Learning Algorithms for Multiple Instance Regression

Aaryan Gupta[1] **Rishi Saket**[1]

[1]Google DeepMind , {aaryangupta,rishisaket}@google.com

Abstract

Multiple instance regression, introduced by Ray and Page [2001], is a generalisation of supervised regression in which the training data is available as a bag of feature-vectors (instances) and for each bag there is a bag-label which matches the label of one (unknown) primary instance from that bag. The goal is to compute a hypothesis regressor consistent with the underlying instance-labels. While most works on MIR focused on training models on such training data, computational learnability of MIR was only recently explored by Chauhan et al. [UAI 2024] who showed worst case intractability of properly learning *linear regressors* in MIR by showing a inapproximability bound. However, their work did not rule out efficient algorithms for this problem on natural distributions and randomly chosen labels. In this work we show that it is indeed possible to efficiently learn linear regressors in MIR when given access to random bags of uniformly randomly sampled primary instance chosen as the bag-label in which the feature vectors are independently sampled from Gaussian distributions. This is achieved by optimizing a certain bag-level loss which, via concentration bounds, yields a close approximation to the target linear regressor. Lastly, we show that the bag-level loss is also useful for learning general concepts (e.g. neural networks) in this setting: an optimizer of the loss on sampled bags is, w.h.p. a close approximation of a scaled version of the target function. We include experimental evaluations of our learning algorithms on synthetic and real-world datasets showing that our method outperforms the baseline MIR methods.

1 INTRODUCTION

In probably approximately correct (PAC) model of learning [Valiant, 1984], we are given distribution $\mathcal{D}$ over feature-vectors and label pairs $(\mathbf{x}, y)$ which are consistent with some unknown function f from a concept class of functions i.e., $y = f(\mathbf{x})$. The goal is to sample iid examples from $\mathcal{D}$ and efficiently compute a hypothesis h which approximates the target function. However, in many applications the labels of individual feature-vectors may not be available due lack of instrumentation, uncertainty in the data or privacy constraints. Instead, we are only given *bag-labels* for *bags* i.e., a subsets of feature-vectors. These bag-labels are derived from the labels of the feature-vectors via some aggregation function. The goal remains the same, to find a hypothesis which accurately predicts the feature-vector labels.

When the aggregation function is sum (equivalently avg, since bag-sizes are known) the setting is known as *learning from label proportions* (LLP) while the $\{0, 1\}$-label setting with OR aggregation function is called *multiple instance learning* (MIL). Previous works have studied the computational and statistical learning aspects of LLP [Yu et al., 2014, Brahmbhatt et al., 2023] as well as MIL [Blum and Kalai, 1998].

Our focus in this work is *multiple instance regression* (MIR) [Ray and Page, 2001] in which the labels are real-valued, obtained by choosing the label of some (undisclosed) feature-vector in the bag, and the goal is to find a regressor with low error w.r.t. the underlying feature-vector labels. Recent work of Chauhan et al. [2024], to the best of our knowledge, is the first to study MIR from the statistical and computational perspective. Chauhan et al. [2024] considered the case of fixed-sized MIR bags each consisting of iid sampled feature-vectors with the bag-label being the label of a uniformly sampled feature-vector from the bag, and showed the first bag-to-instance generalization error bounds. More specifically, they showed that a regressor with a low value of a certain bag-attribution loss (which they define as the minimum distance between the bag-label and the prediction on any

of the bag's feature-vectors) on sampled bags also has low regression loss over the feature-vector distribution. Their work also showed the NP-hardness of even approximately optimizing a linear regressor on arbitrary bag distributions. We note however that the specific bag-attribution loss used by Chauhan et al. [2024] in their generalization error bound is non-convex in the regressor predictions and thus is not practical to optimize efficiently. This state of affairs indicates a lack of algorithmic results for learning in MIR with provable guarantees under reasonable distributional assumptions.

Our Contributions. Our results substantially bridge the gaps in our understanding of MIR.

Specifically, for the random MIR bags considered by Chauhan et al. [2024] as described above, with feature-vectors being Gaussian, we provide an efficient learning algorithm for the realizable setting that can recover the unknown regressor when the latter is a linear function f.

Our results – stated as Theorem 1.1 in Section 1.2 – is the first efficient PAC learning algorithm for MIR, even for learning linear regressors.

The key idea is to use the bag-level loss on MIR bags which for each bag in the sample, assigns its bag-label to all feature-vectors in the bag, and then optimizes the squared-Euclidean bag-level loss on the resultant labeled feature-vectors. This is convex in the regressor predictions and thus over the weights of the linear regressor. We show that in the linear regression case, optimizing this loss yields, using concentration bounds w.h.p. over the sampled bags, an arbitrarily close approximation to a linearly transformed version of the target regressor, where the linear transformation is invertible and can be explicitly computed (more details are in Section 1.3).

While the above results clarify the learnability of linear regressors in MIR, practical applications often require neural regression, and one would wish to extend the above results to general regressors like neural networks. Unfortunately, since neural networks are not necessarily convex in their weights, our approach of optimizing a bag-level loss does not yield an efficient algorithm for general regressor classes which contain neural networks. Setting aside this issue, we do however prove (stated formally as Theorem 1.2 in Sec. 1.2) that any regressor which does optimize the bag-level loss must be a uniformly scaled and translated version of the target regressor. The scaling and translation factors can be estimated efficiently, allowing us to learn the original regressor.

It is pertinent to note that the bag-level loss that we optimize in our results is essentially same as that in the Instance-MIR method [Wang et al., 2008] where the bag-label is assigned to the feature-vectors in the bag and the resultant labeled set of feature-vectors is used for optimizing a regression loss, in our case we use the squared-Euclidean loss. Thus, our results theoretically justify the efficacy of Instance-MIR which has been observed in practice (see [Wang et al., 2008]). However, our algorithms also involve a linear transformation step which makes them distinct from vanilla Instance-MIR.

Our experimental evaluations compare our algorithms for different scenarios to previous baselines such as Instance-MIR, and demonstrate the practical applicability and improved performance of our methods.

Previous Related Work. Multiple instance learning (MIL), specifically its classification setting, was proposed by Dietterich et al. [1997] to model drug activity detection where the bag-label is an OR of its (unknown) instance-labels (all labels are $\{0, 1\}$-valued), with the goal being to train an instance-label classifier. MIL has subsequently been used in various other applications such as medical image [Wu et al., 2015] and videos [Sikka et al., 2013] analysis, time series prediction [Maron, 1998], and information retrieval [Lozano-Pérez and Yang, 2000].

In MIR i.e., multiple instance regression, introduced by Ray and Page [2001], the underlying task is regression over the real-valued labels. For each bag, the label of a *primary* instance from it is its bag-label. The earliest applications of MIR formulations have been in remote sensing such as aerosol optical depth prediction [Wang et al., 2008] and crop yield prediction [Wagstaff and Lane, 2007]. More recently, for applications like assessing image quality depending on that of a constituent prime image, Liang et al. [2021] modeled the problem as MIR to develop model training methods. Another image analysis task of facial age estimation has also been studied in the work of Liu et al. [2019] using MIR techniques while MIR has also recently been used to model the continuous response of bags of neoantigens [Park et al., 2020]. Other applications of MIR are possible in user modeling for online advertising, where due to privacy considerations, an online purchase or conversion event cannot be linked to a unique user clicks, rather we have a subset or bag of clicks which could have resulted in the conversion (see Section 2.1 of [O'Brien et al., 2022]).

Loss based methods which transform the problem into instance-level regression include Aggregated-MIR which assigns the average feature-vectors in each bag the bag-label, and Instance-MIR in which the bag-label is assigned to each instance in a bag (see Wang et al. [2008]). More sophisticated EM based methods are primary-MIR (PIR) [Ray and Page, 2001], pruning MIR [Wang et al., 2008] and mixture-model MIR Wang et al. [2012], while Wagstaff et al. [2008], Trabelsi and Frigui [2018] proposed clustering based methods for MIR. However, the work of Chauhan et al. [2024] is (to the best of our knowledge) the first that investigated in detail the learning theoretic aspects of MIR, showing (i) error bounds for generalizing regressors trained on randomly sampled bags with iid feature-vectors to the underlying

feature-vector distribution, and (ii) the NP-hardness of even approximately optimizing a linear regressor on arbitrary bag distributions. Additionally Chauhan et al. [2024] provided an optimization based model training approach for the MIR problem, albeit without any performance guarantees.

1.1 PROBLEM DEFINITION

A *bag* is a finite subset of feature-vectors. Specifically, if $\boldsymbol{\mathcal{X}}$ is the universe of possible feature-vectors, then a q-sized bag B is a subset of $\boldsymbol{\mathcal{X}}$ s.t. $|B| = q$, for $q \in \mathbb{Z}^+$. In this work, $\boldsymbol{\mathcal{X}} = \mathbb{R}^d$ for some $d \in \mathbb{Z}^+$. A labeling function $f : \boldsymbol{\mathcal{X}} \to \mathbb{R}$ defines the labels of the feature-vectors. We will use $y_B \in \mathbb{R}$ to denote the *bag-label*, which in the MIR setting is an element of $\{f(\mathbf{x})\}_{\mathbf{x}\in B}$. Next we define the random bag distribution (also studied by Chauhan et al. [2024]).

Bag Distribution. Given a distribution $\mathcal{D}$ over $\mathbb{R}^d$ for some $d \in \mathbb{Z}^+$, a target concept $f : \mathbb{R}^d \to \mathbb{R}$, and a bag-size $q \in \mathbb{Z}^+$, the bag distribution $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$ is defined by the following sampling procedure: generate a labeled bag (B, y_B) where $B = \{\mathbf{x}_j\}_{j=1}^q$ such that $\mathbf{x}_j$ is independently sampled from $\mathcal{D}$ for $j \in [q]$, and y_B is chosen uniformly at random from $\{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_q)\}$.

For any two functions $f, h : \mathbb{R}^d \to \mathbb{R}$, we define the ℓ_2^2-error under distribution $\mathcal{D}$ as: $\text{err}_2(\mathcal{D}, f, h) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[(f(\mathbf{x}) - h(\mathbf{x}))^2\right]$.

We will consider *concept classes* of functions mapping $\mathbb{R}^d$ to real-values. In particular, the class of linear regressors Lin over $\mathbb{R}^d$ is given by functions of the form $f(\mathbf{x}) := \mathbf{r}^\mathsf{T}\mathbf{x}$ for some $\mathbf{r} \in \mathbb{R}^d$. Note that we can incorporate a constant term by appending 1 to the feature-vectors and an extra-coordinate to $\mathbf{r}$ and therefore we can use the homogeneous formulation of linear regressors in the rest of the paper.

For a concept class $\mathcal{C}$ of real-valued functions over $\mathbb{R}^d$, and parameters $\varepsilon, \delta > 0$, we define the proper MIR learning problem PAC-MIR $[\mathcal{C}, \mathcal{D}, q, \varepsilon, \delta]$ as follows: for any function $f \in \mathcal{C}$, given access to iid samples (B, y_B) from $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$, with probability $1 - \delta$ over the samples, output a hypothesis $h \in \mathcal{C}$ such that $\text{err}_2(\mathcal{D}, f, h) \le \varepsilon$. We desire that the algorithm for PAC-MIR $[\mathcal{C}, \mathcal{D}, q, \varepsilon, \delta]$ has sample as well as time complexity polynomial in $d, (1/\varepsilon)$, and $\log(1/\delta)$ along with dependence on the parameters of $\mathcal{D}$ and properties of the target regressor f.

In our results stated in the next section, $\mathcal{D}$ is taken to be $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We assume that the second moment matrix $(\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T} + \boldsymbol{\Sigma})$ is of full rank (i.e., invertible) otherwise one can use its pesudo-inverse (see Appendix A) in our analysis.

