Anomaly Detection with Multiple Reference Datasets

Mayee F. Chen
Department of Computer Science
Stanford University
Stanford, CA 94305
mfchen@stanford.edu

Benjamin P. Nachman
Physics Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
bpnachman@lbl.gov

Frederic Sala
Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706
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 Classification 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 real and synthetic data. As an added benefit, our generalizations enable us to provide finite-sample guarantees, improving on existing asymptotic analyses.

1 Introduction

Due to the vast parameter space of Standard Model extensions and to the lack of significant 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. [11]) and the first results with collision data are now available [3, 1]. One way to characterize AD methods is based on their physics assumption of the new phenomena [12]. Strategies that assume the new physics is "rare" [14] estimate (explicitly or implicitly) the data 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*, where one reasonant 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 defined using one reference dataset and one target dataset. 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 [13]. Second, for the same dataset, multiple, independent Standard Model simulators can produce a reference dataset (e.g. Pythia [21] and Herwig [4]). Using multiple reference datasets may improve performance, but it is

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

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 Classification Without Labels (CWoLa) [16, 6, 7], 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 different 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 an overall prediction. We demonstrate MULTI-CWOLA's performance on the LHS Olympics Dataset [13].

Second, we study Simulation Assisted Likelihood-free Anomaly Detection (SALAD) [2]. 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 different 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 finding that combining data helps when each simulator approximates different components of the background well. We demonstrate MULTI-SALAD's performance on synthetic data.

Finally, we study the finite sample guarantees of our proposed methods. Many resonant AD methods have optimality guarantees in some asymptotic limit, but there is no first-principles understanding of the methods' performance with finite samples. 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 finite sample properties of AD and related methods.

2 Background

We have an input space of *discriminating features* $x \in \mathcal{X}$ and k *resonant features* $m = [m^1, \ldots, m^k] \in \mathbb{R}^k$. Associated with a point (x, m) is an unknown label $y \in \mathcal{Y}$ for $\mathcal{Y} = \{0, 1\}$ (background vs. signal). Points (x, m, y) are drawn from a distribution $\mathcal{P}$ with density $p(\cdot)$. For a resonant feature $m^i \in \mathbb{R}$, an interval $\mathcal{I}_{m^i} \in \mathbb{R}$ is used to define a signal region $SR = \{(x, m) : m^i \in \mathcal{I}_{m^i}\}$ and a sideband region $SB = \{(x, m) : m^i \notin \mathcal{I}_{m^i}\}$. 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 $f : \mathcal{X} \to \mathcal{Y}$, outputting a value $\hat{y}$ given (x, m), 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 [19, 10] .

Standard CWOLA We have one unlabeled dataset $\mathcal{D} = \{(x_i, m_i)\}_{i=1}^n$ with one resonant feature $(k = 1)$ that we want to use to learn f. We use m to construct the signal and sideband regions, $\mathcal{D}_{SR}, \mathcal{D}_{SB} \subset \mathcal{D}$, where $\mathcal{D}_{SR} = \mathcal{D} \cap SR$ and $\mathcal{D}_{SB} = \mathcal{D} \cap SB$, 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 $0 \leq \eta_{SR}, \eta_{SB} \leq 1$: $p_{SR}(x) = \eta_{SR} p(x|y = 1) + (1 - \eta_{SR}) p(x|y = 0)$, and $p_{SB}(x) = \eta_{SB} p(x|y = 1) + (1 - \eta_{SB}) p(x|y = 0)$.

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 $\frac{p_{SR}(x)}{p_{SB}(x)} = \frac{\eta_{SR} r(x) + 1 - \eta_{SR}}{\eta_{SB} r(x) + 1 - \eta_{SB}}$. Assuming $\eta_{SR} > \eta_{SB}$ (e.g. more signal in the signal region), the mixture ratio is monotonically increasing in $r(x)$. Therefore, we train a classifier f to learn $\frac{p_{SR}(x)}{p_{SB}(x)}$ by distinguishing between $\mathcal{D}_{SR}$ and $\mathcal{D}_{SB}$, and this f provides information about $r(x)$ and thus can be used for anomaly detection.

MULTI-CWOLA Method Intuitively, CWOLA uses the resonant feature m as a noisy label that identifies the signal vs sideband region and then trains a classifier using these through the mixture component membership. 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 [19, 20, 10].

In our approach, we split $\mathcal{D}$ along each resonant feature m^i to produce multiple pairs of datasets $\mathcal{D}_{SB_i}$ and $\mathcal{D}_{SR_i}$ for each $i \in [k]$ based on membership in I_{m^i}. A straightforward way to use all datasets $(\mathcal{D}_{SB_1}, \mathcal{D}_{SR_1}), \ldots, (\mathcal{D}_{SB_k}, \mathcal{D}_{SR_k})$ is to apply standard CWoLa k times to train k classifiers that we can then ensemble. Instead, in MULTI-CWOLA, we directly aggregate the noisy membership labels provided by thresholding resonant features and training a model on the aggregated label. Specifically, suppose that the mixture membership for a datapoint (x, m) is given by the label $M_i(m) \in \{0, 1\}$, where $M_i(m) = 0$ if $x \in \mathcal{D}_{SB_i}$ and 1 otherwise. To aggregate all noisy labels $\mathbf{M}(m) = \{M_1(m), \ldots, M_k(m)\}$, we draw on weak supervision, a recent class of techniques that combines the outputs of multiple weak sources using learned weight parameters and produces estimates on the true label without requiring any labeled data. That is, we can aggregate $\mathbf{M}(m)$ according to each resonant feature's quality, without needing to know the true presence of an anomaly. In particular, we first learn the probabilistic graphical model for the binary distribution $p(y|\mathbf{M}(m))$ [10]. Since y is unknown, we learn the parameters of the distribution using latent variable estimation on $\mathcal{D}$. Then, weak labels $\hat{y}$ are produced from our estimated distribution $\hat{p}(y|\mathbf{M}(m))$ for $x \in \mathcal{D}$. Finally, an end model $\hat{f}$ is trained with $(x, \hat{y})$ to perform anomaly detection. See Algorithm 1 in Appendix C for explicit algorithm statements.

Theoretical Results Assuming $p(y, \mathbf{M}(m))$ can be parametrized as a class of graphical models (see Equation (1)), MULTI-CWOLA offers *finite-sample generalization* guarantees. Suppose the downstream model $\hat{f}$ trained on $\hat{y}$ belongs to class $\mathcal{F}$. Define a loss function $\ell_C : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and let the expected loss of f be $L_C(f) := \mathbb{E}_{\mathcal{P}_{\text{data}}}[\ell_C(f(x), y)]$ on true labels. Then, the optimal classifier is $f^\star = \text{argmin}_{f \in \mathcal{F}} L_C(f)$, 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 classifier we learn is $\hat{f} = \text{argmin}_{f \in \mathcal{F}} \hat{L}_C(f)$. Note that this construction is different 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* $L_C(\hat{f}) - L_C(f^\star)$. In Theorem 1 (Appendix D), we present a bound on $L_C(\hat{f}) - L_C(f^\star)$. We find that there are four quantities controlling the bound: 1) the *Rademacher complexity* of $\mathcal{F}$, which describes the model's expressivity and can be readily computed for a variety of classes (e.g., decision trees, two-layer feedforward networks); 2) the noise from learning with n points; 3) the noise from using $\hat{y}$ rather than true y; and 4) the gap between $p(y|m)$ and $p(y|\mathbf{M}(m))$ (e.g. how much we lose about the resonant features by only modeling mixture membership). Surprisingly, we find that 2) and 3) both scale in $\mathcal{O}(n^{-1/2})$ and go to 0 for large n.

