Machine Learning Methods for Cross Section Measurements

by

Krish Desai

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Physics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Doctor Benjamin Nachman, Co-chair
Professor Uros Seljak, Co-chair
Professor Joshua Bloom
Professor Saul Perlmutter

Spring 2025

Machine Learning Methods for Cross Section Measurements

Abstract

Machine Learning Methods for Cross Section Measurements

by

Krish Desai

Doctor of Philosophy in Physics

University of California, Berkeley

Doctor Benjamin Nachman, Co-chair

Professor Uros Seljak, Co-chair

Precise differential cross section measurements are indispensable for testing Standard Model predictions at the energy frontier and searching for new physics, yet their extraction from collider data is an ill–posed inverse problem. *Unfolding* (also known as deconvolution) is the process of removing detector distortions to reconstruct particle–level truth from detector–level data. Conventional histogram–based, binned unfolding techniques introduce artifacts, impose arbitrary bin edges, and become computationally prohibitive in high–dimensional phase spaces, potentially obscuring underlying physics.

This dissertation develops a unified framework that leverages modern machine learning techniques to surmount these limitations. Beginning with Neural Posterior Unfolding, I demonstrate how conditional normalizing flows can serve as differentiable surrogates of detector response, enabling likelihood–based unfolding through implicit regularization. Building on this foundation, Moment Unfolding directly extracts distribution moments without binning, providing precise experimental predictions for effective field theories and phenomenological models. The framework is further advanced by Reweighting Adversarial Networks (RANs), which perform full spectral unfolding using adversarial training to implement particle–level reweighting steered by detector–level classifiers, offering theoretical and computational advantages over iterative methods. A critical statistical analysis of event correlations in unfolded data reveals systematic misestimation of uncertainties when these correlations are ignored, leading to methodological recommendations that ensure correct coverage for all derived observables.

The methodology is validated using both idealized Gaussian distributions and realistic particle physics data, including proton–proton collision simulations of the CMS detector. These methods are applied to Z+jets events, demonstrating significant improvements in precision, accuracy, and computational efficiency, achieving orders of magnitude speed up while maintaining unbiased recovery of sharp spectral features.

By marrying statistical rigour with powerful machine learning methods, this work establishes a scalable blueprint for precision measurements at current and future colliders. The resulting open-source software enables more reliable extraction of fundamental physics parameters from complex detector data, advancing our ability to test theoretical models and potentially discover new phenomena in high energy physics experiments.

To my family, mentors, and friends

for their unwavering support and encouragement

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisors, Dr. Benjamin Nachman and Professor Uros Seljak, for their invaluable guidance, unwavering support, and endless patience throughout my doctoral journey. Their insights and encouragement have been instrumental in shaping both this work and my development as a researcher.

I am grateful to the members of my dissertation committee, Professor Joshua Bloom and Professor Saul Perlmutter, for their thoughtful feedback, challenging questions, and valuable suggestions that have significantly improved this work.

I would like to thank my collaborators at [Institution/Lab names] for the stimulating discussions and productive partnerships that have enriched this research. Special thanks to [specific names] for their contributions to [specific chapters/projects].

My sincere appreciation goes to the [Department/Group name] at UC Berkeley for providing an intellectually stimulating environment and the resources necessary for this research. I am particularly grateful to [staff members] for their administrative support.

This work was supported by [funding sources, e.g., NSF grant numbers, fellowships, etc.]. I acknowledge the use of computational resources from [computing facilities].

To my fellow graduate students and postdocs, especially [names], thank you for the countless discussions, both scientific and otherwise, that made this journey enjoyable and memorable.

Finally, I am deeply grateful to my family and friends for their love, understanding, and encouragement. To [specific family members], your belief in me has been a constant source of strength. To [partner/spouse name if applicable], thank you for your patience, support, and for being my anchor through the ups and downs of graduate school.

This thesis is as much a product of your collective support as it is of my own efforts. Thank you all.

Chapter 1

Introduction and Physics Background

1.1 The Standard Model: Theoretical framework

The Standard Model of particle physics represents one of the most significant intellectual achievements in modern science. Developed throughout the latter half of the 20^{th} century, it provides a quantum field theory framework that describes three of the four known fundamental forces—the electromagnetic, weak, and strong interactions—along with classifying all known elementary particles. The mathematical formulation of the Standard Model is based on gauge theory, specifically quantum chromodynamics (QCD) and the electroweak theory, underpinned by the gauge symmetry group $SU(3) \times SU(2) \times U(1)$.

The predictive power of the Standard Model has been repeatedly validated through precision experiments across multiple energy scales, from low-energy nuclear phenomena to the highest energy particle collisions achievable at modern accelerators. Its crowning achievement came with the discovery of the Higgs boson in 2012 at the Large Hadron Collider (LHC), confirming the mechanism through which elementary particles acquire mass. A schematic illustration of the standard model can be found in Fig. **??**

Fundamental Particles and Forces

The Standard Model categorizes elementary particles into two main families: fermions, which comprise matter, and bosons, which mediate forces between matter particles.

Fermions: The Building Blocks of Matter

Fermions, characterized by half-integer spin, obey the Pauli exclusion principle and are further classified into quarks and leptons, each arranged in three generations of increasing mass:

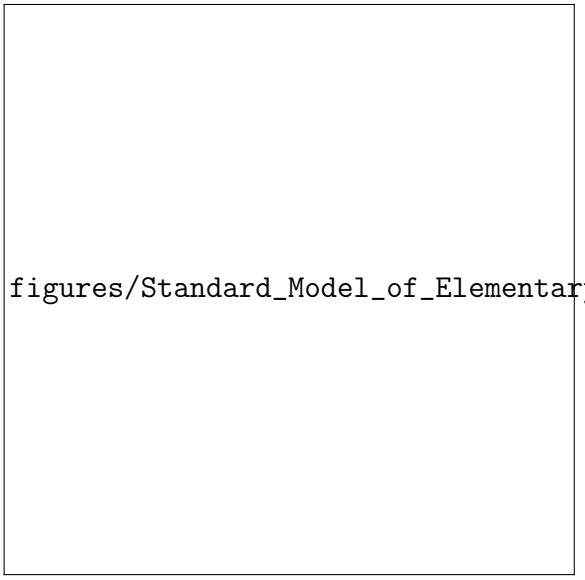


Figure 1.1: A schematic illustration of the Standard Model.

Quarks (spin $= 1/2$) Quarks are catagorized into three generations as follows:

- Up (u) and down (d),
- Charm (c) and strange (s),
- Top (t) and bottom (b).

Quarks carry fractional electric charge and color charge, and experience all fundamental forces. They are confined within hadrons—composite particles categorized as baryons (three–quark states, like protons and neutrons) or mesons (quark–antiquark pairs).

Leptons (spin $= 1/2$) Like quarks, leptons too are catagorized into three generations.

- Electron (e) and electron neutrino (ν_e),
- Muon (μ) and muon neutrino (ν_μ),
- Tau (τ) and tau neutrino (ν_τ).

Electrons, muons, and taus carry unit electric charge and interact via electromagnetic and weak forces, while neutrinos are electrically neutral and interact only through the weak force, making them notoriously difficult to detect.

Bosons: Force Carriers

Bosons, with integer spin values, mediate the fundamental interactions. The Standard Model comprises the following bosons:

- Photon (γ): The massless spin–1 boson mediating the electromagnetic force,
- $W^{\pm}$ and Z bosons: Massive spin–1 bosons mediating the weak force
- Gluons (g): Eight massless spin-1 bosons mediating the strong force
- Higgs boson (H): A massive spin-0 boson associated with the Higgs field that gives mass to elementary particles

Theoretical Framework and Symmetries

The Standard Model is constructed through principles of quantum field theory where particles are excitations of underlying quantum fields. Its mathematical structure is determined by local gauge invariance under the following specific symmetry transformations

- $U(1)_Y$: Associated with weak hypercharge, the symmetry of electroweak theory
- $SU(2)_L$: Describes the weak isospin, acting on left-handed fermions
- $SU(3)_C$: Governs the strong interactions through color charge in QCD

Electroweak unification, demonstrated by Glashow, Weinberg, and Salam**[cite –KD]**, demonstrates how the electromagnetic and weak forces emerge as different aspects of a single electroweak interaction, which undergoes spontaneous symmetry breaking at low energies.

The Higgs Mechanism and Mass Generation

The Higgs mechanism, proposed by several physicists including Peter Higgs in the 1960s**[cite –KD]**, addresses the theoretical inconsistency of massive gauge bosons in a gauge-invariant theory. The mechanism introduces a scalar field—the Higgs field—that permeates space and spontaneously breaks the electroweak symmetry when the universe cooled after the Big Bang. This symmetry breaking generates masses for the W and Z bosons while leaving the photon massless, explaining the significant difference between the electromagnetic and weak forces at ordinary energies. Additionally, the Higgs field couples to fermions through Yukawa interactions**[cite –KD]**, generating their masses with coupling strengths proportional to the particle masses.

The discovery of the Higgs boson at the LHC in 2012, with properties consistent with Standard Model predictions, provided crucial experimental validation of this mechanism and completed the Standard Model's particle roster.

Limitations and Beyond the Standard Model (BSM) Physics

Despite its remarkable success, the Standard Model has several well-recognized limitations:

- It does not incorporate gravity, the fourth fundamental force.
- It fails to explain the observed matter–antimatter asymmetry in the universe.
- It does not account for dark matter or dark energy, which together constitute about 95% of the universe's energy content.
- It requires fine-tuning of parameters, raising theoretical concerns like the hierarchy problem.
- It does not explain neutrino masses, which must exist given observed neutrino oscillations.

These limitations motivate theoretical extensions and experimental searches for physics beyond the Standard Model, including supersymmetry, grand unified theories, and various dark matter candidates. Precision measurements at particle colliders provide one of the most powerful approaches to probe these potential extensions, making analysis techniques like those discussed in this thesis essential for advancing our fundamental understanding of nature.

1.2 Fundamental role of cross section measurements in particle physics

Differential cross section measurements are the fundamental currency of scientific exchange in particle physics, serving as the primary bridge between theoretical predictions and experimental observations. These measurements quantify the probability density of specific particle interactions as a function of kinematic variables, providing the essential link between theoretical predictions and experimental observations. A cross section quantifies the probability of a specific particle interaction occurring and is typically expressed in units of area (barns, where $1\,\text{barn} = 10^{-24}\text{cm}^2$). This seemingly simple concept forms the cornerstone of how we test and validate our understanding of fundamental physics.

The Standard Model of particle physics—our most successful theory describing elementary particles and their interactions—makes precise predictions for cross sections that can be directly tested at collider experiments. Any statistically significant deviation between measured cross sections and theoretical predictions may signal the presence of new physics beyond the Standard Model**[cite –KD]**.

Cross sections are particularly powerful because they encode the underlying quantum field theory structure in a form that can be directly probed by experiment. For instance, measurements of jet production cross sections at different energy scales reveal the running of the strong coupling constant α_S**[cite –KD]**, while precision electroweak cross section measurements constrain the properties of the Higgs boson and other fundamental particles**[cite –KD]**. In searches for physics beyond the Standard Model, differential cross section measurements can reveal subtle deviations that point to new particles or interactions, even when direct observation is beyond experimental reach.

These measurements also serve a crucial role in constraining effective field theories (EFTs) that parameterize potential new physics in a model-independent way. By measuring differential distributions with high precision, experiments can place bounds on EFT coefficients, narrowing the space of viable theoretical extensions to the Standard Model **[cite –KD]**. The ongoing precision program at the Large Hadron Collider (LHC) relies heavily on refined cross section measurements to extract maximum physical insight from collected data.

1.3 Cross Section Measurements: From Theory to Experiment

The measurement of cross sections is a cornerstone of experimental particle physics, providing a direct link between theoretical predictions and observable phenomena. Cross sections quantify the likelihood of specific interactions or scattering processes between particles and are expressed in units of area, typically barns ($1\,\text{barn} = 10^{-28}\,\text{m}^2$).

Theory

The cross section (σ) represents the effective area within which two particles must interact for a particular process to occur. For collisions between discrete particles, the cross section is defined as the area transverse to their relative motion. If the particles were to interact via contact forces (e.g., hard spheres), the cross section corresponds to their geometric size. For long–range forces however, (e.g., electromagnetic or gravitational interactions), the cross section is larger than the physical dimensions of the particles due to action-at-a-distance effects. The differential cross section ($\frac{d\sigma}{d\Omega}$) provides additional

granularity by describing how the probability of scattering depends on specific final-state variables, such as scattering angle (θ) or energy transfer. It is defined as:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega} = \frac{\text{Number of events scattered into } \mathrm{d}\Omega}{\text{Incident flux} \times \text{Target density}}. \tag{1.1}$$

The total cross section can be recovered by integrating over solid angle:

$$\sigma = \int_{4\pi} \frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}\,\mathrm{d}\Omega. \tag{1.2}$$

Differential cross sections are have a long history of providing valuable insights for probing fundamental properties of particles and interactions. For example, Rutherford scattering experiments revealed the existence of atomic nuclei by analyzing angular distributions of scattered alpha particles.**[cite –KD]**

Experimental Measurement

Experimentally, cross sections are determined by measuring the rate of particle interactions under controlled conditions. Consider a beam of incoming particles with flux J (s^{-1}) directed at a target slab with thickness $\mathrm{d}x$ and density n. The probability ($\mathrm{d}p$) of scattering within this slab is proportional to the product of the target density, thickness, and cross section: $\mathrm{d}p = n\,\sigma\,\mathrm{d}x$.

The rate of scattered particles ($J_{\text{scattered}}$) can then be expressed as:

$$J_{\text{scattered}} = J \times \sigma \times n\,\mathrm{d}x. \tag{1.3}$$

$$\therefore \sigma = \frac{J_{\text{scattered}}}{J\,n\,\mathrm{d}x}. \tag{1.4}$$

Modern experiments employ arrays of detectors positioned at various angles around the interaction region to measure differential cross sections. These detectors count scattered particles as a function of solid angle ($d\Omega$), enabling precise determination of angular distributions. Normalization factors are critical for accurate measurements. These include corrections for beam intensity fluctuations, detector efficiency, and background noise. Additionally, advanced simulations are used to account for detector distortions and acceptance effects.