1.2 OUR RESULTS

The first theorem provides an efficient algorithm for PAC-MIR for linear regressors, for random bags over with Gaussian feature-vectors with the bag-label being a random label in the bag.

Theorem 1.1. *For $d \in \mathbb{Z}^+$, let $\mathcal{D}$ be $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ over $\mathbb{R}^d$, $q \in \mathbb{Z}^+$ be the bag-size, $\varepsilon, \delta > 0$ be parameters. Then, there is an algorithm $\mathcal{A}$ for* PAC-MIR $[\mathsf{Lin}, \mathcal{D}, q, \varepsilon, \delta]$ *which samples*

$$m = O\left(\frac{dq^2\|\mathbf{r}\|_2^2 \log\left(\frac{q}{\delta}\right)(\|\boldsymbol{\mu}\| + 1)(\|\boldsymbol{\mu}\|^2 + \lambda_{\max}(\boldsymbol{\Sigma}))^3}{\lambda_{\min}^2(\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T} + \boldsymbol{\Sigma})\varepsilon}\right)$$

bags and runs in time polynomial in the number of sampled bags, where $f(\mathbf{x}) := \mathbf{r}^\mathsf{T}\mathbf{x}$ is the target concept and $\lambda_{\max}$ and $\lambda_{\min}$ yield the maximum and minimum eigenvalues respectively of the operand matrices.

The above results are the first PAC learning algorithm for non-trivial concept classes in the MIR setting. To illustrate the main technical ideas, in Section 3 we prove Theorem 1.1 for the special case of homogeneous regressors i.e., with no constant term, $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \mathbf{I}$, deferring the proof of the general case to Appendix B. While we also provide an overview of the proof techniques later in this section, the main idea is to leverage the following bag-level loss defined for a bag B and label y_B w.r.t. to a hypothesis h as follows:

$$L_{\text{bag}}(B, y_B, h) := \sum_{\mathbf{x}\in B} (h(\mathbf{x}) - y_B)^2 \tag{1}$$

Clearly, the RHS of the above is convex in the weights of h when $h \in \mathsf{Lin}$, as in Theorem 1.1.

However, this approach of optimizing such losses is not tractable for general functions such as neural-networks since their outputs are not necessarily convex in their weights. Nevertheless, neural networks are widely used in ML applications and our next theorem shows that the formulation in (1) is indeed useful for accurately learning neural networks in the MIR setting.

We consider a concept class $\mathcal{F}$ of regressors (e.g. 2-layer neural-networks) with bounded outputs in $[0, 1]$ which is closed under the following transformation: for any $f \in \mathcal{F}$, $f_b = bf + (1 - b)\mathbb{E}[f] \in \mathcal{F}$ for any $b \in [0, 1]$. It can be to seen that value of f_b at any point is in $[0, 1]$, and common neural network models are closed under this transformation (see Appendix C).

Theorem 1.2. *Let $f \in \mathcal{F}$ be any target regressor. Then, for any $q \in \mathbb{Z}^+$ and $\varepsilon, \delta > 0$, if $\mathcal{B}$ is a collection of m bags sampled independently and u.a.r. from $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$, then $h := \text{argmin}_{h'\in\mathcal{F}} \sum_{(B,y_B)\in\mathcal{B}} L_{\text{bag}}(B, y_B, h')$ satisfies $\text{err}_2(\mathcal{D}, f, qh + K) \le \varepsilon$ with probability $(1 - \delta)$, when $m \ge O\left(\frac{rq^2}{\varepsilon^2}\left(\log\left(\frac{rq}{\varepsilon\delta}\right)\right)\right)$, where $r = \mathsf{Pdim}(\mathcal{F})$ is the pseudo-dimension (see Sec. 2.1) of $\mathcal{F}$. Further, K can be efficiently estimated to arbitrary accuracy.*

In effect, the above theorem, proved in Section 4, shows that optimizing the loss in (1) over a large enough sampled set of MIR bags recovers a scaled version of the target concept.

Discussion of Our Results. We would like to note that in [Chauhan et al., 2024] and as well as in our work, the bag distribution is such that each feature-vector in a q-sized bag is chosen iid from the distribution $\mathcal{D}$. The bag-label is the label of a randomly chosen feature-vector in the bag. Such bag distributions occur especially in privacy constrained settings, such as user modeling for online advertising where due to privacy considerations an online purchase or conversion event cannot be linked to a unique user click, rather we have a subset or bag of clicks which could have resulted in the conversion (see Section 2.1 of O'Brien et al. [2022]). Random bags afford more privacy as compared to bags in which feature-vectors are correlated which could induce dependencies between the bag-label and the labels of several feature-vectors within the bag, thus compromising the privacy guarantee. Given the relevance to such revenue critical applications, we believe our algorithmic contributions can have real-world impact. Further, since random bags do not provide any additional information via correlations, from an algorithmic perspective they typically represent a reasonably challenging scenario, and any progress on developing learning techniques on such bags can yield insights which may be generally applicable.

Theorem 1.1 in our work considers Gaussian feature-vectors, which is fairly standard in ML for modeling data to validate algorithmic techniques (see for e.g. Dasgupta [1999], Vempala [2010]). Further, the Gaussianity assumption is only used for estimation bounds to obtain efficient sample complexity, and any sub-Gaussian distribution can also be used to derive similar guarantees. In Theorem 1.2, we extend this to neural regression, in which however the bag-loss function is not convex due to the general non-convexity of neural network outputs in their weights. Instead, we develop pseudo-dimension and covering number based arguments which absorb any distributional assumptions on the feature-vectors. As a result, Theorem 1.2, while relying on black-box optimization of the bag-loss (which is often feasible in practice) is more broadly applicable than Theorem 1.1 which provides a self-contained efficient algorithm. One can also observe that the matrix factor scaling $\hat{\mathbf{v}}_{\text{min}}$ in step 3 of Algorithm 1 for the linear $N(\mathbf{0}, \mathbf{I})$ case of Theorem Theorem 1.1 converges to $q\mathbf{I}$, which corresponds to scaling by factor q obtained in Theorem 1.2. This correspondence is due to the underlying commonality of the main ideas in both theorems.

1.3 OUR TECHNIQUES

In this section we informally describe the techniques used in proving our main results.

Theorem 1.1.

For ease of exposition we shall consider the special case of homogeneous linear regressors $f(\mathbf{x}) = \mathbf{r}^{\mathsf{T}}\mathbf{x}$ in d-dimensional space and $N(\mathbf{0}, \mathbf{I})$ as the feature-vector distribution $\mathcal{D}$. The algorithm is as follows: sample a m-sized collection of iid bags $\mathcal{B}$ from $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$ and minimize the sample loss which is the sum of $L_{\text{bag}}(B, y_B, h)$ over all bags in $\mathcal{B}$, w.r.t. the hypothesis $h(\mathbf{x}) := \mathbf{v}^T\mathbf{x}$. The loss is convex and can be minimized in poly(m, d)-time, and its gradient can be written using sample-dependent matrices (i.e., depending on the sampled bags) as a linear form in $\mathbf{r}$ and $\mathbf{v}$. It can be seen that the loss is minimized at $\mathbf{v}_{\text{min}} = \mathbf{HJr}$, where $\mathbf{H}$ is a matrix that can be derived from the feature-vectors in the sampled bags while $\mathbf{J}$ is a matrix which also depends on the choice of each bag's feature-vector labels chosen to be the bag-label. Crucially however, one can show that $\mathbf{J}$ converges to the identity matrix with the sample size, and therefore one can take $\mathbf{H}^{-1}\mathbf{v}_{\text{min}}$ as the approximate solution. The analysis uses the fact that the sample-dependent matrices are sums of outer products of Gaussian vectors for which the subgaussian concentration inequalities bound the deviation from mean. The general case of non-homogeneous linear regressors and $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be handled similarly, except that matrix factor also depends on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and can be estimated from the sampled bags.

Theorem 1.2. Using algebraic manipulations of the loss expression, we first show that expected loss $L_{\text{bag}}(B, y_B, h')$ over a random bag B from $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$ is greater than the same loss for $\hat{f} := f/q + (1 - 1/q)\mathbb{E}[f]$ by exactly $\text{err}_2(\mathcal{D}, \hat{f}, h')$, for any regressor h. In particular, the expected loss $\mathbb{E}_{B\in\mathcal{D}_{\text{bag}}}[L_{\text{bag}}(B, y_B, h')]$ is minimized by $\hat{f}$. Further, by our assumption on $\mathcal{F}$, $f \in \mathcal{F} \Rightarrow h \in \mathcal{F}$. Applying the generalization error bound on each of the q loss terms in $L_{\text{bag}}(B, y_B, h')$ we obtain generalization error between L_{bag} averaged over sampled bags $\mathcal{B}$ and $\mathbb{E}_{B\in\mathcal{D}_{\text{bag}}}[L_{\text{bag}}(B, y_B, h')]$. Using these bounds for $\hat{f} \in \mathcal{F}$ as well as for the optimizer h of L_{bag} averaged over sampled bags, we obtain that $\mathbb{E}_{B\in\mathcal{D}_{\text{bag}}}[L_{\text{bag}}(B, y_B, h)] \leq \mathbb{E}_{B\in\mathcal{D}_{\text{bag}}}[L_{\text{bag}}(B, y_B, h')] + \varepsilon$. Our previous argment then implies that $\text{err}_2(\mathcal{D}, \hat{f}, h) \leq \varepsilon$. Using $qh - (q-1)K'$ as the hypothesis yields the desired error bound, where K' is an accurate estimate of $\mathbb{E}[f]$ which can be efficiently computed by sampling additional bags.

2 PRELIMINARIES

For $\mathbf{x} \in \mathbb{R}^d$, let $\|\mathbf{x}\|_2$ denote the Euclidean norm. For $\mathbf{A} \in \mathbb{R}^{d\times d}$, let the operator norm of $\mathbf{A}$ be denoted by $\|\mathbf{A}\| = \max_{\|\mathbf{x}\|_2=1}\|\mathbf{Ax}\|_2$. We present the following theorem from Chapter 6 of Wainwright [2019]:

Theorem 2.1. *Consider $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$ in $\mathbb{R}^d$ iid from $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then, for any $\zeta > 0$, we have with probability*

$1 - 2e^{-m\zeta^2/2}$,

$$\left\| \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{X}_i \mathbf{X}_i^T - \mathbb{E}[\mathbf{X}_i \mathbf{X}_i^T] \right) \right\| \leq \left(2\sqrt{\frac{d}{m}} + 2\zeta + \left(\sqrt{\frac{d}{m}} + \zeta \right)^2 \right) \cdot \left(\|\boldsymbol{\mu}\|_2^2 + \lambda_{\max}(\boldsymbol{\Sigma}) \right) \quad (2)$$

2.1 REAL FUNCTIONS FROM A CLASS

For a class $\mathcal{F}$ of real-valued functions (regressors) over $\boldsymbol{\mathcal{X}}$ with values in $[0,1]$, and any $\boldsymbol{\mathcal{X}}' \subseteq \boldsymbol{\mathcal{X}}$ s.t. $|\boldsymbol{\mathcal{X}}'| = N$, let $\mathcal{C}_p(\xi, \mathcal{F}, \boldsymbol{\mathcal{X}}')$ denote a minimum cardinality subset (cover) of $\mathcal{F}$ such that for each $f^* \in \mathcal{F}$, there exists $f \in \mathcal{C}_p(\xi, \mathcal{F}, \boldsymbol{\mathcal{X}}')$ s.t. $\left(\mathbb{E}_{\mathbf{x} \in \boldsymbol{\mathcal{X}}'} \left[|f^*(\mathbf{x}) - f(\mathbf{x})|^p \right]\right)^{1/p} \leq \xi$ for $p \in [1, \infty)$, and $\max_{\mathbf{x} \in \boldsymbol{\mathcal{X}}'} |f^*(\mathbf{x}) - f(\mathbf{x})| \leq \xi$ for $p = \infty$.

