Resonant Anomaly Detection with Multiple Reference Datasets

Mayee F. Chen,[a] Benjamin Nachman,[b;c] and Frederic Sala[d]

[a]Computer Science Department, Stanford University, Stanford, CA 94305, USA
[b]Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[c]Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA
[d]Department of Computer Sciences, University of Wisconsin, Madison, WI 53706, USA

E-mail: mfchen@stanford.edu, bpnachman@lbl.gov, fredsala@cs.wisc.edu

Abstract: An important class of techniques for resonant anomaly detection in high energy physics builds models that can distinguish between reference and target datasets, where only the latter has appreciable signal. Such techniques, including Classication Without Labels (CWoLa) and Simulation Assisted Likelihood-free Anomaly Detection (SALAD) rely on a single reference dataset. They cannot take advantage of commonly-available multiple datasets and thus cannot fully exploit available information. In this work, we propose generalizations of CWoLa and SALAD for settings where multiple reference datasets are available, building on weak supervision techniques. We demonstrate improved performance in a number of settings with realistic and synthetic data. As an added benet, our generalizations enable us to provide nite-sample guarantees, improving on existing asymptotic analyses.

Contents

1 Introduction 2

2 Problem Setup 3

3 Multi-CWoLa: Learning from Multiple Resonant Features 3
- 3.1 Multi-CWoLa Method 4
  - 3.1.1 Model 4
  - 3.1.2 Parameter Estimation 5
  - 3.1.3 Inference and Training 5
- 3.2 Theoretical Results 6
- 3.3 Empirical Results 7

4 Multi-SALAD: Learning from Multiple Simulations 8
- 4.1 Multi-SALAD Method 9
- 4.2 Theoretical Results 10
- 4.3 Empirical Results 11

5 Conclusions and Outlook 13

A Glossary 19

B Additional Algorithmic Details 19
- B.1 Multi-SALAD Algorithm 19

C Additional Theoretical Results 20
- C.1 The Need for 3 Resonant Features 20
- C.2 Rademacher Complexity Bounds 21
- C.3 Asymptotic behavior of SALAD's $\hat{L}_S(h; w)$ 21

D Proofs 22
- D.1 Proof of Theorem 1 22
- D.2 Proof of Theorem 2 23

E Experiment Details 25
- E.1 Multi-CWoLa Experiments 25
- E.2 Multi-SALAD Experiments 25

1 Introduction

Due to the vast parameter space of Standard Model extensions and to the lack of signicant evidence for new particles or forces of nature, a new model-agnostic search paradigm has emerged. Many of these anomaly detection (AD) strategies are enabled by machine learning (see e.g. Ref. [1{4]) and the rst results with collision data are now available [5, 6]. One way to characterize AD methods is based on their physics assumption of the new phenomena [2]. Strategies that assume the new physics is \rare" [7] estimate (explicitly or implicitly) the data probability density and focus on events with low density. In contrast, techniques that assume the new physics will manifest as an overdensity in phase space use likelihood ratio methods to compare a reference dataset to a target dataset. The latter approach has been extensively studied in the context of resonant anomaly detection [8], where one resonant feature (usually a mass) is used to create a sideband region (reference dataset) nearly devoid of any anomalous events and a signal region (target dataset) that may contain anomalies. The reference dataset is used to estimate the presence of anomalies in the target dataset via interpolation.

Many existing approaches are dened using one reference dataset and one target dataset [9{18, 18{24]. However, in practice one can have access to or construct multiple references. First, there may exist multiple resonant features that can be used to construct sideband and signal regions. For instance, when a particle decays into two new particles, the decay products can be used to construct all three intermediate resonances, a setting present in the LHC Olympics Dataset [3]. Second, there may also exist multiple independent Standard Model simulators available for producing a dataset (e.g. Pythia [25], Herwig [26], or Sherpa [27]). Using multiple reference datasets may improve performance, but it is not clear how to incorporate all of their information when using existing methods designed for a single set.

We explore two generalizations of resonant AD to multiple reference datasets. First, we consider Classication Without Labels (CWoLa) [9, 10, 28], in which the reference is simply the sideband region|a form of weak supervision where the noisy label of \signal" is assigned to events in the signal region and the noisy label of 'background' to events in the sideband region. We propose a new method, Multi-CWoLa, that builds multiple reference datasets by constructing signal and sideband regions along dierent resonant features. We consider a point's membership in each feature's signal region as a noisy vote for anomaly, learn weights on each vote, and aggregate them to produce a higher-quality noisy label. We demonstrate Multi-CWoLa's performance on the LHC Olympics Dataset [3].

Next, we study Simulation Assisted Likelihood-free Anomaly Detection (SALAD) [14]. In this method, a reweighting function between a reference simulation dataset and a target dataset is learned in the sideband conditioned on the resonant feature. The simulated events in the signal region are reweighted by interpolating this function and then are used to distinguish anomalies in the target dataset. We extend this to the case of multiple simulated datasets, each of which may make dierent approximation choices and thus provide complementary accuracy when using SALAD. We introduce Multi-SALAD, which

combines the simulated datasets accordingly and then reweights, with the key nding that combining data helps when each simulator approximates dierent components of the background well. We demonstrate Multi-SALAD's performance on synthetic data.

Finally, we study the nite sample guarantees of our proposed methods. Many resonant AD methods have optimality guarantees in some asymptotic limit, but there is no rst-principles understanding of the methods' performance with nite samples. In particular, approaches like the ones described above that use classiers to distinguish a reference dataset from a target dataset approximate the data-to-background likelihood ratio. When the reference (physics) model is correct, this approach will converge to the optimal Neyman-Pearson likelihood ratio test in the limit of innite statistics, complex enough classier architecture, and exible enough training procedure [15, 29]. However, a nite sample understanding of these approaches is lacking. We draw on results from statistical theory to begin a formal study of resonant AD methods with limited data. Our results lay a foundation for future investigations into the nite sample properties of AD and related methods.

This paper is organized as follows. Section 2 briey set up the resonant AD setting and then Multi-CWoLa and Multi-SALAD are introduced in Secs. 3 and 4, respectively. The paper ends with conclusions and outlook in Sec. 5.

2 Problem Setup

We have an input space of discriminating features $x \in X$ and k resonant features $m = [m^1; \ldots; m^k] \in \mathbb{R}^k$. Associated with a point $(x; m)$ is an unknown label $y \in Y$ for $Y = \{0; 1\}$ (background vs. signal). Points $(x; m; y)$ are drawn from a distribution P with density $p()$. For a resonant feature $m^i \in \mathbb{R}$, an interval $I_{m^i} \subset \mathbb{R}$ is used to dene a signal region $SR_i = \{(x; m) : m^i \in I_{m^i}\}$ and a sideband region $SB_i = \{(x; m) : m^i \notin I_{m^i}\}$ (when the resonant feature is obvious, the i is dropped and we use SR and SB). We assume that the sideband region contains little to no signal, i.e., $p(y = 1|(x; m) \in SB) \approx 0$. Our goal is to construct a predictor $f : X \to Y$ to perform anomaly detection.

3 Multi-CWoLa: Learning from Multiple Resonant Features

We introduce Multi-CWoLa, an approach to anomaly detection that uses multiple reference datasets and is built using principles from the area of weak supervision [30, 31].

Standard CWoLa We have one unlabeled dataset $D = \{(x_i; m_i)\}_{i=1}^n$ with one resonant feature ($k = 1$) that we want to use to learn f. We use m to construct the signal and sideband regions, $D_{SR}; D_{SB} \subset D$ where $D_{SR} = D \cap SR$ and $D_{SB} = D \cap SB$, with distributions p_{SR} and p_{SB} respectively. With the intuition that there are more anomalies in the signal region, we express each distribution as a mixture of signal and background components with weight $0 \le {}_{SR}; {}_{SB} \le 1$:

$$p_{SR}(x) = {}_{SR}\, p(x|y = 1) + (1 - {}_{SR})\, p(x|y = 0) \tag{3.1}$$

$$p_{SB}(x) = {}_{SB}\, p(x|y = 1) + (1 - {}_{SB})\, p(x|y = 0) \tag{3.2}$$

Under this construction, the density ratio of the mixtures $\frac{p_{SR}(x)}{p_{SB}(x)}$ can be written in terms of the ratio of the signal and background components, $r(x) = \frac{p(x|y=1)}{p(x|y=0)}$, as $\frac{p_{SR}(x)}{p_{SB}(x)} = \frac{\pi_{SR} r(x) + 1}{\pi_{SB} r(x) + 1} \frac{\pi_{SR}}{\pi_{SB}}$. Assuming $\pi_{SR} > \pi_{SB}$ (e.g. more signal in the signal region), the mixture ratio is monotonically increasing in $r(x)$. Therefore, we train a classier f to learn $\frac{p_{SR}(x)}{p_{SB}(x)}$ by distinguishing between D_{SR} and D_{SB}, and this f provides information about $r(x)$ and can be used for anomaly detection.

3.1 Multi-CWoLa Method

Intuitively, CWoLa uses the resonant feature m as a noisy label that identies the signal versus sideband region and then trains a classier using these. This idea leads to a simple question|if more than one such feature is available ($k > 1$), how can the multiple noisy labels best be utilized? We tackle this question using principles from weak supervision [30{33].

3.1.1 Model

In our approach, we split D along each resonant feature m^i to produce pairs of datasets D_{SB_i} and D_{SR_i} for each $i \in [k]$ based on membership in I_{m^i}. A straightforward way to use all datasets $(D_{SB_1}, D_{SR_1}), \ldots, (D_{SB_k}, D_{SR_k})$ is to apply standard CWoLa k times by training k classiers that we can then ensemble or average. Instead, in Multi-CWoLa, we construct a binary vector per x consisting of k noisy membership labels, $M(m) = \{M_1(m), \ldots, M_k(m)\} \in \{0,1\}^k$, where $M_i(m) = 1$ if $(x, m) \in D_{SR_i}$ and $M_i(m) = 0$ if $(x, m) \in D_{SB_i}$. We propose to directly aggregate these labels $M(m)$ into an estimate of y, $\hat{y}$, and train a classier on the aggregated $\hat{y}$ along with the discriminative features x. Since each $M_i(m)$'s \vote" can have dierent correlation with the true y, we aim to combine the votes in a weighted fashion. We cannot directly measure each membership label's accuracy since the true y is unknown, so we draw on methods from weak supervision.

We model the distribution $p(y, M(m))$ as a probabilistic graphical model with the following parametrization:

$$p(y, M(m); \eta) = \frac{1}{Z} \exp\left(\eta_y \tilde{y} + \sum_{i=1}^{k} \eta_i \tilde{M}_i(m) \tilde{y}\right), \tag{3.3}$$

where $\eta = \{\eta_y, \eta_i \; \forall i \in [k]\}$ are the canonical parameters of the distribution, Z is for normalization, and $\tilde{y}$ and $\tilde{M}_i(m)$ are y and $M_i(m)$ scaled from $\{0,1\}$ to $\{-1,1\}$. Intuitively, η_i represents the (unobserved) strength of the correlation between $M_i(m)$ and y and thus captures a notion of M_i's accuracy. This model also implies, for simplicity, that $M_i(m) \perp M_j(m) | y$; that is, the resonant features are conditionally independent given y.[1]

Our goal is to estimate the parameters of the graphical model and use them to perform inference, producing aggregated weak labels $\hat{y}$ from the distribution $p(y = 1 | M(m); \eta)$ given a vector of noisy labels $M(m)$.