Applications in Particle Physics

Cross section measurements serve multiple roles in particle physics. Comparing measured cross sections with predictions from quantum field theory validates and tests theoretical models like Quantum Chromodynamics (QCD) and electroweak theory. Deviations from

expected cross sections may indicate new phenomena, such as supersymmetric particles or dark matter candidates. Differential cross sections also provide constraints on effective field theories and parton distribution functions (PDFs), essential for understanding the internal structure of hadrons. Unfolded cross section measurements allow comparisons with theoretical models years after data collection, even if detector simulations are no longer available, further enhancing their utility, and future proofing the data.

Cross sections bridge theory and experiment by encapsulating interaction probabilities in a measurable form. Their determination requires careful experimental design and analysis techniques to account for systematic uncertainties introduced by detector effects.

1.4 Detector response in precision measurements

The direct comparison between theoretical predictions and experimental measurements is fundamentally complicated by detector effects. Particle physics detectors are intricate technological marvels surrounding collision points, but they introduce distortions that must be carefully accounted for to extract the true physical distributions of interest. These detector effects include:

- **Finite resolution**: Detector components have limited precision in measuring particle energies, momenta, and positions.
- **Acceptance effects**: Geometric constraints mean that some regions of phase space remain unobserved.
- **Efficiency variations**: The probability of detecting particles varies across phase space and between particle types.
- **Particle misidentification**: The detector may incorrectly classify particle types.
- **Background contamination**: Processes other than the signal of interest contribute to the observed data.

The relationship between the true particle–level distribution (what we want to measure) and the detector–level distribution (what we actually observe) can be mathematically expressed as:

$$p_{\text{detector}}(x) = \int R(x|z)\; p_{\text{particle}}(z)\; \mathrm{d}z \tag{1.5}$$

Here, $R(x|z)$ represents the detector response function that maps particle–level data (Z) to detector–level data X. This function encapsulates all detector effects and is typically

estimated using detailed simulation.**[cite –KD]** The fundamental challenge of experimental particle physics is to invert this kernel to infer $p(Z = z)$ from observed data $p(X = x)$, a process known as unfolding or deconvolution **[cite –KD]**.

1.5 Experimental challenges at modern colliders

The Large Hadron Collider represents the energy and precision frontiers of particle physics, producing collision events of unprecedented complexity at energies up to 13 TeV. Several experimental challenges at the LHC and other modern colliders make cross section measurements particularly demanding.

- **High–dimensional phase spaces**: Modern measurements often involve multiple correlated observables, creating high–dimensional distributions that are difficult to analyze with traditional methods.
- **Limited statistics in extreme regions**: Rare processes or the tails of distributions often contain valuable physics information but suffer from limited statistics.
- **Complex detector effects**: Detectors have non-trivial response functions that can vary significantly across phase space, and are only known implicitly through precision simulations. Their explicit functional form is unknown.
- **Theoretical uncertainties**: Precision measurements are increasingly limited by theoretical uncertainties in both signal and background modeling.
- **Computational constraints**: Detailed simulation of detector response requires substantial computing resources, limiting the statistical precision of response modeling.

These challenges make the unfolding problem increasingly difficult, particularly as measurements probe more complex final states and differential distributions. For example, measurements of jet substructure, which probe the detailed radiation pattern within collimated sprays of particles, involve observables with complex correlations and detector effects that vary based on jet energy, rapidity, and substructure properties themselves. **[cite –KD]**

The need for unfolding arises from the fundamental requirement to present results in a detector–independent form that can be directly compared with theory predictions or results from different experiments. Without this correction, theoretical interpretations would need to incorporate experiment–specific detector simulations, significantly complicating scientific exchange and theoretical analysis.

1.6 Thesis Scope and Physics Impact

This dissertation focuses on developing, analyzing, and applying novel machine learning methods for cross section measurements in particle physics, with particular emphasis on unbinned approaches that overcome limitations of traditional techniques. The work spans the spectrum from improving binned methods with neural posterior estimation to completely binning-free approaches for both full distributions and statistical moments.

The primary contributions of this thesis include:

1. Development of Neural Posterior Unfolding (NPU), enhancing binned approaches through normalizing flows and amortized inference
2. Introduction of Moment Unfolding, directly deconvolving distribution moments without binning
3. Creation of Reweighting Adversarial Networks (RAN), a general framework for unbinned spectrum unfolding
4. Analysis of event correlations in unfolded data and their impact on uncertainty estimation
5. Investigation of symmetry discovery with SymmetryGAN and its connections to measurement constraints

These methodological advances address fundamental challenges in experimental particle physics, potentially enhancing the precision and scope of measurements at the LHC and future colliders. Specific physics impacts include:

- Improved precision in jet substructure measurements, enabling better discrimination between different theoretical models of QCD radiation
- Enhanced sensitivity to effective field theory parameters through direct moment unfolding
- More robust uncertainty quantification in high-dimensional measurements
- Computational efficiency gains allowing for more detailed systematic studies
- Framework for incorporating detector response uncertainties in the unfolding process

By bridging sophisticated machine learning techniques with the specific requirements of particle physics measurements, this work aims to advance our ability to extract fundamental physical insights from complex experimental data. The methods developed here have applications beyond particle physics, potentially benefiting any field where deconvolution of instrumental effects is necessary for scientific interpretation.

Chapter 2

Theoretical Foundations

2.1 Statistical Formulation of the Unfolding Problem

Unfolding, also known as deconvolution, is the process of correcting detector distortions in experimental data to recover the true particle–level distributions. This procedure is critical for comparing experimental results with theoretical predictions and for enabling detector–independent analyses. The unfolding problem is inherently statistical and presents unique challenges due to its ill-posed nature.

The Detector Response and Forward Problem

The relationship between the particle–level truth distribution $p_{\text{truth}}(z)$ and the detector–level measured distribution $p_{\text{measured}}(x)$ is governed by the detector response function $R(x|z)$, which encapsulates the effects of resolution, efficiency, and acceptance:

$$p_{\text{measured}}(x) = \int R(x|z)\; p_{\text{truth}}(z)\, \mathrm{d}z. \tag{2.1}$$

This equation describes the **forward problem**, where the true distribution $p_{\text{truth}}(z)$ is mapped to the measured distribution $p_{\text{measured}}(x)$. The detector response function $R(x|z)$ can often be determined through detailed simulations.

The Inverse Problem: Unfolding

The goal of unfolding is to invert the forward problem and estimate $p_{\text{truth}}(z)$ from observed data $p_{\text{measured}}(x)$. Mathematically, this requires solving:

$$p_{\text{truth}}(z) = \int R^{-1}(z|x)\; p_{\text{measured}}(x)\, \mathrm{d}x, \tag{2.2}$$

where $R^{-1}(z|x)$ represents the inverse response function. However, this inversion is ill–posed because small fluctuations in $p_{\text{measured}}(x)$ can lead to large variations in $p_{\text{truth}}(z)$. Regularization techniques are therefore essential to stabilize the solution.

Likelihood-Based Formulation

In practice, unfolding is performed using statistical inference methods. Given a set of measured data $\mathbf{X}_{i=1}^{N}$, the likelihood function for a proposed truth distribution $p_{\text{truth}}(z;\theta)$, parameterized by θ, is:

$$\mathcal{L}(\theta;\mathbf{X}) = \prod_{i=1}^{N} p_{\text{measured}}(x_i;\ \theta), \tag{2.3}$$

where

$$p_{\text{measured}}(x;\theta) = \int R(x|z)\ p_{\text{truth}}(x;\theta)\,\mathrm{d}z. \tag{2.4}$$

Maximizing this likelihood yields an estimate of the parameters θ, which define the unfolded truth distribution. Regularization can be incorporated into this framework by adding penalty terms to the likelihood or by constraining the parameter space.

Regularization Techniques

Regularization mitigates the instability of unfolding by imposing constraints on the solution. Common approaches include Tikhonov Regularization, in which one adds a penalty term proportional to the norm of the second derivative of $p_{\text{truth}}(z)$, enforcing smoothness, and Iterative Methods which gradually refine estimates of $p_{\text{truth}}(z)$, regularizing by stopping before convergence. These techniques balance fidelity to the measured data (prior independence) with stability of the unfolded solution.

Challenges in High-Dimensional Phase Spaces

Traditional unfolding methods rely on binning to discretize phase spaces into histograms. However, binning introduces artifacts such as bias and loss of resolution, particularly in high-dimensional phase spaces where binning becomes computationally prohibitive. These limitations motivate unbinned approaches that operate directly on event–level data.

Unbinned methods leverage machine learning or statistical techniques to model $R(x|z)$ and infer $p_{\text{truth}}(z)$ without discretization. While promising, these methods must contend with increased computational complexity and sensitivity to model assumptions. This framework lays the foundation for exploring modern machine learning approaches that address these challenges, as discussed in subsequent sections.

2.2 Forward and Inverse Problems in HEP

The measurement process in high-energy physics experiments inherently involves two complementary mathematical challenges: the forward problem of predicting detector responses from particle-level interactions, and the inverse problem of recovering true physics distributions from observed detector measurements. These twin challenges form the conceptual foundation for understanding detector effects and developing unfolding methodologies.

Mathematical Formulation

The relationship between particle–level truth distributions and detector–level observations is governed by the Fredholm integral equation of the first kind**[cite –KD]**:

$$g(s) = \int_{\Omega} K(s, y) f(y)\, \mathrm{d}y + \epsilon(s), \tag{2.5}$$

where $f(y)$ represents the true particle-level distribution, $K(s, y)$ is the detector response kernel encoding resolution effects and acceptance, $g(s)$ is the observed detector-level distribution, and $\epsilon(s)$ accounts for measurement noise.**[cite –KD]** This equation encapsulates the *forward problem* when predicting $g(s)$ given $f(y)$, and the *inverse problem* when estimating $f(y)$ from measurements of $g(s)$.

For discrete histogram representations, this becomes a matrix equation:

$$\boldsymbol{\mu} = \mathbf{K}\boldsymbol{\nu} + \boldsymbol{\epsilon}, \tag{2.6}$$

where $\boldsymbol{\nu}$ and $\boldsymbol{\mu}$ are vectors of true and observed bin counts, respectively, and $\mathbf{K}$ is the smearing matrix containing conditional probabilities $K_{ij} = P(\text{observed bin } i | \text{true bin } j)$**[cite –KD]**.

Challenges in Inverse Problems

The inverse problem in HEP is fundamentally and intrinsically ill-posed. The response kernel is non–injective, i.e. different true distributions can produce identical observed distributions after detector smearing**[cite –KD]**. Furthermore, the distributions are ill–conditioned; small measurement errors $\boldsymbol{\epsilon}$ amplify into large fluctuations in unfolded solutions due to small singular values in $\mathbf{K}$**[cite –KD]**. These intrinsic challenges with inverse problems are compounded by the fact that modern analyses involve a large number of observables, making brute-force phase space discretization computationally prohibitive**[cite –KD]**.

These challenges necessitate regularization techniques that impose physical constraints on solutions, such as Tikhonov regularization or iteration cut-offs before convergence, described above.

HEP Specific Considerations

Three unique aspects complicate inverse problems in particle physics compared to other domains. First, Poisson statistics dominate in low-count regions of histograms.**[cite –KD]**. Second, detector response models contain large numbers of correlated nuisance parameters, complicating the systematics.**[cite –KD]**. Finally, the smearing matrix $\mathbf{R}$ itself depends on Monte Carlo simulations of detector physics.**[cite –KD]**

A representative example is top quark mass measurement, where the true mass m_t affects both production kinematics and decay signatures. The forward problem involves simulating $pp \to t\bar{t}$ events through PYTHIA and detector response via GEANT, while the inverse problem requires extracting m_t from reconstructed jet energies and lepton momenta, while suitably handling the correlated systematics **[cite –KD]**.

Applications and Implications

Proper handling of forward/inverse problems enables precision Standard Model tests. Unfolded W boson mass measurements test electroweak theory predictions**[cite –KD]** and are a classic example of high–precision tests of Standard Model predictions that require unfolding. Unfolding also plays a crucial role in analyses involved in searches for new physics. Resonance searches in $H \to \gamma\gamma$ require correcting detector diphoton mass resolution**[cite –KD]** as a fundamental step in the analysis.

Although in some cases, it can be feasible to instead forward fold theoretical predictions, unfolding is unavoidable in theory agnostic experiment–to–experiment comparisions, and detector-corrected results remain comparable to future theoretical calculations enabling legacy analyses.**[cite –KD]**

Recent advances leverage machine learning to address traditional limitations. For example, Neural Conditional Density Estimators model $p(z|\theta)$ directly using normalizing flows, and differentiable simulation methods enable gradient-based optimization through approximate detector models.**[cite –KD]**

The tension between statistical rigor and computational feasibility remains acute, particularly for high–dimensional phase spaces. Modern solutions increasingly combine traditional regularization with learned representations that encode physical constraints implicitly through neural network architectures.

2.3 Historical development: From matrix inversion to modern approaches

The problem of unfolding has a rich history in high energy physics, with methods evolving alongside computational capabilities and statistical sophistication. Early approaches relied primarily on simple correction factors applied to individual bins of histograms, appropriate only when detector effects were minimal.