Evaluation In Figure 1 (Left), we compare MULTI-CWOLA with standard CWOLA as well as two other baselines. We use simulation data from the LHC Olympics Dataset [13]; in particular from Pythia 8 [21], where the signal is boson decay and the background is generic $2 \to 2$ 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 two other baselines that utilize 3 resonant features: CWOLA-intersect defines the signal region as the intersection of the resonant features' signal regions, e.g. $SR = SR_1 \cap SR_2 \cap SR_3$, but this can be overly conservative. CWOLA $+x$ thresholding keeps one resonant feature as the noisy label $\hat{y} = M_1(m)$, and includes the remaining thresholded features as discriminative features $\{M_2(m), M_3(m), x\}$. 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. We find that MULTI-CWOLA offers a higher AUC and lower variance, especially when there is limited data.

4 MULTI-SALAD: Learning From Multiple Simulations

We introduce MULTI-SALAD, an AD approach that uses multiple simulation datasets.

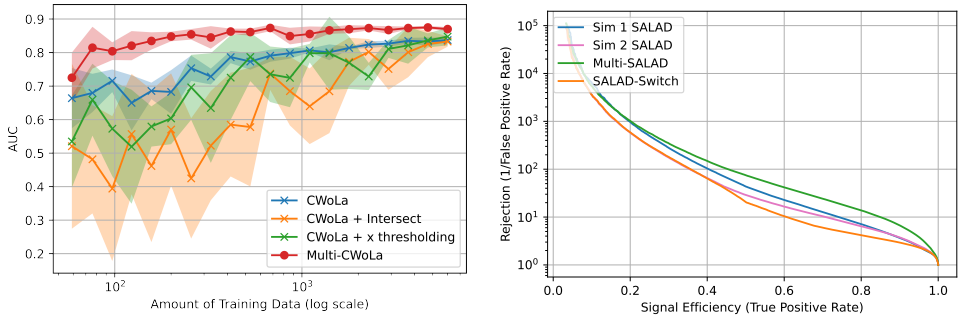

Figure 1: Left: Comparison between CWOLA and MULTI-CWOLA. Using multiple mixed samples helps performance across a range of dataset sizes. Access to multiple weak sources enables better accuracy and lower variance compared to the single-feature version. Right: Signal efficiency to rejection of MULTI-SALAD versus other baselines (weighted and unweighted).

Standard SALAD We have a background simulation dataset $\mathcal{D}^{\text{sim}} = \{(x_i, m_i)\}_{i=1}^{n_{\text{sim}}}$ with $y_i = 0$ for all i in addition to one true dataset $\mathcal{D}$. $\mathcal{D}^{\text{sim}}$ is drawn from some distribution $\mathcal{P}_{\text{sim}}$ with density p_{sim}. While CWoLA learns the likelihood ratio between the signal and sideband regions of $\mathcal{D}$ alone, SALAD utilizes $\mathcal{D}^{\text{sim}}$ as well. Note that if p_{sim} is equal to $p_{\text{data}}(\cdot|y=0)$, we could directly train a model to distinguish between $\mathcal{D}$ and $\mathcal{D}^{\text{sim}}$ in the signal region to get a classifier that could detect anomalies. However, since $\mathcal{D}^{\text{sim}}$ may not match the true background data, we instead first need to learn a reweighting function to capture the differences between $\mathcal{D}^{\text{sim}}$ and $\mathcal{D}$'s background data, and then we train a model to distinguish between $\mathcal{D}$ and the reweighted $\mathcal{D}^{\text{sim}}$ in the signal region.

Formally, given fixed SR and SB for both datasets, the method can be broken into two steps. 1) A classifier $\hat{g}$ is trained to learn the weight $w(x, m) = \frac{p(x,m|y=0)}{p_{\text{sim}}(x,m|y=0)}$ by distinguishing between $\mathcal{D}^{\text{sim}}_{SB} = \mathcal{D}^{\text{sim}} \cap SB$ and $\mathcal{D}_{SB}$. 2) Using a loss function L_S weighted using the estimated $\hat{w}(x, m)$ applied to $\mathcal{D}^{\text{sim}}_{SR} = \mathcal{D}^{\text{sim}} \cap SR$, a classifier $\hat{h}$ is trained to distinguish between $\mathcal{D}_{SR}$ and $\mathcal{D}^{\text{sim}}_{SR}$. 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 $\frac{p(x)}{p(x|y=0)}$, from which one can detect anomalies.

MULTI-SALAD Method Now we have multiple simulation datasets $\mathcal{D}^{\text{sim}}_1, \ldots, \mathcal{D}^{\text{sim}}_k$. 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 $\mathcal{D}^{\text{sim}}_{SR,i}$ 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 $\mathcal{D}$ [9]. To understand why, [8] showed that the generalization error of an empirical loss function with importance weights w depends on the magnitude and variance of w. Applied to our 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 differences between $\mathcal{D}_{SR}$ and the reweighted $\mathcal{D}^{\text{sim}}_{SR}$ that are noise rather than the anomaly. As a result, aggregating individual SALAD outputs can be equivalent to ensembling many poor classifiers.

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 $\mathcal{D}^{\text{sim}}_i$ to produce one large simulation dataset $\widetilde{\mathcal{D}}^{\text{sim}}$ whose distribution best approximates the true background $p(x|y=0)$, and then use standard SALAD with $\widetilde{\mathcal{D}}^{\text{sim}}$ and $\mathcal{D}$. 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 offer such improvements. In Algorithm 2, we write this procedure out where we simply concatenate all $\mathcal{D}^{\text{sim}}_i$ together. However, with domain knowledge on the strengths and weaknesses of each simulation across features, one could produce $\widetilde{\mathcal{D}}^{\text{sim}}$ by sampling accordingly from each. We leave this direction for future work.

Theoretical Results We now present a finite sample generalization error bound on MULTI-SALAD that also applies to SALAD. Define n^{SR} as the number of points from $\mathcal{D}$ and $\widetilde{\mathcal{D}}^{\text{sim}}$ 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 $\widetilde{\mathcal{D}}^{\text{sim}}$ belonging to the signal region. To measure the generalization error, recall $w(x,m) = \frac{p(x,m|y=0)}{p_{\text{sim}}(x,m|y=0)}$ and let $\hat{w}$ be the classifier $\hat{g}$'s estimate. We denote h as the reweighted classifier. Let $h^\star = \text{argmin}_{h\in\mathcal{H}}\ L_S(h,w)$ and let $\hat{h} = \text{argmin}_{h\in\mathcal{H}}\ \hat{L}_S(h,\hat{w})$. We aim to bound $L_S(\hat{h},\hat{w}) - L_S(h^\star,w)$. Define $W = \max_{x,m} w(x,m)$ as the maximum ratio betweeen the simulation and true background. In Theorem 2 in Appendix D, we show an upper bound on generalization error $L_S(\hat{h},\hat{w}) - L_S(h^\star,w)$ that scales in $(n^{SB})^{-1/2}$, and $(n^{SR}_{\text{sim}})^{-1/2}$, where the former comes from the initial reweighting step while the latter comes from the weighted classification step. The bound is also dependent on the Rademacher complexities of both classifiers g and h used. Finally, our result demonstrates that a larger W increases the generalization error bound.