The maximum size of $\mathcal{C}_p(\xi, \mathcal{F}, \boldsymbol{\mathcal{X}}')$ over all choices of $|\boldsymbol{\mathcal{X}}'| = N$ is defined to be $N_p(\xi, \mathcal{F}, N)$. We refer the reader to Sections 10.2-10.4 of Anthony and Bartlett [2009] for more details.

The *pseudo-dimension* of $\mathcal{F}$, $\mathsf{Pdim}(\mathcal{F})$ is a measure of the complexity of $\mathcal{F}$. As described in Sec. 10.4 and 12.3 of Anthony and Bartlett [2009] the pseudo dimension can be used to bound the size of covers of $\mathcal{F}$ as follows:

$$N_1(\xi, \mathcal{F}, N) \leq N_\infty(\xi, \mathcal{F}, N) \leq (eN/\xi p)^p \quad (3)$$

where $p = \mathsf{Pdim}(\mathcal{F})$ and $N \geq p$.

3 LINEAR REGRESSORS OVER $N(\mathbf{0}, \mathbf{I})$

Algorithm 1: PAC Learner for $f(\mathbf{x}) := \mathbf{r}^\mathsf{T}\mathbf{x}$ over $N(\mathbf{0}, \mathbf{I})$

Input: $\mathcal{D}_{\mathsf{bag}}(\mathcal{D} = N(\mathbf{0}, \mathbf{I}), f = \mathsf{Lin}, q), m, q$, where $f(\mathbf{x}) := \mathbf{r}^\mathsf{T}\mathbf{x}$.

1. Sample a collection $\mathcal{B}$ of m iid bags from $\mathcal{D}_{\mathsf{bag}}(\mathcal{D}, f, q)$.
2. Define $\hat{L}(\mathcal{B}, \mathbf{v}) = \frac{1}{m} \sum_{B \in \mathcal{B}} \sum_{\mathbf{x} \in B} (y_B - \mathbf{v}^\mathsf{T}\mathbf{x})^2$, use convex optimisation to find $\hat{\mathbf{v}}_{\min} = \operatorname{argmin}_{\mathbf{v}} \hat{L}(\mathcal{B}, \mathbf{v})$.
3. Output $\hat{\mathbf{r}} = \left(\frac{1}{m} \sum_{B \in \mathcal{B}} \sum_{\mathbf{x} \in B} \mathbf{x}\mathbf{x}^\mathsf{T} \right) \hat{\mathbf{v}}_{\min}$.

For the setting of homogeneous linear regressors over $N(\mathbf{0}, \mathbf{I})$, we provide Algorithm 1. Note that in Step 2 of Algorithm 1, $\hat{L}(\mathcal{B}, \mathbf{v}) = \sum_{B \in \mathcal{B}} L_{\mathsf{bag}}(B, y_B, h)$ where $h(\mathbf{x}) := \mathbf{v}^\mathsf{T}\mathbf{x}$.

Lemma 3.1. *For any $\varepsilon, \delta \in (0,1)$, if $m \geq O\left(dq^2 \log\left(\frac{q}{\delta}\right) \|\mathbf{r}\|_2^2 / \varepsilon\right)$, then $\hat{\mathbf{r}}$ returned in Algorithm 1 satisfies $\|\hat{\mathbf{r}} - \mathbf{r}\|_2 \leq \sqrt{\varepsilon}$ with probability $1 - \delta$.*

We defer the proof of lemma 3.1 to the next subsection.

Lemma 3.2. *Let $\varepsilon, \delta \in (0,1)$ and suppose that $\hat{\mathbf{r}}$ returned in Algorithm 1 satisfies $\|\hat{\mathbf{r}} - \mathbf{r}\|_2 \leq \sqrt{\varepsilon}$, then $h(\mathbf{x}) = \hat{\mathbf{r}}^\mathsf{T}\mathbf{x}$ satisfies $\mathrm{err}_2(\mathcal{D}, f, h) \leq \varepsilon$ with probability $1 - \delta$.*

Proof. (of Lemma 3.2) $\mathrm{err}_2(\mathcal{D}, f, h) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}\left[(f(\mathbf{x}) - h(\mathbf{x}))^2\right] = \mathbb{E}_{\mathbf{x} \sim N(\mathbf{0}, \mathbf{I})}[((\mathbf{r} - \hat{\mathbf{r}})^\mathsf{T}\mathbf{x})^2] = \mathrm{Var}[(\mathbf{r} - \hat{\mathbf{r}})^\mathsf{T}\mathbf{x}] + \mathbb{E}[(\mathbf{r} - \hat{\mathbf{r}})^\mathsf{T}\mathbf{x}]^2$. Now, note that $(\mathbf{r} - \hat{\mathbf{r}})^\mathsf{T}\mathbf{x} \sim N(\mathbf{0}, \|\mathbf{r} - \hat{\mathbf{r}}\|_2^2)$. So we get $\mathrm{err}_2(\mathcal{D}, f, h) = \|\mathbf{r} - \hat{\mathbf{r}}\|_2^2 \leq \varepsilon$. □

Since $\|f\|_2 \geq \|\mathbf{r}\|_2$, $q \geq \log q$, for $m \geq O\left(dq^2 \log\left(\frac{q}{\delta}\right) \|f\|_2^2 / \varepsilon\right)$, we show that Algorithm 1 outputs h such that $\mathrm{err}_2(\mathcal{D}, f, h) \leq \varepsilon$. The convex optimisation subroutine called inside Algorithm 1 is $\mathrm{poly}[d, q, (1/\varepsilon), \log(1/\delta)]$, which makes Algorithm 1 polynomial in $\mathrm{poly}[d, q, (1/\varepsilon), \log(1/\delta), \|f\|_2]$. This completes the proof of Theorem 1.1 for the setting of homogeneous linear regressors and $\mathcal{D} = N(\mathbf{0}, \mathbf{I})$.

3.1 PROOF OF LEMMA 3.1

Taking $B = \{\mathbf{x}_{B1}, \ldots, \mathbf{x}_{Bq}\}$ to be a random bag from $\mathcal{D}_{\mathsf{bag}}(\mathcal{D}, f, q)$, one can assume $y_B = f(\mathbf{x}_{B1}) = \mathbf{r}^\mathsf{T}\mathbf{x}_{B1}$ as each feature-vector in B is iid from $N(\mathbf{0}, \mathbf{I})$. Using this:

$$\begin{aligned} \hat{L}(\mathcal{B}, \mathbf{v}) &= \frac{1}{m} \sum_{B = \{\mathbf{x}_i \mid i \in [q]\} \in \mathcal{B}} [(\mathbf{r}^\mathsf{T}\mathbf{x}_{B1} - \mathbf{v}^\mathsf{T}\mathbf{x}_{B1})^2 \\ &\quad + \sum_{j=2}^{q} (\mathbf{r}^\mathsf{T}\mathbf{x}_{B1} - \mathbf{v}^\mathsf{T}\mathbf{x}_{Bj})^2] \\ &= (\mathbf{r} - \mathbf{v})^\mathsf{T}\mathbf{A}(\mathbf{r} - \mathbf{v}) + (q-1)\mathbf{r}^\mathsf{T}\mathbf{A}\mathbf{r} \\ &\quad + \sum_{j=2}^{q} (\mathbf{v}^\mathsf{T}\mathbf{C}_j\mathbf{v} - \mathbf{r}^\mathsf{T}\mathbf{D}_j^\mathsf{T}\mathbf{v} - \mathbf{v}^\mathsf{T}\mathbf{D}_j\mathbf{r}) \quad (4) \end{aligned}$$

where $\mathbf{A} = \frac{1}{m} \sum_{B = \{\mathbf{x}_i \mid i \in [q]\} \in \mathcal{B}} \mathbf{x}_{B1}\mathbf{x}_{B1}^\mathsf{T}$, $\mathbf{C}_j = \frac{1}{m} \sum_{B = \{\mathbf{x}_i \mid i \in [q]\} \in \mathcal{B}} \mathbf{x}_{Bj}\mathbf{x}_{Bj}^\mathsf{T}$, and $\mathbf{D}_j = \frac{1}{m} \sum_{B = \{\mathbf{x}_i \mid i \in [q]\} \in \mathcal{B}} \mathbf{x}_{Bj}\mathbf{x}_{B1}^\mathsf{T}$.

We define $\hat{\mathbf{v}}_{\min} = \operatorname{argmin}_{\mathbf{v}} \hat{L}(\mathcal{B}, \mathbf{v})$ as used in Algorithm 1. $\hat{L}(\mathcal{B}, \mathbf{v})$ is convex in $\mathbf{v}$, hence $\hat{\mathbf{v}}_{\min}$ can be found by solving $\frac{\partial \hat{L}(\mathcal{B}, \mathbf{v})}{\partial \mathbf{v}} = 0$, which yields (see Appendix A.2),

$$0 = \frac{\partial \hat{L}(\hat{\mathbf{v}})}{\partial \mathbf{v}} = 2\mathbf{A}\left(\hat{\mathbf{v}}_{\min} - \mathbf{r}\right) + \sum_{j=2}^{q} \left(2\mathbf{C}_j\hat{\mathbf{v}}_{\min} - 2\mathbf{D}_j\mathbf{r}\right)$$

$$\begin{aligned} \hat{\mathbf{v}}_{\min} &= \left(\mathbf{A} + \sum_{j=2}^{q} \mathbf{C}_j\right)^{-1} \left(\mathbf{A} + \sum_{j=2}^{q} \mathbf{D}_j\right) \mathbf{r} \\ &= \left(\frac{1}{m} \sum_{B \in \mathcal{B}} \sum_{j=1}^{q} \mathbf{x}_{Bj}\mathbf{x}_{Bj}^\mathsf{T}\right)^{-1} \left(\mathbf{A} + \sum_{j=2}^{q} \mathbf{D}_j\right) \mathbf{r} \end{aligned}$$

Note that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{A}] = \mathbb{E}_{x\sim\mathcal{D}}[\mathbf{C}_j] = \mathbf{I}$ and $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{D}_j] = \mathbf{0}$. As defined in Algorithm 1, $\hat{\mathbf{r}}$ is

$$\hat{\mathbf{r}} = \left(\frac{1}{m}\sum_{B\in\mathcal{B}}\sum_{j=1}^{q}\mathbf{x}_{Bj}\mathbf{x}_{Bj}^{\mathsf{T}}\right)\mathbf{v}_{\min} = \left(\mathbf{A} + \sum_{j=2}^{q}\mathbf{D}_j\right)\mathbf{r}. \tag{5}$$

Note that $\mathbb{E}_{\mathcal{B}}[\mathbf{A}] = \mathbb{E}_{\mathcal{B}}[\mathbf{C}_j] = \mathbf{I}$ and $\mathbb{E}_{\mathcal{B}}[\mathbf{D}_j] = \mathbf{0}$, since $\mathbf{x}_{Bj}$ $(B \in \mathcal{B}, j \in [q])$ are iid $N(\mathbf{0}, \mathbf{I})$. Thus, we have

$$\begin{aligned}\|\hat{\mathbf{r}} - \mathbf{r}\| &\leq \left\|\mathbf{A} - \mathbf{I} + \sum_{j=2}^{q}\mathbf{D}_j\right\| \|\mathbf{r}\| \\ &\leq \|\mathbf{A} - \mathbf{I}\| \|\mathbf{r}\| + \sum_{j=2}^{q}\|\mathbf{D}_j\| \|\mathbf{r}\| \end{aligned} \tag{6}$$

by triangle inequality. As $m \geq O\left(d\log\left(\frac{q}{\delta}\right)\|\mathbf{r}\|_2^2 q^2/\varepsilon\right)$, using using Theorem 2.1 we obtain

$$\Pr\left[\|\mathbf{A} - \mathbf{I}\| \leq \frac{\sqrt{\varepsilon}}{2q\|\mathbf{r}\|}\right] \geq 1 - \frac{\delta}{2q}. \tag{7}$$

