{T-1} \mathbb{E}\left[ \Psi^{t+1} \right] + 5C_T \sum_{t=0}^{T-1} \mathbb{E}\left[ \Upsilon^t \right] + 5C_T \sum_{t=0}^{T-1} \mathbb{E}\left[ \Upsilon^{t+1} \right] \\
- \left( \frac{1}{2} + \frac{3\eta^2 L_{\Phi}^2}{\beta_1^2} \right) \sum_{t=0}^{T-1} \mathbb{E}\left[ \left\| \Psi^t \right\|_2^2 \right] + 5C_T \sum_{t=0}^{T-1} \frac{1}{D_+} \mathbb{E}\left[ \sum_{i \in S^t_+} \left\| s_i^{t+1} - s^{t}_i \right\|^2 \right] + 5C_T \sum_{t=0}^{T-1} \frac{1}{D_-} \mathbb{E}\left[ \sum_{i \in S^t_-} \left\| s_i^{t+1} - s^{t}_i \right\|^2 \right].
\]
Apply Lemma 9 and Lemma 10.

\[
\sum_{t=0}^{T-1} \mathbb{E}\left[ \left\| \nabla \Phi(W^t) \right\|_2^2 \right] \\
\leq \frac{2(\Phi(W^0) - \inf \Phi)}{\eta} + \frac{\Delta^0}{\beta_1} + \frac{20L_P^2}{\eta'} + \frac{20(C_T + 32L_P^2)D_+}{\gamma_0S_+} + \frac{20(C_T + 32L_P^2)D_-}{\gamma_0S_-} \\
+ 80T \gamma_0 (C_T + 32L_P^2) B_j^2 (N - B) \\
+ \frac{5(C_T + 32L_P^2)}{D_+} \mathbb{E}\left[ \sum_{i \in S_t^+} \left\| s_i^t - s_i^0 \right\|_2^2 \right] + \frac{5(C_T + 32L_P^2)}{D_-} \mathbb{E}\left[ \sum_{i \in S_t^-} \left\| s_i^t - s_i^0 \right\|_2^2 \right] \\
- \left( \frac{1}{2} - \frac{3\eta^2 L_G^2}{\beta_1^2} - \frac{100L_P^2\eta^2}{(\eta')^2} - \frac{100\eta^2(C_T + 32L_P^2)C_f^2 D_2^2}{\gamma_0^2 S_+^2} - \frac{100\eta^2(C_T + 32L_P^2)C_f^2 D_2^2}{\gamma_0^2 S_-^2} \right) \sum_{t=0}^{T-1} \mathbb{E}\left[ \left\| v^t \right\|_2^2 \right].
\]

Due to the update formula of \( v^t \), we have \( \left\| v^t \right\|_2 \leq C_v \) for all \( t \geq 0 \), \( C_v := 2C_f C_f (2B_f + B_a + B_b) + 2B_1 C_f C_f + 2(2B_f + B_a + B_b) \). We choose \( s_i^0 = 0 \) for all \( i \in D_+ \cup D_- \) and the step sizes as follows

\[
\beta_1 \leq \frac{\epsilon^2}{22C_G}, \quad \eta' \leq \frac{\epsilon^2}{3520L_P^2(B_f^2 + B_h^2)}, \quad \gamma_0 \leq \frac{\epsilon^2 B(N - 1)}{880(C_T + 32L_P^2)B_f^2 (N - B)}.
\]

After \( T = \max \left\{ \frac{22(\Phi(W^0) - \inf \Phi)}{8 C_G \eta^2}, \frac{11\Delta^0}{\beta_1 \eta^2}, \frac{220L_P^2}{\eta' \epsilon^2}, \frac{220(C_T + 32L_P^2)D_+}{\gamma_0 S_+ \epsilon^2}, \frac{220(C_T + 32L_P^2)D_-}{\gamma_0 S_- \epsilon^2} \right\} \) iterations, we have

\[
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[ \left\| \Phi(W^t) \right\|_2^2 \right] + \frac{1}{4T} \sum_{t=0}^{T-1} \mathbb{E}\left[ \left\| v^t \right\|_2^2 \right] \leq \epsilon^2.
\]

According to Lemma 9, we have \( \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[ Y^t_+ \right] = \mathcal{O}(\epsilon^2), \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[ Y^t_- \right] = \mathcal{O}(\epsilon^2) \) with \( \gamma_0 = \mathcal{O}(\epsilon^2) \). Therefore, \( T = \mathcal{O}\left( \max\left( \frac{D_+}{S_+}, \frac{D_-}{S_-} \right) \epsilon^{-1} \right) \).

C. More Figures
Figure 4. Training and testing convergence of MIDAM (XX) vs DAM (XX). The top is for training AUC, and the bottom is for testing AUC.

Figure 5. Training and testing convergence of MIDAM (XX) vs DAM (MB-XX). The margin $c = 0.1$ and learning rate is tuned in $\{1e-1,1e-2,1e-3\}$. The top is for training AUC, and the bottom is for testing AUC.
Figure 6. Training with different instance-batch sizes

Figure 7. Ablation study for fixing the total budget per-iteration by varying bag-batch size $S_+ = S_- = \tilde{B}$ and instance-batch size $B$ for the proposed MIDAM approaches
Figure 8. Demonstrations for positive and negative examples for Breast Cancer dataset. Left: original image. Middle: prediction scores for each patch. Right: attention weights for each patch.