As measurements became more precise, matrix inversion techniques emerged as the standard approach. These methods discretize both the particle-level and detector-level distributions into bins, relating them through a response matrix $\mathbf{R}_{ij}$ that describes the probability for an event in particle–level bin j to be observed in detector–level bin i. The unfolding problem then becomes

$$\mu_i \text{detector} = \sum_j R_{ij} \nu_j \text{particle} \tag{2.7}$$

where $\mu_{i,\text{detector}}$ is the expected number of counts in detector–level bin i and $\nu_{j,\text{particle}}$ is the expected number of events in particle–level bin j. Naively, one might attempt to solve this system by simply inverting the response matrix:

$$\nu_j^{\text{particle}} = \sum_i (R^{-1})_{ji} \mu_i^{\text{detector}} \tag{2.8}$$

However, this direct inversion leads to wildly oscillating solutions with large variances—a manifestation of the ill-posed nature of the unfolding problem. To address this issue, regularization techniques were developed to stabilize the solution by imposing additional constraints.

Several regularized unfolding methods emerged as standards in the field:

- **Iterative Bayesian unfolding [cite –KD]**: Uses Bayes' theorem to iteratively update the estimate of the true distribution, with the number of iterations controlling regularization strength.
- **SVD unfolding [cite –KD]**: Applies singular value decomposition to the response matrix and suppresses contributions from small singular values that amplify statistical fluctuations.
- **TUnfold[cite –KD]**: Formulates unfolding as a least-squares problem with Tikhonov regularization to penalize large second derivatives, preserving smoothness.

These methods have served the field well for decades, particularly for one–dimensional measurements where binning is manageable. However, they all share the common limitation of requiring discretization of the underlying distributions, which becomes increasingly problematic as measurements probe higher–dimensional spaces and more complex observables.

2.4 Traditional Unfolding Methods in Experimental Analyses

Traditional unfolding methods form the bedrock of detector corrections in high-energy physics (HEP), balancing statistical rigor with computational practicality. This section provides an overview of established techniques, their mathematical foundations, implementation nuances, and limitations.

Bin–by–Bin Correction

The simplest unfolding approach applies multiplicative correction factors to observed bin counts:

$$\hat{\nu}_j = \frac{\mu_j - b_j}{C_j}, \quad C_j = \frac{\nu_j^{\text{MC}}}{\mu_j^{\text{MC}}}, \tag{2.9}$$

where μ_j is the observed count in bin j, b_j the estimated background, and C_j the correction factor derived from Monte Carlo (MC) simulations relating particle–level generated (ν_j^{MC}) and detector–level simulated (μ_j^{MC}) events**[cite –KD]**.

This method has the advantange of being computationally trivial, with no bin-to-bin correlations. However, it assumes perfect MC agreement with data migration patterns and fails to account for a non-diagonal response matrix ($\exists i \neq j : \mathbf{R}_{ij} \neq 0$. This is illustrated most dramatically by the observation that biases persist even with $C_j \to 1$ due to ignored cross-bin migrations.**[cite –KD]**

Used primarily in early LHC analyses (e.g., ATLAS jet cross-sections**[cite –KD]**), bin–by–bin correction remains viable only for coarse binnings with negligible migration ($< 5\%$) between adjacent bins.

Matrix Inversion

When $n_{\text{bins, truth}} = n_{\text{bins, reco}}$, the response matrix

$$\mathbf{R}$$

is square. Formally one can write the unfolded solution as

$$\hat{\boldsymbol{\nu}} = \mathbf{R}^{-1}\boldsymbol{\mu}, \tag{2.10}$$

and even propograte the covariance as

$$V_{\hat{\boldsymbol{\nu}}} = \mathbf{R}^{-1} V_{\boldsymbol{\mu}} (\mathbf{R}^{-1})^T. \tag{2.11}$$

However, in practice, direct inversion is highly pathological. This pathology can be quantified by the condition number

$$\kappa(R^{-1}) = \frac{|\lambda_{\max}(R^{-1})|}{|\lambda_{\min}(R^{-1})|} \sim 10^3 - 10^6 \tag{2.12}$$

where $\lambda_{\max}(R^{-1})$ and $\lambda_{\min}(R^{-1})$ are the largest and smallest eigenvalues of R^{-1} respectively. The condition number measures how much a perturbation in the measured counts $\delta\mu$ perturbs the predicted truth counts $\delta\nu$. The large condition number amplifies statistical fluctuations.**[cite –KD]** Further, unphysical solutions such as negative bin count values can arise from noise-dominated eigenvectors.

Methods have been suggested to control this variance, such as Truncated SVD, involving discard singular values $\sigma_i < \lambda_{\text{cut}}$**[cite –KD]**, and Wiener-SVD, a frequency-domain filtering method to maximize signal-to-noise ratio.**[cite –KD]** Despite this, matrix inversion's instability limits utility to toy models. No modern analysis uses pure inversion without regularization.

Iterative Bayesian Unfolding (D'Agostini)

This Expectation-Maximization (EM) algorithm iteratively updates truth estimates:

$$\nu_j^{(k+1)} = \nu_j^{(k)} \sum_{i=1}^{N_{\text{Data}}} \frac{R_{ij}\ \mu_i}{\sum_{l=1}^{N_{\text{Truth}}} R_{il}\ \nu_l^{(k)}} \tag{2.13}$$

This method regularization via stopping, by terminating at $k \sim 4 - 6$ iterations before noise amplification**[cite –KD]**. The initial guess $\boldsymbol{\nu}^{(0)}$ biases the solution. Some common choices include Generation $\boldsymbol{\nu}_{\text{MC}}$, a uniform distribution, and Data-driven backwards folding $\mathbf{R}^T\mathbf{mu}$.

IBU is Dominant in LHC analyses (e.g., ATLAS top mass measurements**[cite –KD]**) because it balances simplicity with moderate-dimensional phase spaces ($N_{\text{Truth}} \leq 20$).

Tikhonov Regularization

Tikhonov regularization is a penalized least-squares minimization method

$$\hat{\boldsymbol{\nu}} = \arg\min_{\boldsymbol{\nu}} \left[||\boldsymbol{\mu} - \mathbf{R}\boldsymbol{\nu}||^2 + \lambda ||\mathbf{L}(\boldsymbol{\nu} - \boldsymbol{\nu}_0)||^2 \right] \tag{2.14}$$

where $\mathbf{L}$ is typically the discrete curvature operator and $\boldsymbol{\mu}_0$ a prior estimate.**[cite –KD]** L-curve optimization balances residual norm vs. solution norm to choose λ.**[cite –KD]** The choice of λ sets the bias variance tradeoff. $\lambda \to 0$ represents the high variance, low bias limit and $\lambda \to \infty (\implies \hat{\boldsymbol{\nu}} \to \boldsymbol{\nu}_0)$ represents the low variance, high bias limit. This method is implemented through the TUnfold package, which also provides automated λ tuning via global correlation minimization.**[cite –KD]**

Tikhonov regularization is the preferred method for precision SM measurements (e.g., W boson mass**[cite –KD]**), because it excels when smooth spectra are expected. However this method struggles with non-differentiable features like threshold effects.

Template Fitting

Template fitting is a method suitable in cases where $N_{\text{Data}} \gg N_{\text{Truth}}$. In this case, one can construct detector–level templates for each truth bin:

$$\mu_i = \sum_{j=1}^{N_{\text{Truth}}} R_{ij}\nu_j + b_i \tag{2.15}$$

with χ^2 minimization:

$$\chi^2 = \sum_{i=1}^{N_{\text{Data}}} \frac{(\mu_i - \sum_j R_{ij}\nu_j - b_i)^2}{\sigma_i^2} \tag{2.16}$$

The solution then is overconstrained, since we leverage $N_{\text{Data}}/N_{\text{Truth}} \sim 2-3$ for stability.**[cite –KD]** Nuisance parameters are systematically modeled via template morphing.**[cite –KD]** Template fitting requires dense detector-level binning, which inflates statistical uncertainties. Template fitting is commonly used in Higgs coupling measurements where broad mass resolutions necessitate wide truth bins.

Summary

Table **??** summarizes the strengths and limitations of the methods discussed above. These limitations motivated the use of machine learning in unfolding, a transition explored in subsequent sections. However, traditional methods remain indispensable for validation and low–dimensional precision measurements where interpretability is crucial.

Method	Dimensionality	MC Dependence	Uncertainty Propagation
Bin-by-bin	1D	Extreme	Underestimated
Matrix Inversion	$\leq$10D	None	Exact but unstable
D'Agostini	$\leq$20D	Moderate	Partial
Tikhonov	$\leq$50D	Moderate	Full
Template Fit	$\leq$10D	Low	Full

Table 2.1: Traditional unfolding method capabilities. Dimensionality refers to practical limits.

Regularization: Need, Approaches, and Limitations

The inherent ill–posedness of unfolding necessitates regularization to stabilize solutions against statistical fluctuations while preserving physical meaning. This section systematically examines the theoretical justification for regularization, surveys dominant methodologies, and critically evaluates their limitations in high energy physics applications.

The Necessity of Regularization

As discussed earlier, unfolding inverse problems in HEP exhibit pathological characteristics that demand regularization. Regularization counteracts these issues by introducing prior knowledge about $p(z)$, typically favoring smoothness or similarity to Monte Carlo (MC) predictions. However, as Zech emphasizes in **[cite –KD]**, this unavoidably discards information—regularized solutions cannot resolve features finer than the detector resolution or distinguish theories predicting distributions within the regularization bias.

Established Regularization Approaches

Tikhonov Regularization This widespread method minimizes the penalized least-squares functional:

$$\min_{\boldsymbol{\mu}} ||\mathbf{mu} - \mathbf{R}\boldsymbol{\nu}||^2 + \lambda||\mathbf{L}(\boldsymbol{\nu} - \boldsymbol{\nu}_{\text{MC}})||^2 \tag{2.17}$$

where $\mathbf{L}$ imposes smoothness (e.g., discrete second derivatives) and $\boldsymbol{\nu}_{\text{MC}}$ anchors solutions to MC predictions.**[cite –KD]**. Singular Value Decomposition (SVD) implementations truncate small singular values $\sigma_i < \lambda$, suppressing high-frequency noise**[cite –KD]**. While effective for moderate dimensions, Tikhonov methods introduce model dependence through directly on $\boldsymbol{\mu}_{\text{MC}}$ (rather than for instance on $\mathbf{R}$)**[cite –KD]**, struggle with sharply falling spectra due to biased curvature penalties **[cite –KD]**, and require ad hoc

$$\lambda$$

selection, often via L-curve curvature maximization**[cite –KD]**.

Bayesian Unfolding Bayesian methods regularize through prior distributions $p(z)$, yielding posterior estimates:

$$p(z|x) \propto \mathcal{L}(x|z)p(z) \tag{2.18}$$

Common priors include entropy maximization $p(z) \propto \exp(-\sum z_j \log z_j)$**[cite –KD]** and Gaussian processes enforcing smoothness**[cite –KD]**. These provide natural uncertainty quantification but suffer from high computational cost, scaling poorly with dimensionality **[cite –KD]**,sensitivity to prior misspecification, especially in low-statistics regions**[cite –KD]**, and difficulty interpreting credible intervals as frequentist coverage**[cite –KD]**

Iterative Methods The D'Agostini algorithm**[cite –KD]**, also known as Iterative Bayesian Unfolding (IBU)**[cite –KD]** applies expectation–maximization iterations:

$$z_j^{(k+1)} = z_j^{(k)} \sum_i \frac{R_{ij}\ x_i}{\sum_l R_{il}\ z_l^{(k)}} \tag{2.19}$$

Early termination (typically $k \sim 4-6$) acts as implicit regularization by preventing overfitting**[cite –KD]**. While computationally efficient, this approach lacks objective stopping criteria, requiring heuristic cross-validation**[cite –KD]** and underestimates uncertainties due to ignored iteration-dependent covariance**[cite –KD]**.

Regularized Poisson Likelihood For low-statistics regions, Gaponenko**[cite –KD]** advocates minimizing:

$$-\log \mathcal{L}(x|z) + \lambda S(z) \tag{2.20}$$

$S(z)$ penalizes non-monotonicity in sharply falling spectra. Using cubic B-splines with entropy regularization, this method avoids binning artifacts through continuous representations**[cite –KD]**. However, it requires careful basis function placement to prevent endpoint spikes**[cite –KD]** and demands specialized optimization protocols (e.g., cooling schedules for λ).**[cite –KD]**

Limitations and Practical Challenges

Subjectivity-Objectivity Trade-off All regularization methods inject subjective choices—smoothness scales, prior distributions, stopping criteria and so on—that bias results. As shown in Figure 3 of **[cite –KD]**, Tikhonov regularization artificially suppresses true peaks when λ over-penalizes curvature. Zech**[cite –KD]** and Kuusela**[cite –KD]** argue this necessitates publishing unregularized results for theory comparisons, reserving regularization only for visualization.

High-Dimensional Regimes Traditional methods fail catastrophically in

$$d \geq 4$$

phase spaces for multiple reasons. Binned approaches require n^d histogram bins, running up against memory limits**[cite –KD]**, and struggling to effectively sample an increasingly sparse phase space. Global smoothness assumptions become untenable for multi-scale features**[cite –KD]** straining regularization methods that rely on them.

As the number of dimensions increases, the binning also increasingly distorts error propagation. Bayesian credible intervals exhibit poor frequentist coverage (Figure 4 in **[cite –KD]**), and correlated systematic uncertainties (e.g., jet energy scale) introduce non-convex likelihoods.**[cite –KD]**

Spectrum-Dependent Biases Sharply falling spectra (e.g., proton momentum in **[cite –KD]**) exacerbate regularization artifacts. Entropic priors overweight high-

$$z$$

regions, distorting tails**[cite –KD]**, Finite sample sizes truncate measurable phase space, creating cutoff-induced spikes,**[cite –KD]** and curvature penalties conflict with natural spectral shapes, requiring physics-informed $S(z)$**[cite –KD]**.

Recent advances aim to mitigate these limitations in various ways. For example, adversarial regularization involves training discriminators to enforce physical consistency rather than explicit smoothness**[cite –KD]**. Differentiable unfolding methods embed detector response in neural networks enabling gradient-based λ optimization**[cite –KD]**. However, no universal solution exists. The choice of regularization must align with analysis–specific priorities: theory comparison, visualization, or parameter extraction. As detector granularity increases, developing dimension–agnostic regularization schemes remains an open challenge requiring collaboration between statisticians and physicists.