Further, since for any fixed $j \in \{2, \ldots, k\}$, $\{(\mathbf{x}_{B1} - \mathbf{x}_{Bj})\}_{B\in\mathcal{B}} \sim N(0, 2\mathbf{I})$ iid , and $\{(\mathbf{x}_{B1} + \mathbf{x}_{Bj})\}_{B\in\mathcal{B}} \sim N(0, 2\mathbf{I})$ iid, we have

$$\begin{aligned}&\Pr\left[\left\|\frac{1}{m}\sum_{B\in\mathcal{B}}(\mathbf{x}_{B1} + \mathbf{x}_{Bj})(\mathbf{x}_{B1} + \mathbf{x}_{Bj})^{\mathsf{T}} - 2\mathbb{I}\right\| \leq \frac{\sqrt{\varepsilon}}{q\|\mathbf{r}\|}\right] \\ &\Pr\left[\left\|\frac{1}{m}\sum_{B\in\mathcal{B}}(\mathbf{x}_{B1} - \mathbf{x}_{Bj})(\mathbf{x}_{B1} - \mathbf{x}_{Bj})^{\mathsf{T}} - 2\mathbb{I}\right\| \leq \frac{\sqrt{\varepsilon}}{q\|\mathbf{r}\|}\right] \\ &\geq 1 - \frac{\delta}{4q}. \end{aligned} \tag{8}$$

Observe that $(\mathbf{x}_{B1} + \mathbf{x}_{Bj})(\mathbf{x}_{B1} + \mathbf{x}_{Bj})^{\mathsf{T}} - (\mathbf{x}_{B1} - \mathbf{x}_{Bj})(\mathbf{x}_{B1} - \mathbf{x}_{Bj})^{\mathsf{T}} = 4\mathbf{x}_{B1}\mathbf{x}_{Bj}^{\mathsf{T}}$. Thus,

$$\begin{aligned}4\mathbf{D}_j =& \frac{1}{m}\sum_{B\in\mathcal{B}}(\mathbf{x}_{B1} + \mathbf{x}_{Bj})(\mathbf{x}_{B1} + \mathbf{x}_{Bj})^{\mathsf{T}} - 2\mathbb{I} \\ &- \left[\frac{1}{m}\sum_{B\in\mathcal{B}}(\mathbf{x}_{B1} - \mathbf{x}_{Bj})(\mathbf{x}_{B1} - \mathbf{x}_{Bj})^{\mathsf{T}} - 2\mathbb{I}\right]\end{aligned}$$

The above, using (8) along with the triangle inequality on the operator norm of matrices gives us

$$\Pr\left[\|\mathbf{D}_j\| \leq \frac{\sqrt{\varepsilon}}{2q\|\mathbf{r}\|}\right] \geq 1 - \frac{\delta}{2q}.$$

Combining (7), (8) along with (6) and a union bound over j, we obtain

$$\Pr[\|\hat{\mathbf{r}} - \mathbf{r}\| \leq \sqrt{\varepsilon}] \geq 1 - \delta. \tag{9}$$

4 PROOF OF THEOREM 1.2

For convenience, let us define

$$\Delta(h) := \mathbb{E}_{(B,y_B)\leftarrow\mathcal{D}_{\text{bag}}(\mathcal{D},f,q)} L_{\text{bag}}(B, y_B, h). \tag{10}$$

Take $B = \{\mathbf{x}_{B1}, \ldots, \mathbf{x}_{Bq}\}$ to be a random bag from $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$ where $y_B = f(\mathbf{x}_{B1})$ as each feature-vector in B is independent and u.a.r. from $\mathcal{D}$. Thus,

$$\begin{aligned}\Delta(h) &:= \mathbb{E}_{\{\mathbf{x}_{Bj}\leftarrow\mathcal{D}\,|\,j=1,\ldots,q\}}\sum_{j=1}^{q}\left[(h(\mathbf{x}_{B1}) - f(\mathbf{x}_{Bj}))^2\right] \\ &= \mathbb{E}_{\mathbf{x}_{B1},\mathbf{x}_{B2}\leftarrow\mathcal{D}}\left[(h(\mathbf{x}_{B1}) - f(\mathbf{x}_{B1}))^2\right. \\ &\quad \left. +(q-1)(h(\mathbf{x}_{B2}) - f(\mathbf{x}_{B1}))^2\right] \end{aligned} \tag{11}$$

where $\mathbf{x}_{B1}, \mathbf{x}_{B2}$ are iid from $\mathcal{D}$.

We will first do the analysis for unbiased target concept f i.e., satisfying $\mathbb{E}_{\mathbf{x}\leftarrow\mathcal{D}}[f] = 0$. The following lemma shows that any regressor h for which $\Delta(h)$ is close to its optimal value, must have low error w.r.t. to a scaled version of f.

Lemma 4.1. *Consider any $f \in \mathcal{F}$ s.t. $\mathbb{E}_{\mathbf{x}\leftarrow\mathcal{D}}[f(\mathbf{x})] = 0$, then, letting $\tilde{f} := f/q$, for any $h : \boldsymbol{\mathcal{X}} \to R$,*

$$\Delta(h) = \Delta(\tilde{f}) + \text{err}_2(D, \tilde{f}, h). \tag{12}$$

In particular, $\tilde{f}$ minimizes $\Delta(h)$ over all regressors h.

Proof. From (11) we have

$$\begin{aligned}&\Delta(h) \\ &= \mathbb{E}\left[(h(\mathbf{x}_{B1}) - f(\mathbf{x}_{B1}))^2 + (q-1)(h(\mathbf{x}_{B2}) - f(\mathbf{x}_{B1}))^2\right] \\ &= \mathbb{E}\left[\left((h(\mathbf{x}_{B1}) - \tilde{f}(\mathbf{x}_{B1})) + (\tilde{f}(\mathbf{x}_{B1}) - f(\mathbf{x}_{B1}))\right)^2\right. \\ &\left. +(q-1)\left((h(\mathbf{x}_{B2}) - \tilde{f}(\mathbf{x}_{B2})) + (\tilde{f}(\mathbf{x}_{B2}) - f(\mathbf{x}_{B1}))\right)^2\right] \\ &= \mathbb{E}\left[\left(h(\mathbf{x}_{B1}) - \tilde{f}(\mathbf{x}_{B1})\right)^2 + \left(\tilde{f}(\mathbf{x}_{B1}) - f(\mathbf{x}_{B1})\right)^2\right. \\ &\quad +2\left(h(\mathbf{x}_{B1}) - \tilde{f}(\mathbf{x}_{B1})\right)\left(\tilde{f}(\mathbf{x}_{B1}) - f(\mathbf{x}_{B1})\right) \\ &\quad + (q-1)\left[\left(h(\mathbf{x}_{B2}) - \tilde{f}(\mathbf{x}_{B2})\right)^2 + \left(\tilde{f}(\mathbf{x}_{B2}) - f(\mathbf{x}_{B1})\right)^2\right. \\ &\quad \left.\left. + 2\left(h(\mathbf{x}_{B2}) - \tilde{f}(\mathbf{x}_{B2})\right)\left(\tilde{f}(\mathbf{x}_{B2}) - f(\mathbf{x}_{B1})\right)\right]\right]. \end{aligned}$$

Using the fact that $\mathbf{x}_{B1}$ and $\mathbf{x}_{B2}$ are iid, $\mathbb{E}[f(\mathbf{x})] = 0$, and $\text{err}_2(\mathcal{D}, \tilde{f}, h) = \mathbb{E}\left[\left(h(\mathbf{x}) - \tilde{f}(\mathbf{x})\right)^2\right]$, the above simplifies

to

$$\begin{aligned}\Delta(h) &= \Delta(\tilde{f}) + \mathrm{err}_2(\mathcal{D}, \tilde{f}, h) \\ &\quad +2\mathbb{E}\left[(1/q-1)\left(h(\mathbf{x}_{B1}) - \tilde{f}(\mathbf{x}_{B1})\right) f(\mathbf{x}_{B1})\right] \\ &\quad +2(q-1)\mathbb{E}\left[(1/q)\left(h(\mathbf{x}_{B2}) - \tilde{f}(\mathbf{x}_{B2})\right) f(\mathbf{x}_{B2})\right] \\ &= \Delta(\tilde{f}) + \mathrm{err}_2(\mathcal{D}, \tilde{f}, h) \\ &\quad +2\mathbb{E}_{\mathbf{x}\leftarrow\mathcal{D}}\left[(1/q-1)\left(h(\mathbf{x}) - \tilde{f}(\mathbf{x})\right) f(\mathbf{x})\right. \\ &\quad \left. + (1-1/q)\left(h(\mathbf{x}) - \tilde{f}(\mathbf{x})\right) f(\mathbf{x})\right] \\ &= \Delta(\tilde{f}) + \mathrm{err}_2(\mathcal{D}, \tilde{f}, h), \end{aligned} \tag{13}$$

completing the proof. □

We now move to a general target f which may have non-zero expectation. Observing that $(h(\mathbf{x}_{Bj}) - f(\mathbf{x}_{B1}))^2 = ((h(\mathbf{x}_{Bj}) - \mathbb{E}[f]) - (f(\mathbf{x}_{B1}) - \mathbb{E}[f]))^2$ and applying the previous lemma, we obtain

$$\Delta(h) = \Delta(\hat{f}) + \mathrm{err}_2(D, \hat{f}, h). \tag{14}$$

where $\hat{f} = f/q + (1-1/q)\mathbb{E}[f]$. We will now show that the optimizer of the loss on the sampled bags, w.h.p., yields an approximation to $\hat{f}$. As per our assumptions, $\mathsf{Pdim}(\mathcal{F}) = r$ defined over $\boldsymbol{\mathcal{X}}$ with range $[0,1]$ that contains f as well as $\hat{f}$. For the rest of the proofs we shall fix $\mathcal{B}$ to be a collection of m bags sampled from $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$. The loss corresponding to $\Delta(h)$ on $\mathcal{B}$ is given by:

$$\Delta(\mathcal{B}, h) := \frac{1}{m} \sum_{B=\{\mathbf{x}_{Bj} \mid j\in[q]\}\in\mathcal{B}} \sum_{i=1}^{q} (h(\mathbf{x}_{B1}) - f(\mathbf{x}_{Bj}))^2 \tag{15}$$

Lemma 4.2. *With probability at least* $1 - 4q\left(\frac{32emq}{\varepsilon r}\right)^r \exp\left(-\frac{(\varepsilon/q)^2 m}{32}\right)$ *over the choice of* $\mathcal{B}$*, for any* $h \in \mathcal{F}$*,* $|\Delta(\mathcal{B}, h) - \Delta(h)| \le \varepsilon$*.*

