Split-N-Fit: A Differentiable Maximum Likelihood Fit for training neural networks and performing anomaly detection on data

Philip Harris[1,2] **Andrzej Novak**[1,2]
[1]Massachusetts Institute of Technology, Cambridge, MA, USA
[2] Institute for Artificial Intelligence and Fundamental Interactions (IAIFI)
{novaka,pcharris}@mit.edu

Abstract

In particle physics, machine learning algorithms are often trained on simulations to codify a selection algorithm or perform a reconstruction task. When these algorithms are then applied to recorded data, their performance incurs a cost in terms of modeling uncertainties due to the limitations of the simulation. Training on *Data* would resolve this issue, but existing approaches require regions with high sample purity, which are nearly impossible to obtain. Instead, we present an approach to construct a fully differentiable maximum likelihood fit, enabling differentiable hypothesis testing directly in data. Our approach, dubbed "Split-N-Fit" can adapt to all forms of hypothesis tests and includes the ability to model systematic uncertainties. We demonstrate the use of Split-N-Fit in particle physics, specifically covering simulation mismodeling in a supervised measurement of the di-Higgs boson. We find that our approach outperforms conventional supervised learning approaches when trained on samples that are mismodeled w.r.t reference samples, and that Split-N-Fit adapts to these "poor" training data, achieving near-optimal performance.

1 Introduction

Machine learning (ML) methods in particle physics are often trained on simulated data rather than directly on experimental measurements. While simulations provide access to truth-level labels and can, in principle, generate arbitrarily large datasets, they are inherently approximations. They rely on perturbative QCD for hard scattering, phenomenological models for parton showers and hadronization, and detailed but imperfect detector response simulations [1, 2, 3]. These modeling assumptions inevitably lead to discrepancies between simulated and real collision data. Therefore, training solely on simulation risks systematic biases when models are applied to experimental data. Conversely, training on real data is limited by the absence of ground-truth labels and by statistical and selection constraints. This tension motivates the development of hybrid approaches—such as domain adaptation [4, 5], weak supervision [6, 7], and simulation-based inference [8]—that can mitigate mismodeling while leveraging the complementary advantages of both real and simulated data.

Weakly supervised resonant anomaly searches learn background structure from sidebands to detect signals. Notable examples include CATHODE [9], which uses conditional density estimation, and CURTAINs [10], which transports sideband events via invertible maps; extensions such as LaCATHODE [11] and CURTAINs Flows [12] have set strong benchmarks. Despite their success, these methods share limitations: two-stage workflows prevent optimizing representations for the final test statistic, template transport can introduce covariate-shift or coverage errors, and non-differentiable binning blocks gradient flow and enforces ad-hoc hyperparameter choices.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

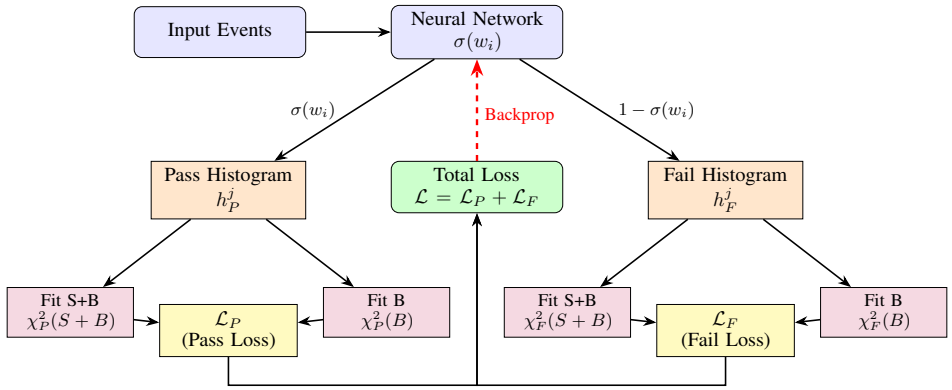

Figure 1: Schematic overview of the Split-N-Fit method. Events are split into pass and fail histograms. Both S+B and B functions are fit to each histogram, yielding χ^2 values that define individual loss terms $\mathcal{L}_P$ and $\mathcal{L}_F$, which combine to form the total differentiable loss.