2.5 Limitations of Traditional Binned Methods

While binned unfolding methods remain workhorses of experimental particle physics, they face significant limitations that have motivated the development of new approaches:

- **Curse of dimensionality**: As the dimensionality of the measurement increases, the number of bins required to cover the phase space grows exponentially, leading to sparsely populated bins and unstable unfolding.

- **Binning bias**: The subjective choice of binning scheme biases the result and can obscure fine features in the distributions.
- **Binning artifacts**: Binning inherently discards information about the precise values of observables within each bin, reducing statistical power, and introducing binning artifacts.
- **Regularization ambiguity**: The choice of regularization scheme and strength significantly impacts results, with no universally accepted procedure for regularizing the unfolding.
- **Cross–bin correlations**: Binned methods often struggle to properly account for correlations between bins as the dimension of the phase space increases.

These limitations become particularly acute in modern analyses that target complex, high dimensional observables with non–trivial correlations. For instance, in jet substructure measurements, the relationship between different substructure variables contains valuable information about the underlying physics that can be obscured by independent binning of each variable **[cite –KD]**.

Moreover, many theoretical predictions in particle physics are at the level of statistical moments or other distribution properties rather than full differential spectra. Traditional unfolding methods require first unfolding the full distribution and then calculating these properties, which can lead to reduced precision in the moment predictions.

2.6 From Binned to Unbinned Methods: Statistical Considerations

The evolution from binned to unbinned unfolding methodologies represents a paradigm shift in high-energy physics, driven by the need to preserve fine-grained kinematic information while managing the statistical and computational complexities of high-dimensional phase spaces. This section systematically analyzes the theoretical foundations, practical challenges, and performance trade-offs that this transition entails.

Statistical Foundations of Binned Unfolding

Traditional binned methods discretize particle-level (Z) and detector-level (X) observables into histograms, reducing the Fredholm integral equation to matrix form:

$$\boldsymbol{\mu} = \mathbf{R}\boldsymbol{\nu} + \boldsymbol{\epsilon}, \tag{2.21}$$

where $\boldsymbol{\nu} \in \mathbb{R}^{N_{\text{Truth}}}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\text{Data}}}$ are truth and data histogram counts, $\mathbf{R}$ is the smearing matrix, and $\boldsymbol{\epsilon}$ models statistical noise.**[cite –KD]** Maximum likelihood estimation via Poisson statistics remains the gold standard:

$$\mathcal{L}(\boldsymbol{\nu}) = \prod_{i=1}^{N_{\text{Data}}} \frac{(\mathbf{R}\boldsymbol{\nu})_i^{\mu_i} e^{-(\mathbf{R}\boldsymbol{\nu})_i}}{\mu_i!}. \tag{2.22}$$

In addition to the limitations discussed in earlier sections, integrated correlations between observables due to binning obscure multi-differential features.**[cite –KD]** In binned analyses, sharp spectral features (e.g., resonance peaks) are artificially broadened.**[cite –KD]** In the case of TUnfold, Tikhonov smoothing also conflates detector resolution effects with physical spectral curvature,**[cite –KD]** further compounding the artifacts.

Sparse bin populations in high dimensions amplify statistical uncertainties, destabilizing inversion:

$$\text{Relative uncertainty} \propto \frac{1}{\sqrt{N_{\text{events}}} \cdot \prod_{i=1}^{d} \Delta z_i}, \tag{2.23}$$

where Δz_i are bin widths. This forces analysts to integrate over variables, discarding critical correlations needed for precision Standard Model tests or new physics searches.**[cite –KD]**

Unbinned Methodologies: Principles and Implementations

Unbinned unfolding operates directly on event tuples $\{(z_1, x_1), ..., (z_N, x_N)\}$, preserving full kinematic information. One class of implementations leverages machine learning to estimate probability density ratios:

$$w(z) = \frac{p_{\text{Truth}}(z)}{p_{\text{Gen.}}(z)}, \quad \nu(m) = \frac{p_{\text{Data}}(x)}{p_{\text{Sim.}}(x)}, \tag{2.24}$$

where $w(z)$ reweights particle–level MC to match data, and $v(x)$ corrects detector–level distributions.**[cite –KD]** A classic example of such an approach is OmniFold.

OmniFold (Iterative Reweighting)

OmniFold trains classifiers iteratively in an IBU–style, unbinned, expectation–maximization algorithm:

$$w^{(k+1)}(z) = \left.\frac{p_{\text{Data}}(x)}{p^{(k)}_{\text{Sim.}}(x)}\right|_{x=f_{\text{det}}(z)}, \tag{2.25}$$

$$p_{(k+1)}(z) = w^{(k+1)}(z)\, p^{(k)}(z), \tag{2.26}$$

where $f_{\text{det}}(z)$ is the function that maps a particular z to the corresponding x via detector simulation.**[cite –KD]**

cINN (Conditional Invertible Networks)

A distinct class of methods seek to train generative models to learn the diffeomorphic mapping

$$g_\theta : \mathcal{Z} \to \mathcal{X}, \tag{2.27}$$

with a tractable Jacobian,

$$p(z|x) = p(x|z)\frac{p(z)}{p(x)} = \left|\det \frac{\partial g_\theta}{\partial z}\right|^{-1} p(g_\theta(z)). \tag{2.28}$$

This enables direct sampling from $p(z|m)$ without iterations.**[cite –KD]** Generative models however, are susceptible to mode collapse.**[cite –KD]** Schrödinger Bridge Unfolding is an method deleloped to address mode collapse.**[cite –KD]** It is an optimal transport formulation minimizing KL divergence between joint distributions:

$$\inf_{p(z,x)} D_{\text{KL}}\Big(p(z,x) \parallel p_{\text{MC}}(z,x)\Big) \text{ s.t. } p(x) = p_{\text{Data}}(x). \tag{2.29}$$

Statistical Considerations in Unbinned Regimes

Neural networks implicitly regularize via inductive controls. E.g., convolutional layers enforce translational symmetry in jet images.**[cite –KD]** However, this introduces model-dependent smoothing scales requiring careful validation against closure tests.**[cite –KD]**

Reweighting based methods propagate uncertainties through event weights:

$$\text{Cov}[O] = \sum_{i=1}^{N} w_i^2 O(z_i)^2 - \left(\sum_{i=1}^{N} w_i O(z_i)\right)^2, \tag{2.30}$$

for observable $O(z)$**[cite –KD]**. While avoiding binning-induced correlations, unbinned inference requires re-conceptualizing what the unbinned equivalent of the covariance matrix would be in order to appropriately account for correlations in the unfolded data.

Acceptance Effects Detector fiducial volumes impose phase space cuts $A(z) \in \{0, 1\}$. Unbinned methods can correct acceptance via reweighting:

$$w_{\text{acc}}(z) = \frac{A_{\text{data}}(z)}{A_{\text{MC}}(z)}, \tag{2.31}$$

but require dense coverage of Z near boundaries to avoid edge artifacts.**[cite –KD]**

Limitations in Complex Phase Spaces

Model Misspecification

Generative models assume $p_{\text{Gen.}}(z) > 0 \implies p_{\text{Truth}}(z) > 0$. New physics signatures outside the support of Generation ($p_{\text{Gen.}}(z) = 0$) is difficult to detect.**[cite –KD]** Hybrid approaches combining discriminative and generative components show promise for anomaly detection.**[cite –KD]**

Computational Scaling

While unbinned methods avoid exponential bin scaling, neural network training time grows super-linearly with event multiplicity. For $N_{\text{events}} \sim 10^8$, distributed training across GPU clusters becomes essential.**[cite –KD]** Noise-smeared likelihoods in $d \geq 4$ phase spaces also complicate posterior calibration. The *statistical coverage gap*–where 68% credible intervals contain true values only 50% of the time–persists despite adaptive methods.**[cite –KD]**

Future Directions

Machine Learning for unfolding is a rapidly evolving field, and emerging techniques might have the potential to mitigate current limitations. Some examples are

- **Differentiable detectors**: End-to-end gradient propagation through approximate detector simulations enables precise Jacobian calculations.**[cite –KD]**
- **Equivariant architectures**: Built-in symmetry constraints (e.g., Lorentz invariance) reduce hypothesis space while preserving physical consistency.**[cite –KD]**
- **Foundational models**: Pre-trained on broad MC datasets, these allow rapid fine-tuning for specific analyses, amortizing computational costs.**[cite –KD]**

A summary comparison between binned and unbinned methods is provided in Table **??**. The values in the table should be treated as order of magnitude estimates provided to suggest rough scales, rather than precise reports. The transition to unbinned methodologies represents not merely a technical advancement but a fundamental reorientation toward maximal information preservation. As experimental statistics grow at the HL-LHC and future colliders, these methods will become indispensable tools for precision physics.**[cite. –KD]**

Metric	**Binned**	**Unbinned**
Computational complexity	$\mathcal{O}(n^{2d})$	$\mathcal{O}(N_{\text{events}})$
Memory footprint (4D, $n = 10$)	100 GB	1 GB
Systematic uncertainty propagation	Analytical	Monte Carlo
Feature resolution limit	$\sim 2\sigma_{\text{det}}$	$\sim \sigma_{\text{det}}/\sqrt{N}$
Multi-observable correlations	Integrated	Preserved

Table 2.2: Comparative performance of unfolding methods in high-dimensional regimes

2.7 Evaluation metrics for unfolding

The evaluation of unfolding methods presents unique challenges due to the ill-posed nature of the inverse problem. While the goal of unfolding is conceptually straightforward, to recover the true particle–level distribution from detector–level observations, quantifying the success of this recovery requires careful consideration. This chapter discusses various metrics and approaches for evaluating unfolding performance, considering both traditional binned techniques and modern unbinned methods. We examine metrics for assessing accuracy, precision, and uncertainty quantification, as well as practical considerations for their application in high-energy physics analyses.

Statistical Metrics for Evaluating Point Estimates

Residual-Based Metrics

The most intuitive approach to evaluating an unfolding method is to compare the unfolded distribution to the true distribution when it is known (e.g., in simulation studies). Simple residual–based metrics quantify the difference between the estimated and true distributions. For binned methods, the bin–by–bin residual is defined as:

$$\delta_i = \hat{t}_i - t_i \tag{2.32}$$

where $\hat{t}_i$ is the unfolded count in bin i and t_i is the true count. Various summary statistics of these residuals can be computed, including:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{t}_i - t_i)^2 \tag{2.33}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{t}_i - t_i)^2} \tag{2.34}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{t}_i - t_i| \tag{2.35}$$

While these metrics are straightforward, they have limitations in the context of unfolding. In particular, they treat all bins equally, even though certain regions of phase space may be physically more significant than others. Additionally, these metrics do not account for the correlations between bins introduced by the unfolding process.

For unbinned methods, where the output is a set of weights or a continuous probability density, these metrics must be adapted. One approach is to bin the unbinned unfolded distributions and then apply the above metrics, though this introduces binning artifacts that the unbinned method was designed to avoid.

Distributional Distance Metrics

Given the limitations of simple residual metrics, distributional distance measures provide a more comprehensive assessment of unfolding performance. These metrics compare the entire unfolded distribution to the true distribution.

The Kullback-Leibler (KL) Divergence measures the information lost when using the unfolded distribution to approximate the true distribution.

$$D_{\text{KL}}(p||q) = \int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x, \tag{2.36}$$

where p is the true distribution and q is the unfolded distribution. While theoretically sound, KL divergence can be numerically unstable when the support of the distributions differs.

The Vincze-Le Cam (VLC) Divergence is symmetric alternative to KL divergence that is both bounded and highly convex.

$$\Delta(p,q) = \frac{1}{2}\int \frac{(p(\lambda)-q(\lambda))^2}{p(\lambda)+q(\lambda)}\,\mathrm{d}\lambda. \tag{2.37}$$

Table 2.3: Distributional distance metrics for unfolding evaluation

	Symmetric	Bounded	Support sensitivity	Compute Cost
D_{KL}	No	No	High	Low
Δ_{VLC}	Yes	Yes	Medium	Low
W_2	Yes	No	Low	High

This metric is particularly useful for comparing unfolding methods as it provides a balanced assessment of differences across the entire distribution and has been used in comparative analyses of various unfolding approaches.

The Wasserstein Distance, also known as the “Earth Mover’s Distance," provides a measure of the minimum “work" required to transform one distribution into another.

$$W_p(p,q) = \left(\inf_{\gamma \in \Gamma(p,q)} \int \int |x-y|^p d\gamma(x,y) \right)^{1/p}, \tag{2.38}$$

where $\Gamma(p,q)$ is the set of all joint distributions with marginals p and q. This metric is particularly useful for unfolding evaluations as it accounts for both the magnitude and location of discrepancies between distributions.

Table **??** summarizes the various distributional metrics and their relative strengths for unfolding evaluation.

Uncertainty Quantification Metrics

Beyond point estimates, properly evaluating unfolding methods requires assessing the accuracy of their uncertainty estimates. This is particularly important in particle physics, where uncertainties propagate to downstream analyses such as parameter fitting.

Pull Distributions

Pull distributions offer a rigorous way to evaluate the calibration of reported uncertainties. For a given unfolded bin or parameter θ, the pull is defined as:

$$\text{Pull}\,\theta = \frac{\hat{\theta} - \theta_{\text{true}}}{\sigma_{\hat{\theta}}} \tag{2.39}$$

where $\hat{\theta}$ is the unfolded estimate, θ_{true} is the true value, and $\sigma_{\hat{\theta}}$ is the reported uncertainty. For a well-calibrated method, the pull distribution across many pseudo–experiments should

follow a standard normal distribution, $\mathcal{N}(0, 1)$. Deviations from this indicate either overestimation or underestimation of uncertainties.

In the context of binned unfolding, pull distributions can be computed for each bin, while for unbinned methods, they can be applied to derived quantities or parameters of interest. For Bayesian methods, pulls can be calculated using the mean and standard deviation of the posterior distribution.