Evaluation 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 datasets.[1] The anomaly distribution is only slightly different from the background data, which follows a symmetric Gaussian mixture. On the other hand, each simulation has a distribution that matches only one mixture component of the background data. We assume that the signal and sideband regions are the same over x (e.g. m and x are independent). A visualization is shown in Figure 2 and details are in Appendix E.

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 $\mathcal{P}(\cdot|y=1)$ well, as suggested by Figures 3 and 4. 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. In Figure 1 (Right), we present the signal efficiency to rejection rate of four methods: 1) MULTI-SALAD; 2) SALAD on the first simulation; 3) SALAD on the second simulation; and 4) SALAD-SWITCH, where we learn k separate w_i functions and switch among them in the reweighted loss function. Table 1 contains the accuracy and AUC scores for each method. Averaged over 10 random seeds, MULTI-SALAD outperforms other methods. The signal efficiency 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	$43.8_{\pm 2.2}$	$62.5_{\pm 8.8}$	$42.7_{\pm 3.6}$	$64.3_{\pm 12.3}$	$50.0_{\pm 0.0}$	$54.3_{\pm 6.2}$	$\mathbf{64.8}_{\pm 9.3}$
AUC	$28.5_{\pm 4.2}$	$80.7_{\pm 14.5}$	$27.4_{\pm 4.5}$	$78.7_{\pm 18.2}$	$15.4_{\pm 5.3}$	$74.7_{\pm 17.0}$	$\mathbf{90.8}_{\pm 10.2}$

Table 1: Accuracy and AUC scores (%) for MULTI-SALAD on two simulation datasets. We compare to SALAD-SWITCH (different 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 Conclusion

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 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, effectively utilizing as much information as possible.

[1] 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.

6 Acknowledgements

BN was supported by the Department of Energy, Office 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, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of NIH, ONR, or the U.S. Government.

References

[1] Anomaly detection search for new resonances decaying into a Higgs boson and a generic new particle X in hadronic final states using $\sqrt{s}$ = 13 TeV pp collisions with the ATLAS detector. Technical report, CERN, Geneva, 2022. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CONFNOTES/ATLAS-CONF-2022-045.

[2] Anders Andreassen, Benjamin Nachman, and David Shih. Simulation Assisted Likelihood-free Anomaly Detection. *Phys. Rev. D*, 101(9):095004, 2020.

[3] ATLAS Collaboration. Dijet resonance search with weak supervision using 13 TeV pp collisions in the ATLAS detector. 2020.

[4] Johannes Bellm et al. Herwig 7.0/Herwig++ 3.0 release note. *Eur. Phys. J. C*, 76(4):196, 2016.

[5] Francois Chollet et al. Keras, 2015.

[6] Jack H. Collins, Kiel Howe, and Benjamin Nachman. Anomaly Detection for Resonant New Physics with Machine Learning. *Phys. Rev. Lett.*, 121(24):241803, 2018.

[7] Jack H. Collins, Kiel Howe, and Benjamin Nachman. Extending the search for new resonances with machine learning. *Phys. Rev.*, D99(1):014038, 2019.

[8] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[9] Sanjoy Dasgupta and Philip M. Long. Boosting with diverse base classifiers. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 273–287, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[10] Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher R/'e. Fast and three-rious: Speeding up weak supervision with triplet methods. In *International Conference on Machine Learning*, 2020.

[11] HEP ML Community. A Living Review of Machine Learning for Particle Physics.

[12] Georgia Karagiorgi, Gregor Kasieczka, Scott Kravitz, Benjamin Nachman, and David Shih. Machine Learning in the Search for New Fundamental Physics. 12 2021.

[13] Gregor Kasieczka et al. The LHC Olympics 2020: A Community Challenge for Anomaly Detection in High Energy Physics. 1 2021.

[14] Gregor Kasieczka, Radha Mastandrea, Vinicius Mikuni, Benjamin Nachman, Mariel Pettee, and David Shih. Anomaly Detection under Coordinate Transformations. 9 2022.

[15] Gregor Kasieczka, Ben Nachman, and David Shih. R&D Dataset for LHC Olympics 2020 Anomaly Detection Challenge, April 2019.

[16] Eric M. Metodiev, Benjamin Nachman, and Jesse Thaler. Classification without labels: Learning from mixed samples in high energy physics. *JHEP*, 10:174, 2017.

[17] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[19] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 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.

[20] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Jul 2019.

[21] Torbjorn Sjostrand, Stephen Mrenna, and Peter Z. Skands. A Brief Introduction to PYTHIA 8.1. *Comput. Phys. Commun.*, 178:852–867, 2008.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We report errors for MULTI-CWOLA. For MULTI-SALAD, standard deviation on the signal efficiency to rejection rate can be extremely large (as it is computed over the reciprocal of the FPR, which is expected to be noisy across runs). Instead, we have included Figure 5, which shows several individual runs of our experiment.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Data used is publicly available [13]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Potential Broader Impacts

Anomaly detection is used throughout science and industry and so our observations may be useful to setups beyond collider physics. These tools are designed to find group anomalies and so even when applied to society at large, there is no way to single out individuals.

A.1 Code and Data

The code for this paper can be found at `https://github.com/mayeechen/anomaly-detection-multi` and the physics data sets are hosted on Zenodo at [15].

B Appendix

We provide algorithmic details for MULTI-CWOLA and MULTI-SALAD in Section C. In section D, we provide proofs for our theoretical results. In section E, we provide experimental details.

C Additional Algorithmic Details

C.1 MULTI-CWOLA Algorithm

MULTI-CWOLA is described formally in Algorithm 1. Given a dataset $\mathcal{D}$ with k resonant features with corresponding thresholds $\mathcal{I}_m$ for signal vs sideband region, we can construct a binary vector per (x, m), $\mathbf{M}(m) = [M_1(m), \dots M_k(m)] \in \{0,1\}^k$, where $M_i(m) = 1$ if x belongs to the ith resonant feature's signal region. The goal is to learn how to produce a label from k noisy "votes" $\mathbf{M}(m)$.

To aggregate $\mathbf{M}(m)$ into a label, we draw from weak supervision and aim to compute $p(y|\mathbf{M}(m))$, the distribution from which we produce an estimate $\hat{y}$. We assume $p(y, \mathbf{M}(m))$ can be written as a graphical model as follows:

$$p(y, \mathbf{M}(m)) = \frac{1}{Z} \exp\left(\theta_y \widetilde{y} + \sum_{i=1}^{k} \theta_i \widetilde{M_i}(m) \widetilde{y}\right), \tag{1}$$

where θ_y, θ_i for $i \in [k]$ are the canonical parameters of the distribution, Z is for normalization, and $\widetilde{y}$ and $\widetilde{M_i}(m)$ are y and $M_i(m)$ scaled from $\{0,1\}$ to $\{-1,1\}$. Intuitively, θ_i represents the strength of the correlation between $M_i(m)$ and y. This model also implies that $M_i(m) \perp M_j(m)|y$; that is, the mixture membership components (i.e., the resonant features) are conditionally independent given y.