We propose replacing such weakly supervised strategies with a fully differentiable binned likelihood fit trained end-to-end. We call this approach Split-N-Fit. Our work builds on recent auto-differentiable methods that aim to connect physics results, uncertainty knowledge with the learning process [13, 14, 15, 16, 17], which include the first application in a full-fledged HEP analysis [18]. Concretely, we embed (i) a differentiable histogrammer (via smooth bin surrogates with straight-through gradients [19]), (ii) a likelihood with profileable nuisance parameters, and (iii) the final likelihood-ratio objective directly into the computational graph. A diagram is shown in Figure 1.

This allows the feature extractor to be jointly optimized against the actual test statistic while propagating gradients through binning and systematic morphing. Prior work shows that differentiable binning is feasible at high precision [19], enabling stable gradient-based training; integrating differentiable binning in the likelihood lets us learn categorizations, integrate analytic models for backgrounds, calibrate nuisance pulls, and align transfer models to the inference objective—closing the data/simulation mismatch, and avoiding two-stage procedures used in weak supervision.

2 Method

At a high level, Split-N-Fit works as follows: a neural network outputs a score w_i for each event, which is converted to a probability $\sigma(w_i)$ via a sigmoid function. This probability determines how each event is split between two histograms: "pass" and "fail". Analytic functions are then fit to each histogram (signal+background for pass, background-only for fail) and the quality of these fits is used to define a loss function. This loss is minimized to encourage the network to place signal events in the pass category and background events in the fail category, thereby maximizing statistical separation.

The core of Split-N-Fit is a loss function with an embedded differentiable hypothesis test. Once such a loss function is constructed, a variety of applications are possible. We emphasize that this loss function can be applied to any neural network architecture. The Split-N-Fit loss can be written by constructing two differentiable histograms, labelled pass (P) and fail (F). The histograms can be written in terms of a binning b_j of a variable x with each event having value x_i and neural network output w_i

$$h_P^j = \sum_{x_i \in b_j} \sigma(w_i) \qquad h_F^j = \sum_{x_i \in b_j} 1 - \sigma(w_i) \tag{1}$$

Where σ here denotes a sigmoid applied to the output w_i of the neural network, which takes event features as input. To compute this differentiable histogram, we use a Pytorch histogram, which builds on core ideas from Ref. [19]. We then perform a maximum likelihood fit of a function to the histogram yields in both the pass and fail categories. Our maximum likelihood fit yields a function

$f_{P/F}(x|\theta_i)$, where θ_i are the parameters of the fitting function, and in the fit, we minimize

$$\chi^2_P(\theta_{\min}) = \frac{1}{h^j_P}\left(h^j_P - f_P(b_j|\theta_{\min})\right)^2 \quad (2)$$

$$\chi^2_F(\theta_{\min}) = \frac{1}{h^j_F}\left(h^j_F - f_F(b_j|\theta_{\min})\right)^2. \quad (3)$$

The fit function $f_{P/F}(x|\theta_i)$ is completely arbitrary and is chosen a priori by the user. This choice can be reasonably generic, e.g., low-order polynomials for smooth backgrounds, but can also encode physics knowledge, such as modeling the signal as a Breit-Wigner distribution for resonant processes. In this case, we use a Levenberg–Marquardt implementation `lmfit`, which calls `scipy.optimize.leastsq`, though more efficient minimizations can be done[20].

To perform the full hypothesis test, we fit two functions, a signal+background (S+B) function, and a background (B) function in both regions and aim to maximize the signal in the pass region, and minimize the signal in the failing region.