[1] We can model some dependencies among resonant features if desired (see [31] for a method and see [34] for how to learn if resonant features are not conditionally independent). However, we need at least three conditionally independent subsets of resonant features in $M(m)$ in order for the estimation method from [31] to recover the correct parameters.

3.1.2 Parameter Estimation

We first learn the parameters of $p(y, M(m); \theta)$ as defined in (3.3). Of key interest is the accuracy parameter $\theta_i = p(M_i(m) = 1 | y = 1) = p(M_i(m) = 0 | y = 0)$ of the ith resonant feature, which corresponds to the canonical parameter η_i (see [35] for more background on probabilistic graphical models). We estimate the accuracy parameters by adapting the triplet approach from [31]. First, we draw triplets of resonant features $a, b, c \in [k]$.[2] If the distribution on $y, M(m)$ follows the graphical model in (3.3), it holds that $y \perp M_a(m) \perp y \perp M_b(m)$ if $M_a(m) \perp M_b(m) | y$. Then, we have that $E[\tilde{y} \tilde{M}_a(m)] E[\tilde{y} \tilde{M}_b(m)] = E[\tilde{M}_a(m) \tilde{M}_b(m)]$ since $\tilde{y}^2 = 1$. Writing one such equation for each pair in the triplet (a, b, c), we have that

$$
\begin{aligned}
E[\tilde{y} \tilde{M}_a(m)] E[\tilde{y} \tilde{M}_b(m)] &= E[\tilde{M}_a(m) \tilde{M}_b(m)] \\
E[\tilde{y} \tilde{M}_a(m)] E[\tilde{y} \tilde{M}_c(m)] &= E[\tilde{M}_a(m) \tilde{M}_c(m)] \\
E[\tilde{y} \tilde{M}_b(m)] E[\tilde{y} \tilde{M}_c(m)] &= E[\tilde{M}_b(m) \tilde{M}_c(m)].
\end{aligned}
$$

Solving this system, we obtain

$$
|E[\tilde{y} \tilde{M}_a(m)]| = \sqrt{\frac{E[\tilde{M}_a(m) \tilde{M}_b(m)] E[\tilde{M}_a(m) \tilde{M}_c(m)]}{E[\tilde{M}_b(m) \tilde{M}_c(m)]}},
$$

and similarly for b and c. We assume that each signal region is positively correlated with the true signal, which allows for us to ignore the absolute value and uniquely recover $E[\tilde{y} \tilde{M}_a(m)]$. Next, we can use $E[\tilde{y} \tilde{M}_a(m)] = 2p(\tilde{y} = \tilde{M}_a(m)) - 1$ to obtain θ_i using properties of the graphical model in (3.3). Note that in practice, all of these quantities are empirical estimates, with terms such as $\hat{E}[\tilde{M}_a(m) \tilde{M}_b(m)] = \frac{1}{n} \sum_{i=1}^{n} \tilde{M}_a(m_i) \tilde{M}_b(m_i)$.

3.1.3 Inference and Training

After we learn the accuracy parameters, we use them to estimate $p(y = 1 | M(m))$ for a given $M(m)$. We use Bayes' rule and the conditional independence among $M(m)$ to write $p(y | M(m)) = \frac{\prod_{i=1}^{m} p(M_i(m) | y=1) p(y=1)}{p(M(m))}$. We assume that the class balance $p(y = 1)$ is known; otherwise, it can be estimated via tensor decomposition [33]. $p(M_i(m) | y = 1)$ is either equal to θ_i if $M_i(m) = 1$ or $1 - \theta_i$ if $M_i(m) = 0$, and the denominator $p(M(m))$ can be either directly estimated since all quantities are observable or computed as $\prod_{i=1}^{m} p(M_i(m) | y = 1) p(y = 1) + \prod_{i=1}^{m} p(M_i(m) | y = 0) p(y = 1)$ using the estimated accuracies and class balance.

Once $p(y = 1 | M(m))$ is estimated for all $M(m) \in \{0, 1\}^k$, the aggregated weak label $\hat{y}$ is drawn from such distribution. With labels $\hat{y}$ for each $(x, m) \in D$, we train a classifier $f_{\hat{\theta}}$ on the weakly labeled dataset $\{(x, \hat{y})\}_{i=1}^{n}$. This procedure is summarized in Algorithm 1.

[2] We assume that $k \geq 3$. In Lemma 1, we discuss why having $k = 1$ or $k = 2$ resonant features does not recover a unique model.

Algorithm 1 Multi-CWoLa

1: Input: Dataset $D = \{(x_i, m_i)\}_{i=1}^n$; thresholds l_{m^i} that split D into signal and sideband regions, D_{SR_i} and D_{SB_i} respectively, for each m^i; class balance probability of anomaly $p(y = 1)$

2: Construct noisy label $M_i(m) = \begin{cases} 1 & (x, m) \in D_{SR_i} \\ 0 & (x, m) \in D_{SB_i} \end{cases}$ for each resonant feature m^i.

3: for each triplet $a, b, c \in [k]$ do

4:

$$\epsilon_a := \sqrt{\frac{\hat{E}[\tilde{M}_a(m)\tilde{M}_b(m)]\hat{E}[\tilde{M}_a(m)\tilde{M}_c(m)]}{\hat{E}[\tilde{M}_b(m)\tilde{M}_c(m)]}} \tag{3.4}$$

$$\epsilon_b := \sqrt{\frac{\hat{E}[\tilde{M}_a(m)\tilde{M}_b(m)]\hat{E}[\tilde{M}_b(m)\tilde{M}_c(m)]}{\hat{E}[\tilde{M}_a(m)\tilde{M}_c(m)]}} \tag{3.5}$$

$$\epsilon_c := \sqrt{\frac{\hat{E}[\tilde{M}_a(m)\tilde{M}_c(m)]\hat{E}[\tilde{M}_b(m)\tilde{M}_c(m)]}{\hat{E}[\tilde{M}_a(m)\tilde{M}_b(m)]}}; \tag{3.6}$$

where $\hat{E}$ is an empirical estimate of the expectation over D, and $\tilde{M}(m)$ indicates $M(m)$ scaled to $\{-1, 1\}$.

5: end for

6: Set accuracy parameter $\epsilon_i = \hat{p}(M_i(m) = 1|y = 1) = \hat{p}(M_i(m) = 0|y = 0) = \hat{p}(M_i(m) = y) = \frac{\epsilon_i + 1}{2}$.

7: Compute estimate $\hat{p}(y = 1|M(m)) \propto \prod_{i=1}^m \hat{p}(M_i(m)|y = 1)p(y = 1)$.

8: Construct $\hat{y} \sim \hat{p}(y = 1|M(m))$ for each $(x, m) \in D$.

9: Output: Classier $\hat{f}$ for anomaly detection trained on $\{(x_i, m_i, \hat{y}_i)\}_{i=1}^n$.

3.2 Theoretical Results

Under (3.3), Multi-CWoLa oers nite-sample generalization guarantees. Suppose the downstream model $\hat{f}$ trained on $\hat{y}$ belongs to class F. Dene a loss function $\ell_C : Y \times Y \to \mathbb{R}$ and let the expected loss of f be $L_C(f) := E[\ell_C(f(x), y)]$ on true labels. Then, the optimal classier is $f^* = \operatorname{argmin}_{f \in F} L_C(f)$, which is achieved with unlimited labeled data. Let the empirical loss of f on $\hat{y}$ be $\hat{L}_C(f) := \frac{1}{n} \sum_{i=1}^n \ell_C(f(x_i), \hat{y}_i)$. Then, the $\hat{f}$ we learn is constructed from $\hat{f} = \operatorname{argmin}_{f \in F} \hat{L}_C(f)$, which is learned on nite and noisily labeled data. Note that this construction is dierent from the standard empirical risk minimization (ERM) loss on labeled data, and thus $\hat{L}_C(f)$ does not asymptotically equal $L_C(f)$. We aim to minimize the generalization error $L_C(\hat{f}) - L_C(f^*)$.

We now present our result on an upper bound for $L_C(\hat{f}) - L_C(f^*)$. Dene the Rademacher complexity of F as $R_n(\ell \circ F) = E \sup_{f \in F} \frac{1}{n} \sum_{i=1}^n \sigma_i \ell(f(x_i), y_i)$ with random variables $\Pr(\sigma = 1) = \Pr(\sigma = -1) = \frac{1}{2}$. Dene e_{min} as the minimum eigenvalue of the covariance matrix on $[y, M_1(m), \ldots, M_k(m)]$, and let a_{min} be the minimum value of $E[\tilde{M}_i(m)y]$ over all i.

Theorem 1. Assume that $p(y, M(m))$ can be parametrized according to (3.3) and that ℓ is scaled to be bounded in $[0, 1]$. Assume that the class balance $p(y)$ is known (if not, there are ways to estimate it [33]), and that $k \geq 3$. Then, with probability at least $1 - \delta$, the

generalization error of Multi-CWoLa on D is at most

$$L_C(\hat{f}) \quad L_C(f^?) \quad 4R_n(' F) + \sqrt{\frac{\log 2=}{2n}} + \frac{c_1}{e_{min} a_{min}^5} \sqrt{\frac{k}{n}} + \frac{c_2 k}{p \overline{n}}.$$

where $c_1; c_2$ are positive constants.

We observe that there are three quantities controlling the above bound:

- The Rademacher complexity of F: this term describes the model's expressivity. Smaller Rademacher complexity means that the model is easier to learn and that our $\hat{f}$ will be closer to the best model in F. This quantity can be readily computed for a variety of function classes F, such as decision trees, linear models, and two-layer feedforward networks, which makes our bound in Theorem 1 tractable. See Appendix C.2 for exact values.
- Using n nite samples: as the amount of data increases, the error decreases in $O(n^{1=2})$.
- Using noisy labels $\hat{y}$ instead of y: for our weak supervision algorithm and graphical model, using $\hat{y}$ rather than y contributes an additional $O(n^{1=2})$ error. Asymptotically, our approach thus does no worse than training with labeled data.

By contrast, the standard CWoLa approach with k = 1 does not utilize any aggregation or weak supervision, which requires k 3. For standard CWoLa, the second term in the generalization error is irreducible due to the fact that using any single resonant feature in place of y is biased. On the other hand, Multi-CWoLa corrects for some of this bias; the second term asymptotically approaches 0 with more data.

3.3 Empirical Results

In Figure 1, we compare Multi-CWoLa with standard CWoLa as well as two other baselines. We use simulation data from the LHC Olympics Dataset [3]; in particular from Pythia 8 [25], where the signal is boson decay and the background is generic 2 ! 2 parton scattering. This dataset contains 5 features; in the standard CWoLa setup, we use one thresholded resonant feature (k = 1) and use 4 discriminative features as x. For Multi-CWoLa, we have generated k = 3 mixtures by varying how the 3 resonant features (the jet masses in addition to the dijet mass) are thresholded and use 2 discriminative features as x. We have three other baselines that utilize 3 resonant features:

- CWoLa + intersect denes the signal region as the intersection of the resonant features' signal regions, e.g. $SR = SR_1 \setminus SR_2 \setminus SR_3$, but this can be overly conservative.
- CWoLa +x thresholding has one resonant feature as the noisy label $\hat{y} = M_1(m)$, and includes the remaining thresholded features as discriminative $fM_2(m); M_3(m); xg$.
- CWoLa + average runs standard CWoLa three times, once per resonant feature and with the 2 discriminative features. The three model scores are averaged to produce the nal output.