Proof. Consider a random bag $B = \{\mathbf{x}_{B1}, \ldots, \mathbf{x}_{Bq}\}$. For each $j \in [q]$, applying Theorem 17.1 of Anthony and Bartlett [2009] to the marginal distribution of $(\mathbf{x}_{Bj}, f(\mathbf{x}_{B1}))$, we obtain that w.p. $1 - 4\left(\frac{32emq}{\varepsilon r}\right)^r \exp\left(-\frac{(\varepsilon/q)^2 m}{32}\right)$ over $\mathcal{B}$,

$$\begin{aligned}&\Bigg|\mathbb{E}_{(B=\{\mathbf{x}_{B1},\ldots,\mathbf{x}_{Bq}\}}\left[(h(\mathbf{x}_{B1}) - f(\mathbf{x}_{Bj}))^2\right] \\ &-\frac{1}{m}\sum_{B=\{\mathbf{x}_{Bj}\mid j\in[q]\}\in\mathcal{B}} (h(\mathbf{x}_{B1}) - f(\mathbf{x}_{Bj}))^2\Bigg| \le \varepsilon/q\end{aligned} \tag{16}$$

where the expectation on the LHS is over a random bag B from $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$. Thus, in the following we use a union bound to obtain

$$\begin{aligned}&|\Delta(\mathcal{B}, h) - \Delta(h)| \\ &= \Bigg|\sum_{i=1}^{q}\Bigg[\mathbb{E}_{(B=\{\mathbf{x}_{B1},\ldots,\mathbf{x}_{Bq}\}}\left[(h(\mathbf{x}_{B1}) - f(\mathbf{x}_{Bj}))^2\right] \\ &\quad -\frac{1}{m}\sum_{B=\{\mathbf{x}_{Bj}\mid j\in[q]\}\in\mathcal{B}} (h(\mathbf{x}_{B1}) - f(\mathbf{x}_{Bj}))^2\Bigg]\Bigg| \\ &\le \sum_{i=1}^{q}\Bigg|\mathbb{E}_{(B=\{\mathbf{x}_{B1},\ldots,\mathbf{x}_{Bq}\}}\left[(h(\mathbf{x}_{B1}) - f(\mathbf{x}_{Bj}))^2\right] \\ &\quad -\frac{1}{m}\sum_{B=\{\mathbf{x}_{Bj}\mid j\in[q]\}\in\mathcal{B}} (h(\mathbf{x}_{B1}) - f(\mathbf{x}_{Bj}))^2\Bigg| \\ &\le q\left(\frac{\varepsilon}{q}\right) = \varepsilon\end{aligned}$$

with probability $1 - 4q\left(\frac{32emq}{\varepsilon r}\right)^r \exp\left(-\frac{(\varepsilon/q)^2 m}{32}\right)$. □

For convenience we define $\zeta := 4q\left(\frac{32emq}{\varepsilon r}\right)^r \exp\left(-\frac{(\varepsilon/q)^2 m}{32}\right)$. Using the above we prove the following lemma.

Lemma 4.3. *With probability* $1 - \zeta$*, any* $h \in \mathcal{F}$ *s.t.* $\Delta(\mathcal{B}, h) \le \Delta(\mathcal{B}, \hat{f})$ *satisfies,* $\Delta(h) \le \Delta(\hat{f}) + 3\varepsilon$*.*

Proof. From Lemma 4.2 we have that with probability $1-\zeta$,

$$|\Delta(\mathcal{B}, h) - \Delta(h)| \le \varepsilon, \left|\Delta(\mathcal{B}, \hat{f}) - \Delta(\hat{f})\right| \le \varepsilon, \forall h \in \mathcal{F}. \tag{17}$$

Suppose for a contradiction that there is some $h' \in \mathcal{F}$ s.t.

$$\Delta(\mathcal{B}, h') \le \Delta(\mathcal{B}, \hat{f}) \tag{18}$$

and

$$\Delta(h') > \Delta(\hat{f}) + 3\varepsilon. \tag{19}$$

Using (19) along with (17) yields $\Delta(\mathcal{B}, h') > \Delta(\mathcal{B}, \hat{f}) + \varepsilon$ which is a contradiction to (18). □

Proof. (of Theorem 1.2). Observe that if h minimizes $\Delta(\mathcal{B}, h')$ over all choices of $h' \in \mathcal{F}$, then $\Delta(\mathcal{B}, h) \le \Delta(\mathcal{B}, \hat{f})$ since by our assumption, $\hat{f} \in \mathcal{F}$. Thus, applying Lemma 4.3 we obtain $\Delta(h) \le \Delta(\hat{f}) + 3\varepsilon$ which by (14) implies that $\mathrm{err}_2(D, \hat{f}, h) \le 3\varepsilon$. The theorem statement is obtained by replacing ε with $\varepsilon/3$ in the above proof, and substituting the value of m as in the statement of the theorem so that $\zeta \le \delta$. The estimation of $\mathbb{E}[f]$ can be done using additional bag samples and we defer the details to Appendix D. □

5 EXPERIMENTAL RESULTS

We evaluate our approach over both synthetically generated data and real datasets and compare against baselines for different bag sizes.

Baseline Methodogies. The following baselines are included as part of our experiments:

1. Instance-MIR [Ray and Craven, 2005] in which all the feature-vectors in a bag are labeled with the bag-label and the model is trained on the resultant data.
2. Aggregation-MIR [Wang et al., 2008] in which the feature-vectors in a bag are averaged into a single feature-vector which is assigned the bag label and the model is trained on this aggregated dataset.
3. Prime-MIR [Ray and Page, 2001] which is an EM based method which iteratively selects and updates the best instance in a bag as primary and trains the model on the selected primary instances.
4. BP-MIR [Wang et al., 2008] in which those instances in a bag are removed which are farthest from the median prediction over the nonpruned bags. This is a more sophisticated, as well as empirically better performing, of the pruning based methods.

Training and Evaluation. Our model training uses the above baselines and our proposed algorithms in a mini-batch loop. For the optimisation step, we use the Adam optimiser and do a hyper-parameter search over the learning rate $= \{1\text{e}{-2}, 1\text{e}{-3}, 1\text{e}{-4}, 1\text{e}{-5}, 1\text{e}{-6}\}$ for each configuration (specific dataset, methodology and bag size). For each configuration, we run the same experiment 25 times and report the average mse score. Note that the instances in the specific dataset are randomly bagged for each run. A different random seed is chosen for each trial.

Linear Regression over Synthetic Data. We empirically evaluate Algorithm 1 for linear regression over $N(\mathbf{0}, \mathbf{I})$ (which we refer to as $\mathcal{A}$ for brevity) along with Instance-MIR, Aggregation-MIR, Prime-MIR, BP-MIR baselines. For $d \in \{5, 25\}$, bag size $q \in \{2, 5, 10, 20\}$, and number of bags $m = 5000$, we sample iid instances from $N(\mathbf{0}, \mathbf{I})$ and do a 80/20 split into the training and test sets respectively, whose instance-wise labels are given by $f(\mathbf{x}) = \mathbf{r}^\mathsf{T}\mathbf{x}$ for a randomly sampled regression vector $\mathbf{r}$ from $N(\mathbf{0}, \mathbf{I})$. The train-set is partitioned into training bags of size q and each bag is assigned a bag-label uniformly chosen from its instance-labels. We then compare the instance-wise mse loss on the test set of Algorithm 1 with Instance-MIR, Aggregation-MIR, Prime-MIR, BP-MIR.

Linear Regression over Real Data. We evaluate Algorithm 2 (denoted by $\mathcal{A}$) for linear regression over $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ along with Instance-MIR, Aggregation-MIR, BP-MIR baselines on the *Wine Quality* dataset (Cortez et al. [2009]). We do not include Prime-MIR in the evaluation as it does not converge in sufficient time. Two seperate datasets are in-

Table 1: Linear Regression MIR over $N(\mathbf{0}, \mathbf{I})$ synthetic data

Algorithm	d	q	Test Loss (mse)
$\mathcal{A}$	5	2	0.0093 ± 0.0047
Instance-MIR	5	2	1.2 ± 0.57
Aggregated-MIR	5	2	0.0051 ± 0.0033
Prime-MIR	5	2	$3.2\text{e}{-14} \pm 1.0\text{e}{-14}$
BP-MIR	5	2	1.23 ± 0.09
$\mathcal{A}$	5	5	0.021 ± 0.013
Instance-MIR	5	5	2.7 ± 0.52
Aggregated-MIR	5	5	0.019 ± 0.0099
Prime-MIR	5	5	4.72 ± 4.92
BP-MIR	5	5	0.70 ± 0.07
$\mathcal{A}$	5	10	0.041 ± 0.021
Instance-MIR	5	10	3.2 ± 0.40
Aggregated-MIR	5	10	0.040 ± 0.024
Prime-MIR	5	10	13.82 ± 6.50
BP-MIR	5	10	0.38 ± 0.09
$\mathcal{A}$	5	20	0.034 ± 0.028
Instance-MIR	5	20	1.3 ± 0.16
Aggregated-MIR	5	20	0.029 ± 0.016
Prime-MIR	5	20	0.004 ± 0.013
BP-MIR	5	20	0.092 ± 0.04
$\mathcal{A}$	25	2	0.13 ± 0.040
Instance-MIR	25	2	3.7 ± 0.59
Aggregated-MIR	25	2	0.082 ± 0.023
Prime-MIR	25	2	$1.48\text{e}{-12} \pm 1.05\text{e}{-12}$
BP-MIR	25	2	3.77 ± 0.29
$\mathcal{A}$	25	5	0.45 ± 0.10
Instance-MIR	25	5	10.0 ± 0.52
Aggregated-MIR	25	5	0.38 ± 0.10
Prime-MIR	25	5	2.18 ± 3.67
BP-MIR	25	5	2.72 ± 0.35
$\mathcal{A}$	25	10	1.1 ± 0.26
Instance-MIR	25	10	14.0 ± 0.37
Aggregated-MIR	25	10	0.93 ± 0.27
Prime-MIR	25	10	3.33 ± 6.77
BP-MIR	25	10	2.09 ± 0.34
$\mathcal{A}$	25	20	2.0 ± 0.58
Instance-MIR	25	20	16.0 ± 0.32
Aggregated-MIR	25	20	1.7 ± 0.49
Prime-MIR	25	20	2.80 ± 3.60
BP-MIR	25	20	1.97 ± 0.53

Table 2: Linear Regression MIR over *red wine quality* data

Algorithm	q	Test Loss(mse)
$\mathcal{A}$	5	0.82 ± 0.097
Instance-MIR	5	0.87 ± 0.079
Aggregated-MIR	5	1.5 ± 0.32
BP-MIR	5	0.82 ± 0.07
$\mathcal{A}$	10	1.40 ± 0.34
Instance-MIR	10	0.94 ± 0.057
Aggregated-MIR	10	1.89 ± 0.68
BP-MIR	10	1.30 ± 0.33

Table 3: Linear Regression MIR over *white wine quality* data

Algorithm	q	Test Loss (mse)
$\mathcal{A}$	5	0.77 ± 0.038
Instance-MIR	5	0.88 ± 0.044
Aggregated-MIR	5	1.1 ± 0.17
BP-MIR	5	0.81 ± 0.054
$\mathcal{A}$	10	1.0 ± 0.16
Instance-MIR	10	0.92 ± 0.045
Aggregated-MIR	10	1.9 ± 0.45
BP-MIR	10	0.92 ± 0.13

Table 4: Neural Network MIR over synthetic data

Algorithm	d	q	Test Loss (mse)
$\mathcal{A}_2$	5	5	0.014 ± 0.0028
Instance-MIR	5	5	0.031 ± 0.0026
Aggregated-MIR	5	5	0.070 ± 0.021
Prime-MIR	5	5	0.081 ± 0.023
BP-MIR	5	5	0.014 ± 0.0023

cluded in this one dataset, related to *red* and *white* vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests. The *red* wine dataset has 1599 wine samples and the *white* wine dataset has 4898 wine samples. For both *red* and *white* wines, we use the feature QUALITY as the label and regress on the rest of the features. We pre-process the data by standardising each feature column and label. We randomly shuffle the samples into an 80/20 split into training and test data. We use bag sizes $q \in \{5, 10\}$ and for each bag size, we assign a bag-label uniformly chosen from its instance-labels for both wines. We try and find the optimal linear regressor $\mathbf{r}$ for the features $\mathbf{x}$, $f(\mathbf{x}) = \mathbf{r}^\mathsf{T}\mathbf{x}$. We then compare the instance-wise mse loss on the test set of Algorithm 2 with Instance-MIR, Aggregation-MIR, BP-MIR.