With the fitted functions, and computed $\chi^2_{P/F}$ values for $f_{S+B}(b_i|\theta_i)$, we add regularizing terms $\lambda_{\rm v}$ to improve the diversity of the inputs in the training, and λ_χ to penalize poor quality fits. In practice, we typically set these parameters $\lambda_{\rm v} = \lambda_\chi = 0$, but we list them here since they have been found to be useful. Our final loss can be written as

$$\mathcal{L} = \mathcal{L}_\mathcal{P} + \mathcal{L}_\mathcal{F} + \lambda_{\rm v}\sigma^2(w_i x_m) \quad (4)$$

$$\mathcal{L}_\mathcal{P} = \left(\chi^2_{S+B} - \chi^2_B + \lambda_\chi\left(R\left(\chi^2_{S+B} - \kappa_{S+B}\right) + R\left(\chi^2_B - \kappa_B\right)\right)\right) \quad (5)$$

$$\mathcal{L}_\mathcal{F} = \left(\chi^2_B - \chi^2_{S+B} + \lambda_\chi\left(R\left(\chi^2_{S+B} - \kappa_{S+B}\right) + R\left(\chi^2_B - \kappa_B\right)\right) + \delta\right)/\langle 1 - \sigma(w_i)\rangle \quad (6)$$

where $\lambda_{\rm v}$ is a regularization to ensure the output discriminator relies on all inputs, we typically set this value to 0, λ_χ is needed when the fit function does not describe the data well, it ensures that the fit converges with a reasonable χ^2 denoted by κ_{S+B} for the signal+background fit, and κ_B for the background fit. R denotes a SiLU (sigmoid linear unit, also known as Swish) activation function, which is a smooth, continuous variant of ReLU, and enables a turn on of the λ_χ to penalize poor fit quality above a desired χ^2, we typically set λ_χ to 0 for well behaved fits and $\lambda_\chi = 2$ for poorly behaved fits. Finally, $\langle 1 - \sigma(w_i)\rangle$ denotes the average output in the failing category; this term in the denominator is added to put pressure on the loss to avoid the minimization where the neural network puts all events in the passing category.

We want to stress that ultimately this loss is focused on categorizing events with an output probability $\sigma(w_i)$. We use $\sigma(w_i)$ to split the events, fit them, and perform a hypothesis test that then is used to improve the w_i splitting, hence the name Split-N-Fit. Given that the loss function is defined on the output of the neural network, this loss can be applied to any neural network architecture for tasks including classification, anomaly detection, and regression.

The core assumption for this analysis to work is that a function can be written, which can fit the data with a well-defined goodness of fit. For signals and backgrounds that are well-modeled by simulation, this is often achieved by using simulated shapes with additional nuisances to capture the modeling differences.

3 Results

To illustrate the Split-N-Fit procedure, we utilize the di-Higgs production sample with $h \to \bar{b}b$ and $h \to \gamma\gamma$ decays. This di-Higgs channel is considered one of the most critical channels for measuring the Higgs boson self-coupling, which is one of the major measurements of future HL-LHC running [21, 22]. All samples are produced with Madgraph 3.2 and showered with Pythia [23, 24], and detector effects are added with the Delphes toy simulation [25]. For background [26], we produce two QCD di-photon samples, one with the default Pythia shower, and another with Pythia Tune 14 [27, 28]. Additionally, the simulation is modified to reflect mismodeled detectors. For the default shower, the CMS Delphes card is used. For the modified shower, the ATLAS Delphes card is used, and then b-quarks are further smeared by an additional 10 percent, and scaled by 2 percent to reflect a typical mismodeling of the b-jet energy scale. These two different backgrounds provide a basis to demonstrate how the Split-N-Fit procedure can recover sensitivity. To understand how we can