With the above construction, we learn $\hat{y}$ as follows. First, we estimate $p(M_i(m)|y)$, which corresponds to the ith resonant feature's accuracy. This can be done by adapting the *triplet* approach from [10]. First, we draw triplets of resonant features $a, b, c \in [k]$. If the distribution on $y, \mathbf{M}(m)$ follows the graphical model in (1), it holds that $\mathbb{E}[\widetilde{y}\widetilde{M_a}(m)]\mathbb{E}[\widetilde{y}\widetilde{M_b}(m)] = \mathbb{E}[\widetilde{M_a}(m)\widetilde{M_b}(m)]$. Writing one such equation for each pair in the triplet (a, b, c), we have that

$$\begin{aligned}
\mathbb{E}[\widetilde{y}\widetilde{M_a}(m)]\mathbb{E}[\widetilde{y}\widetilde{M_b}(m)] &= \mathbb{E}[\widetilde{M_a}(m)\widetilde{M_b}(m)] \\
\mathbb{E}[\widetilde{y}\widetilde{M_a}(m)]\mathbb{E}[\widetilde{y}\widetilde{M_c}(m)] &= \mathbb{E}[\widetilde{M_a}(m)\widetilde{M_c}(m)] \\
\mathbb{E}[\widetilde{y}\widetilde{M_b}(m)]\mathbb{E}[\widetilde{y}\widetilde{M_c}(m)] &= \mathbb{E}[\widetilde{M_b}(m)\widetilde{M_c}(m)],
\end{aligned}$$

since $\widetilde{y}^2 = 1$. Solving this system, we obtain

$$|\mathbb{E}[\widetilde{y}\widetilde{M_a}(m)]| = \sqrt{\left|\frac{\mathbb{E}[\widetilde{M_a}(m)\widetilde{M_b}(m)]\mathbb{E}[\widetilde{M_a}(m)\widetilde{M_c}(m)]}{\mathbb{E}[\widetilde{M_b}(m)\widetilde{M_c}(m)]}\right|},$$

and similarly for b and c. We assume that each signal region is positively correlated with the true signal, which allows for us to uniquely recover $\mathbb{E}[\widetilde{y}\widetilde{M_a}(m)]$. Next, we can use $\mathbb{E}[\widetilde{y}\widetilde{M_a}(m)]$ to obtain $p(M_a(m)|y)$ due to the structure of the graphical model in (1). From these, we use Bayes' rule

Algorithm 1 MULTI-CWOLA

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

For each resonant feature m^i, define $M_i(m) = \begin{cases} 1 & x \in \mathcal{D}_{SR_i} \\ 0 & x \in \mathcal{D}_{SB_i} \end{cases}$.

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

$$\alpha_a := \sqrt{\left|\hat{\mathbb{E}}[\widetilde{M}_a(m)\widetilde{M}_b(m)]\hat{\mathbb{E}}[\widetilde{M}_a(m)\widetilde{M}_c(m)]/\hat{\mathbb{E}}[\widetilde{M}_b(m)\widetilde{M}_c(m)]\right|} \tag{2}$$

$$\alpha_b := \sqrt{\left|\hat{\mathbb{E}}[\widetilde{M}_a(m)\widetilde{M}_b(m)]\hat{\mathbb{E}}[\widetilde{M}_b(m)\widetilde{M}_c(m)]/\hat{\mathbb{E}}[\widetilde{M}_a(m)\widetilde{M}_c(m)]\right|} \tag{3}$$

$$\alpha_c := \sqrt{\left|\hat{\mathbb{E}}[\widetilde{M}_a(m)\widetilde{M}_c(m)]\hat{\mathbb{E}}[\widetilde{M}_b(m)\widetilde{M}_c(m)]/\hat{\mathbb{E}}[\widetilde{M}_a(m)\widetilde{M}_b(m)]\right|}, \tag{4}$$

where $\hat{\mathbb{E}}$ is an empirical estimate of the expectation over $\mathcal{D}$.

end for

Set $\hat{p}(M_i(m) = 1|y = 1) = \hat{p}(M_i(m) = 0|y = 0) = \hat{p}(M_i(m) = y) = \frac{\alpha_i+1}{2}$.

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

Construct $\hat{y}$ for each $(x, m) \in \mathcal{D}$.

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

to produce an estimate of $p(y|\mathbf{M}(m)) = \frac{\prod_{i=1}^m p(M_i(m)|y=1)p(y=1)}{p(\mathbf{M}(m))}$, where we assume that the class balance $p(y = 1)$ is known; otherwise, it can be estimated [20]. In practice, all of these quantities are empirical estimates, starting with terms such as $\hat{\mathbb{E}}[\widetilde{M}_a(m)\widetilde{M}_b(m)] = \frac{1}{n}\sum_{i=1}^n \widetilde{M}_a(x_i)\widetilde{M}_b(x_i)$.

Finally, with labels $\hat{y}$ for each $(x, m) \in \mathcal{D}$, we train a classifier $\hat{f}$ on points $(x, \hat{y})$. This procedure is summarized in Algorithm 1.

C.2 MULTI-SALAD Algorithm

MULTI-SALAD is described formally in Algorithm 2. We have simulation datasets $\mathcal{D}_1^{\text{sim}}, \ldots \mathcal{D}_k^{\text{sim}}$, where $\mathcal{D}_i^{\text{sim}} = \{(x_j, m_j)\}_{j=1}^{n_{\text{sim}}}$ 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 $\mathcal{D}^{\text{sim}}$ (whether it be with uniform or stratified 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 $\mathcal{D}_{SR}^{\text{sim}}, \mathcal{D}_{SB}^{\text{sim}}$ and $\mathcal{D}_{SR}$ and $\mathcal{D}_{SB}$. We train a classifier over $\mathcal{D}_{SB}^{\text{sim}}$ and $\mathcal{D}_{SB}$ to distinguish between simulation and real data in the sideband region. That is, we train a binary classifier $\hat{g}$ over points (x, m, z) in the sideband where x, m is either from $p_{\text{sim}}(\cdot|y = 0)$ $(z = 0)$ or $p(\cdot|y = 0)$ $(z = 1)$, 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 define the weight as the estimated likelihood ratio

$$\hat{w}(x, m) = \frac{\hat{g}(x, m)}{1 - \hat{g}(x, m)} \approx \frac{q(z = 1|x, m)}{q(z = 0|x, m)} = \frac{q(x, m|z = 1)}{q(x, m|z = 0)} \cdot \frac{q(z = 1)}{q(z = 0)} \tag{5}$$

$$= \frac{q(x, m|z = 1)}{q(x, m|z = 0)} = \frac{p(x, m|y = 0)}{p_{\text{sim}}(x, m|y = 0)}. \tag{6}$$

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 defined on the sideband region. Next, we interpolate and correct the bias of the simulation in the signal region. Let $\mathcal{D}_{SR}^{\text{sim}}$ be the set of simulation data in the signal region of size n_{sim}^{SR}, and let $\mathcal{D}_{SR}$ be the set of true data in the signal region of size n_{data}^{SR}, for a total