Coverage Properties

Related to pulls but more direct is the evaluation of coverage properties of confidence or credible intervals. For a nominal 68% confidence interval, approximately 68% of intervals computed across many pseudo-experiments should contain the true value. Systematic deviations from nominal coverage indicate issues with the uncertainty estimation. Coverage can be assessed through closure tests. These involve generating multiple datasets from a known truth, applying the unfolding procedure, and checking the fraction of times the true value falls within the reported confidence intervals. Coverage plots plot the actual coverage versus the nominal coverage across different confidence levels. The example in Figure **[find a suitable figure –KD]** shows expected coverage properties for both well-calibrated and miscalibrated unfolding methods.

Variance and Bias Decomposition

The total error of an unfolding method can be decomposed into bias and variance components,

$$\text{MSE}(\hat{t}) = \text{Bias}^2(\hat{t}) + \text{Var}(\hat{t});, \tag{2.40}$$

where $\text{Bias}(\hat{t}) = \mathbb{E}[\hat{t}] - t$ and $\text{Var}(\hat{t}) = \mathbb{E}[(\hat{t} - \mathbb{E}[\hat{t}])^2]$.

This decomposition is particularly valuable for understanding the trade–offs inherent in regularized unfolding methods, where stronger regularization typically reduces variance at the expense of increased bias. Different applications might prioritize minimizing one component over the other, making this decomposition essential for method selection.

Evaluation of Correlation Structure

Traditional evaluation metrics often focus on marginal distributions, overlooking an important aspect of unfolding: the correlation structure between different bins or events. Properly accounting for these correlations is crucial for downstream analyses.

Covariance Matrix Assessment

For binned methods, the full covariance matrix of the unfolded distribution provides information about bin-to-bin correlations. A useful visualization is the correlation matrix, defined as:

$$\text{Corr}_{ij} = \frac{\text{Cov}_{ij}}{\sqrt{\text{Cov}_{ii}\text{Cov}_{jj}}} \tag{2.41}$$

Comparing the correlation structure of the unfolded distribution to that of the true distribution (when known) can reveal systematic distortions introduced by the unfolding procedure.

Event-to-Event Correlation Metrics

For unbinned methods event–to–event correlations in the unfolded weights can significantly impact downstream inference. These correlations can be quantified by studying the weight correlation as a function of distance. For any pair of events, one can compute the correlation between their weights as a function of their distance in feature space. One can also estimate the reduction in statistical power due to correlated weights:

$$N_{\text{eff}} = \frac{\left(\sum_i w_i\right)^2}{\sum_i w_i^2 + 2\sum_{i<j} w_i w_j \text{Corr}(w_i, w_j)} \tag{2.42}$$

Figure **[Pick a figure –KD]** illustrates how event correlations typically decay with distance, with the correlation length scale increasing with detector resolution effects.

Method specific evaluation metrics

Iterative Methods

For iterative methods like Iterative Bayesian Unfolding (IBU) or OmniFold, convergence behavior provides important diagnostic information. To study the convergence behavior, we plot metric values (e.g., χ^2 or NLL) as a function of iteration number. We then compare unfolded distributions at different iterations to assess stability and analyze how the bias–variance tradeoff evolves with iteration number.

Bayesian Methods

For Bayesian unfolding methods such as Fully Bayesian Unfolding (FBU) or Neural Posterior Unfolding (NPU), additional posterior–specific metrics are relevant. We can compare detector–level data to detector–level predictions generated from the posterior. We assess convergence using standard MCMC based diagnostics like Gelman-Rubin statistics

or effective sample size. The width of the posterior allows us to evaluate the posterior uncertainty in relation to the true frequentist variance.

Practical Considerations

In general, when comparing different unfolding methods, a structured evaluation framework ensures fair and comprehensive assessment. Such a framework should consider

- Computational efficiency: Measure training time, inference time, and memory requirements.
- Dimensionality scaling: Assess how performance metrics change as the dimensionality of the problem increases.
- Prior dependence: Evaluate robustness to different initial simulations.
- Regularization parameter sensitivity: Compare how performance varies with changes in regularization strength.

In real experimental settings where the truth is unknown, evaluation presents additional challenges that require alternative pragmatic approaches. For example, when the true distribution is unavailable, data-splitting techniques can provide useful validation. The two most commonly used techniques are cross-validation, where we split the detector–level data, unfold one portion, then refold it, and compare predictions against the held–out portion; and boostrapping where we generate multiple resampled datasets to assess the stability of the unfolding procedure.

Closure tests involve applying the full analysis chain (forward model followed by unfolding) to a known input distribution. While not a direct evaluation of performance on real data, closure tests provide confidence in the methodology. The simplest kinds of closure tests involve apply detector simulation to a known particle–level distribution, then unfolding the resulting detector–level distribution and compare with the original input. This procedure can then be modified by using a different particle-level input than the one used to train the unfolding method, testing robustness to prior misspecification.

Evaluating how unfolding methods propagate systematic uncertainties is crucial for real-world applications. We can test the sensitivity of the method to systematic uncertainties by applying variations to the response matrix based on known systematic uncertainties and assessing the impact on unfolded distributions. For methods that support nuisance parameter profiling, evaluating how effectively nuisance parameters are profiled out is a gold standard test for the effectiveness of the method.

Rigorous evaluation of unfolding methods requires a multi–faceted approach that considers accuracy, uncertainty quantification, and computational performance. The metrics

and frameworks presented in this section provide a comprehensive foundation for assessing both traditional and machine learning-based unfolding techniques. For binned methods, established metrics like χ^2 and coverage tests remain valuable, while for unbinned approaches, distributional metrics like Wasserstein distance and VLC divergence offer more appropriate evaluation. Regardless of the method, uncertainty calibration through pull distributions and correlation structure assessment are important to validate any measurement. As unfolding methods continue to evolve, particularly with the advent of machine learning approaches, evaluation metrics must adapt accordingly. The framework presented here is designed to be extensible, accommodating new methods and application domains while maintaining rigor and comparability.

Chapter 3

Machine Learning for Unfolding

3.1 The Emergence of Machine Learning in Particle Physics

A Paradigm Shift in Data Analysis

The past decade has witnessed a remarkable transformation in how particle physicists analyze data, driven by the adoption and adaptation of machine learning (ML) techniques. This revolution has been catalyzed by the confluence of three factors: the growing volume and complexity of data from modern collider experiments, the substantial increase in available computing resources, and the rapid advancement of ML algorithms and frameworks. The Large Hadron Collider (LHC) alone produces petabytes of data annually, with individual collisions generating thousands of particles across multiple detector subsystems—creating a rich, high-dimensional dataset that traditional analysis methods struggle to fully exploit.**[cite –KD]** Particle physics presents unique analytical challenges that align particularly well with the strengths of modern machine learning approaches. The field routinely deals with high–dimensional feature spaces, complex non–linear correlations between observables, rare signal processes buried within overwhelming backgrounds, and some of the largest scientific datasets in existence.**[cite –KD]** These characteristics create an ideal testbed for advanced ML techniques, which excel at discovering patterns in precisely such environments.

The relationship between particle physics and machine learning is not entirely new. Neural networks were first applied to high energy physics problems in the late 1980s and 1990s.**[cite –KD]** However, these early applications were limited by computational resources and algorithmic capabilities. The contemporary renaissance began around 2012–2014, coinciding with the broader deep learning revolution across computer science.**[cite –KD]**

This timing allowed particle physicists to leverage developments in computer vision, reinforcement learning, and generative modeling, adapting these techniques to the unique requirements of high-energy physics.

Evolution of ML Applications in HEP

Classification Tasks

The first wave of modern ML applications in particle physics focused predominantly on classification tasks, particularly signal/background discrimination and particle identification. These problems are naturally framed as binary or multi–class classification, making them accessible entry points for machine learning techniques. Boosted decision trees initially dominated these applications, particularly in the analysis chains that led to the Higgs boson discovery.**[cite –KD]** However, deep neural networks quickly demonstrated superior performance by automatically discovering complex patterns in high–dimensional data, often outperforming carefully hand–crafted physics–inspired variables.**[cite –KD]** For example, neural networks trained on low–level detector information have been shown to match or exceed the performance of approaches using physics–motivated high–level features for tasks like quark–gluon discrimination**[cite –KD]** and top quark tagging.**[cite –KD]** The ATLAS and CMS experiments have now incorporated deep learning based taggers for identifying hadronically decaying W/Z bosons, top quarks, and Higgs bosons, significantly enhancing their sensitivity to new physics.**[cite –KD]** These applications benefit from convolutional neural networks' ability to exploit spatial correlations in calorimeter deposits, and recurrent or graph neural networks' capacity to handle variable–length, unordered collections of particles.

Regression and Anomaly Detection

As the field matured, ML applications expanded beyond classification to include regression tasks, such as energy calibration**[cite –KD]**, pileup mitigation**[cite –KD]**, and particle momentum reconstruction**[cite –KD]**. These applications require models to predict continuous quantities rather than discrete classes, introducing additional complexity but extendomg ML based methods to a larger class of problems. Simultaneously, unsupervised and weakly–supervised learning techniques emerged as powerful tools for anomaly detection, identifying potential new physics without explicit models**[cite –KD]**. These methods include autoencoders that flag events with high reconstruction loss, density estimation techniques that identify low–probability regions of phase space, and weakly–supervised classifiers that can distinguish data mixtures without event–by–event labels.

Generative Models

Perhaps the most significant recent development in ML for HEP has been the adoption of generative models, which learn to produce samples from complex probability distributions. This capability is particularly valuable in particle physics, where accurate simulation is crucial but computationally expensive.

Generative adversarial networks (GANs)**[cite –KD]**, variational autoencoders (VAEs)**[cite –KD]**, and normalizing flows**[cite –KD]** have all been successfully applied to particle physics problems. These models can generate synthetic collision events**[cite –KD]**, simulate detector responses**[–KD]**, and model complex differential distributions,**[–KD]** often accelerating these processes by orders of magnitude compared to traditional Monte Carlo techniques.

The development of these generative models opened new possibilities for addressing inverse problems in particle physics, including the unfolding problem at the center of this dissertation. By learning complex, high–dimensional probability distributions directly from data, these methods offer a natural framework for tackling unfolding without the limitations of binning.

Machine Learning Approaches to Unfolding

The application of machine learning to unfolding represents a particularly promising frontier. Traditional unfolding methods face significant challenges when dealing with high–dimensional spaces, correlated variables, and complex detector effects. Machine learning approaches offer several potential advantages compared to traditional statistical methods.

Neural networks excel at capturing patterns in high-dimensional spaces, helping to mitigate the curse of dimensionality that plagues binned methods. While traditional approaches become computationally prohibitive beyond a few dimensions, ML-based methods can effectively unfold many variables simultaneously.**[–KD]** This capability enables jointly unfolding multi–differential measurements that would be impractical with conventional techniques.

Many ML–based unfolding methods operate directly on continuous distributions, eliminating binning bias and the need to predetermine bin boundaries. This approach preserves fine–grained information that might otherwise be lost through discretization, and it enables the extraction of arbitrary derived observables from the unfolded distribution.**[–KD]** Furthermore, the architecture and training procedure of neural networks provide natural regularization without requiring explicit constraints. Rather than manually tuning regularization parameters as in traditional methods, ML approaches incorporate regularization through network depth, width, dropout rates, and early stopping.**[–KD]** This implicit regularization can adapt more naturally to the varying complexity across different regions of phase space.

ML methods can directly optimize for the quantities of interest, potentially reducing error propagation between steps. While traditional unfolding can involve separate stages for inversion and regularization, neural networks can learn the transformation from detector–level to particle–level in a single end–to–end process.[–KD] Once trained, many ML models also enable rapid inference on new data without retraining. This amortized approach is particularly valuable for unfolding tasks that need to be repeated with slight variations, such as systematic uncertainty studies or detector condition changes.[–KD]

Early Successes and Current Challenges

Machine learning approaches to unfolding, such as OmniFold[–KD], have demonstrated promising results on jet substructure measurements, showing improved performance in high-dimensional spaces compared to traditional methods. OmniFold has been successfully applied in experimental analyses at H1[–KD], ATLAS[–KD], CMS[–KD], and LHCb[–KD], validating its practical utility. However, these methods also face significant challenges. Unlike traditional techniques with decades of validation, ML–based unfolding is relatively new and requires careful validation to ensure physical consistency. Key concerns include proper uncertainty quantification, interpretability of the results, sensitivity to the training data, and potential biases introduced by the neural network architecture or training process.[–KD] Additionally, the field is still developing consensus on best practices for hyperparameter selection, architecture design, and evaluation metrics specific to unbinned unfolding. The balance between flexibility and stability remains an active area of research, with new approaches continuing to emerge.[–KD]

As machine learning continues to evolve, both within particle physics and in the broader scientific community, we can expect further innovations in unfolding techniques. Promising directions include physics–informed neural networks that incorporate known conservation laws or symmetries[–KD], uncertainty–aware models that provide more reliable error estimates[–KD], and techniques that combine the strengths of traditional and ML-based approaches[–KD]. The rapid progress in this field demonstrates the synergistic relationship between particle physics and machine learning. Particle physics provides challenging problems and unique datasets that drive methodological innovations; machine learning offers powerful tools that enable new measurements and insights. This dissertation builds upon this foundation, exploring novel machine learning approaches to improve the precision and scope of differential cross–section measurements through novel unfolding techniques.

3.2 Introduction to Neural Networks

Neural networks are the fundamental building blocks of modern machine learning applications. This section provides a rigorous mathematical framework for understanding neural networks, from their basic architecture to training methodologies.