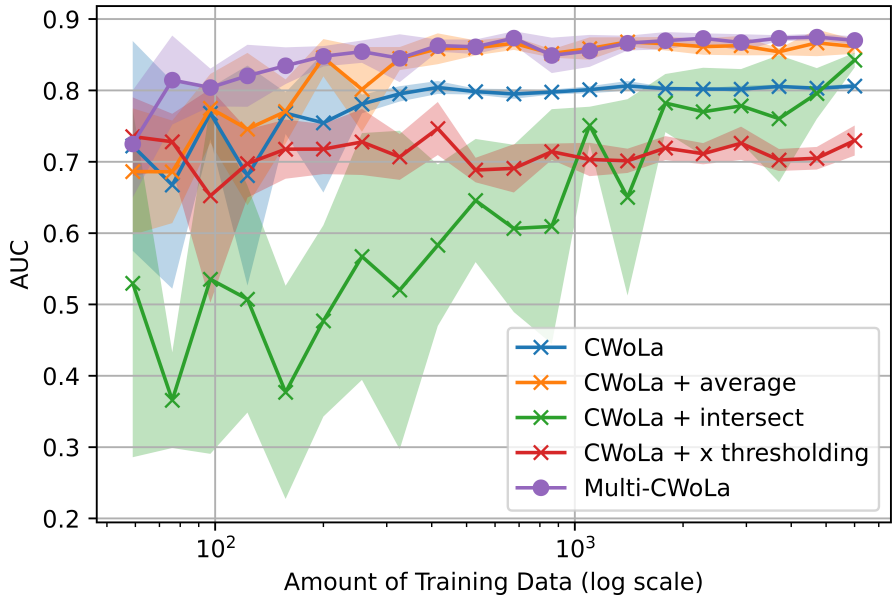

Figure 1. Comparison between CWoLa and Multi-CWoLa. Using multiple mixed samples helps performance across a range of dataset sizes. Access to multiple weak sources enables better AUC and lower variance compared to the single-feature version.

We vary the number of samples available on a logarithmic scale from n = 59 to 6003 and plot the AUC averaged over 5 runs per sample size in 1. We nd that Multi-CWoLa oers a higher AUC and lower variance, especially when there is limited data. We also plot the SI curves averaged over 5 runs for n = 59; 530; 6003 in 2.

4 Multi-SALAD: Learning from Multiple Simulations

We often have access to a(n approximate) simulation of the background process. We rst provide an overview of SALAD, which reweighs samples from the simulation to better assist with classication on the real dataset. Then, we present Multi-SALAD, a variant of SALAD that uses multiple simulations.

Standard SALAD We have a background simulation dataset $D^{sim} = f(x_i; m_i)g_{i=1}^{n_{sim}}$ with $y_i = 0$ for all i in addition to one true dataset $D = f(x_i; m_i)g^{n}{}_{1\cdot i}$. D^{sim} is drawn from some distribution P_{sim} with density p_{sim}. While CWoLA learns the likelihood ratio between the signal and sideband regions of D alone, SALAD utilizes D^{sim} as well. Note that if p_{sim} is equal to p(jy = 0), we could directly train a model to distinguish between D and D^{sim} in the signal region to get a classier that could detect anomalies. However, since D^{sim} may not match the true background data, we instead rst need to learn a reweighting function that captures the dierences between D^{sim} and D's background data, and then we train a model to distinguish between D and the reweighted D^{sim} in the signal region. Formally, given xed SR and SB for both datasets, the method can be broken into two steps:

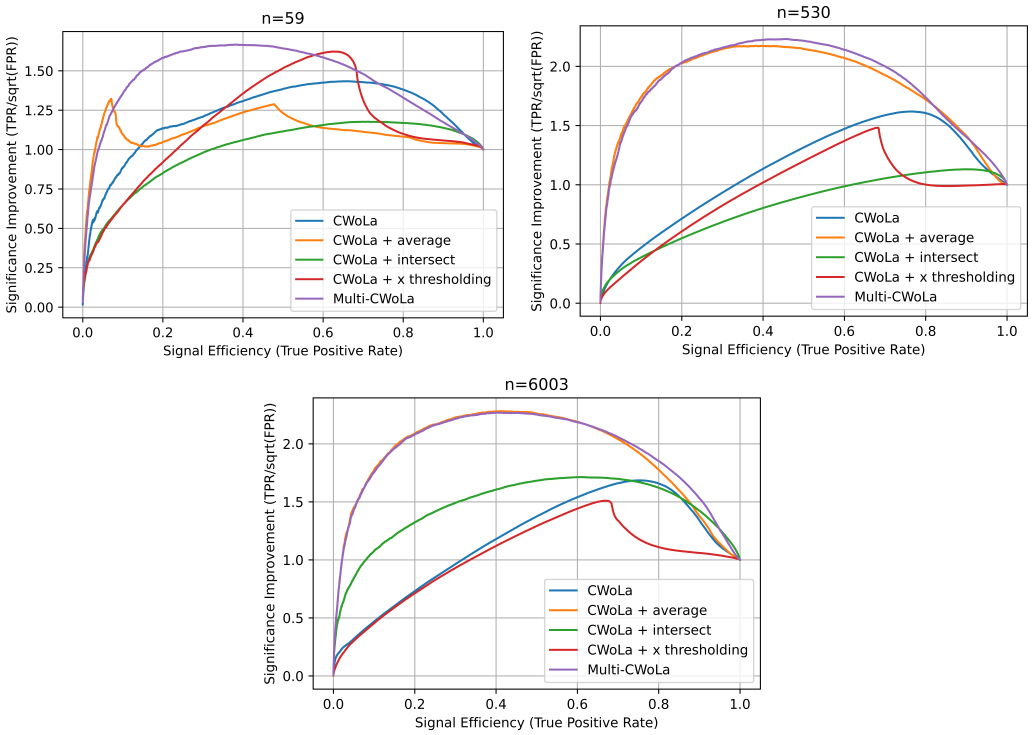

Figure 2. Signicance Improvement (SI) curve for Multi-CWoLa at sizes n = 59; 530; and 6003.

1. Reweighting: a classier $\hat{g}$ is trained to distinguish between $D^{sim}_{SB} = D^{sim} \setminus SB$ and D_{SB}. Assuming that the sideband region has no anomalies, this $\hat{g}$ is able to produce an estimate of the weight ratio[3] $w(x; m) = \frac{p(x;m|y=0)}{p^{sim}(x;m|y=0)} \approx \frac{\hat{g}(x;m)}{1-\hat{g}(x;m)}$, assuming that the datasets are the same size ($|D^{sim}_{SB}| = |D_{SB}|$).

2. Detection: Using a loss function L_S with estimated weights $\hat{w}(x; m)$ applied to $D^{sim}_{SR} = D^{sim} \setminus SR$, a classier $\hat{h}$ is trained to distinguish between D_{SR} and D^{sim}_{SR}.

If the estimate $\hat{w}(x; m)$ is exactly equal to $w(x; m)$ (e.g. $\hat{g}$ is Bayes-optimal), then the second step will be equivalent in expectation to learning the ratio $\frac{p(x)}{p(x|y=0)}$ (see Lemma 2 in Appendix C.3), from which one can detect anomalies.

4.1 Multi-SALAD Method

Now, we have multiple simulation datasets $D^{sim}_1, \ldots, D^{sim}_k$. One approach would be to maintain distinctions among simulations by reweighing each pair to learn k weight functions $w_i(x; m)$, and then using one overall loss function that weights points from each $D^{sim}_{SR;i}$ with w_i. However, it has been shown that importance reweighting, despite working in expectation, can be highly unstable and result in poor performance of tasks on the target data D [40]. To understand why, Ref. [41] showed that the generalization error of an empirical loss function with importance weights w depends on the magnitude of w. Applied to our

[3] This is with the binary cross entropy loss function (also works for other functions [36]). This likelihood-ratio trick is well-known (see e.g. Ref. [37, 38]), also in high-energy physics (see e.g. Ref. [39]).

setting, it suggests that the more inaccurate the simulation is, the less the reweighted loss recovers the true $\frac{p(x)}{p(x|y=0)}$, and the model may instead pick up on dierences between D_{SR} and the reweighted D^{sim}_{SR} that are noise rather than the anomaly. As a result, aggregating individual SALAD outputs can be equivalent to ensembling many poor classiers.

Given these observations, Multi-SALAD uses multiple simulation datasets in a very simple yet theoretically principled way: control the magnitude of the overall w by combining all the D^{sim}_i to produce one large simulation dataset D^{sim} whose distribution best approximates the true background p(xjy = 0), and then use standard SALAD with D^{sim} and D. Note that this approach both improves sample complexity and can \suppress" a simulation that on its own has high w, while the approach of learning k weight functions would not oer such improvements. In Algorithm 2 and Appendix B.1, we write this procedure out where we simply concatenate all D^{sim}_i together. However, with domain knowledge on the strengths and weaknesses of each simulation across features, one could produce D^{sim} by sampling accordingly from each. We leave this direction for future work.

4.2 Theoretical Results

We now present a nite sample generalization error bound on Multi-SALAD that also applies to SALAD. To measure the generalization error, recall $w(x;m) = \frac{p(x;m|y=0)}{p_{sim}(x;m|y=0)}$ and let $\hat{w}$ be the classier g's estimate. We denote h as the reweighted classier. Let $h^? = \text{argmin}_{h\in H} L_S(h;w)$ and let $\hat{h} = \text{argmin}_{h\in H} L_S(h;\hat{w})$. We aim to bound $L_S(\hat{h};w) - L_S(h^?;w)$.

We rst set up some denitions. Dene n^{SR} as the number of points from D and D^{sim} belonging to the signal region, and n^{SB} as the number of points belonging to the sideband. Let n^{SR}_{sim} be the number of points in D^{sim} belonging to the signal region. Let $\hat{g}(x) \in [\hat{g}_{min};\hat{g}_{max}]$ and $g^?(x) \in [g^?_{min};g^?_{max}]$, where $g^?_m$ is the optimal classier. Let $R_{n^{SR}}(`_S \circ \{H;G\})$ be the Rademacher complexity of the overall loss $L_S(h;w)$ across function classes $h \in H; g \in G$. Dene $W = \max_{x;m} w(x;m)$ as the maximum ratio between the simulation and true background. Let $B_1 = \max\{-\log h^?(x;m); -\log(1-h^?(x;m))\}$ be based on the most extreme value of $h^?$ (i.e. how far apart p and p(jy = 0) can be). Let $\gamma = \max(-\log(1-h^?(x;m)))$ for $x;m \in D^{sim}_{SR}$. Let $R_{n^{SB}}(`\circ G)$ is the Rademacher complexity of the loss function class used for learning the reweighting, where ` is point-wise cross-entropy. Finally, let $B_2 = -\log(\min\{\hat{g}_{min}; g^?_{min}\})$.

Theorem 2. With probability at least $1-\delta$, there exists a constant $c > 0$ such that the generalization error of Multi-SALAD on D^{sim} and D is at most

$$L_S(\hat{h};\hat{w}) - L_S(h^?;w) \le 2R_{n^{SR}}(`_S \circ \{H;G\}) + (1+WB_1)\sqrt{\frac{\log 8/\delta}{2n^{SR}}} + \frac{n^{SR}_{sim}}{(1-\hat{g}_{max})(1-g^?_{max})n^{SR}}\left(4cR_{n^{SB}}(`\circ G) + 2c\sqrt{\frac{\log 4/\delta}{2n^{SB}}} + B_2\sqrt{\frac{\log 8/\delta}{2n^{SR}_{sim}}}\right). \quad (4.1)$$

We make several observations about this bound:

- The bound scales in $(n^{SB})^{1=2}$ and $(n^{SR}_{sim})^{1=2}$, where the former comes from the initial reweighting step while the latter comes from the weighted classication step.
- The bound is also dependent on the Rademacher complexities of both classiers g and h used.
- The bound depends on the dierence between the simulation and data distributions through quantities W, $B_1; B_2; ; g_{max}; g_{max}$. If the distributions have very dierent densities, these quantities will all be large, increasing the generalization error.