Neural Regression over Synthetic Data. We conduct synthetic experiments for a neural network architecture with a 5-neuron ReLU-activated hidden layer and a final linear activation. Since the final layer is linear, for any network f, $f_b = bf + (1-b)\mathbb{E}[f]$ can also be achieved by this architecture. For the experiments we fix dimension $d = 5$, bag size $q = 5$, number of bags $m = 1000$, and do 5. To generate the synthetic data, we sample $\mathbf{x}$ from $N(\mathbf{0}, \mathbf{I})$, but this distribution is unknown to the algorithm. We initialize a random neural network f with weights of each layer initialised from He-Normal and the biases of each layer set to zero. We then obtain the labels for each instance, and perform $80/20$ test-train splits and create the bags as described above. Our goal is to recover the weights and biases of the neural network used to generate the bag labels, given $\mathbf{x}$ and the architectures.

We train a neural network h to minimises the sample loss $\Delta(\mathcal{B}, h)$ from (15) and estimate $\mathbb{E}_{\mathcal{D}}[f(\mathbf{x})]$ by simply averaging over the bag labels. Let the neural network for returned by the optimiser be h and let the weights and biases of the last (linear) layer of h be $\mathbf{w}, b$ respectively. We then replicate the neural network h to form $\tilde{f}$, and then modify the weights and biases of the last layer of $\tilde{f}$ to be $\mathbf{w}' = q\mathbf{w}$ and $b' = qb - (q-1)\mathbb{E}_{\mathcal{D}}[f(\mathbf{x})]$. Our algorithm outputs the scaled neural network $\tilde{f}$ and we compare test losses with the Instance-MIR, Aggregation-MIR, Prime-MIR, BP-MIR baselines. We refer to our algorithm as $\mathcal{A}_2$ for convenience.

Results. Table 1 contains our experimental results for linear regression on MIR over $N(\mathbf{0}, \mathbf{I})$ synthetic data, Tables 2, 3 contain the results for linear regression on MIR over a real dataset, and Table 4 contains the results for the synthetic neural network regression experiment. For the linear synthetic data experiments, Prime-MIR performed exceedingly well on bag size 2 as there are much fewer assignments of prime instances as compared to datasets with larger bag sizes and it is unstable for larger bags, giving rise to a high variance term. This instability and variance of performance across bag sizes is also noted by Ray and Page [2001]. Other than for bag size 2, we see that our algorithm outperforms all baselines except Aggregated-MIR, which performs equally well. However, in *wine quality* linear regression, we see that Aggregated-MIR performs worse than $\mathcal{A}$, Instance-MIR, BP-MIR, all of which perform equally well. We observe that the test loss for $\mathcal{A}_2$ for synthetic neural network regression performs the best among all baselines along with BP-MIR. Since Instance-MIR is simply our algorithm without the scaling step, these results validate our theoretical analysis, and and confirm that the scaling step in our algorithm is crucial for accurately recovering the target regressor.

The experimental code is available at `https://github.com/google-deepmind/mir_uai25`. The implementations of the algorithms in this paper are in python using the TensorFlow library. Our experiments were run on a system with standard 8-core CPU, 64GB of memory with one 16 GB RAM GPU.

6 CONCLUSIONS

Our work is the first to study computational learning in MIR, providing a PAC learning algorithm for the linear regression task on random bags and bag-labels over Gaussian feature-vectors. Our algorithm recovers the target regressor to arbitrary accuracy by optimizing a bag-level squared-Euclidean loss. This is in contrast to previous work of Chauhan et al. [2024] who showed that linear MIR is NP-hard to approximate on arbitrary bags. We also show the applicability of our loss formulation to neural regression tasks. We conduct experimental evaluations which show that our techniques significantly outperform popular baselines, validating our theoretical insights. Open directions on this topic would be to develop techniques for more complicated bag constructions and more general feature-vector distributions.

References

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, USA, 1st edition, 2009. ISBN 052111862X.

Avrim Blum and Adam Kalai. A note on learning from multiple-instance examples. *Machine learning*, 30:23–29, 1998.

Anand Paresh Brahmbhatt, Rishi Saket, and Aravindan Raghuveer. PAC learning linear thresholds from label proportions. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5Gw9YkJkFF.

Kushal Chauhan, Rishi Saket, Lorne Applebaum, Ashwinkumar Badanidiyuru, Chandan Giri, and Aravindan Raghuveer. Generalization and learnability in multiple instance regression. In *UAI*, 2024.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553, 2009.

Sanjoy Dasgupta. Learning mixtures of gaussians. In *Proc. FOCS*, 1999.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, 1997.

Dong Liang, Xinbo Gao, Wen Lu, and Jie Li. Deep blind image quality assessment based on multiple instance regression. *Neurocomputing*, 431:78–89, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.12.009. URL https://www.sciencedirect.com/science/article/pii/S0925231220318907.

Jianyi Liu, Rui Qiao, Yueying Li, and Sheng Li. Witness detection in multi-instance regression and its application for age estimation. *Multimedia Tools and Applications*, 78:33703–33722, 2019.

T. Lozano-Pérez and C. Yang. Image database retrieval with multiple-instance learning techniques. In *Proc. ICDE*, page 233, 2000.

O. Maron. *Learning from ambiguity*. PhD thesis, Massassachusetts Institute of Technology, 1998.

Conor O'Brien, Arvind Thiagarajan, Sourav Das, Rafael Barreto, Chetan Verma, Tim Hsu, James Neufield, and Jonathan J Hunt. Challenges and approaches to privacy preserving post-click conversion prediction. *arXiv preprint arXiv:2201.12666*, 2022.

Seongoh Park, Xinlei Wang, Johan Lim, Guanghua Xiao, Tianshi Lu, and Tao Wang. Bayesian multiple instance regression for modeling immunogenic neoantigens. *Statistical Methods in Medical Research*, 29(10):3032–3047, 2020.

S. Ray and D. Page. Multiple instance regression. In *Proc. ICML*, pages 425–432, 2001.

Soumya Ray and Mark Craven. Supervised versus multiple instance learning: An empirical comparison. In *Proceedings of the 22nd international conference on Machine learning*, pages 697–704, 2005.

Karan Sikka, Abhinav Dhall, and Marian Bartlett. Weakly supervised pain localization using multiple instance learning. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8, 2013.

Mohamed Trabelsi and Hichem Frigui. Fuzzy and possibilistic clustering for multiple instance linear regression. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7, 2018.

Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

Santosh S. Vempala. Learning convex concepts from gaussian distributions with PCA. In *Proc. FOCS*, 2010.

K. L. Wagstaff and T. Lane. Salience assignment for multiple-instance regression. In *Workshop on Constrained Optimization and Structured Output (ICML)*, 2007.

K. L. Wagstaff, T. Lane, and A. Roper. Multiple-instance regression with structured data. In *Workshops Proceedings of the 8th IEEE ICDM*, pages 291–300, 2008.

Martin J. Wainwright. *High Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, USA, 1st edition, 2019.

Z. Wang, V. Radosavljevic, B. Han, Z. Obradovic, and S. Vucetic. Aerosol optical depth prediction from satellite observations by multiple instance regression. In *Proceedings of the 8th SIAM International Conference on Data Mining (SDM)*, pages 165–176, 2008.

Z. Wang, L. Lan, and S. Vucetic. Mixture model for multiple instance regression and applications in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 50(6):2226–2237, 2012.

J. Wu, Yinan Yu, Chang Huang, and Kai Yu. Deep multiple instance learning for image classification and auto-annotation. In *Proc. CVPR*, pages 3460–3469, 2015.

F. X. Yu, K. Choromanski, S. Kumar, T. Jebara, and S. F. Chang. On learning from label proportions. *CoRR*, abs/1402.5902, 2014. URL http://arxiv.org/abs/1402.5902.

A PRELIMINARIES FOR APPENDIX

A.1 HOEFFDING'S INEQUALITY

We state the well known Hoeffding's inequality.

Theorem A.1. *Let $X_1, X_2, \ldots, X_m$ be independent random variables such that $a_i \le X_i \le b_i$. Consider the sum of these random variables $S_m = X_1 + X_2 + \cdots + X_m$. Then we have for all $t > 0$, $\Pr(|S_m - \mathbb{E}[S_m]| \ge t) \le 2\exp\left(-\frac{2t^2}{\sum_{i=1}^m (b_i - a_i)^2}\right)$*

A.2 DIFFERENTIATION W.R.T. A VECTOR

We state basic identities for differentiation with respect to vectors. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and symmetric $A \in \mathbb{R}^{d \times d}$, we have:

$$\frac{\partial}{\partial \mathbf{x}^\mathsf{T}}(\mathbf{x}^\mathsf{T} A \mathbf{y}) = A\mathbf{y}, \qquad \frac{\partial}{\partial \mathbf{x}^\mathsf{T}}(\mathbf{x}^\mathsf{T} A \mathbf{x}) = 2A\mathbf{x}$$

For reference see Appendix C of [W. Yang, W. Cao, T. Chung, J. Morris: Applied Numerical Methods Using Matlab, 2007].

A.3 CASE OF SINGULAR COVARIANCE MATRIX

If $\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T} + \boldsymbol{\Sigma}$ is not invertible, observe that any $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is in the linear space spanned by the eigen-vectors of $\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T} + \boldsymbol{\Sigma}$ corresponding to non-zero eigenvalues. Thus, one can consider this reduced space in which case the minimimum non-zero eigenvalue is given by the operator norm of its pseudo-inverse (see Section A.5.4 of [S. P. Boyd and L. Vandenberghe. Convex Optimization, 2014]). The projection of $\boldsymbol{\mu}$ into that space yields the new mean vector.

A.4 GAUSSIAN RANDOM VECTORS AND THEIR CONCENTRATION

We also state the equivalent of Theorem 2.1 for Gaussian distributions as given in Wainwright [2019].

Lemma A.2. *Consider $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$ in $\mathbb{R}^d$ iid from $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then we have with probability $1 - \delta$,*

$$\left\| \frac{1}{m} \sum_{i=1}^m \left(\mathbf{X}_i \mathbf{X}_i^T - \mathbb{E}[\mathbf{X}_i \mathbf{X}_i^T] \right) \right\| \le O\left(\|\boldsymbol{\Sigma}\| \sqrt{\log\left(\frac{1}{\delta}\right)} \sqrt{\frac{d}{m}} \right).$$

B PROOF OF THEOREM 1.1

For the rest of the proof we assume that $f(\mathbf{x}) := \mathbf{r}^\mathsf{T}\mathbf{x}$ and $\mathcal{D} := N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

In addition, we assume $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ are unknown and estimate them in one step of the algorithm. Let $\boldsymbol{\Gamma} = \boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T} + \boldsymbol{\Sigma}$ be the second moment matrix of $\mathcal{D}$.