Essential Concepts

Neural Network Fundamentals

At its core, a neural network is a parametric function approximator loosely inspired by biological neural systems. The fundamental unit, a neuron or node, computes a weighted sum of its inputs followed by a non-linear transformation:

$$y = \sigma(w^T x + b) \tag{3.1}$$

where $x \in \mathbb{R}^n$ is the input vector, $y \in \mathbb{R}^m$ is the output vector, $w \in \mathbb{R}^{n \times m}$ is a weight matrix, $b \in \mathbb{R}^m$ is a bias term, and $\sigma(\cdot)$ is a (typically non–linear) activation function.

The *depth* of a neuron is defined as the length of the longest path between the input and the neuron. The set of all neurons at depth l is called *layer* l of the network. This organization of neurons into layers, creates a hierarchical structure. The *depth*, $d = l_{\max}$, of a neural network is defined as the depth of its deepest layer.**[add figure to explain –KD]**

Any layer of a neural network except the input layer (often labeled layer 0) and the output layer ($l_{\max}$ is called a hidden layer. A *deep neural network* is a neural network with depth, $d > 2$. That is to say standard feedforward neural network consists of an input layer, *more than one* hidden layers, and an output layer.

For a network with d layers, at each layer l the neural network computes

$$y^{(l)} = \sigma^{(l)}(W^{(l)} y^{(l-1)} + b^{(l)}) \tag{3.2}$$

where $y^{(l)}$ represents the output of layer l, $W^{(l)}$ is the weight matrix, $b^{(l)}$ is the bias vector, and $\sigma^{(l)}$ is the activation function of later l. By convention, $y^{(0)} = x$ is the input, and $y^{(d)} = y$ is the output.

This architecture enables neural networks to approximate arbitrarily complex functions, as formalized in the Universal Approximation Theorem, which states that a deep neural network containing a finite number of neurons can approximate any continuous function on compact subsets of $\mathbb{R}^n$ given certain mild conditions on the activation function.

Forward Propagation

The process of computing the network's output given an input is called forward propagation. From its input $y^{(0)} = x$ the network sequentially computes the output of each

layer

$$z^{(1)} = W^{(1)} y^{(0)} + b^{(1)} \tag{3.3}$$
$$y^{(1)} = \sigma^{(1)}(z^{(1)}) \tag{3.4}$$
$$\vdots$$
$$z^{(d)} = W^{(d)} y^{(d-1)} + b^{(d)} \tag{3.5}$$
$$y^{(d)} = \sigma^{(d)}(z^{(d)}) \tag{3.6}$$

where $z^{(l)}$ represents the pre–activation values at layer l

Training objectives and loss functions

In order to quantify the accuracy with which a neural network f models a desired target function g for any particular input x, a suitable measure of the error between $f(x)$ and $g(x)$ is required. The *divergence* $\mathrm{Div}(x)$ is a continuous valued proxy for this error. The goal of training can then be defined to be to learn the values of W that minimize the expected divergence. (For notational convenience, from here on out, b will be collapsed into W as its 0^{th} column; the corresponding input, $x_0 = 1$.)

$$\widehat{W} = \arg\min_{W} \int_X \mathrm{Div}\Big(f(x;\, W), g(x)\Big)\, p_X(x)\, \mathrm{d}x \tag{3.7}$$

In practice however, g is often not known for all $x \in X$, it is only known at some subset $\{x_i\} \subseteq X$. The *Loss function* $\mathcal{L}$ is defined as the unbiased empirical average divergence between the neural network output and the target output over all training instances.

$$\mathcal{L}(W) = \frac{1}{N_X} \sum_{i=1}^{N_X} \mathrm{Div}\Big(f(x_i;\, W), g(x_i)\Big). \tag{3.8}$$

The expected value of the loss function can be proved to be the expected divergence, which is precisely what it means for the loss function to be an unbiased estimate of the expected divergence. However, this does not guarantee that minimizing the loss function will minimize the expected divergence.

Backpropagation and Parameter Learning

The training of neural networks revolves around finding the optimal values for weights and biases that minimize the loss function $\mathcal{L}$ Backpropagation is the algorithm used to efficiently compute gradients of the loss function with respect to each network parameter.

The core idea of backpropagation relies on the chain rule of calculus. For a loss function $\mathcal{L}$ and parameters $\{W^{(l)}\}_{l=1}^{d}$ the network computes

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \frac{\partial \mathcal{L}}{\partial z^{(l)}} \frac{\partial z^{(l)}}{\partial W^{(l)}} = \delta^{(l)} (y^{(l-1)})^T \tag{3.9}$$

where $\delta^{(l)} = \frac{\partial \mathcal{L}}{\partial z^{(l)}}$ is the error term for layer l. These error terms are computed recursively, starting from the output layer.

$$\delta^{(L)} = \frac{\partial \mathcal{L}}{\partial y^{(L)}} \odot \sigma'^{(L)}(z^{(L)}) \tag{3.10}$$

$$\delta^{(l)} = ((W^{(l+1)})^T \delta^{(l+1)}) \odot \sigma'^{(l)}(z^{(l)}) \tag{3.11}$$

where $\odot$ denotes the Hadamard (element-wise) product and σ' is the derivative of the activation function. These derivatives are thus propagated backwards through the network, beginning at the output layer, propagated to the input layer.

Batching and Training Dynamics

In practice, neural networks are trained using mini–batch gradient descent, where parameter updates are performed using gradients computed on subsets (batches) of the training data. For a batch of size B, the loss is

$$\mathcal{L}_{\text{batch}} = \frac{1}{B} \sum_{i=1}^{B} \mathcal{L}(x_i, y_i, W) \tag{3.12}$$

Training proceeds in *epochs*, where each epoch represents one complete pass through the training dataset. To prevent overfitting, a separate validation dataset is typically used to monitor performance, and techniques such as early stopping may be applied when validation performance plateaus or degrades.

Neural Networks as Universal Approximators

The power of neural networks in physics applications stems from their ability to approximate arbitrary functions. This property is formalized in the Universal Approximation Theorem, which has several variations but fundamentally states that feed-forward networks with a single hidden layer can approximate any continuous function on compact subsets of $\mathbb{R}^n$ to arbitrary precision, given sufficient depth and width.

Theorem 3.1 (Universal Approximation Theorem)**.**

$$\forall \sigma \in C^0(\mathbb{R})\ \forall g \in C^0([0,1]^n)\ \forall \epsilon > 0\ \exists M \in \mathbb{R} \tag{3.13}$$

$$\Bigg[\forall x \in \mathbb{R}\ |\sigma(x)| < M \implies \exists N \in \mathbb{N}\ \exists c, b \in \mathbb{R}^N\ \exists W \in \mathbb{R}^{N \times n} \tag{3.14}$$

$$\Big\{ \forall z \in [0,1]^n | c^T \sigma(Wz + b) - f(z)| < \epsilon \Big\} \vee \exists A \in \mathbb{R}\ \forall x \in \mathbb{R}\ \sigma(x) = A \Bigg]. \tag{3.15}$$

This theorem was first proved by Cybenko for sigmoid activation functions[–KD] and later extended by Hornik to include a broader class of activation functions.[–KD] Modern versions of the theorem have relaxed various assumptions and expanded the results to deeper networks.

Relevance to Physics Applications

The universal approximation property is particularly significant in physics. Many physical systems are governed by complex, non-linear differential equations that lack closed-form solutions. Neural networks can approximate these solutions without explicitly knowing the underlying equations, hence serving as excellent implicit simulators. Additionally, physics problems often involve high–dimensional spaces (e.g., the phase space for jet physics). Neural networks excel at learning in such spaces, where traditional numerical methods become computationally intractable. The universal approximation property makes neural networks well–suited for these tasks, where the mappings to be learned are highly complex.

When theoretical descriptions are incomplete, neural networks can discover patterns in data that suggest new physical models or refinements to existing ones. Neural networks as universal approximators are also helpful for simulation based inference. Modern physics often relies on computer simulations rather than closed–form likelihood functions. Neural networks can approximate these implicit models for fast posterior inference.

While the Universal Approximation Theorem guarantees the existence of a neural network capable of approximating any continuous function, it does not provide guidance on architecture design or training procedures. Nor does the theorem provide any guarantees about the rate of convergence of training procedures. In practice, deeper networks with multiple hidden layers have provide faster convergence, and greater parameter efficiency. However, the benefits of increasing depth are bounded, and decay exponentially, and hence the choice of architecture can involve carefully balancing the increased computational cost of a deeper network with the gains it provides. Extensions of the theorem show that depth can exponentially reduce the number of neurons required to approximate certain function classes, explaining the empirical success of deep learning in physics applications. These

theoretical insights have motivated the development of specialized architectures tailored to specific physics problems, as will be discussed in subsequent sections.

Activation Functions, Optimization, and Regularization

Activation Functions

Activation functions introduce non–linearity into neural networks, enabling them to learn complex patterns. Several activation functions are commonly used in high energy physics applications.

- Sigmoid:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.16}$$

  An activation function that outputs values between 0 and 1, it is most often used in problems involving binary classification. The sigmoid activation function has a rich and important history as one of the earliest activation functions to be used in machine learning applications. However, it suffers from vanishing gradient problems, where training instances far from its inflection point do not provide much information to the system.

- Hyperbolic Tangent (tanh):

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{3.17}$$

  The hyperbolic tangent function is very similar to sigmoid, except that it outputs values between -1 and 1. It is preferred in some applications because it is zero–centered, which can help in training dynamics, but it also suffers from the same vanishing gradient problems as the sigmoid.

- Rectified Linear Unit (ReLU):

$$\text{ReLU}(z) = \max(0, z) \tag{3.18}$$

  Simple, and incredibly computationally efficient, ReLU has become one of the most frequently used activation functions both within HEP and in the AI/ML space more broadly. It addresses the vanishing gradient problem for positive inputs. However, since it is simply zero for half its support, it can suffer from the "dying ReLU" problem where units can become permanently inactive.

- Leaky ReLU:
$$\text{LeakyReLU}(z) = \max(\alpha z, z) \tag{3.19}$$
α is a small positive constant. LeakyReLU is a modification of ReLU to allow small negative valued outputs, which helps mitigate the dying ReLU problem

- Softmax:
$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{3.20}$$
Softmax is most often used for multi–class classification output layers because it produces a probability distribution over classes

These a but a few examples of the most common activation functions. The choice of activation function significantly impacts network performance, convergence speed, and the occurrence of issues like vanishing or exploding gradients.

Optimization Techniques

As we discussed, neural network training is fundamentally an optimization problem. Various algorithms have been developed to find optimal parameters efficiently.

Gradient Descent is the oldest and simplest approach to optimization. It involves updating parameters in the direction of the negative gradient:

$$W_{t+1} = W_t - \eta \nabla_W \mathcal{L}(W_t) \tag{3.21}$$

η is the learning rate. While conceptually simple, gradient descent can be slow to converge and may get trapped in suboptimal solutions such as local minima.

Stochastic Gradient Descent (SGD) is an adaptation of gradient descent that updates parameters using gradients computed on mini–batches, introducing noise that can help escape local minima.

$$W_{t+1} = W_t - \eta \nabla_W \mathcal{L}_{\text{batch}}(W_t) \tag{3.22}$$

Momentum based methods accelerate convergence by accumulating a “velocity" vector in directions of persistent reduction in the loss function.

$$v_{t+1} = \gamma v_t + \eta \nabla_W \mathcal{L}(W_t) \tag{3.23}$$
$$W_{t+1} = W_t - v_{t+1} \tag{3.24}$$

where γ is the momentum coefficient. In HEP applications, a typical value of γ is 0.9.

Adaptive Methods are methods that vary the learning rate η over the course of the training rather than keeping it constant, and allow different learning rates for different dimensions of the input, rather than the traditional scalar learning rate that was constant across dimensions. One of the earliest adaptive methods proposed was AdaGrad, which adapts learning rates per-parameter based on historical gradients. AdaGrad was however limited by its aggressive learning rate reduction that slowed down training considerably. It was soon replaced in almost all applications by RMSProp, which addresses AdaGrad's aggressive learning rate reduction. Today, RMSProp is a widely used optimizer in ML applications, and is especially effective at stabilizing training when gradients are sparse and/or the loss function is dominated by saddle points.

Adam combines momentum based methods and RMSProp, and is currently the most widely used optimizer in ML applications. Adam's optimization procedure is can be described as

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_W \mathcal{L}(W_t) \tag{3.25}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla_W \mathcal{L}(W_t))^2 \tag{3.26}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{3.27}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{3.28}$$

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t \tag{3.29}$$

Typical values in HEP applications are $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

Learning Rate Scheduling is the process of dynamically adjusting the learning rate during training. Some of the more common variants are step decay (reducing learning rate by a fixed factor after a set number of epochs), exponential decay ($\eta_t = \eta_0 e^{-kt}$, and cosine annealing($\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(\frac{t\pi}{T}))$).

Regularization Techniques

To improve stabilize the training, prevent overfitting, and increase generalization, various regularization methods are employed.

L^p Regularization involves adding penalty terms to the loss function to penalize large weights.

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \lambda_p \|W\|^p. \tag{3.30}$$

The most common values of p are 1 and 2. For certain applications, a hybrid $L1 - L2$ regulariztion may also be appropriate.**[cite –KD]**

Dropout is a form of regularization that involves randomly setting a fraction of neuron outputs to zero during training:

$$y^{(l)}_{\text{dropout}} = m \odot y^{(l)} \tag{3.31}$$

where m is a binary mask with entries drawn from a Bernoulli distribution with parameter $p \in [0, 1)$

Batch Normalization is the process of normalizing the inputs of a layer to have zero mean and unit variance.

$$\hat{y}^{(l)} = \frac{y^{(l)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3.32}$$

$$z^{(l+1)}_{\text{BN}} = W^T \hat{y}^{(l)} + b \tag{3.33}$$

where μ_B and σ_B^2 are the batch mean and variance.

These regularization techniques, combined with appropriate optimization strategies and activation functions, form the foundation for effectively training neural networks for high energy physics applications. The principles outlined here will be applied in subsequent sections focusing on specific neural network architectures and their applications to unfolding problems.