We comment how this bound is dierent when instantiated for SALAD versus Multi-SALAD. The following example shows how SALAD with one simulation can result in a large W (and other large constants), while Multi-SALAD with two simulations combined can reduce W in the bound.

Example 1. Let $P_{sim}(xjy = 0) = N(;^2), ^2P_{sim}(xjy = 0) = N(;^2)$ be Gaussian distributions on x with $;^2 2 R$, and let the true background distribution $P(jy = 0)$ be a mixture of the Gaussians on x, $P(xjy = 0) = \frac{1}{2}P^1_{sim} + \frac{1}{2}P^2_{sim}$. Let $P^1_{sim}; P^2_{sim};$ and P have the same marginal distribution over m with x ? mjy. Then, if we only use one simulation P^1_{sim},

$$w(x; m) = \frac{p(x; mjy = 0)}{p^1_{sim}(x; mjy = 0)} = \frac{p(xjy = 0)}{p^1_{sim}(xjy = 0)}$$

$$= \frac{\frac{1}{2}\frac{1}{\sqrt{2}} \exp \frac{(x)^2}{2^2} + \frac{1}{2}\frac{1}{\sqrt{2}} \exp \frac{(x+)^2}{2^2}}{\frac{1}{\sqrt{2}} \exp \frac{(x)2}{2^2}}$$

$$= \frac{1}{2} + \frac{1}{2} \exp \frac{(x)^2}{2 2} \frac{(x)^2 +}{2 2} = \frac{1}{2} + \frac{1}{2} \exp \frac{2}{2}:$$

Therefore, as x ! 1, W ! 1. However, if we dene P_{sim} as the distribution of the two simulation datasets concatenated, we have that $p_{sim}(xjy = 0) = p(xjy = 0)$, and as a result, W ! 1, making the generalization error bound smaller.

From this example, we can see that signicantly diering simulation and data distributions can result in large, unbounded weight ratios, which are correlated with poor performance.[4] This concretely motivates our algorithmic objective to combine multiple simulation datasets as to closely approximate the true data.

4.3 Empirical Results

To demonstrate how Multi-SALAD can improve over using only one simulation and over using simulations separately, we consider a synthetic experiment with two simulation

[4] The bound in Theorem 2 is meant to provide a general understanding of SALAD's performance. It can be made tighter by replacing terms that are maxima like M and B_2 with terms that are based on the overall data distributions (e.g. variance, as in Ref. [41]). Variance-based bounds are less likely to be vacuous, but will still demonstrate how performance is dependent on the intrinsic dierences between the two distributions.

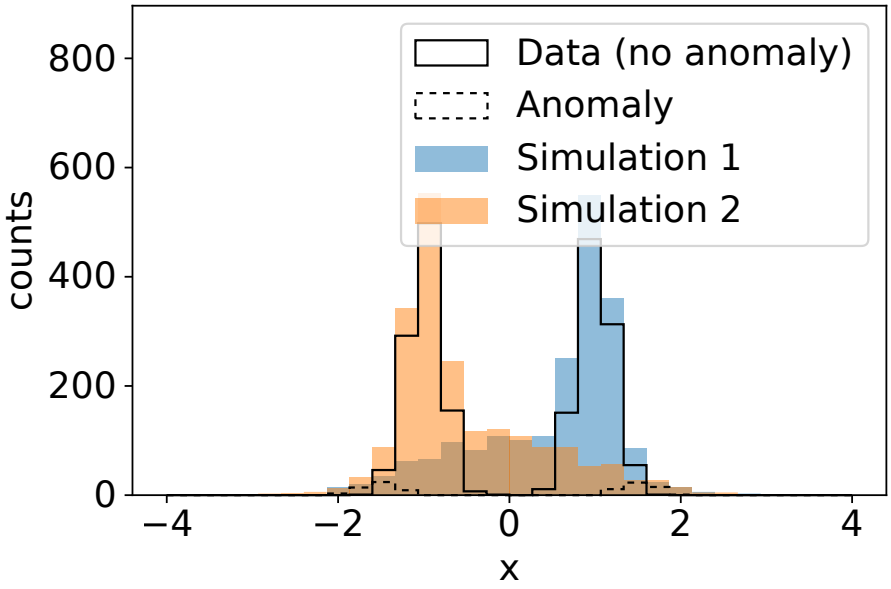

Figure 3. Synthetic data for evaluating Multi-SALAD.

datasets[5]. The true background is $P(y = 0) = \frac{1}{2}\mathcal{N}(-1; 0.2) + \frac{1}{2}\mathcal{N}(1; 0.2)$, and the anomaly is $P(y = 1) = \frac{1}{2}\mathcal{N}(-2; 0.2) + \frac{1}{2}\mathcal{N}(2; 0.2)$. Simulation 1 is $P_{sim}^1 = \frac{1}{2}\mathcal{N}(-1; 0.2) + \frac{1}{2}\mathcal{N}(0; 1)$, and simulation 2 is $P_{sim}^2 = \frac{1}{2}\mathcal{N}(-1; 0.2) + \frac{1}{2}\mathcal{N}(0; 1)$. We generate 2000 points from the true background and 100 points that are anomalies to form D, and 2000 points each from P_{sim}^1 and P_{sim}^2 to form D_1^{sim} and D_2^{sim}. We construct signal and sideband regions from these by splitting datasets in half randomly, assuming they follow the same distribution over x (i.e., m is independent of x) except that there is no anomaly in the sideband regions. A visualization is shown in Figure 3.

Intuitively, the anomaly is only slightly different from the background data, which makes it important to learn a good reweighting function from the simulations. Because each simulation alone diverges greatly from the data for one mode, each individual reweighting may not approximate the true $P(y = 1)$ well. On the other hand, if we combine both simulation datasets together, the aggregate distribution has smaller weights with lower variance, which can allow for more accurate reweighting. This is demonstrated in Figure 4, which depicts the reweighting in the sideband region. Figure 5 depicts the reweighting's interpolation into the signal region, where we introduce an additional baseline SALAD-Switch, which uses k separate weight functions $w_i(x; m)$ and switches among them in the reweighted loss function L_S. In all but the bottom right subfigure in both figures, the reweighted simulation data poorly approximates the true background data. As a result, a classifier trained to distinguish between the high-variance reweighted simulation and the true background data plus some small anomaly will more likely learn the distinctions coming from poor approximation, rather than anomaly. In particular, note that SALAD-

[5] We find that the differences between the simulations in the LHC Olympics are not enough to see a noticeable gain from Multi-SALAD over SALAD.

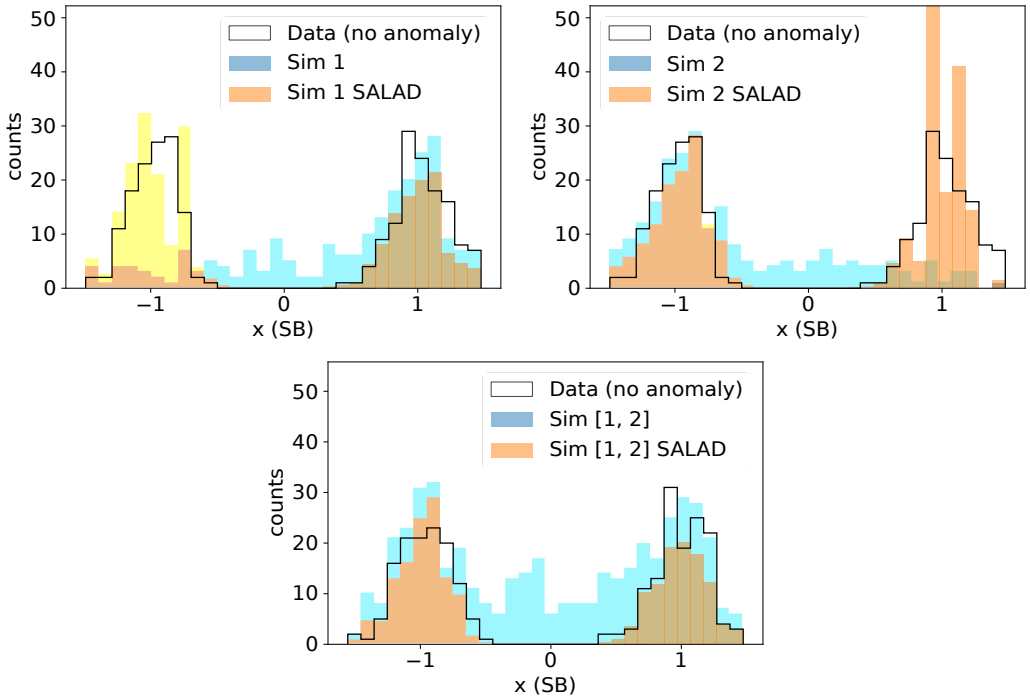

Figure 4. Top left: SALAD reweighting using simulation 1 on sideband region. Top right: reweighting using simulation 2. Bottom: reweighting using simulation 1 and 2 combined.

Switch results in signicant overweighting in Figure 5.

With these observations, we present the signal eciency to rejection rate of each method in Figure 6, where we compare Multi-SALAD against SALAD using simulation 1 only, SALAD using simulation 2 only, and SALAD-Switch. Table 1 contains the accuracy and AUC scores for each method. Averaged over 10 random seeds, Multi-SALAD outperforms other methods. The signal eciency to rejection rate for each of the 10 runs is available in Appendix E.

	Simulation 1		Simulation 2		Simulation 1 and 2		
Method	None	SALAD	None	SALAD	None	SALAD-Switch	Multi-SALAD
Accuracy	$43{:}8_{2:2}$	$62{:}5_{8:8}$	$42{:}7_{3:6}$	$64{:}3_{12:3}$	$50{:}0_{0:0}$	$54{:}3_{6:2}\,74{:}7_{17:0}$	$64{:}8_{9:3}\,90{:}8_{10:2}$
AUC	$28{:}5_{4:2}$	$80{:}7_{14:5}$	$27{:}4_{4:5}$	$78{:}7_{18:2}$	$15{:}4_{5:3}$		

Table 1. Accuracy and AUC scores (%) for Multi-SALAD on two simulation datasets. We compare to SALAD-Switch (dierent reweighting), as well as standard SALAD on individual simulations and no reweighting. Performance is averaged over 10 random runs with one standard deviation reported.