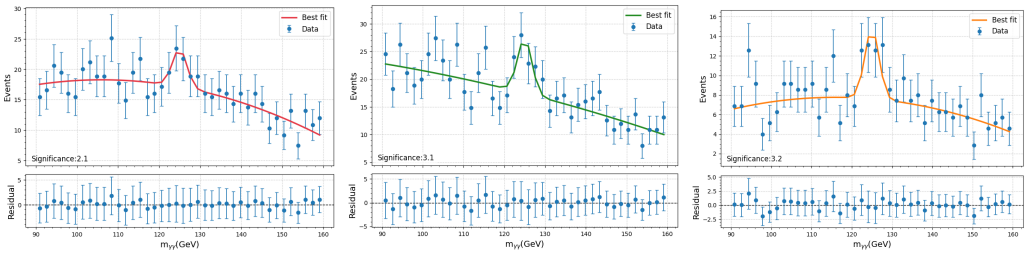

Figure 2: Visualization of the resulting di-Higgs signal in the fitted di-photon mass for (left) selecting events with an MLP trained on the incorrect background, (center), the same incorrect MLP as the left diagram fine-tuned with Split-N-Fit, and (right), an MLP trained with the correct background.

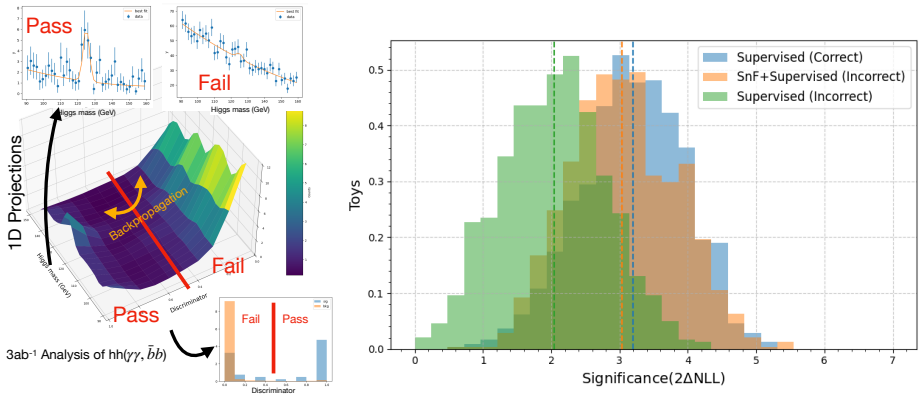

Figure 3: (Left) Diagram Illustrating Split-N-Fit on the di-Higgs dataset, showing the learned 2D discriminator di-photon mass space. The individual panels depict the 1D projections. (Right) Comparison of the significance over 1000 pseudoexperiments between the Supervised algorithm trained on the correct dataset (blue), the Supervised algorithm trained on the modified dataset (green), and Split-N-Fit finetuned on the Supervised algorithm. The vertical lines denote the median of each distribution.

fine-tune on data, we train two networks: one separates the smeared, mis-tuned background from the signal (incorrect), and the other is trained against the true background (correct). For both networks, we use the same number of events, the same batch size, and the same architecture. In all cases, this is a 4-layer MLP with 32 hidden parameters and swish activation. The input variables are 10 high-level observables documented in a recent related paper [28]. Lastly, the samples are scaled to correspond with the full luminosity of a future LHC data-taking period ($3ab^{-1}$). With the current dataset, the expected sensitivity, including full uncertainties, using the ATLAS detector is approximately 2.2 standard deviations. We find a similar sensitivity with our analysis. However, we stress that our analysis is only an approximation of an LHC analysis.

Figure 2 demonstrates the Split-N-Fit procedure on an example event. For all events, the significance is computed by calculating the 2 Δ-log-likelihood for a B-only null hypothesis and a S+B hypothesis. A second-order polynomial is used for the background model, and a Gaussian with fixed mass and width is used for the signal. The signal region is defined as events with a discriminator value greater than 0.5. From the plots, it's clear that Split-N-Fit restores the significance of the incorrectly trained model back to roughly the same sensitivity as the ideal trained model. The increase in significance is substantial, corresponding to a sensitivity of 2 standard deviations to 3 standard deviations. An equivalent improvement from data alone would require a more than doubling of the data (or an additional 6 years of LHC operation).