3.3 Supervised Learning Approaches for Unfolding

Supervised learning provides a powerful framework for addressing the unfolding problem in particle physics. Unlike traditional matrix inversion methods, supervised learning approaches can leverage the flexibility and expressiveness of machine learning models to handle high-dimensional data and complex detector responses without requiring explicit binning schemes.

Mathematical Framework

Before exploring specific applications to unfolding, it is essential to establish what supervised learning is from a mathematical and statistical perspective. Supervised learning represents one of the primary paradigms in machine learning where an algorithm learns a

mapping from inputs to outputs using labeled training data. Supervised learning involves using a dataset, $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ consisting of n input–output pairs. Each input $x_i \in \mathcal{X}$ is a feature vector, and each output $y_i \in \mathcal{Y}$ is a label or target value. The objective is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ that approximates the true relationship between inputs and outputs.

Supervised learning searches over a space of functions $f_\theta$parameterized by θ that minimizes the expected risk,

$$\mathcal{R}(f_\theta) = \mathbb{E}_{(x,y)\sim P(X,Y)}[\mathrm{Div}(y, f_\theta(x))] \tag{3.34}$$

where Div is the divergence between the predicted output $f_\theta(x)$ and the true output y y, and $P(X, Y)$ is the joint probability distribution of inputs and outputs. Since the true distribution $P(X, Y)$ is unknown, we typically minimize the empirical risk

$$\hat{\mathcal{R}}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y_i, f_\theta(x_i)) \tag{3.35}$$

Where $\mathcal{L}$ is the loss function, the empirical expected divergence. This empirical risk minimization is accomplished through the optimization of the loss functions described in the previous section using techniques like stochastic gradient descent and backpropagation.

Statistical Interpretation

From a statistical standpoint, supervised learning can be viewed as conditional density estimation. For classification problems, we estimate $P(Y|X)$, the probability of a particular class given the input features. For regression problems, we estimate the conditional expectation $\mathbb{E}[Y|X]$ or the full conditional distribution $P(Y|X)$.

The choice of divergence function directly relates to the statistical assumptions being made. The mean squared error, $\mathrm{Div}(x) = (y - f_\theta(x))^2$, corresponds to maximum likelihood estimation under Gaussian noise assumptions. The cross–entropy, $-y \log f_\theta(x) - (1 - y) \log(1 - f_\theta(x))$ corresponds to maximum likelihood estimation for categorical distributions. Quantile divergences correspond to estimating specific quantiles of the conditional distribution

Types of Supervised Learning

Supervised learning problems broadly fall into two categories:

1. **Classification**: When the output space $\mathcal{Y}$ is discrete (categorical), the task is to predict the class label of a given input. Common loss functions include cross–entropy and hinge loss.

2. **Regression**: When the output space $\mathcal{Y}$ is continuous, the task is to predict a real–valued output. Common loss functions include mean squared error, mean absolute error, and Huber loss.

For unfolding problems, both paradigms can be relevant. Classification approaches often estimate density ratios between distributions, while regression approaches directly estimate mappings between detector–level and particle–level observables.

Learning and Generalization

A fundamental aspect of supervised learning is generalization i.e. the ability of the model to perform well on unseen data. This requires balancing two competing concerns, underfitting, when the model is too simple to capture the underlying structure of the data, and overfitting, when the model captures noise in the training data rather than the underlying structure. Regularization techniques help prevent overfitting by constraining the complexity of the model. For unfolding problems, regularization is particularly important due to the ill–posed nature of the inverse problem, where small variations in the data can lead to large variations in the prediction.

Evaluation

Supervised learning models are typically evaluated by measuring their performance on a separate test set not used during training. Common evaluation metrics include accuracy, precision, recall, F1-scores, and ROC curves for classification problems; mean squared error, mean absolute error, and R-squared for regression problems

In the context of unfolding, additional evaluation criteria can include physical consistency, preservation of known symmetries, and robustness to statistical fluctuations.

Unfolding as a Supervised Learning Problem

At its core, unfolding seeks to recover the mapping from detector–level distributions to particle–level truth distributions. This can be naturally framed as a supervised learning task where the model learns from pairs of detector–level and particle–level data generated through simulation. The supervised learning approach to unfolding typically uses paired data (z, x), where z represents particle–level quantities and x represents detector–level observations. The forward problem (detector simulation) maps $z \mapsto x$ through some response function, while unfolding attempts to estimate the inverse mapping.

Classification–Based Approaches

A prominent class of supervised learning methods for unfolding uses binary classification as its foundation. This approach leverages the ability of classifiers to estimate likelihood ratios between distributions. Of this class, OmniFold is perhaps the most widely adopted classification–based unfolding method and has been applied to several experimental measurements. It uses an iterative procedure inspired by Iterative Bayesian Unfolding (IBU), generalizing it to the unbinned case.

First, a classifier is trained to distinguish detector–level data from detector–level simulation. The classification outputs are used to reweight simulation events. Then another classifier is trained at particle level to transfer these weights back to the particle–level simulation. These steps are repeated for some fixed number of iterations.

This approach is particularly powerful for handling high–dimensional phase spaces where traditional binned methods become impractical. OmniFold effectively retains the full phase space information without requiring dimensionality reduction.

Regression–Based Approaches

While classification methods focus on reweighting simulation events, regression–based approaches aim to directly predict particle–level quantities from detector–level inputs. Neural networks can be trained to directly predict particle–level quantities from detector–level observations. This approach attempts to learn the function $f : X \to Z$ that maps detector–level features to particle–level truth values. However, this direct approach often struggles with the ill–posed nature of the unfolding problem.

More sophisticated regression approaches employ conditional density estimation to predict the full distribution of possible particle–level values given detector measurements. These methods recognize that the detector response introduces uncertainty, and hence the mapping from detector to particle level should be probabilistic rather than deterministic. Techniques such as Mixture Density Networks, **[cite –KD]** Normalizing Flows,**[–KD]** and Gaussian processes**[–KD]** have been explored to model these conditional distributions, providing both point estimates and uncertainty quantification.

Regularization Strategies

Supervised learning approaches to unfolding must address the inherently ill–posed nature of the inverse problem. This requires effective regularization strategies to prevent the amplification of statistical fluctuations.

Neural network architectures themselves provide implicit regularization. The choice of network depth, width, activation functions, and training protocols significantly impacts the

solution's smoothness properties. Convolutional layers, for instance, can encode physical priors such as translational invariance, constraining the space of possible solutions.

In iterative approaches like OmniFold, early stopping serves as a form of regularization. By halting the iteration process before full convergence, the method prevents overfitting to statistical fluctuations in the data.

Domain–specific physics knowledge can also be incorporated into the learning process through custom loss functions, model architectures, or constraints. For example, conservation laws, symmetries, or known theoretical behaviors can be encoded to guide the supervised learning process toward physically plausible solutions.

Advantages and Challenges

Supervised learning approaches to unfolding can operate directly on unbinned data, avoiding information loss and artifacts from binning. They scale better to high–dimensional observables and can effectively capture complex non–linear detector responses. Additionally, they provide flexible regularization through model architecture and training.

However, supervised approaches also face challenges. They require large amounts of simulated training data with accurate detector modeling. Unless designed carefully for interpretability, the black–box nature of neural networks can obscure the physical interpretation of the results. For unbinned unfolding methods, uncertainty quantification remains challenging, particularly in capturing the correlations between events when computing confidence intervals. Finally, validating the unfolding performance requires careful cross–checks and closure tests.

Despite these challenges, supervised learning approaches represent a significant advancement in unfolding methodology, enabling measurements that would be impractical with traditional binned techniques, in high–dimensional phase spaces relevant to modern particle physics analyses.

3.4 Deep Learning Architectures

High Energy Physics (HEP) presents unique data analysis challenges that have inspired the adoption and adaptation of specialized deep learning architectures. These architectures are designed to handle the distinctive properties of particle physics data, including high dimensionality, sparsity, permutation invariance, and complex correlations. This section explores the most prominent deep learning architectures employed in HEP applications, with particular emphasis on their relevance to unfolding tasks.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have found extensive application in HEP, particularly for analyzing data with inherent spatial or geometric structure. Originally developed for image recognition tasks, CNNs are well–suited for handling detector data that can be represented in image–like formats.

A prominent application of CNNs in HEP is in the analysis of "'jet images" — representations of energy deposits in calorimeters. Treating jet constituents as "pixels" in a coordinate system of pseudorapidity (η) and azimuthal angle (ϕ) creates image–like representations that CNNs can process effectively. For unfolding applications, CNNs can learn complex mappings between detector–level jet images and their corresponding particle–level representations. The translation invariance property of CNNs naturally encodes physical symmetries, providing implicit regularization that is beneficial for the ill–posed unfolding problem.**[–KD]**

CNNs have also been successfully applied to model energy depositions in calorimeters.**[–KD]** These architectures can capture the spatial correlations in shower development patterns, enabling more accurate unfolding of particle energies and types from detector responses. Three–dimensional CNNs have been employed to handle the full volumetric nature of calorimeter data, treating detector cells as voxels in a 3D space.**[–KD]** These approaches have shown promising results in reconstructing particle properties from complex shower patterns.**[–KD]**

Recurrent Neural Networks

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), have been applied to sequential aspects of HEP data analysis. In many HEP applications, particles can be naturally ordered by properties such as transverse momentum or angular distance from a reference axis. RNNs can process these ordered sequences while maintaining information about dependencies between particles.

For unfolding tasks involving sequential data, RNNs can model the mapping between detector–level sequences and their particle–level counterparts, capturing complex ordered dependencies that might be lost in other architectures. RNNs are also applicable to time–dependent detector responses or beam conditions that evolve over time. By incorporating time information, these models could help unfold distributions that are affected by time–varying detector effects.

Graph Neural Networks

Graph Neural Networks (GNNs) have emerged as one of the most promising architectural paradigms for HEP applications. These networks explicitly model relationships between particles or detector elements as graphs, with nodes representing individual entities and edges representing their interactions or relationships.

Particle interactions naturally form graph–like structures, where each particle can be considered a node with properties (features) such as momentum, energy, and charge. GNNs can process these "particle clouds" directly, without requiring conversion to image or sequence formats. The key advantage of GNNs for unfolding is their ability to preserve the permutation invariance of particle collections; the physical properties of a jet should not depend on the arbitrary order in which particles are processed. By operating directly on graphs, GNNs respect this physical constraint.

GNNs have also been applied to model the complex geometry of particle detectors, where detector elements can be represented as nodes and their physical or electronic connections as edges.**[–KD]** This approach enables more accurate modeling of detector responses, which is crucial for reliable unfolding. A specialized class of GNNs called Message Passing Neural Networks (MPNNs) has shown particular promise in physics applications.**[–KD]** These networks iteratively update node representations by passing information along edges, mimicking the way physical interactions propagate through a system.

Transformer-Based Architectures

Transformer architectures, which have revolutionized natural language processing, are increasingly being applied to HEP problems due to their ability to model complex dependencies through self-attention mechanisms.**[–KD]** Transformers can naturally process sets of particles by using self–attention to capture relationships between all pairs of particles. This makes them particularly well–suited for unfolding tasks involving collections of particles with complex inter–relationships.

The Particle Transformer (ParT) architecture,**[–KD]** specifically designed for HEP applications, incorporates physics–motivated constraints and has demonstrated strong performance on jet classification tasks.**[–KD]** Similar principles can be applied to develop transformer–based unfolding methods.

The attention mechanisms in transformers provide an additional benefit. They can highlight which detector–level features are most informative for reconstructing particle–level properties. This interpretability is valuable for understanding the unfolding process and identifying potential biases or limitations.

Physics-Informed Neural Networks

A growing trend in HEP applications is the development of physics–informed neural networks that explicitly incorporate domain knowledge about physical laws, conservation principles, and symmetries. Networks that preserve known physical symmetries, such as Lorentz invariance or gauge symmetry, can provide more physically plausible unfolding results. Architectures such as Lorentz Group Equivariant Networks**[–KD]** ensure that the neural network respects these fundamental symmetries by design. For unfolding problems where energy conservation is essential, specialized architectures can enforce this constraint by design. These networks ensure that the total energy is preserved between detector–level and particle–level representations, reducing the space of possible solutions and improving physical consistency. Beyond specialized architectures, physical constraints can be incorporated into the loss function or network design. For example, constraints on momentum conservation, charge conservation, or known physical boundaries of observables can guide the network toward physically plausible solutions.

Energy Flow Networks and Particle Flow Networks

Energy Flow Networks (EFNs) and Particle Flow Networks (PFNs)**[–KD]** represent specialized architectures developed specifically for HEP applications, grounded in the theoretical framework of Energy Flow Polynomials and the infrared and collinear safety properties essential for jet physics. EFNs and PFNs are based on the principle that jets can be represented as collections of particles, each characterized by its energy (or transverse momentum) and angular coordinates. These architectures directly incorporate the Energy Flow basis, a complete, linear basis for infrared and collinear safe observables, into their design. The key insight of these networks is the decomposition of jet observables into products of per–particle functions (capturing individual particle features) and pairwise or multi–particle correlators (capturing relationships between particles). This decomposition aligns with how QCD radiation patterns physically manifest in jet structure.

Both EFNs and PFNs operate on sets of particles with the following structure:

1. **Per–particle feature extraction:** Each particle is processed independently through a shared neural network Φ (called the “particle embedding network"), mapping individual particle features to a latent representation.

2. **Permutation–invariant aggregation:** The individual particle embeddings are combined through a permutation–invariant operation, typically summation, weighted by particle energies in the case of EFNs.

3. **Global processing:** The aggregated representation is processed by another neural network F (the “latent space network") to produce the final output.

EFNs process only the geometric information of particles (η, ϕ coordinates), weighted by their energies or transverse momenta during aggregation. This architecture is specifically designed for infrared and collinear safe observables. PFNs can incorporate additional per–particle features beyond just angular coordinates (e.g., particle type, charge, or identification probabilities), making them more flexible but potentially less theoretically constrained.