Algorithm 2 MULTI-SALAD

1: **Input:** Simulation datasets $\mathcal{D}_1^{\text{sim}}, \ldots, \mathcal{D}_k^{\text{sim}}$ and real dataset $\mathcal{D}$.
2: Construct overall simulation dataset $\mathcal{D}^{\text{sim}} = \bigcup_{i=1}^{k} \mathcal{D}_i^{\text{sim}}$.
3: Split each dataset into signal region and sideband region using resonant feature m to get $\{\mathcal{D}_{SR}^{\text{sim}}, \mathcal{D}_{SB}^{\text{sim}}\}$ and $\{\mathcal{D}_{SR}, \mathcal{D}_{SB}\}$.
4: Learn weight $\hat{w}(x, m) = \frac{\hat{g}(x,m)}{1-\hat{g}(x,m)}$, where $\hat{g}$ is a classifier that distinguishes data $\mathcal{D}_{SB}$ from simulation $\mathcal{D}_{SB}^{\text{sim}}$ in the sideband region.
5: Train a new classifier $\hat{h}$ on the signal region to distinguish between points in $\mathcal{D}_{SR}$ and points in $\mathcal{D}_{SR}^{\text{sim}}$ reweighted by $\hat{w}$, using the following loss:

$$\hat{L}_S(h, \hat{w}) = -\frac{1}{n^{SR}} \Bigg(\sum_{x \in \mathcal{D}_{SR}} \log h(x, m) + \sum_{x \in \mathcal{D}_{SR}^{\text{sim}}} \hat{w}(x, m) \log(1 - h(x, m)) \Bigg). \tag{8}$$

6: **Output:** Classifier output $\hat{h}(x, m)$ for anomaly detection.

of n^{SR} points. We train a classifier h to distinguish between the reweighted simulated data, which approximates true background data, and the true data. In particular, the loss function used is

$$\hat{L}_S(h, \hat{w}) = -\frac{1}{n^{SR}} \Bigg(\sum_{x \in \mathcal{D}_{SR}} \log h(x, m) + \sum_{x \in \mathcal{D}_{SR}^{\text{sim}}} \hat{w}(x, m) \log(1 - h(x, m)) \Bigg). \tag{7}$$

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 $p(\cdot|y = 0)$ in the signal region. Therefore, h can be used for anomaly detection. The procedure is summarized in Algorithm 2.

D Theoretical Results

We first present our generalization error bound on MULTI-CWOLA.

We define the following terms. Define the Rademacher complexity of $\mathcal{F}$ as $\mathfrak{R}_n(\ell \circ \mathcal{F}) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i \ell(f(x_i), y_i)\right]$ with Rademacher random variables $\Pr(\varepsilon = 1) = \Pr(\varepsilon = -1) = \frac{1}{2}$. Define $e_{\min}$ as the minimum eigenvalue of the covariance matrix on $[y, M_1(x), \ldots, M_k(x)]$, and let $a_{\min}$ be the minimum value of $\mathbb{E}[\widetilde{M}_i(x)y]$ over all i.

Theorem 1. *Assume that $p(y, \mathbf{M}(x))$ can be parametrized according to* (1) *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 [20]), and that $k \geq 3$. Then, with probability at least $1 - \delta$, the generalization error of* MULTI-CWOLA *on $\mathcal{D}$ is at most*

$$\begin{aligned} L_C(\hat{f}) - L_C(f^\star) \leq 4\mathfrak{R}_n(\ell \circ \mathcal{F}) + \sqrt{\frac{\log 2/\delta}{2n}} + \frac{c_1}{e_{\min} a_{\min}^5} \left(\sqrt{\frac{k}{n}} + \frac{c_2 k}{\sqrt{n}} \right) \\ + D_{\text{KL}}(p(y|x)||p(y|\mathbf{M}(m))), \end{aligned}$$

where c_1, c_2 are positive constants.

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

- The *Rademacher complexity* of $\mathcal{F}$: 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 $\mathcal{F}$. This quantity can be readily computed for a variety of function classes $\mathcal{F}$, such as decision trees, linear models, and two-layer feedforward networks, which makes our bound in Theorem 1 tractable.
- Using n finite samples: as the amount of data increases, the error decreases in $\mathcal{O}(n^{-1/2})$.

- 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 $\mathcal{O}(n^{-1/2})$ error. Asymptotically, our approach thus does no worse than training with labeled data.
- The irreducible gap between $p(y|m)$ and $p(y|\mathbf{M}(m))$: we lose information about m by only modeling mixture membership $\mathbf{M}(m)$.

Proof. From Theorem 3 of [10], we have that $L_C(\hat{f}) - L_C(f^\star)$ is bounded by the traditional ERM generalization gap of $L_C(\bar{f}) - L_C(f^\star)$, where $\bar{f} = \text{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{m} \ell(f(x_i), y_i)$ is the classifier learned on labeled data, plus the terms $\frac{c_1}{e_{\min} a_{\min}^5} \Big(\sqrt{\frac{k}{n}} + \frac{c_2 k}{\sqrt{n}} \Big) + D_{\text{KL}}(p(y|x)||p(y|\mathbf{M}(m)))$.

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

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

where we have used the fact that $\hat{L}_C(\bar{f}) \leq \hat{L}_C(f^\star)$. Then, using uniform convergence bounds, such as Theorem 3.3 of [17], we have

$$L_C(\bar{f}) - L_C(f^\star) \leq 2\big(2\mathfrak{R}_n(\ell \circ \mathcal{F}) + \sqrt{\frac{\log 2/\delta}{2n}}\big).$$

This gives us our desired result.

□

Next, we present our theoretical result on MULTI-SALAD.

We first set up some definitions. Let $\hat{g}(x) \in [\hat{g}_{\min}, \hat{g}_{\max}]$ and $g^\star(x) \in [g^\star_{\min}, g^\star_{\max}]$, where $g^\star$ is the optimal classifier. Let $\mathfrak{R}_{n^{SR}}(\ell_S \circ \{H, G\})$ be the Rademacher complexity of the overall loss $L_S(h, w)$ across function classes $h \in \mathcal{H}, g \in \mathcal{G}$. Define $W = \max_{x,m} w(x, m)$ as the maximum ratio between the simulation and true background. Let $B_1 = \max\{-\log h^\star(x, m), -\log(1 - h^\star(x, m))\}$ be based on the most extreme value of $h^\star$ (i.e. how far apart p and $p(\cdot|y = 0)$ can be). Let $\eta = \max(-\log(1 - h^\star(x, m)))$ for $x, m \in \mathcal{D}^{\text{sim}}_{SR}$. Let $\mathfrak{R}_{n^{SB}}(\ell \circ \mathcal{G})$ is the Rademacher complexity of the loss function class used for learning the reweighting, where ℓ is point-wise cross-entropy. Finally, let $B_2 = -\log(\min\{\hat{g}_{\min}, g^\star_{\min}\})$.