5 Conclusions and Outlook

We extend two resonant AD approaches to incorporate multiple reference datasets. For Multi-CWoLa, we draw from weak supervision models to handle multiple resonant features. For Multi-SALAD, we combine multiple simulation datasets to best approximate

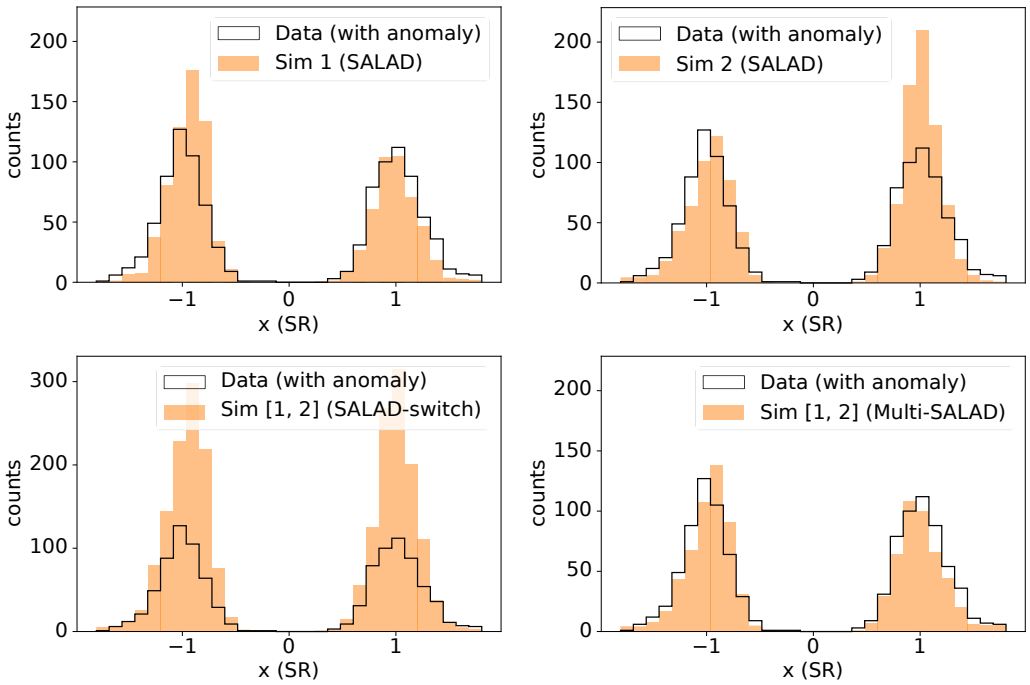

Figure 5. Top left: SALAD reweighting using simulation 1 on signal region. Top right: reweighting using simulation 2. Bottom left: using both simulation 1 and 2 weights separately. Bottom right: reweighting using simulation 1 and 2 combined.

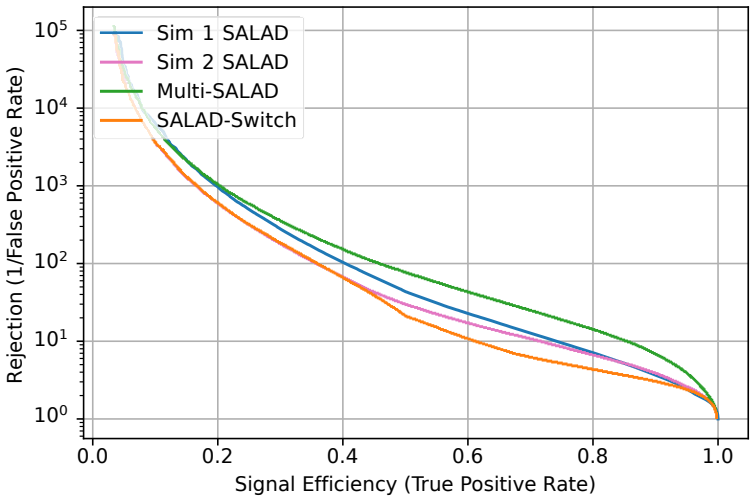

Figure 6. Signal eciency to rejection of Multi-SALAD versus other baselines (weighted and unweighted).

the background process. Future work includes 1) exploring Multi-SALAD's applicability on real data and algorithms for sampling from simulation datasets 2) extending Multi-

CWoLa to model more complex relationships among resonant features and 3) using such approaches together over multiple simulations and resonant features, eectively utilizing as much information as possible.

Acknowledgments

We thank David Shih and Jesse Thaler for useful discussions and comments about the manuscript. BN was supported by the Department of Energy, Oce of Science under contract number DE-AC02-05CH11231. FS is grateful for the support of the NSF under CCF2106707 and the Wisconsin Alumni Research Foundation (WARF). We gratefully acknowledge the support of NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF1763315 (Beyond Sparsity), CCF1563078 (Volume to Velocity), and 1937301 (RTML); ARL under No. W911NF-21-2-0251 (Interactive Human-AI Teaming); ONR under No. N000141712266 (Unifying Weak Supervision); ONR N00014-20-1-2480: Understanding and Applying Non-Euclidean Geometry in Machine Learning; N000142012275 (NEPTUNE); NXP, Xilinx, LETI-CEA, Intel, IBM, Microsoft, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, Google Cloud, Salesforce, Total, the HAI-GCP Cloud Credits for Research program, the Stanford Data Science Initiative (SDSI), and members of the Stanford DAWN project: Facebook, Google, and VMWare. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, ndings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reect the views, policies, or endorsements, either expressed or implied, of NIH, ONR, or the U.S. Government.

References

[1] HEP ML Community, \A Living Review of Machine Learning for Particle Physics."

[2] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman and D. Shih, Machine Learning in the Search for New Fundamental Physics, 2112.03769.

[3] G. Kasieczka et al., The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics, 2101.08320.

[4] T. Aarrestad et al., The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classication for the Large Hadron Collider, 2105.14027.

[5] ATLAS Collaboration, Dijet resonance search with weak supervision using 13 TeV pp collisions in the ATLAS detector, 2005.02983.

[6] ATLAS Collaboration collaboration, Anomaly detection search for new resonances decaying into a Higgs boson and a generic new particle X in hadronic nal states using $\sqrt{s}$ = 13 TeV pp collisions with the ATLAS detector, tech. rep., CERN, Geneva, 2022.

[7] G. Kasieczka, R. Mastandrea, V. Mikuni, B. Nachman, M. Pettee and D. Shih, Anomaly Detection under Coordinate Transformations, 2209.06225.

[8] G. Kasieczka, B. Nachman and D. Shih, New Methods and Datasets for Group Anomaly Detection From Fundamental Physics, ANDEA (2021) , [2107.02821].

[9] J. H. Collins, K. Howe and B. Nachman, Anomaly Detection for Resonant New Physics with Machine Learning, Phys. Rev. Lett. 121 (2018) 241803, [1805.02664].

[10] J. H. Collins, K. Howe and B. Nachman, Extending the search for new resonances with machine learning, Phys. Rev. D99 (2019) 014038, [1902.02634].

[11] R. T. D'Agnolo and A. Wulzer, Learning New Physics from a Machine, Phys. Rev. D99 (2019) 015014, [1806.02350].

[12] R. T. D'Agnolo, G. Grosso, M. Pierini, A. Wulzer and M. Zanetti, Learning Multivariate New Physics, 1912.12155.

[13] K. Benkendorfer, L. L. Pottier and B. Nachman, Simulation-Assisted Decorrelation for Resonant Anomaly Detection, 2009.02205.

[14] A. Andreassen, B. Nachman and D. Shih, Simulation Assisted Likelihood-free Anomaly Detection, Phys. Rev. D 101 (2020) 095004, [2001.05001].

[15] B. Nachman and D. Shih, Anomaly Detection with Density Estimation, Phys. Rev. D 101 (2020) 075042, [2001.04990].

[16] O. Amram and C. M. Suarez, Tag N' Train: A Technique to Train Improved Classiers on Unlabeled Data, 2002.12376.

[17] A. Hallin, J. Isaacson, G. Kasieczka, C. Krause, B. Nachman, T. Quadfasel et al., Classifying Anomalies THrough Outer Density Estimation (CATHODE), 2109.00546.

[18] J. A. Raine, S. Klein, D. Sengupta and T. Golling, CURTAINs for your Sliding Window: Constructing Unobserved Regions by Transforming Adjacent Intervals, 2203.09470.

[19] R. T. d'Agnolo, G. Grosso, M. Pierini, A. Wulzer and M. Zanetti, Learning New Physics from an Imperfect Machine, 2111.13633.

[20] P. Chakravarti, M. Kuusela, J. Lei and L. Wasserman, Model-Independent Detection of New Physics Signals Using Interpretable Semi-Supervised Classier Tests, 2102.07679.

[21] B. M. Dillon, R. Mastandrea and B. Nachman, Self-supervised Anomaly Detection for New Physics, 2205.10380.

[22] M. Letizia, G. Losapio, M. Rando, G. Grosso, A. Wulzer, M. Pierini et al., Learning new physics eciently with nonparametric methods, 2204.02317.

[23] K. Krzyzanska and B. Nachman, Simulation-based Anomaly Detection for Multileptons at the LHC, 2203.09601.

[24] S. Alvi, C. Bauer and B. Nachman, Quantum Anomaly Detection for Collider Physics, 2206.08391.

[25] T. Sjostrand, S. Mrenna and P. Z. Skands, A Brief Introduction to PYTHIA 8.1, Comput. Phys. Commun. 178 (2008) 852{867, [0710.3820].

[26] J. Bellm et al., Herwig 7.0/Herwig++ 3.0 release note, Eur. Phys. J. C 76 (2016) 196, [1512.01178].

[27] Sherpa collaboration, E. Bothmann et al., Event Generation with Sherpa 2.2, SciPost Phys. 7 (2019) 034, [1905.09127].

[28] E. M. Metodiev, B. Nachman and J. Thaler, Classication without labels: Learning from mixed samples in high energy physics, JHEP 10 (2017) 174, [1708.02949].

[29] J. Neyman and E. S. Pearson, On the problem of the most ecient tests of statistical hypotheses, Phil. Trans. R. Soc. Lond. A 231 (1933) 289.

[30] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu and C. Re, Snorkel: Rapid training data creation with weak supervision, in Proceedings of the 44th International Conference on Very Large Data Bases (VLDB), (Rio de Janeiro, Brazil), 2018.

[31] D. Y. Fu, M. F. Chen, F. Sala, S. M. Hooper, K. Fatahalian and C. R/'e, Fast and three-rious: Speeding up weak supervision with triplet methods, in International Conference on Machine Learning, 2020, https://arxiv.org/pdf/2002.11955.pdf.

[32] A. Ratner, C. D. Sa, S. Wu, D. Selsam and C. Re, Data programming: Creating large training sets, quickly, in Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, (Red Hook, NY, USA), p. 3574{3582, Curran Associates Inc., 2016.

[33] A. Ratner, B. Hancock, J. Dunnmon, F. Sala, S. Pandey and C. Re, Training complex models with multi-task weak supervision, in Proceedings of the AAAI Conference on Articial Intelligence, Jul, 2019.

[34] P. Varma, F. Sala, A. He, A. Ratner and C. Re, Learning dependency structures for weak supervision models, in Proceedings of the 36th International Conference on Machine Learning, 2019.

[35] M. J. Wainwright, M. I. Jordan et al., Graphical models, exponential families, and variational inference, Foundations and Trends® in Machine Learning 1 (2008) 1{305.

[36] B. Nachman and J. Thaler, E Pluribus Unum Ex Machina: Learning from Many Collider Events at Once, 2101.07263.

[37] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[38] M. Sugiyama, T. Suzuki and T. Kanamori, Density Ratio Estimation in Machine Learning. Cambridge University Press, 2012, 10.1017/CBO9781139035613.

[39] K. Cranmer, J. Pavez and G. Louppe, Approximating Likelihood Ratios with Calibrated Discriminative Classiers, 1506.02169.

[40] S. Dasgupta and P. M. Long, Boosting with diverse base classiers, in Learning Theory and Kernel Machines (B. Scholkopf and M. K. Warmuth, eds.), (Berlin, Heidelberg), pp. 273{287, Springer Berlin Heidelberg, 2003.

[41] C. Cortes, Y. Mansour and M. Mohri, Learning bounds for importance weighting, in Advances in Neural Information Processing Systems (J. Laerty, C. Williams, J. Shawe-Taylor, R. Zemel and A. Culotta, eds.), vol. 23, Curran Associates, Inc., 2010, https://proceedings.neurips.cc/paper/2010/le/59c33016884a62116be975a9bb8257e3-Paper.pdf.