Finally, we perform this same test over 1000 randomly subsampled pseudo-experiments and compute the observed significance. The performance is shown in Figure 3. We find that we can increase the performance of the incorrectly trained model by roughly one standard deviation over all the toys. Moreover, if we consider the supervised classifier as the theoretical limit of discrimination power, we find that we recover nearly all the missing discrimination power. In a more realistic version of this

analysis, we would have profiled additional nuisances in the fit and added additional backgrounds. However, we note that there is no technical limitation to adding these parameters.

In summary, we have presented a new differentiable fitting approach, Split-N-Fit. Our approach builds upon the construction of a differentiable histogram and interfaces for fitting within it. We have demonstrated that this method enables fine-tuning of poor training on simulation or control regions by making the final fitted observable differentiable.

A limitation of this method is that a robust fitting function and fitting variable must be determined at the start of training. The function, in particular, needs to be able to converge to a large variety of possible shapes to ensure that the training can adapt to large steps away from the original pre-training. This algorithm is designed to find a signal in unlabeled data, and, as such, can produce an excess of events from random background fluctuations. To mitigate this "overtraining", various approaches can be taken, such as k-fold cross-validation, early stopping, and regularization via $\lambda_\chi > 0$ to penalize poorly-fit distributions.

For larger-scale fine-tunings or complete training tasks, a validation sample on the fitted region is needed to ensure that the network doesn't overtrain. This can be achieved by splitting the dataset into two or other k-folds and comparing the loss on these k-folds to prevent divergence. We note that, even then, some care needs to be taken, since the k-folding procedure has its own intrinsic biases [29]. Moreover, this method is subject to limitations of the central limit theorem and will become more biased towards random overdensities as more inputs are added.

A future direction of study will be to understand the size of networks that can be handled with Split-N-Fit. When acting as a fine-tuning step, large initial networks can be pre-trained, leaving a smaller set of updates to fine-tune on. We suggest that contrastive pre-training can aid fine-tuning by leading to an interpretable, possibly fitable space that captures complex particle/other information in a reduced-dimensional dataset [30, 31]. Beyond fine-tuning, there is a wide range of applications for this approach. In particular, this approach can be used in place of conventional weak supervision methods [32, 9, 33] since it performs training on data, which simultaneously maximizes a signal, and controls the backgrounds through a fit. The advantage of Split-N-Fit over other approaches is that it can be performed in a single step, thereby avoiding algorithmic complications such as sidebands, model ensembling, and issues with systematic uncertainties. Ultimately, this work bridges the gap between simulation and data, taking a step closer to the paradigm of automatically tuning our physics model to the data we observe.

References

[1] Andy Buckley et al. General-purpose event generators for lhc physics. *Phys. Rept.*, 504:145–233, 2011.

[2] Torbjörn Sjöstrand, Stefan Ask, et al. An introduction to PYTHIA 8.2. *Comput. Phys. Commun.*, 191:159–177, 2015.

[3] J. de Favereau et al. DELPHES 3: a modular framework for fast simulation of a generic collider experiment. *JHEP*, 02:057, 2014.

[4] Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

[5] Yaroslav Ganin et al. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:1–35, 2016.

[6] Glen Cowan, Kyle Cranmer, Eilam Gross, and Ofer Vitells. Asymptotic formulae for likelihood-based tests of new physics. *Eur. Phys. J. C*, 71:1554, 2011.

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

[8] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proc. Natl. Acad. Sci. USA*, 117(48):30055–30062, 2020.

[9] Anna Hallin, Joshua Isaacson, Gregor Kasieczka, Claudius Krause, Benjamin Nachman, Tobias Quadfasel, Matthias Schlaffer, David Shih, and Manuel Sommerhalder. Classifying anomalies through outer density estimation (cathode). *Phys. Rev. D*, 106(5):055006, 2022.

[10] Anna Hallin, Gregor Kasieczka, David Shih, et al. Curtains for your sliding window: Constructing unobserved regions with transport and invertible networks. *SciPost Physics*, 2023.