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

$$\begin{aligned} L_S(\hat{h}, \hat{w}) - L_S(h^\star, w) \leq\ & 2\mathfrak{R}_{n^{SR}}(\ell_S \circ \{\mathcal{H}, \mathcal{G}\}) + (1 + WB_1)\sqrt{\frac{\log 8/\delta}{2n^{SR}}} \\ & + \frac{\eta n^{SR}_{\text{sim}}}{(1 - \hat{g}_{\max})(1 - g^\star_{\max}) n^{SR}} \left(4c\mathfrak{R}_{n^{SB}}(\ell \circ \mathcal{G}) + 2c\sqrt{\frac{\log 4/\delta}{2n^{SB}}} + B_2\sqrt{\frac{\log 8/\delta}{2n^{SR}_{\text{sim}}}} \right). \end{aligned} \tag{9}$$

We make several observations about this bound:

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

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

$$L_S(h,w) = -\Pr(z'=1)\mathbb{E}_{z'=1}\left[\log h(x,m)\right] - \Pr(z'=0)\mathbb{E}_{x,m\in\mathcal{P}_{\text{sim}}^{SR}}\left[w(x,m)\log(1-h(x,m))\right]$$

where $z'=1$ for $x,m\sim\mathcal{P}$ and 0 for $x,m\sim\mathcal{P}(\cdot|y=0)$. Next, define $w(x,m)=\frac{q(x,m|z=1)}{q(x,m|z=0)}$ and let $\hat{w}$ be the weight ratio learned by our model. Let $\hat{h}=\text{argmin}_{h\in\mathcal{H}}\ \hat{L}_S(h,\hat{w})$, and let $h^\star=\text{argmin}_{h\in\mathcal{H}}\ L(h,w^\star)$. Intuitively, $h^\star$ corresponds to the true difference between $\mathcal{P}_{\text{data}}^{SR}$ and $\mathcal{P}_{\text{data}}^{SR}(\cdot|y=0)$. We can first decompose the generalization error as

$$L_S(\hat{h},\hat{w}) - L_S(h^\star,w) = [L_S(\hat{h},\hat{w}) - \hat{L}_S(\hat{h},\hat{w})] + [\hat{L}_S(\hat{h},\hat{w}) - \hat{L}_S(h^\star,\hat{w})] \tag{10}$$

$$+ [\hat{L}_S(h^\star,\hat{w}) - \hat{L}_S(h^\star,w)] + [\hat{L}_S(h^\star,w) - L_S(h^\star,w)] \tag{11}$$

We know that $\hat{L}_S(\hat{h},\hat{w})\le\hat{L}_S(h^\star,\hat{w})$, so

$$\begin{aligned}L_S(\hat{h},\hat{w}) - L_S(h^\star,w) &\le |L_S(\hat{h},\hat{w}) - \hat{L}_S(\hat{h},\hat{w})| + |\hat{L}_S(h^\star,w) - L_S(h^\star,w)| \\ &\quad + \hat{L}_S(h^\star,\hat{w}) - \hat{L}_S(h^\star,w) \\ &\le \sup_{h,w}|L_S(h,w) - \hat{L}_S(h,w)| + |\hat{L}_S(h^\star,w) - L_S(h^\star,w)| + \hat{L}_S(h^\star,\hat{w}) - \hat{L}_S(h^\star,w)\end{aligned}$$

We first bound $\sup_{h,w}|L_S(h,w)-\hat{L}_S(h,w)|$. For notation, we rewrite $L_S(h,w)$ as $L_S(h,g)$, where $w(x,m)=\frac{g(x,m)}{1-g(x,m)}$ and g belongs to some function class $\mathcal{G}$. Then, using Theorem 3.3 from [17], we get that $\sup_{h,w}|L_S(h,w)-\hat{L}_S(h,w)|\le 2\mathfrak{R}_{n^{SR}}(\ell_S\circ\{\mathcal{H},\mathcal{G}\})+\sqrt{\frac{\log 1/\delta}{2n^{SR}}}$ with probability at least $1-\delta$, where $\ell_S\circ\{\mathcal{H},\mathcal{G}\}$ is defined as satisfying $\ell_S(h(x,m),g(x,m),y) = -y\log h(x,m) - (1-y)\frac{g(x,m)}{1-g(x,m)}\log(1-h(x,m))$ for $h\in\mathcal{H},g\in\mathcal{G}$.

Next, we bound $|\hat{L}_S(h^\star,w)-L_S(h^\star,w)|$. Let $W=\max w(x,m)<\infty$ be the maximum density ratio, and let $B_1=\max_{x,m}\{-\log h^\star(x,m),-\log(1-h^\star(x,m))\}$. Assume that $B_1<\infty$. We can apply standard concentration inequalities here (Hoeffding) to get that $|\hat{L}_S(h^\star,w)-L_S(h^\star,w)|\le WB_1\sqrt{\frac{\log 2/\delta}{2n^{SR}}}$ with probability at least $1-\delta$.

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

$$\hat{L}_S(h^\star,\hat{w}) - \hat{L}_S(h^\star,w) = \frac{1}{n^{SR}}\sum_{x\in\mathcal{D}_{SR}^{\text{sim}}}(\hat{w}(x,m)-w(x,m))\cdot(-\log(1-h^\star(x,m))) \tag{12}$$

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

$$|\hat{L}_S(h^\star,\hat{w}) - \hat{L}_S(h^\star,w)| \le \frac{\eta}{n^{SR}}\sum_{x,m\in\mathcal{D}_{SR}^{\text{sim}}}|\hat{w}(x,m)-w(x,m)| \tag{13}$$

Recall that $\hat{w}(x,m)=\frac{\hat{g}(x,m)}{1-\hat{g}(x,m)}$ and $w(x,m)=\frac{g^\star(x,m)}{1-g^\star(x,m)}$ where $g^\star(x,m)=\Pr(z=1|x,m)$, so $|\hat{w}(x,m)-w(x,m)| = \frac{|\hat{g}(x,m)-g^\star(x,m)|}{(1-\hat{g}(x,m))(1-g^\star(x,m))}$. This denominator is greater than $(1-\hat{g}_{\max})(1-g^\star_{\max})$. Then,

$$|\hat{L}_S(h^\star,\hat{w}) - \hat{L}_S(h^\star,w)| \le \frac{\eta}{(1-\hat{g}_{\max})(1-g^\star_{\max})n^{SR}}\sum_{x,m\in\mathcal{D}_{SR}^{\text{sim}}}|\hat{g}(x,m)-g^\star(x,m)| \tag{14}$$

We now look at the classifier 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, $g^\star(x,m) > \hat{g}(x,m)$. Then $|\ell(g^\star(x,m),1) - \ell(\hat{g}(x,m),1)| =$

$\log \frac{g^\star(x,m)}{\hat{g}(x,m)} = \log\left(1 + \left(\frac{g^\star(x,m)}{\hat{g}(x,m)} - 1\right)\right) \geq \frac{g^\star(x,m)/\hat{g}(x,m)-1}{g^\star(x,m)/\hat{g}(x,m)} = \frac{g^\star(x,m)-\hat{g}(x,m)}{g^\star(x,m)} \geq |g^\star(x,m) - \hat{g}(x,m)|$ and $|\ell(g^\star(x,m),0) - \ell(\hat{g}(x,m),0)| = \log\frac{1-\hat{g}(x,m)}{1-g^\star(x,m)} = \log\left(1 + \left(\frac{1-\hat{g}(x,m)}{1-g^\star(x,m)} - 1\right)\right) \geq \frac{(1-\hat{g}(x,m))/(1-g^\star(x,m))-1}{(1-\hat{g}(x,m))/(1-g^\star(x,m))} = \frac{g^\star(x,m)-\hat{g}(x,m)}{1-\hat{g}(x,m)} \geq |g^\star(x,m) - \hat{g}(x,m)|$, where we use the inequality $\log(1+x) \geq \frac{x}{1+x}$ for $x > -1$. Therefore, with probability $1-\delta$,