Algorithm 2: PAC Learner for $f(\mathbf{x}) := \mathbf{r}^\mathsf{T}\mathbf{x}$ over $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Input: $\mathcal{D}_{\mathsf{bag}}(\mathcal{D} = N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), f = \mathsf{Lin}, q), m, q$, where $f(\mathbf{x}) := \mathbf{r}^\mathsf{T}\mathbf{x}$.
1. Sample a collection $\mathcal{B}$ of m iid bags from $\mathcal{D}_{\mathsf{bag}}(\mathcal{D}, f, q)$.
2. Define $\hat{L}(\mathcal{B}, \mathbf{v}) = \frac{1}{m}\sum_{B \in \mathcal{B}} \sum_{\mathbf{x} \in B} (y_B - \mathbf{v}^\mathsf{T}\mathbf{x})^2$, use convex optimisation to find $\hat{\mathbf{v}}_{\min} = \operatorname{argmin}_{\mathbf{v}} \hat{L}(\mathcal{B}, \mathbf{v})$.
3. Estimate the sample mean $\hat{\boldsymbol{\mu}} := \frac{1}{mq}\sum_{B \in \mathcal{B}} \sum_{\mathbf{x} \in B} \mathbf{x}$, and sample second moment $\hat{\boldsymbol{\Gamma}} := \frac{1}{mq}\sum_{B \in \mathcal{B}} \sum_{\mathbf{x} \in B} \mathbf{x}\mathbf{x}^\mathsf{T}$.
4. Output $\hat{\mathbf{r}} = ((q-1)\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T} + \hat{\boldsymbol{\Gamma}})^{-1} \left(\frac{1}{m}\sum_{B \in \mathcal{B}} \sum_{\mathbf{x} \in B} \mathbf{x}\mathbf{x}^\mathsf{T}\right) \hat{\mathbf{v}}_{\min}$.

Lemma B.1. *For any $\varepsilon, \delta \in (0, 1)$, if $m \ge O\left(\frac{dq^2 \|\mathbf{r}\|_2^2 \log(\frac{q}{\delta})(\|\boldsymbol{\mu}\|+1)(\|\boldsymbol{\mu}\|^2 + \lambda_{\max}(\boldsymbol{\Sigma}))^3}{\lambda_{\min}^2(\boldsymbol{\Gamma})\varepsilon}\right)$, then $\hat{\mathbf{r}}$ returned in Algorithm 1 satisfies $\|\hat{\mathbf{r}} - \mathbf{r}\|_2 \le \sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2 + \lambda_{\max}(\boldsymbol{\Sigma})}}$ with probability $1 - \delta$.*

We defer the proof of lemma 3.1 to the next subsection.

Lemma B.2. *Let $\varepsilon, \delta \in (0,1)$ and suppose that $\hat{\mathbf{r}}$ returned in Algorithm 2 satisfies $\|\hat{\mathbf{r}} - \mathbf{r}\|_2 \leq \sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2 + \lambda_{max}(\boldsymbol{\Sigma})}}$, then $h(\mathbf{x}) = \hat{\mathbf{r}}^\mathsf{T}\mathbf{x}$ satisfies $\text{err}_2(\mathcal{D}, f, h) \leq \varepsilon$ with probability $1 - \delta$.*

Proof. (of Lemma B.2) $\text{err}_2(\mathcal{D}, f, h) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[(f(\mathbf{x}) - h(\mathbf{x}))^2\right] = \mathbb{E}_{\mathbf{x}\sim N(\boldsymbol{\mu},\boldsymbol{\Sigma})}[((\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}\mathbf{x})^2] = \text{Var}[(\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}\mathbf{x}] + \mathbb{E}[(\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}\mathbf{x}]^2$. Now, note that $(\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}\mathbf{x} \sim N((\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}\boldsymbol{\mu}, (\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}\boldsymbol{\Sigma}(\mathbf{r}-\hat{\mathbf{r}}))$. So we get

$$\begin{aligned}
\text{err}_2(\mathcal{D}, f, h) &= (\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}\boldsymbol{\Sigma}(\mathbf{r}-\hat{\mathbf{r}}) + ((\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}\boldsymbol{\mu})^2 \\
&= (\mathbf{r}-\hat{\mathbf{r}})^\mathsf{T}(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T})(\mathbf{r}-\hat{\mathbf{r}}) \\
&\leq (\lambda_{\max}(\boldsymbol{\Sigma}) + \|\boldsymbol{\mu}\|^2)\|\mathbf{r}-\hat{\mathbf{r}}\|^2 \leq \varepsilon.
\end{aligned}$$

□

B.1 PROOF OF LEMMA B.1

Taking $B = \{\mathbf{x}_{B1}, \ldots, \mathbf{x}_{Bq}\}$ to be a random bag from $\mathcal{D}_{\text{bag}}(\mathcal{D}, f, q)$, one can assume $y_B = f(\mathbf{x}_{B1}) = \mathbf{r}^\mathsf{T}\mathbf{x}_{B1}$ as each feature-vector in B is iid from $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Using this:

$$\begin{aligned}
\hat{L}(\mathcal{B}, \mathbf{v}) &= \frac{1}{m}\sum_{B=\{\mathbf{x}_i \mid i\in[q]\}\in\mathcal{B}}\left[(\mathbf{r}^\mathsf{T}\mathbf{x}_{B1} - \mathbf{v}^\mathsf{T}\mathbf{x}_{B1})^2 + \sum_{j=2}^{q}(\mathbf{r}^\mathsf{T}\mathbf{x}_{B1} - \mathbf{v}^\mathsf{T}\mathbf{x}_{Bj})^2\right] \\
&= (\mathbf{r}-\mathbf{v})^\mathsf{T}\mathbf{A}(\mathbf{r}-\mathbf{v}) + (q-1)\mathbf{r}^\mathsf{T}\mathbf{A}\mathbf{r} + \sum_{j=2}^{q}(\mathbf{v}^\mathsf{T}\mathbf{C}_j\mathbf{v} - \mathbf{r}^\mathsf{T}\mathbf{D}_j^\mathsf{T}\mathbf{v} - \mathbf{v}^\mathsf{T}\mathbf{D}_j\mathbf{r}) \qquad (20)
\end{aligned}$$

where $\mathbf{A} = \frac{1}{m}\sum_{B=\{\mathbf{x}_i \mid i\in[q]\}\in\mathcal{B}}\mathbf{x}_{B1}\mathbf{x}_{B1}^\mathsf{T}$, $\mathbf{C}_j = \frac{1}{m}\sum_{B=\{\mathbf{x}_i \mid i\in[q]\}\in\mathcal{B}}\mathbf{x}_{Bj}\mathbf{x}_{Bj}^\mathsf{T}$, and $\mathbf{D}_j = \frac{1}{m}\sum_{B=\{\mathbf{x}_i \mid i\in[q]\}\in\mathcal{B}}\mathbf{x}_{Bj}\mathbf{x}_{B1}^\mathsf{T}$. We define $\hat{\mathbf{v}}_{\min} = \text{argmin}_{\mathbf{v}}\hat{L}(\mathcal{B}, \mathbf{v})$ as used in Algorithm 2. $\hat{L}(\mathcal{B}, \mathbf{v})$ is convex in $\mathbf{v}$, hence $\hat{\mathbf{v}}_{\min}$ can be found by solving $\dfrac{\partial\hat{L}(\mathcal{B}, \mathbf{v})}{\partial\mathbf{v}} = 0$.

$$0 = \frac{\partial\hat{L}(\hat{\mathbf{v}})}{\partial\mathbf{v}} = 2\mathbf{A}\left(\hat{\mathbf{v}}_{\min} - \mathbf{r}\right) + \sum_{j=2}^{q}\left(2\mathbf{C}_j\hat{\mathbf{v}}_{\min} - 2\mathbf{D}_j\mathbf{r}\right)$$

$$\hat{\mathbf{v}}_{\min} = \left(\mathbf{A} + \sum_{j=2}^{q}\mathbf{C}_j\right)^{-1}\left(\mathbf{A} + \sum_{j=2}^{q}\mathbf{D}_j\right)\mathbf{r} = \left(\frac{1}{m}\sum_{B\in\mathcal{B}}\sum_{j=1}^{q}\mathbf{x}_{Bj}\mathbf{x}_{Bj}^\mathsf{T}\right)^{-1}\left(\mathbf{A} + \sum_{j=2}^{q}\mathbf{D}_j\right)\mathbf{r}$$

Note that $\mathbb{E}_{\mathcal{B}}[\mathbf{A}] = \mathbb{E}_{\mathcal{B}}[\mathbf{C}_j] = \boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T} + \boldsymbol{\Sigma}$ and $\mathbb{E}_{\mathcal{B}}[\mathbf{D}_j] = \boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}$. As defined in Algorithm 2, $\hat{\mathbf{r}}$ is

$$\hat{\mathbf{r}} = ((q-1)\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T} + \hat{\boldsymbol{\Gamma}})^{-1}\left(\frac{1}{m}\sum_{B\in\mathcal{B}}\sum_{j=1}^{q}\mathbf{x}_{Bj}\mathbf{x}_{Bj}^\mathsf{T}\right)\hat{\mathbf{v}}_{\min} = ((q-1)\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T} + \hat{\boldsymbol{\Gamma}})^{-1}\left(\mathbf{A} + \sum_{j=2}^{q}\mathbf{D}_j\right)\mathbf{r}. \qquad (21)$$

So we have

$$\begin{aligned}
\|\hat{\mathbf{r}} - \mathbf{r}\| &\leq \left\|((q-1)\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T} + \hat{\boldsymbol{\Gamma}})^{-1}(\mathbf{A} + \sum_{j=2}^{q}\mathbf{D}_j) - \mathbf{I}\right\|\|\mathbf{r}\| \\
&\leq \left\|((q-1)\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T} + \hat{\boldsymbol{\Gamma}})^{-1}\right\|\left\|\mathbf{A} + \sum_{j=2}^{q}\mathbf{D}_j - (q-1)\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T} - \hat{\boldsymbol{\Gamma}}\right\|\|\mathbf{r}\|
\end{aligned}$$

Clearly, $\left\|((q-1)\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T}+\hat{\boldsymbol{\Gamma}})^{-1}\right\| \leq \frac{1}{\lambda_{\min}(\hat{\boldsymbol{\Gamma}})}$. Using this we obtain,

$$\begin{aligned}
&\|\hat{\mathbf{r}}-\mathbf{r}\| \\
\leq\ & \frac{1}{\lambda_{\min}(\hat{\boldsymbol{\Gamma}})}\left\|\mathbf{A}+\sum_{j=2}^{q}\mathbf{D}_j-(q-1)\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T}-\hat{\boldsymbol{\Gamma}}\right\|\|\mathbf{r}\| \\
\leq\ & \frac{1}{\lambda_{\min}(\hat{\boldsymbol{\Gamma}})}\left\|\mathbf{A}-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}-\boldsymbol{\Sigma}+\sum_{j=2}^{q}(\mathbf{D}_j-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T})\right\|\|\mathbf{r}\|+\frac{\|\hat{\boldsymbol{\Gamma}}-(q\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}+\boldsymbol{\Sigma})\|\|\mathbf{r}\|}{\lambda_{\min}(\tilde{\boldsymbol{\Sigma}})} \\
\leq\ & \frac{\|\mathbf{A}-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}-\boldsymbol{\Sigma}\|\|\mathbf{r}\|}{\lambda_{\min}(\hat{\boldsymbol{\Gamma}})}+\sum_{j=2}^{q}\frac{\|\mathbf{D}_j-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}\|\|\mathbf{r}\|}{\lambda_{\min}(\hat{\boldsymbol{\Gamma}})}+\frac{\|\hat{\boldsymbol{\Gamma}}-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}-\boldsymbol{\Sigma}\|\|\mathbf{r}\|}{\lambda_{\min}(\hat{\boldsymbol{\Gamma}})}+\frac{(q-1)\|\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}-\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T}\|}{\lambda_{\min}(\hat{\boldsymbol{\Gamma}})} \qquad (22)
\end{aligned}$$

From the lower bound on m in the statement of Lemma B.1 and using Theorem 2.1 we bound the first term on the RHS of (22) as follows.

$$\Pr\left[\|\mathbf{A}-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}-\boldsymbol{\Sigma}\| \leq \frac{\lambda_{\min}(\boldsymbol{\Gamma})}{8q\|\mathbf{r}\|_2}\sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2+\lambda_{\max}(\boldsymbol{\Sigma})}}\right] \geq 1-\frac{\delta}{4q} \qquad (23)$$