[11] Anna Hallin, Gregor Kasieczka, Tobias Quadfasel, David Shih, and Manuel Sommerhalder. Resonant anomaly detection without background sculpting. 2022.

[12] Tiecheng Gong, Gregor Kasieczka, Benjamin Nachman, David Shih, et al. Curtains flows for flows: Constructing unobserved regions with conditional normalizing flows. 2023.

[13] Pablo De Castro and Tommaso Dorigo. INFERNO: Inference-Aware Neural Optimisation. *Comput. Phys. Commun.*, 244:170–179, 2019.

[14] Nathan Simpson and Lukas Heinrich. neos: End-to-End-Optimised Summary Statistics for High Energy Physics. *J. Phys. Conf. Ser.*, 2438(1):012105, 2023.

[15] Aishik Ghosh, Benjamin Nachman, and Daniel Whiteson. Uncertainty-aware machine learning for high energy physics. *Phys. Rev. D*, 104:056026, Sep 2021.

[16] Christoph Englert, Peter Galler, Philip Harris, and Michael Spannowsky. Machine Learning Uncertainties with Adversarial Neural Networks. *Eur. Phys. J. C*, 79(1):4, 2019.

[17] Lukas Heinrich, Matthew Feickert, and Giordon Stark. scikit-hep/pyhf: v0.7.6, 2024. Accessed: 2025-08-30.

[18] CMS Collaboration. Measurement of the Higgs boson production via vector boson fusion and its decay into bottom quarks in proton-proton collisions at $\sqrt{s} = 13$ TeV. CMS Physics Analysis Summary CMS-PAS-MLG-23-005, CERN, 2024.

[19] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss, 2016.

[20] Reuben Feinman. Pytorch-minimize: A library for numerical optimization with autograd, 2021. Accessed: 2025-08-30.

[21] ATLAS and CMS Collaborations. Highlights of the HL-LHC physics projections by ATLAS and CMS, 2025. CERN Technical Report ATL-PHYS-PUB-2025-018.

[22] ATLAS Collaboration. HL-LHC prospects for the measurement of Higgs boson pair production. *CERN Document Server*, 2022.

[23] Olivier Mattelaer. Madgraph5_amc@nlo 3.2.0, 2021. Accessed: 2025-08-30.

[24] Johan Alwall, Michel Herquet, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. Madgraph 5: Going beyond. *JHEP*, 06:128, 2011.

[25] Andrea Giammanco, Vincent Lemaitre, Alexandre Mertens, and Michele Selvaggi. DELPHES 3: A modular framework for fast simulation of generic collider experiments, 2014. arXiv:1307.6346.

[26] Georges Aad et al. Search for Higgs boson pair production in the two bottom quarks plus two photons final state in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. *Phys. Rev. D*, 106(5):052001, 2022.

[27] Peter Skands, Stefano Carrazza, and Juan Rojo. Tuning PYTHIA 8.1: the Monash 2013 Tune. *Eur. Phys. J. C*, 74(8):3024, 2014.

[28] Oz Amram and Manuel Szewc. Data-Driven High-Dimensional Statistical Inference with Generative Models. 6 2025.

[29] Prasanth Shyamsundar, Nicholas Smith, and Manuel Szewc. Unaccounted-for look-elsewhere effect in k-fold cross adaptive anomaly searches. Presented at the Anomaly Detection for High Energy Physics (AD4HEP) Workshop, Columbia University, Nevis Laboratories, June 2025. Slide deck available at the AD4HEP workshop Indico site.

[30] Philip Harris, Jeffrey Krupa, Michael Kagan, Benedikt Maier, and Nathaniel Woodward. Resimulation-based self-supervised learning for pretraining physics foundation models. *Phys. Rev. D*, 111(3):032010, 2025.

[31] Kyle Metzger, Lana Xu, Mia Sodini, Thea K. Arrestad, Katya Govorkova, Gaia Grosso, and Philip Harris. Anomaly preserving contrastive neural embeddings for end-to-end model-independent searches at the LHC. 2 2025.

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

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