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

$$\leq \frac{\eta n^{SR}_{\text{sim}}}{(1-\hat{g}_{\max})(1-g^\star_{\max})n^{SR}} \left(\mathbb{E}\left[|\ell(\hat{g}(x,m),z) - \ell(g^\star(x,m),z)|\right] + B_2\sqrt{\frac{\log 2/\delta}{2n^{SR}_{\text{sim}}}}\right)$$

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

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

where $L(g) = \mathbb{E}_{x,m\in SR}[\ell(g(x,m),z)]$. Since $g^\star(x,m)$ is Bayes optimal, $|L(\hat{g}) - L(g^\star)| = L(\hat{g}) - L(g^\star) = L(\hat{g}) - \hat{L}(\hat{g}) + \hat{L}(\hat{g}) - \hat{L}(g^\star) + \hat{L}(g^\star) - L(g^\star) \leq 2\sup_{g\in\mathcal{G}}|L(g) - \hat{L}(g)|$. From Theorem 3.3 in [17], this is bounded by $2\mathfrak{R}_{n^{SB}}(\ell\circ\mathcal{G}) + \sqrt{\frac{\log 1/\delta}{2n^{SB}}}$ with probability at least $1-\delta$. Then, applying a union bound, with probability $1-\delta$, we have

$$|\hat{L}_S(h^\star,\hat{w}) - \hat{L}_S(h^\star,w)|$$

$$\leq \frac{\eta n^{SR}_{\text{sim}}}{(1-\hat{g}_{\max})(1-g^\star_{\max})n^{SR}} \left(4c\mathfrak{R}_{n^{SB}}(\ell\circ\mathcal{G}) + 2c\sqrt{\frac{\log 2/\delta}{2n^{SB}}} + B_2\sqrt{\frac{\log 4/\delta}{2n^{SR}_{\text{sim}}}}\right).$$

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

$$L_S(\hat{h},\hat{w}) - L_S(h^\star,w) \leq 2\mathfrak{R}_{n^{SR}}(\ell_S\circ\{\mathcal{H},\mathcal{G}\}) + (1+WB_1)\sqrt{\frac{\log 8/\delta}{2n^{SR}}} \tag{15}$$

$$+ \frac{\eta n^{SR}_{\text{sim}}}{(1-\hat{g}_{\max})(1-g^\star_{\max})n^{SR}} \left(4c\mathfrak{R}_{n^{SB}}(\ell\circ\mathcal{G}) + 2c\sqrt{\frac{\log 4/\delta}{2n^{SB}}} + B_2\sqrt{\frac{\log 8/\delta}{2n^{SR}_{\text{sim}}}}\right) \tag{16}$$

□

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 [13]. We have $k = 3$, and construct $M_i(m)$ based on the thresholds $[[3.3, 3.7], [0.09, 0.13], [0.3, 0.35]]$. In standard CWOLA, only the first feature is regarded as the resonant feature, and it is thresholded with the interval $[3.3, 3.7]$. We measure the AUC of the MULTI-CWOLA versus CWOLA classifier and baselines in Figure 1 (Left). We constructed training datasets of varying sizes from $n = 59$ to 6003. 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 MLPClassifier with max_iter=5000. For MULTI-CWOLA's weak supervision step, we learn the parameters of the graphical model using SGD and PyTorch [18] with class balance $\Pr(y=1) = 0.25$, 30000 epochs, and learning rate $= 1e-6$.

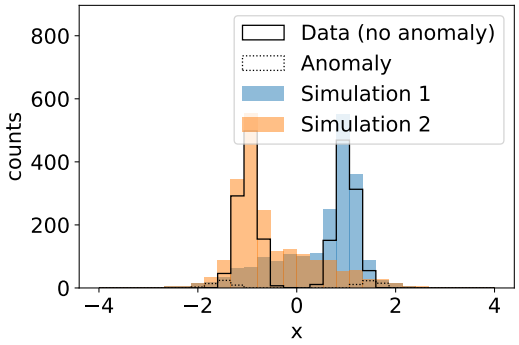

Figure 2: Synthetic data for evaluating MULTI-SALAD.

E.2 MULTI-SALAD Experiments

Setup For the synthetic, the true background is $\mathcal{P}(\cdot|y=0) = \frac{1}{2}\mathcal{N}(-1, 0.2) + \frac{1}{2}\mathcal{N}(1, 0.2)$, and the anomaly is $\mathcal{P}(\cdot|y=1) = \frac{1}{2}\mathcal{N}(-2, 0.2) + \frac{1}{2}\mathcal{N}(2, 0.2)$. Simulation 1 is $\mathcal{P}^1_{\text{sim}} = \frac{1}{2}\mathcal{N}(1, 0.2) + \frac{1}{2}\mathcal{N}(0, 1)$, and simulation 2 is $\mathcal{P}^2_{\text{sim}} = \frac{1}{2}\mathcal{N}(-1, 0.2) + \frac{1}{2}\mathcal{N}(0, 1)$. We generate 2000 points from the true background and 100 points that are anomalies to form $\mathcal{D}$, and 2000 points each from $\mathcal{P}^1_{\text{sim}}$ and $\mathcal{P}^2_{\text{sim}}$ to form $\mathcal{D}^{\text{sim}}_1$ and $\mathcal{D}^{\text{sim}}_2$. 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.

We use MLPs from Keras [5], 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 efficiency to rejection rate. All experiments were run on a personal laptop.

Additional Results In Figure 2, we plot the synthetic example we evaluate on. In Figure 3, we look at the reweighting applied to the sideband region, and in Figure 4 we compare our reweighting on simulation data to the true $p(\cdot|y=0)$ in the signal region. In Figure 5, we show our results on three individual runs. This is because computing the confidence intervals of these curves averaged across the 10 random runs is too noisy due to the magnitude of the reciprocal 1/FPR.