Further, since for any fixed $j \in \{2,\ldots,k\}$, $\{(\mathbf{x}_{B1}-\mathbf{x}_{Bj})\}_{B\in\mathcal{B}} \sim N(\mathbf{0},2\boldsymbol{\Sigma})$ iid , and $\{(\mathbf{x}_{B1}+\mathbf{x}_{Bj})\}_{B\in\mathcal{B}} \sim N(2\boldsymbol{\mu},2\boldsymbol{\Sigma})$ iid, we have using Theorem 2.1

$$\Pr\left[\left\|\frac{1}{m}\sum_{B\in\mathcal{B}}(\mathbf{x}_{B1}+\mathbf{x}_{Bj})(\mathbf{x}_{B1}+\mathbf{x}_{Bj})^\mathsf{T}-2\boldsymbol{\Sigma}-4\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}\right\| \leq \frac{\lambda_{\min}(\boldsymbol{\Gamma})}{2q\|\mathbf{r}\|_2}\sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2+\lambda_{\max}(\boldsymbol{\Sigma})}}\right] \geq 1-\frac{\delta}{8q}$$

$$\Pr\left[\left\|\frac{1}{m}\sum_{B\in\mathcal{B}}(\mathbf{x}_{B1}-\mathbf{x}_{Bj})(\mathbf{x}_{B1}-\mathbf{x}_{Bj})^\mathsf{T}-2\boldsymbol{\Sigma}\right\| \leq \frac{\lambda_{\min}(\boldsymbol{\Gamma})}{2q\|\mathbf{r}\|_2}\sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2+\lambda_{\max}(\boldsymbol{\Sigma})}}\right] \geq 1-\frac{\delta}{8q}$$

Observe that $(\mathbf{x}_{B1}+\mathbf{x}_{Bj})(\mathbf{x}_{B1}+\mathbf{x}_{Bj})^\mathsf{T}-(\mathbf{x}_{B1}-\mathbf{x}_{Bj})(\mathbf{x}_{B1}-\mathbf{x}_{Bj})^\mathsf{T}=4\mathbf{x}_{B1}\mathbf{x}_{Bj}^\mathsf{T}$. Thus,

$$4\mathbf{D}_j-4\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}=\frac{1}{m}\sum_{B\in\mathcal{B}}(\mathbf{x}_{B1}+\mathbf{x}_{Bj})(\mathbf{x}_{B1}+\mathbf{x}_{Bj})^\mathsf{T}-2\boldsymbol{\Sigma}-4\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}-\left[\frac{1}{m}\sum_{B\in\mathcal{B}}(\mathbf{x}_{B1}-\mathbf{x}_{Bj})(\mathbf{x}_{B1}-\mathbf{x}_{Bj})^\mathsf{T}-2\boldsymbol{\Sigma}\right]$$

Using the above along with the triangle inequality of the operator norm on matrices, and a union bound gives us

$$\Pr\left[\|\mathbf{D}_j-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}\| \leq \frac{\lambda_{\min}(\boldsymbol{\Gamma})}{8q\|\mathbf{r}\|_2}\sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2+\lambda_{max}(\boldsymbol{\Sigma})}}\right] \geq 1-\frac{\delta}{4q}. \qquad (24)$$

We again use Theorem 2.1, leveraging the lower bound on m, to bound $\|\hat{\boldsymbol{\Gamma}}-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}-\boldsymbol{\Sigma}\|$, thus obtaining

$$\Pr\left[\|\hat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}\| \leq \frac{\lambda_{\min}(\hat{\boldsymbol{\Gamma}})}{8\|\mathbf{r}\|_2}\sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2+\lambda_{max}(\boldsymbol{\Sigma})}}\right] \geq 1-\frac{\delta}{8} \qquad (25)$$

Now, to bound the last term $\|\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T}-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}\|$, we first bound $\|\hat{\boldsymbol{\mu}}-\boldsymbol{\mu}\|$. Note that $\hat{\boldsymbol{\mu}}-\boldsymbol{\mu} \sim N(\mathbf{0},\frac{\boldsymbol{\Sigma}}{m})$. We use Gaussian concentration (A.2) to obtain for $m \geq O\left(\frac{dq\|\mathbf{r}\|_2(\|\boldsymbol{\mu}\|+1)\log(\frac{1}{\delta})\lambda_{\max}(\boldsymbol{\Sigma})}{\lambda_{\min}(\boldsymbol{\Gamma})}\sqrt{\frac{\|\boldsymbol{\mu}\|^2+\lambda_{\max}(\boldsymbol{\Sigma})}{\varepsilon}}\right)$,

$$\Pr\left[\|\hat{\boldsymbol{\mu}}-\boldsymbol{\mu}\| \leq \frac{\lambda_{\min}(\boldsymbol{\Gamma})}{16q\|\mathbf{r}\|_2(\|\boldsymbol{\mu}\|_2+1)}\sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2+\lambda_{\max}(\boldsymbol{\Sigma})}}\right] \geq 1-\frac{\delta}{8} \qquad (26)$$

Now, we use this to upper-bound the last term as follows,

$$\begin{aligned}
\|\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^\mathsf{T}-\boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}\| &\leq \|(\hat{\boldsymbol{\mu}}-\boldsymbol{\mu})(\hat{\boldsymbol{\mu}}-\boldsymbol{\mu})^\mathsf{T}+\boldsymbol{\mu}(\hat{\boldsymbol{\mu}}-\boldsymbol{\mu})^\mathsf{T}+(\hat{\boldsymbol{\mu}}-\boldsymbol{\mu})\boldsymbol{\mu}^\mathsf{T}\| \\
&\leq \|\hat{\boldsymbol{\mu}}-\boldsymbol{\mu}\|^2+2\|\boldsymbol{\mu}\|\|\hat{\boldsymbol{\mu}}-\boldsymbol{\mu}\| \qquad (27)
\end{aligned}$$

Substituting the bound in equation (26), we get that for $m \geq O\left(\frac{dq\|\mathbf{r}\|_2(\|\boldsymbol{\mu}\|+1)\log(\frac{1}{\delta})\lambda_{\max}(\boldsymbol{\Sigma})}{\lambda_{\min}(\boldsymbol{\Gamma})}\sqrt{\frac{\|\boldsymbol{\mu}\|^2+\lambda_{\max}(\boldsymbol{\Sigma})}{\varepsilon}}\right)$,

$$\Pr\left[\|\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^{\mathsf{T}} - \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}}\| \leq \frac{\lambda_{\min}(\boldsymbol{\Gamma})}{8q\|\mathbf{r}\|_2}\sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2 + \lambda_{max}(\boldsymbol{\Sigma})}}\right] \geq 1 - \frac{\delta}{8} \tag{28}$$

Combining the bounds in (23), (24), (25), (28) we obtain that

$$\Pr\left[\|\hat{\mathbf{r}} - \mathbf{r}\| \leq \frac{\lambda_{\min}(\boldsymbol{\Gamma})}{2\lambda_{\min}(\hat{\boldsymbol{\Gamma}})}\sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2 + \lambda_{max}(\boldsymbol{\Sigma})}}\right] \geq 1 - \frac{3\delta}{4}. \tag{29}$$

We use Weyl's inequality on perturbation of eigenvalues as mentioned in Equation (6.7) of [Wainwright, 2019] along with Theorem 2.1 applied to iid samples from $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, to conclude that for $m \geq O\left(\frac{d\log(\frac{1}{\delta})(\|\boldsymbol{\mu}\|^2+\lambda_{\max}(\boldsymbol{\Sigma}))^2}{\lambda_{\min}^2(\boldsymbol{\Gamma})}\right)$, we have $\lambda_{\min}(\hat{\boldsymbol{\Gamma}}) \geq \lambda_{\min}(\boldsymbol{\Gamma})/2$ with probability at least $1 - \frac{\delta}{4}$. Combining this with equation (29), we get that for m as lower bounded in the statement of Lemma B.1

$$\Pr\left[\|\hat{\mathbf{r}} - \mathbf{r}\| \leq \sqrt{\frac{\varepsilon}{\|\boldsymbol{\mu}\|_2^2 + \lambda_{\max}(\boldsymbol{\Sigma})}}\right] \geq 1 - \delta. \tag{30}$$

C CLOSURE OF NEURAL NETWORKS UNDER TRANSFORMATION

We consider a concept class $\mathcal{F}$ of regressors with bounded outputs in $[0, 1]$ which is closed under the following transformation: for any $f \in \mathcal{F}$, $f_b = bf + (1 - b)\mathbb{E}[f] \in \mathcal{F}$ for any $b \in [0, 1]$. Common regression neural networks that have a final activation which is relu are closed under this transformation. Their output can be multiplicatively scaled by simply scaling its input weights of the final layer uniformly. A scalar translation can be achieved by adding a constant to the output.

D ESTIMATION FOR THEOREM 1.2

The estimation of $\mathbb{E}[f]$ can be done using averaging the bag label of m' bag samples. As $f(\mathbf{x}) \in [0, 1]$, we can use Hoeffding's inequality (Theorem A.1) to bound the error in the estimate $E_{m'}$. We get $\Pr(|E_{m'} - \mathbb{E}[f]| \geq t) \leq 2\exp(-2t^2/m')$. To get an absolute error of t with a probability of $1 - \delta$, we would need $m' = 2t^2/\log(2/\delta)$ many bag samples. We can estimate $\mathbb{E}[f]$ very accurately with a high probability with a relatively small number of samples. Hence, we exclude the error in the estimation of $\mathbb{E}[f]$ in the analysis of Theorem 1.2 and assume that this constant is known exactly for simplicity.

E EXPERIMENTS OVER NOISY SYNTHETIC DATA

We conduct experiments on data generated by adding Gaussian noise $N(0, \sigma^2)$ to linear synthetic labels generated using the same methodology as before for bag size $q = 5$, dimension $d = 10$. We compare our algorithm's robustness to Gaussian noise against the Instance-MIR, Aggregation-MIR in Table 5. We conduct more experiments adding $N(0, \sigma^2)$ Gaussian noise to 2-layer neural network synthetic labels generated above for bag size $q = 5$, dimension $d = 5$ and report the results in Table 6. In Tables 5 and 6, we observe that our algorithm performs favorably under Gaussian noise and is robust.

Table 5: Linear Regression MIR over noisy synthetic data

σ^2	Instance-MIR	$\mathcal{A}$	Agg-MIR
0.0	7.771 ± 0.109	0.166 ± 0.068	0.184 ± 0.101
0.1	7.787 ± 0.208	0.159 ± 0.060	0.125 ± 0.046
0.5	7.728 ± 0.225	0.170 ± 0.053	0.105 ± 0.054
1.0	7.711 ± 0.196	0.193 ± 0.067	0.159 ± 0.071
2.0	7.757 ± 0.231	0.165 ± 0.088	0.175 ± 0.058
5.0	7.698 ± 0.184	0.364 ± 0.153	0.346 ± 0.107
10.0	8.349 ± 0.467	1.574 ± 0.580	1.469 ± 0.815

Table 6: Neural Network MIR over noisy synthetic data

σ^2	Instance-MIR	$\mathcal{A}$	Agg-MIR
0.0	0.446 ± 0.025	0.168 ± 0.040	0.427 ± 0.076
0.01	0.432 ± 0.021	0.206 ± 0.051	0.369 ± 0.088
0.05	0.465 ± 0.039	0.193 ± 0.078	0.345 ± 0.092
0.1	0.442 ± 0.019	0.160 ± 0.027	0.363 ± 0.025
0.5	0.467 ± 0.026	0.235 ± 0.045	0.399 ± 0.101
1.0	0.472 ± 0.029	0.441 ± 0.042	0.558 ± 0.080