[42] T. Ma, Lecture notes for machine learning theory (cs229m/stats214), June, 2022.

[43] P. L. Bartlett and S. Mendelson, Rademacher and gaussian complexities: Risk bounds and structural results, Journal of Machine Learning Research 3 (2002) 463{482.

[44] M. Mohri, A. Rostamizadeh and A. Talwalkar, Foundations of machine learning. MIT press, 2018.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., Pytorch: An imperative style, high-performance deep learning library, in Advances in Neural Information Processing Systems 32, pp. 8024{8035. Curran Associates, Inc., 2019.

[46] F. Chollet et al., Keras, 2015.

Appendix

We provide a glossary of notation in A. We provide algorithmic details for Multi-SALAD in Section B. We present additional theoretical results on Rademacher complexities and the asymptotic behavior of SALAD in Section C. In section D, we provide proofs for our theoretical results. In section E, we provide additional experimental details.

A Glossary

The glossary is given in Table 2.

B Additional Algorithmic Details

B.1 Multi-SALAD Algorithm

Multi-SALAD is described in Algorithm 2. We have simulation datasets $D_1^{sim}; \dots : D_k^{sim}$, where $D_i^{sim} = f(x_j; m_j) g_{j=1}^{n_{sim}}$ and all points belong to the background ($y = 0$). As discussed in Section 4, we propose using these simulation datasets by aggregating them into a single simulation dataset D^{sim} (whether it be with uniform or stratied sampling, etc.) Then the rest of this section proceeds as follows and is a review of the standard SALAD method.

Reweighting First, we learn weights to correct for the bias of the simulated background data. We split the both simulation and true data along m to produce sets $D_{SR}^{sim}; D_{SB}^{sim}$ and D_{SR} and D_{SB}. We train a classier over D^{sim}_{SB} and D_{SB} to distinguish between simulation and real data in the sideband region. That is, we train a binary classier $\hat{g}$ over points $(x; m; z)$ in the sideband where $x; m$ is either from $p_{sim}(jy = 0)(z = 0)$ or $p(jy = 0)(z = 1)$, where we recall that simulation data only contains $y = 0$, and no anomalies are present in the sideband. Denote q as the joint density of $(x; m; z)$. We dene the weight as the estimated likelihood ratio

$$\hat{w}(x; m) = \frac{\hat{g}(x; m)}{1 \quad \hat{g}(x; m)} \quad \frac{q(z = 1jx; m)}{q(z = 0jx; m)} = \frac{q(x; mjz = 1)}{q(x; mjz = 0)} \quad \frac{q(z = 1)}{q(z = 0)}$$
$$= \frac{q(x; mjz = 1)}{q(x; mjz = 0)} = \frac{p(x; mjy = 0)}{p_{sim}(x; mjy = 0)}:$$

Here, we assume that $q(z = 1) = q(z = 0)$ (i.e. balanced simulation and real dataset, which we can always ensure by generating more or less simulation data). Equality is obtained in the expression above when $\hat{g}$ is Bayes-optimal.

Training The above $\hat{w}(x; m)$ is dened on the sideband region. Next, we interpolate and correct the bias of the simulation in the signal region. Let D_{SR}^{sim} be the set of simulation data in the signal region of size n_{sim}^{SR}, and let D_{SR} be the set of true data in the signal region of size n_{data}^{SR}, for a total of n^{SR} points. We train a classier h to distinguish between the reweighted simulated data, which approximates true background data, and the true data. In particular, the loss function used is

$$\hat{L}_S(h; \hat{w}) = \frac{1}{n^{SR}} \sum_{x \in D_{SR}} \log h(x; m) + \sum_{x \in D_{sim}^{SR}} \hat{w}(x; m) \log(1 \quad h(x; m)) \quad : \qquad (B.1)$$

Algorithm 2 Multi-SALAD

1: Input: Simulation datasets $D_1^{sim}, \ldots, D_k^{sim}$ and real dataset D.
2: Construct overall simulation dataset $D^{sim} = \bigcup_{i=1}^{k} D_i^{sim}$.
3: Split each dataset into signal region and sideband region using resonant feature m to get $\{D_{SR}^{sim}, D_{SB}^{sim}\}$ and $\{D_{SR}, D_{SB}\}$.
4: Learn weight $\hat{w}(x, m) = \frac{\hat{g}(x,m)}{1 - \hat{g}(x,m)}$, where $\hat{g}$ is a classier that distinguishes data D_{SB} from simulation D_{SB}^{sim} in the sideband region.
5: Train a new classier $\hat{h}$ on the signal region to distinguish between points in D_{SR} and points in D_{SR}^{sim} reweighted by $\hat{w}$, using the following loss:

$$\hat{L}_S(h; \hat{w}) = \frac{1}{n} \sum_{x \in D_{SR}} \log h(x, m) + \sum_{x \in D_{SR}^{sim}} \hat{w}(x, m) \log(1 - h(x, m)). \quad (B.2)$$

6: Output: Classier output $\hat{h}(x, m)$, which yields a score that is thresholded for anomaly detection.

In expectation with an optimal w, we can see that minimizing this loss is equivalent to minimizing the cross-entropy loss on a task that distinguishes between points drawn from p and points drawn from $p(y \mid m = 0)$ in the signal region. Therefore, h can be used for anomaly detection. The procedure is summarized in Algorithm 2.

C Additional Theoretical Results

C.1 The Need for 3 Resonant Features

We show that to identify the model (3.3), we need at least $k = 3$ resonant features.

Lemma 1. If $k = 1$ or $k = 2$ in model (3.3), the parameters α_1 and α_2 cannot be recovered from the observable quantities.

Proof. The strategy we use to show that the model cannot be identied for $k = 1$ or $k = 2$ is to prove that the observable distributions $P(M_1^f(m), \ldots, M_k^f(m))$ are consistent with multiple values of α. We do so by direct calculation.

First, consider the case of $k = 1$. Set $\alpha_y = 0$ for simplicity. Then, the model is $\frac{1}{Z}\exp(M_1^f(m)\alpha)$. Then $Z = 2\exp(\alpha) + 2\exp(-\alpha)$, and

$$P(M_1^f(m) = 1) = \frac{\exp(\alpha) + \exp(-\alpha)}{2\exp(\alpha) + 2\exp(-\alpha)} = \frac{1}{2}.$$

Thus, any α value produces the same observable distribution, so that we cannot identify α.

Next, we consider $k = 2$. Again, set $\alpha_y = 0$. The model is now $\frac{1}{Z}\exp(\alpha_1 M_1^f(m)\alpha + \alpha_2 M_2^f(m)\alpha)$. We similarly compute

$$Z = 2(\exp(\alpha_1 + \alpha_2) + \exp(-\alpha_1 + \alpha_2) + \exp(\alpha_1 - \alpha_2) + \exp(-\alpha_1 - \alpha_2)).$$

The observable distribution is now $P(\hat{M}_1(m); \hat{M}_2(m))$. We have that

$$P(\hat{M}_1(m) = 1; \hat{M}_2(m) = 1) = \frac{1}{Z}(\exp(\theta_1 + \theta_2) + \exp(-\theta_1 - \theta_2));$$

and

$$P(\hat{M}_1(m) = 1; \hat{M}_2(m) = -1) = \frac{1}{Z}(\exp(\theta_1 - \theta_2) + \exp(-\theta_1 + \theta_2)):$$

Note that we have $P(\hat{M}_1(m) = -1; \hat{M}_2(m) = -1) = P(\hat{M}_1(m) = 1; \hat{M}_2(m) = 1)$ and $P(\hat{M}_1(m) = -1; \hat{M}_2(m) = 1) = P(\hat{M}_1(m) = 1; \hat{M}_2(m) = -1)$.

As a result, we have the same distribution $P(M_1(m); M_2(m))$ for the parameters $\theta_1; \theta_2 = a; b$ and for $\theta_1; \theta_2 = b; a$, where $a; b$ are some non-negative values. If $a = b$, we end up with at least two solutions that cannot be distinguished, completing the proof. □

C.2 Rademacher Complexity Bounds

We present bounds on the Rademacher complexity $R_n(F)$ of various models F. For all of the F below, we obtain $R_n(\ell \circ F)$ by computing $R_n(F)$. These two Rademacher complexities are equal when we assume that ℓ is 1-Lipschitz and apply Talagrand's lemma.

- Linear models: We define $f(x) = \theta^\top x$ with $\|\theta\|_2 \le B$ and $E[\|x\|_2^2] \le C^2$, $R_n(F) \le \frac{BC}{\sqrt{n}}$ [42, Theorem 5.5].
- Two-layer feed-forward neural networks (MLPs): We define $f(x)$ where $\theta = (U; w)$ are the parameters for the weights for the two layers of an MLP. Here $U \in \mathbb{R}^{md}$ and $w \in \mathbb{R}^m$. Suppose ReLU is the activation function, $\|w\|_2 \le B_w$, $\|u_i\|_2 \le B_u$ for all $1 \le i \le m$, and that $E[\|x\|_2] \le C^2$. Then, $R_n(F) \le 2B_w B_u C \sqrt{\frac{m}{n}}$ [42, Theorem 5.9].
- Kernels: Let $k : X \times X \to \mathbb{R}$ be a continuous symmetric function so that for $x_1; \ldots; x_n$, the matrix given by $K_{ij} = k(x_i; x_j)$ is positive semidefinite. The class of kernel estimators consists of functions $f(x) = \sum_{i=1}^n \alpha_i k(X_i; x)$. Suppose that $\sum_{i,j} \alpha_i \alpha_j k(X_i; X_j) \le B^2$; then, from [43], $R_n(F) \le 2B\sqrt{\frac{E[k(X;X)]}{n}}$. For particular kernels it is easy to bound the term in the numerator above. For example, we consider the RBF kernel which has maximum one, yielding $R_n(F) \le \frac{2B}{\sqrt{n}}$.

C.3 Asymptotic behavior of SALAD's $\hat{L}_S(h; w)$

Lemma 2. Assume that the reweighting function is Bayes-optimal, meaning that $w^*(x; m) = w(x; m)$. Then,

$$\lim_{n \to \infty} \hat{L}_S(h; w^*) \to L_{CE}(h);$$

where $L_{CE}(h) = E_{x;m;z'=1}[-\log h(x; m)] + E_{x;m;z'=0}[-\log(1 - h(x; m))]$ is the cross entropy loss on label $z' = \begin{cases} 1 & x; m \sim P \\ 0 & x; m \sim p(y = 0) \end{cases}$.