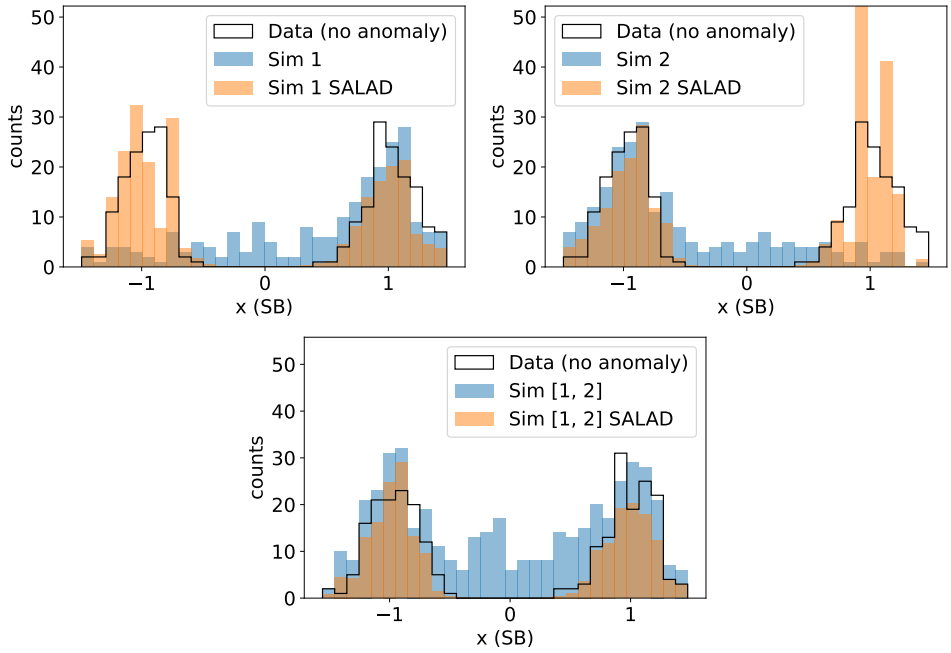

Figure 3: Top left: SALAD reweighting using simulation 1 on sideband region. Top right: reweighting using simulation 2. Bottom: reweighting using simulation 1 and 2 combined.

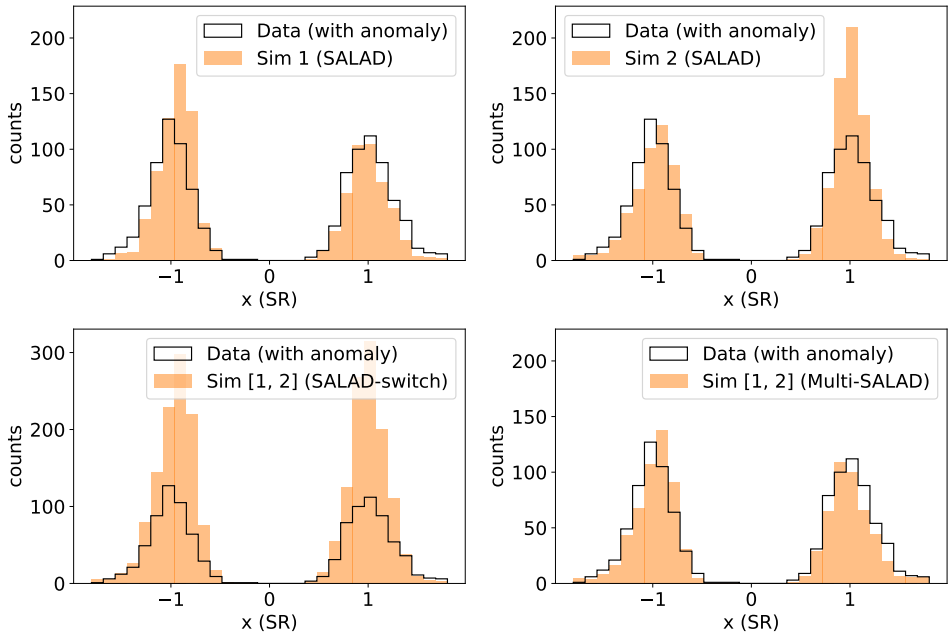

Figure 4: Top left: SALAD reweighting using simulation 1 on signal region. Top right: reweighting using simulation 2. Bottom left: using both simulation 1 and 2 weights separately. Bottom right: reweighting using simulation 1 and 2 combined.

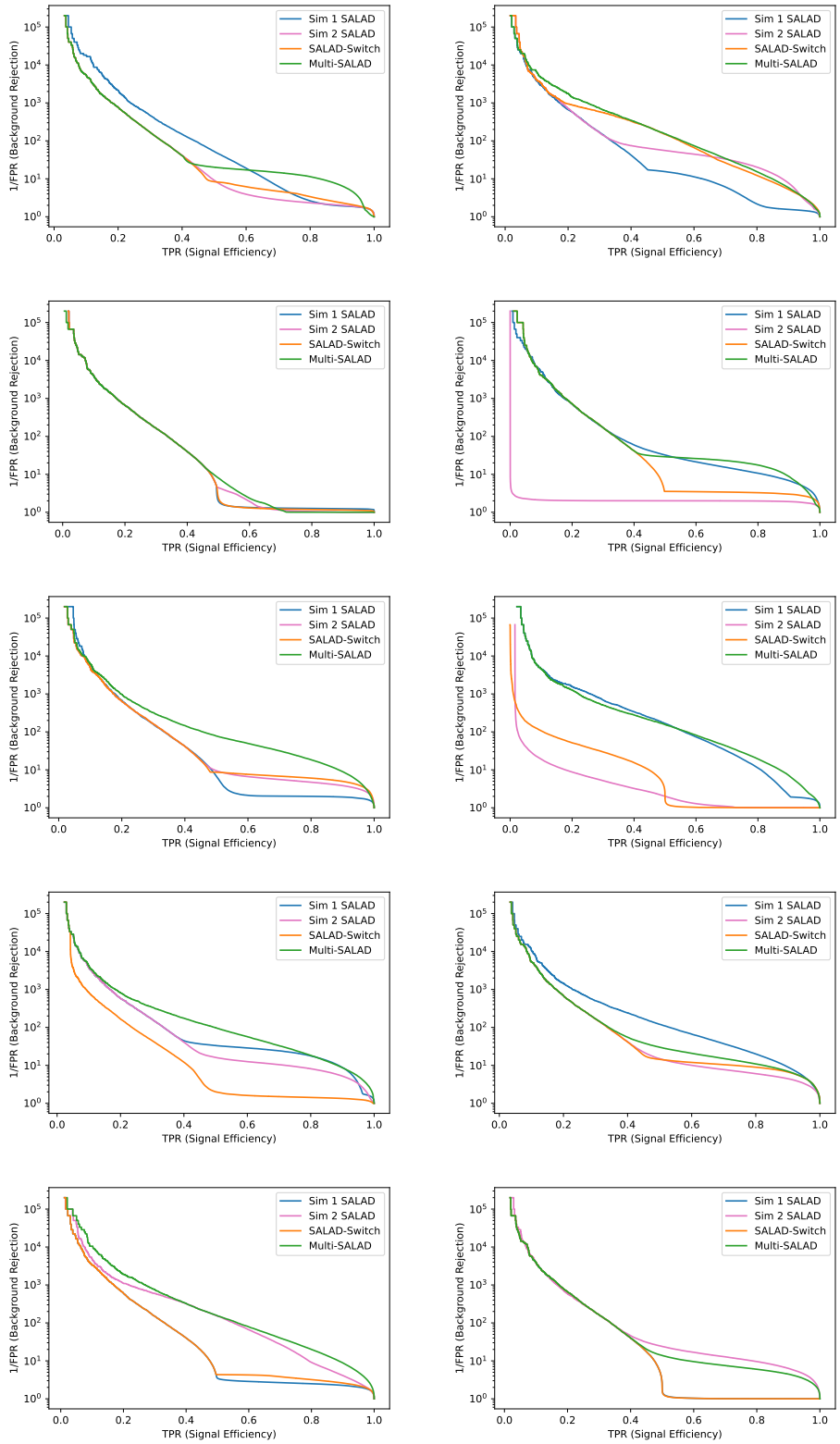

Figure 5: Results on individual runs.