Proof. Let n_{data}^{SR} be the number of points from D that belong to the signal region. Under our assumptions, the empirical loss function can be written as

$$\hat{L}(h; w) \propto \frac{n_{data}^{SR}}{n^{SR}} \frac{1}{n_{data}^{SR}} \sum_{x \in D_{SR}} \log h(x; m) + \frac{n_{sim}^{SR}}{n^{SR}} \frac{1}{n_{sim}^{SR}} \sum_{x \in D_{SR}^{sim}} \frac{p(x; m|y = 0)}{p_{sim}(x; m|y = 0)} \log(1 - h(x; m)).$$

As $n^{SR} \to \infty$, the first term approaches $\Pr(z^0 = 1) E_{x;m \sim P}[\log h(x; m)] = \Pr(z^0 = 1) E_{x;m|z^0=1}[\log h(x; m)]$. For the second term, we can construct $n_{data}^{SR;0}$, the amount of data where x is from $p(x|y = 0)$, to be equal to n_{sim}^{SR} such that the expression asymptotically approaches $\Pr(z^0 = 0) E_{x;m \sim P_{sim}}\left[\frac{p(x;m|y=0)}{p_{sim}(x;m|y=0)} \log(1 - h(x; m))\right]$. Performing a change of expectation, this is equal to $\Pr(z^0 = 0) E_{x;m|z^0=0}[\log(1 - h(x; m))]$. Putting this together, we have that

$$\lim_{n^{SR} \to \infty} \hat{L}(h; w) \propto \Pr(z^0 = 1) E_{x;m|z^0=1}[\log h(x; m)] - \Pr(z^0 = 0) E_{x;m|z^0=0}[\log(1 - h(x; m))] = L_{CE}(h).$$

□

D Proofs

D.1 Proof of Theorem 1

Proof. From Theorem 3 of [31], we have that $L_C(\hat{f}) - L_C(f^*)$ is bounded by the traditional ERM generalization gap of $L_C(f) - L_C(f^*)$, where $f = \operatorname{argmin}_{f \in F} \frac{1}{n} \sum_{i=1}^{m} \ell(f(x_i; m_i); y_i)$ is the classier learned on labeled data, plus the term $\frac{c_1}{e_{min} a_{min}^5} \sqrt{\frac{k}{n} + \frac{c_2 k}{\rho n}}$.

We can apply standard learning theory bounds on $L_C f) - L_C(f^*)$. In particular, this quantity is equal to

$$\begin{aligned} L_C(f) - L_C(f^*) &= (L_C(f) - \hat{L}_C(f)) + (\hat{L}_C(f) - \hat{L}_C(f^*)) + (\hat{L}_C(f^*) - L_C(f^*)) \\ &\leq L_C(f) - L_C(\hat{f}) + L_C(\hat{f}^*) - L_C(f^*) \\ &\leq 2 \sup_{f \in F} |L_C(f) - \hat{L}_C(f)|; \end{aligned}$$

where we have used the fact that $\hat{L}_C(f) \leq L_C(\hat{f}^*)$. Then, using uniform convergence bounds, such as Theorem 3.3 of [44], we have

$$L_C(f) - L_C(f^*) \leq 2 R_n(\ell \circ F) + \sqrt{\frac{\log 2/\delta}{2n}}$$

This gives us our desired result.

□

D.2 Proof of Theorem 2

Proof. We define the true (cross-entropy) loss as

$$L_S(h; w) = \Pr(z^0 = 1) E_{z^0=1}[\log h(x; m)] \quad \Pr(z^0 = 0) E_{x;m \in P^{SR}_{sim}}[w(x; m) \log(1 \quad h(x; m))];$$

where $z_0 = 1$ for $x; m \in P$ and 0 for $x; m \in P(jy = 0)$. Next, define $w(x; m) = \frac{q(x;mjz=1)}{q(x;mjz=0)}$ and let $\hat{w}$ be the weight ratio learned by our model. Let $\hat{h} = \operatorname{argmin}_{h \in H} \hat{L}_S(h; \hat{w})$, and let $h^? = \operatorname{argmin}_{h \in H} L(h; w^?)$. Intuitively, $h^?$ corresponds to the true difference between P^{SR}_{data} and $P^{SR}_{data}(jy = 0)$. We can first decompose the generalization error as

$$L_S(\hat{h}; \hat{w}) \quad L_S(h^?; w) = [L_S(\hat{h}; \hat{w}) \quad \hat{L}_S(\hat{h}; \hat{w})] + [\hat{L}_S(\hat{h}; \hat{w}) \quad \hat{L}_S(h^?; \hat{w})] \tag{D.1}$$

$$+ [\hat{L}_S(h^?; \hat{w}) \quad \hat{L}_S(h^?; w)] + [\hat{L}_S(h^?; w) \quad L_S(h^?; w)]: \tag{D.2}$$

We know that $\hat{L}_S(\hat{h}; \hat{w}) \quad L_S(h^?; \hat{w})$, so

$$L_S(\hat{h}; \hat{w}) \quad L_S(h^?; w) \quad jL_S(h; \hat{w}) \quad \hat{L}_S(\hat{h}; \hat{w})j + j\hat{L}_S(h^?; w) \quad L_S(h^?; w)j$$

$$+ \hat{L}_S(h^?; \hat{w}) \quad \hat{L}_S(h^?; w)$$

$$\sup_{h;w} jL_S(h; w) \quad \hat{L}_S(h; w)j + j\hat{L}_S(h^?; w) \quad L_S(h^?; w)j + \hat{L}_S(h^?; \hat{w}) \quad \hat{L}_S(h^?; w):$$

We first bound $\sup_{h;w} jL_S(h; w) \quad \hat{L}_S(h; w)j$. For notation, we rewrite $L_S(h; w)$ as $L_S(h; g)$, where $w(x; m) = \frac{g(x;m)}{1 \quad g(x;m)}$ and g belongs to some function class G. Then, using Theorem 3.3 from [44], we get that $\sup_{h;w} jL_S(h; w) \quad \hat{L}_S(h; w)j \quad 2R_{n^{SR}}(`_S fH; Ggg) + \sqrt{\frac{\log 1=}{2n^{SR}}}$ with probability at least $1 \quad$, where $`_S fH; Ggg$ is defined as satisfying $`_S(h(x; m); g(x; m); y) = \quad y \log h(x; m) \quad (1 \quad y) \frac{g(x;m)}{1 \quad g(x;m)} \log(1 \quad h(x; m))$ for $h \in H; g \in G$.

Next, we bound $j\hat{L}_S(h^?; w) \quad L_S(h^?; w)j$. Let $W = \max w(x; m) < 1$ be the maximum density ratio, and let $B_1 = \max_{x;m} f \log h^?(x; m); \log(1 \quad h^?(x; m))g$. Assume that $B_1 < 1$. We can apply standard concentration inequalities here (Hoeding) to get that $j\hat{L}_S(h^?; w) \quad L_S(h^?; w)j \quad W B_1 \sqrt{\frac{\log 2=}{2n^{SR}}}$ with probability at least $1 \quad$.

Finally, we bound $\hat{L}_S(h^?; \hat{w}) \quad \hat{L}_S(h^?; w)$. We can write $\hat{L}_S(h^?; \hat{w}) \quad \hat{L}_S(h^?; w)$ as

$$\hat{L}_S(h^?; \hat{w}) \quad \hat{L}_S(h^?; w) = \frac{1}{n^{SR}} \sum_{x \in D^{SR}_{sim}} (\hat{w}(x; m) \quad w(x; m)) (\quad \log(1 \quad h^?(x; m))): \tag{D.3}$$

Define $= \max(\quad \log(1 \quad h^?(x; m))) \quad 0$ for $x; m \in D^{sim}_{SR}$, which is small as long as $h^?(x; m)$ sufficiently classifies x and is hence a property of how separated the reweighted simulation and true data is. Then,

$$j\hat{L}_S(h^?; \hat{w}) \quad \hat{L}_S(h^?; w)j \quad \frac{SR}{n} \sum_{x;m \in D^{sim}_{SR}} j\hat{w}(x; m) \quad w(x; m)j: \tag{D.4}$$

Recall that $\hat{w}(x; m) = \frac{\hat{g}(x;m)}{1 \quad \hat{g}(x;m)}$ and $w(x; m) = \frac{g^?(x;m)}{1 \quad g^?(x;m)}$ where $g^?(x; m) = \Pr(z = 1jx; m)$, so $j\hat{w}(x; m) \quad w(x; m)j = \frac{j\hat{g}(x;m) \quad g^?(x;m)j}{(1 \quad \hat{g}(x;m))(1 \quad g^?(x;m))}$. This denominator is greater than

$(1 - \hat{g}_{max})(1 - g^*_{max})$. Then,

$$|\hat{L}_S(h^*; w) - \hat{L}_S(\hat{h}; w)| \leq \frac{1}{(1 - \hat{g}_{max})(1 - g^*_{max}) n^{SR}} \sum_{x,m \in D^{sim}_{SR}} |\hat{g}(x,m) - g^*(x,m)|. \quad (D.5)$$

We now look at the classier for training g. The per-point cross entropy loss for $(x; m; z)$ is $\ell(g(x; m); z) = -\log g(x; m)$ for $z = 1$ and $-\log(1 - g(x; m))$ for $z = 0$. WLOG, assume for some x and m, $g^*(x; m) > \hat{g}(x; m)$. Then $|\ell(g^*(x; m); 1) - \ell(\hat{g}(x; m); 1)| = \log \frac{g^*(x;m)}{\hat{g}(x;m)} = \log\left(1 + \frac{g^*(x;m)}{\hat{g}(x;m)} - 1\right) \leq \frac{g^*(x;m)/\hat{g}(x;m) - 1}{g^*(x;m)/\hat{g}(x;m)} = \frac{g^*(x;m) - \hat{g}(x;m)}{g^*(x;m)} \leq |g^*(x; m) - \hat{g}(x; m)|$ and $|\ell(g^*(x; m); 0) - \ell(\hat{g}(x; m); 0)| = \log \frac{1-\hat{g}(x;m)}{1-g^*(x;m)} = \log\left(1 + \frac{\hat{g}(x;m) - g^*(x;m)}{1-g^*(x;m)}\right) \leq \frac{\hat{g}(x;m) - g^*(x;m)}{(1-g^*(x;m))/(1-\hat{g}(x;m))} \leq \frac{g^*(x;m) - \hat{g}(x;m)}{1 - g(x;m)} \leq |g^*(x; m) - \hat{g}(x; m)|$, where we use the inequality $\log(1 + x) \leq \frac{x}{1+x}$ for $x > -1$. Therefore, with probability $1 - \delta$,

$$|\hat{L}_S(h^*; w) - \hat{L}_S(\hat{h}; w)| \leq \frac{1}{(1 - \hat{g}_{max})(1 - g^*_{max}) n^{SR}} \sum_{x,m \in S_R} |\ell(\hat{g}(x; m); z) - \ell(g^*(x; m); z)|$$

$$\leq (1 - \hat{g}_{max})(1 - g^*_{max}) n^{SR} E[|\ell(\hat{g}(x; m); z) - \ell(g^*(x; m); z)|] + B_2 \sqrt{\frac{\log 2/\delta}{2n^{SR}_{sim}}},$$

where $B_2 = \max_{x,y} |\ell(\hat{g}(x; m); z), \ell(g^*(x; m); z)| = -\log(\min\{\hat{g}_{min}, g^*_{min}\})$. We assume that B_2 is nite, so there exists a constant c such that

$$|\hat{L}_S(h^*; w) - \hat{L}_S(\hat{h}; w)| \leq \frac{n^{SR}_{sim}}{(1 - \hat{g}_{max})(1 - g^*_{max}) n^{SR}} c|L(\hat{g}) - L(g^*)| + B_2 \sqrt{\frac{\log 2/\delta}{2n^{SR}_{sim}}},$$

where $L(g) = E_{x,m \in S_R}[\ell(g(x; m); z)]$. Since $g^*(x; m)$ is Bayes optimal, $|L(\hat{g}) - L(g^*)| = L(\hat{g}) - L(g^*) = L(\hat{g}) - \hat{L}(\hat{g}) + \hat{L}(\hat{g}) - \hat{L}(g^*) + \hat{L}(g^*) - L(g^*) \leq L(\hat{g}) - \hat{L}(\hat{g}) + \hat{L}(g^*) - L(g^*) \leq 2 \sup_{g \in G} |L(g) - \hat{L}(g)|$. From Theorem 3.3 in [44], this is bounded by $2R_{n^{SB}}(\ell \circ G) + \sqrt{\frac{\log 1/\delta}{2n^{SB}}}$ with probability at least $1 - \delta$. Then, applying a union bound, with probability $1 - \delta$, we have

$$|\hat{L}_S(h^*; w) - \hat{L}_S(\hat{h}; w)| \leq \frac{n^{SR}_{sim}}{(1 - \hat{g}_{max})(1 - g^*_{max}) n^{SR}} 4cR_{n^{SB}}(\ell \circ G) + 2c \sqrt{\frac{\log 2/\delta}{2n^{SB}}} + B_2 \sqrt{\frac{\log 4/\delta}{2n^{SR}_{sim}}}.$$

Putting everything together with another union bound, with probability $1 - \delta$, the generalization error is at most

$$L_S(\hat{h}; w) - L_S(h^*; w) \leq 2R_{n^{SR}}(\ell \circ \{f \in H; G\}) + (1 + W B_1) \sqrt{\frac{\log 8/\delta}{2n^{SR}}} \quad (D.6)$$

$$+ \frac{n^{SR}_{sim}}{(1 - \hat{g}_{max})(1 - g^*_{max}) n^{SR}} 4cR_{n^{SB}}(\ell \circ G) + 2c \sqrt{\frac{\log 1/\delta}{2n^{SB}}} + B_2 \sqrt{\frac{\log 8/\delta}{2n^{SR}_{sim}}}. \quad (D.7)$$

□

E Experiment Details

E.1 Multi-CWoLa Experiments

For the Multi-CWoLa experiment, we used the anomaly and simulation data from the Pythia 8 simulations in the LHC Olympics Dataset to create an unlabeled dataset we want to perform anomaly detection on [3]. We have k = 3, and construct $M_i(m)$ based on the thresholds [[3:3; 3:7]; [0:09; 0:13]; [0:3; 0:35]] on the rst three features. For standard CWoLa, only the third feature is regarded as the resonant feature, and it is thresholded with the interval [0:3; 0:35]. We constructed training datasets of varying sizes with class balance Pr(y = 1) = 0:149. We used one test dataset with 65755 randomly sampled anomaly points and 161658 randomly sampled background points.

All methods were trained using scikit-learn's MLPClassier with max_iter=5000. For Multi-CWoLa's weak supervision step, we learn the parameters of the graphical model using SGD and PyTorch [45] with class balance Pr(y = 1) = 0:25, 30000 epochs, and learning rate = 1e 6.

E.2 Multi-SALAD Experiments

Setup We use MLPs from Keras [46], each with 3 hidden layers of dimension 32, ReLu activation, and trained with cross-entropy loss and the Adam optimizer. We train for 50 epochs, batch size 200, and default parameters otherwise. Finally, we evaluate our approach on a new test set containing 200000 background points and 200000 anomaly points. This test set is used to produce the signal eciency to rejection rate. All experiments were run on a personal laptop.

Additional Results In Figure 7, we show our results on individual runs. This is because computing the condence intervals of these curves averaged across the 10 random runs is too noisy due to the magnitude of the reciprocal 1/FPR.

Symbol	Used for
x	Discriminative feature $x \in \mathcal{X}$.
m	Resonant feature vector of length k, $m = [m^1, \ldots, m^k] \in \mathbb{R}^k$.
y	True unknown label $y \in \mathcal{Y} = \{0, +1\}$, where 0 is background and 1 is signal.
$P; p$	Distribution and density of data (x, m, y).
I_{m^i}	Interval along which ith resonant feature m^i is thresholded to produce signal region and sideband.
$SR; SB$	Signal region and sideband. For an interval I_{m^i}, $SR_i = \{(x, m) : m^i \in I_{m^i}\}$ and $SB_i = \{(x, m) : m^i \notin I_{m^i}\}$.
f	Classier $f : \mathcal{X} \to \mathcal{Y}$ used for anomaly detection.
D	Unlabeled dataset $D = \{(x_i, m_i)\}_{i=1}^n$ of discriminative and resonant features.
$D_{SR}; D_{SB}$	Signal region and sideband of D, $D_{SR} = D \setminus SR$, $D_{SB} = D \setminus SB$.
$\alpha_{SR}; \alpha_{SB}$	Mixture weights corresponding to $p(y = 1 \mid x \in SR)$ and $p(y = 1 \mid x \in SB)$. It is assumed that $\alpha_{SR} > \alpha_{SB}$.
$M_i(m)$	Noisy membership label for the ith resonant feature, equal to 0 if $x \in D_{SB_i}$ and 1 if $x \in D_{SR_i}$. $M(m) = M_1(m), \ldots, M_k(m)$.
$\hat{y}_{y,i}$	Weak label drawn from estimated distribution on $p(y \mid M(m))$.
$\eta_{y,i}$	Canonical parameters of graphical model on $y, M(m)$ in (3.3).
Z	η_y scales with the class balance of y and η_i scales with the accuracy of $M_i(m)$.
$\eta_{y,M(m)}$	Partition function used for normalizing distribution $p(y, M(m))$ in (3.3).
$\tilde{y}; \tilde{M}(m)$	y and $M(m)$ scaled from $\{0, 1\}$ to $\{-1, 1\}$.
η_i	Accuracy parameter $\eta_i = p(M_i(m) = 1 \mid y = 1)$ for the membership label of the ith resonant feature.
ℓ_C	Loss function $\ell_C : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ for training classier f. Expected
f^*	loss on labeled data using f, $L_C(f) = \mathbb{E}[\ell_C(f(x), y)]$.
$L_C(f)$	Optimal classier trained on innite labeled data, $f^* = \operatorname{argmin}_{f \in \mathcal{F}} L_C(f)$.
$\hat{f}$	Empirical loss on D with weak labels using f, $\hat{L}_C(f) = \frac{1}{n} \sum_{i=1}^n \ell_C(f(x_i), \hat{y}_i)$.
D^{sim}	Classier learned using Multi-CWoLa, $f \triangleq \operatorname{argmin}_{f \in \mathcal{F}} \hat{L}_C(f)$.
$D_{SB}^{sim}, D_{SR}^{sim}$	Simulation dataset used in standard SALAD, $D^{sim} = \{(x_i, m_i)\}_{i=1}^{n_{sim}}$.
$w(x, m)$	Has distribution P_{sim} and density $p_{sim}()$.
$\hat{g}$	$D_{SB}^{sim} = D^{sim} \setminus SB$, $D_{SR}^{sim} = D^{sim} \setminus SR$.
$L_S(h, w)$	Density ratio between D_{SB}^{sim} and D_{SB} used for reweighting, $w(x, m) = \frac{p(x, m \mid y=0)}{p_{sim}(x, m \mid y=0)}$.
$\hat{h}$	Classier trained to classify D_{SB}^{sim} vs D_{SB}, used for approximating $w(x, m)$ when $\lvert D_{SB}^{sim} \rvert = \lvert D_{SB} \rvert$.
$D_1^{sim}, \ldots, D_k^{sim}$	Cross-entropy loss function used to classify D_{SR}^{sim} reweighted with w vs D_{SR}.
D_{sim}	Classier trained using L_S.
n^{SR}	k multiple simulation datasets used in Multi-SALAD.
n^{SB}	Dataset aggregated from $D_1^{sim}, \ldots, D_k^{sim}$.
n_{sim}^{SR}	$n^{SR} = \lvert D_{SR} \rvert$.
h^*	$n^{SB} = \lvert D_{SB} \rvert$.
W	$n_{sim}^{SR} = \lvert D_{SR}^{sim} \rvert$.
	The optimal classier $h^* = \operatorname{argmin}_{h \in \mathcal{H}} L_S(h, w)$.
	The maximum ratio between the simulation and true background, $W = \max_{x,m} w(x, m)$.

Table 2. Glossary of variables and symbols used in this paper.

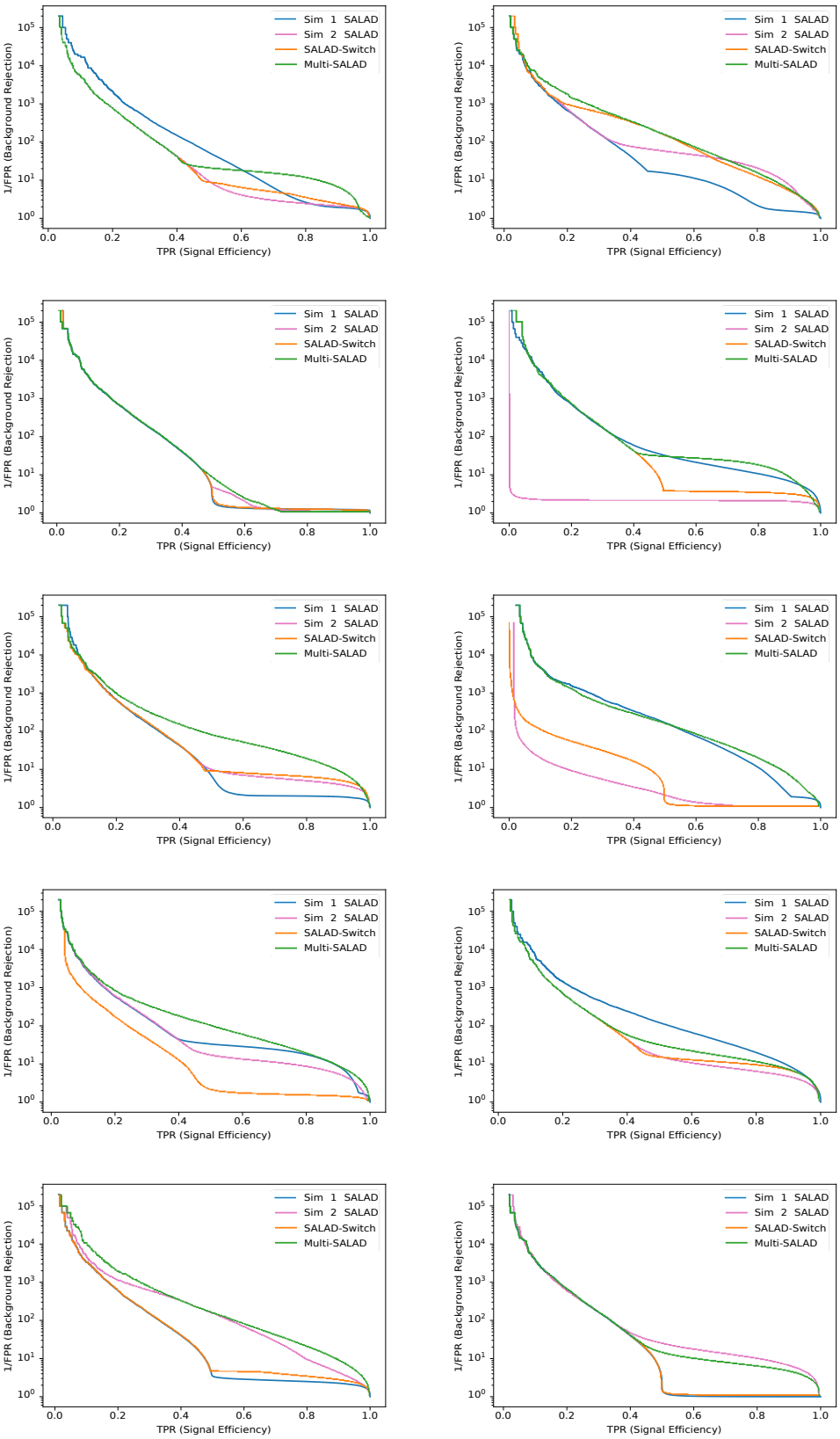

Figure 7. Results on individual runs.