For unfolding applications, by respecting infrared and collinear safety in their design, EFNs ensure that unfolded distributions preserve important theoretical properties of QCD. Jets naturally contain varying numbers of particles. EFNs and PFNs handle this variable–length input natively without padding or truncation. The architectural constraints of EFNs/PFNs serve as implicit regularization aligned with physical principles, helping to constrain the ill–posed nature of unfolding.

In unfolding contexts, EFNs and PFNs can be employed either as components within classification-based approaches (like OmniFold) or as direct mappings between detector-level and particle-level representations. EFNs and PFNs can be understood as specialized implementations of the Deep Sets paradigm, which provides a theoretical foundation for processing sets of varying sizes. This connection to a broader machine learning framework can facilitate theoretical analysis of these architectures and inspire further developments.

The deep learning architectures discussed in this section provide a rich toolkit for addressing the challenges of unfolding in HEP. By selecting and adapting architectures based on the specific properties of the data and the physics requirements of the analysis, researchers can develop more accurate and physically consistent unfolding methods. The next section will explore advanced machine learning frameworks that build upon these architectures to provide even more powerful approaches to unfolding.

3.5 Modern Machine Learning Frameworks

This section explores sophisticated machine learning frameworks that have demonstrated significant potential for addressing the unfolding problem in high energy physics. We examine three principal categories, generative models, discriminative models, and adversarial frameworks, each offering distinct approaches to learning complex distributions and relationships in particle physics data.

Generative Models

Generative models represent a powerful class of machine learning techniques that aim to learn and characterize the underlying probability distribution of observed data. Unlike discriminative models that focus on decision boundaries between classes, generative models capture the full data–generation process, enabling them to produce new samples that

resemble the training distribution **[cite –KD]**. The defining characteristic of these models is their ability to generate new, synthetic data points that follow the same statistical patterns as the training data. This capability makes them particularly valuable for applications in particle physics where simulation of complex physical processes is essential **[cite –KD]**. In the context of unfolding, generative models offer a natural framework for modeling the mapping between particle-level and detector-level distributions.

Generative models typically approach the learning task either by modeling the probability density function of the data explicitly, or by learning to generate samples from the target distribution without explicitly estimating the density. Training the model involves maximizing the likelihood of the data.

$$\hat{\theta} = \arg\max_{\theta} \prod_{i=1}^{n} p(x_i|\theta) \tag{3.36}$$

where $p(x|\theta)$ is the conditional probability density function at x given θ and $\{x_i\}_{i=1}^{n}$ are the training data.

Variational Autoencoders and Normalizing Flows are two commonly used generative models that have demonstrated significant potential for unfolding applications in particle physics.

Variational Autoencoders (VAEs)

Variational Autoencoders represent a class of deep generative models that combine the principles of variational inference with neural network-based autoencoders **[cite –KD]**. First introduced by Kingma and Welling in 2013 **[cite –KD]**, VAEs provide a principled probabilistic framework for learning complex data distributions while enabling both generation of new samples and inference of latent representations. Fundamentally, VAEs extend traditional autoencoders by imposing a probabilistic structure on the latent space. While standard autoencoders learn deterministic encodings, VAEs encode inputs as probability distributions in the latent space. This probabilistic approach enables principled sampling and uncertainty quantification, which are critical requirements for unfolding applications.

The VAE architecture consists of two primary components, an encoder network $q_\phi(z|x)$ which maps inputs x to a latent space distribution, typically parameterized as a multivariate Gaussian:

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \text{diag}(\sigma^2_\phi(x))) \tag{3.37}$$

where $\mu_\phi(x)$ and $\sigma^2_\phi(x)$ are parameters learned by the network, and a decoder network $p_\theta(x|z)$, which reconstructs inputs from latent samples, often parameterized as a Gaussian for continuous data,

$$p_\theta(x|z) = \mathcal{N}(x|\mu_\theta(z), \text{diag}(\sigma^2_\theta(z))), \tag{3.38}$$

or a multivariate Bernoulli distribution**[cite –KD]** for categorical data.

VAEs are trained by maximizing the Evidence Lower Bound (ELBO), which serves as a tractable lower bound on the log-likelihood of the data:

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)|p(z)) \tag{3.39}$$

This objective balances two competing terms, the *reconstruction term* $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ encourages accurate reconstruction, and the KL divergence term $D_{KL}(q_\phi(z|x)\|p(z))$ that acts as a regularizer, encouraging the learned latent distribution to match a prior distribution (typically a standard normal) The ELBO has deep connections to statistical mechanics and information theory. The KL divergence term bears striking resemblance to the free energy minimization principle, where $p(z)$ serves as an analog to the “ground state" or “vacuum state" distribution **[cite –KD]**.

Once trained, VAEs enable principled sampling by first sample from the prior distribution: $z \sim p(z) = \mathcal{N}(0, 1)$, and then generating a new observation by passing z through the decoder: $x \sim p_\theta(x|z)$ This generative capability is particularly valuable for modeling the detector response in particle physics, where the mapping from particle–level to detector–level observables involves complex transformations and uncertainties. Therefore in high energy physics, VAEs have found numerous applications including

- Fast detector simulation, where the VAE learns to map particle–level quantities to detector–level observables, **[cite –KD]**
- Anomaly detection for beyond Standard Model physics searches,**[cite –KD]**
- Dimensionality reduction of for processing collider data, **[cite –KD]**
- Unfolding detector effects.**[cite –KD]**

For unfolding applications specifically, several features of VAEs make them naturally suited for the task. VAEs model the uncertainty in the inverse mapping from detector-level to particle-level distributions providing a direct probabilistic framework for downstream analysis. One can regularize the problem by constraining the latent space to accord with the physics of the problem, which helps stabilize the training and reduce the variance. Once trained, VAEs provide fast amortized inference, enabling efficient processing of large collision datasets Recent work has demonstrated VAEs’ effectiveness for unfolding tasks, particularly when combined with adversarial training components to enhance the physical consistency of the unfolded distributions **[cite –KD]**.

Certain challenges posed by VAEs must be considered when selecting the appropriate model for a problem. VAEs with Gaussian latent spaces can have limited expressivity because the latent space may be too restrictive for complex physical distributions. Conversely,

selecting a feature dense latent space (a) can deregularize the problem and (b) cause a loss of interpretability. VAEs also tend to produce smoothed–out samples, potentially losing fine details in particle distributions. Finally, balancing the reconstruction and KL terms requires careful tuning, and small changes in the hyperparameters can lead to significant changes in the learned function.

Normalizing Flows

Normalizing flows represent a powerful class of generative models that learn the exact likelihood computation through a series of invertible, differentiable transformations **[cite –KD]**. Unlike VAEs, which approximate the likelihood, normalizing flows directly learn a bijective mapping between the data distribution and a simple base distribution, typically a multivariate Gaussian. The core principle behind normalizing flows is the change of variables formula from probability theory. Given a random variable z with density $p_Z(z)$ and an invertible, differentiable transformation $f : \mathbb{R}^d \to \mathbb{R}^d$ the density of the transformed variable $x = f(z)$ is

$$p_X(x) = p_Z(f^{-1}(x)) \cdot \left| \det \left(\frac{\partial f^{-1}(x)}{\partial x} \right) \right| \tag{3.40}$$

where $\left| \det \left(\frac{\partial f^{-1}(x)}{\partial x} \right) \right|$ is the absolute value of the determinant of the Jacobian of f^{-1}, which accounts for the change in volume elements due to the transformation.

Normalizing flows chain together multiple such transformations to create highly expressive mappings while maintaining invertibility:

$$f = f_K \circ f_{K-1} \circ \cdots \circ f_1 \tag{3.41}$$

The resulting density is

$$p_X(x) = p_Z(z) \cdot \prod_{k=1}^{K} \left| \det \left(\frac{\partial f_k^{-1}}{\partial f_{k-1}} \right) \right| \tag{3.42}$$

where $z = f^{-1}(x) = f_1^{-1} \circ \cdots \circ f_K^{-1}(x)$.

Since they were first proposed, several architectural innovations have expanded the expressivity and computational efficiency of normalizing flows. Coupling Layers (e.g. NICE **[cite –KD]**, RealNVP **[cite –KD]**) partition the input dimensions and apply transformations to one part conditioned on the other, ensuring tractable Jacobian determinants

$$y_1 = x_1 \tag{3.43}$$

$$y_{d+1} = x_{d+1} \odot \exp(s(x_1)) + t(x_1) \tag{3.44}$$

where s and t are scale and translation networks. Autoregressive Flows (e.g. IAF **[cite –KD]**, MAF **[cite –KD]**) model dependencies between dimensions through an autoregressive structure

$$y_i = x_i \cdot \exp(s_i(x_1)) + t_i(x_1). \tag{3.45}$$

Continuous Normalizing Flows **[cite –KD]** formulate the transformation as an ordinary differential equation (ODE), offering increased flexibility

Each version of the model makes different trade–offs between expressivity, computational efficiency, and ease of training. In general, normalizing flows are trained by directly maximizing the log-likelihood of the data:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log p_X(x_i; \theta) \tag{3.46}$$

where θ parameterizes the flow transformations. This direct likelihood maximization contrasts with the variational approach of VAEs and the adversarial approach of GANs, offering a more stable optimization objective in many cases.

Normalizing flows have gained significant traction in particle physics applications, including

- Simulation-based inference: Using flows to perform likelihood–free inference for physics parameters, **[cite –KD]**
- Fast detector simulation: Modeling the detector response through invertible mappings, **[cite –KD]**
- Phase space integration: Computing complex multi–dimensional integrals in perturbative calculations, **[cite –KD]**
- Unfolding detector effects.**[cite –KD]**

For unfolding specifically, normalizing flows provide several compelling advantages. The exact likelihood computation provided by the model enables principled uncertainty quantification in the unfolded distributions. The chain of bijective mappings also naturally aligns with the physical process of detector effects and their inversion. Capable of capturing complex, multi–modal distributions common in particle physics, flow–based models have been shown to be effective vehicles for unfolding, showing improved performance particularly for high–dimensional and complex distributions **[cite –KD]**.

A particularly promising approach for unfolding involves conditional normalizing flows, where the transformation depends on additional conditioning variables:

$$p_X(x|c) = p_Z(f^{-1}(x; c)) \cdot \left| \det \left(\frac{\partial f^{-1}(x; c)}{\partial x} \right) \right| \tag{3.47}$$

In the unfolding context, the detector–level observables can serve as the conditioning variables, with the flow mapping from a base distribution to the particle-level distribution conditioned on detector measurements **[cite –KD]**. This approach enables explicit modeling of the posterior distribution $p(\text{particle}|\text{detector})$ that unfolding aims to estimate.

Despite their theoretical elegance, normalizing flows face several practical challenges in HEP applications. The invertibility requirement limits the types of transformations that can be used. The computational cost of evaluating Jacobian determinants for high–dimensional data can impose practical limits on the dimensionality of the unfolding problem flow based methods can be applied to. Ongoing research continues to address these challenges through improved architectures and training procedures.

Discriminative Models

Discriminative models directly learn the mapping from input features to output labels, modeling the conditional distribution $p(y|x)$ rather than the joint distribution. In the context of particle physics, these models excel at classification, regression, and decision tasks that underpin many key analyses, from particle identification to unfolding. Discriminative models parameterize the conditional distribution, $p(y|x)$ using a function $f_\theta(x)$, where θ represents the model parameters.

$$p(y|x) = p(y|f_\theta(x)) \tag{3.48}$$

For classification tasks with K classes, this typically takes the form of a categorical distribution,

$$p(y = k|x) = \frac{\exp\left(f_\theta^k(x)\right)}{\sum_{j=1}^{K} \exp\left(f_\theta^j(x)\right)}. \tag{3.49}$$

The optimization objective is to maximize the conditional likelihood

$$\mathcal{L}(\theta) = \prod_{i=1}^{N} p(y_i|x_i;\theta) \tag{3.50}$$

or equivalently, minimizing the negative log–likelihood

$$-\log \mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(y_i|x_i;\theta). \tag{3.51}$$

This formulation provides a principled approach for both probabilistic classification and regression tasks that dominate HEP analyses.

Modern discriminative models in particle physics typically employ neural networks with various architectures tailored to specific data structures. The most common of these are fully connected networks, traditional deep neural networks with dense connections that transform inputs through a series of nonlinear functions,

$$y_l = \sigma(W_l y_{l-1} + b_l) \tag{3.52}$$

Convolutional Neural Networks (CNNs) are useful for exploiting translational invariance in detector data through localized feature extraction.

$$y^k_{l,i,j} = \sigma\left(\sum_m \sum_{p,q} W^{k,m}_{p,q} y^m_{l-1,i+p,j+q} + b^k_l\right) \tag{3.53}$$

where $y^k_{l,i,j}$ represents the output of the $k-$th feature map at position (i, j) in layer l. Recurrent Neural Networks (RNNs) are most often used to capture sequential information in particle trajectories.

$$y_t = \sigma(W_x x_t + W_y y_{t-1} + b) \tag{3.54}$$

where y_t is the hidden state at step t.

Training Dynamics and Optimization

Training discriminative models involves gradient–based optimization with regularization techniques to prevent overfitting. The cross–entropy loss described in Sec. **??**, derived from maximum likelihood is the most commonly used loss function to train discriminative models. In some cases, specialized losses are like focal loss are used address class imbalances in rare physics processes.

$$L_{focal} = -\sum_{i=1}^{N} (1 - p_i)^\gamma \log(p_i) \tag{3.55}$$

where p_i is the predicted probability of the correct class and γ is a focusing parameter.

Applications in HEP and Unfolding

Discriminative models serve multiple critical functions in particle physics analyses.

- Event Classification: Separating signal from background events, with area under the ROC curve (AUC) values exceeding 0.99 in many recent LHC analyses,
- Particle Identification: Distinguishing particle types based on detector signatures with significantly improved efficiency compared to cut–based methods,

- Jet Tagging: Identifying jets originating from specific particles with significant performance gains over traditional approaches,
- Unfolding: Estimating reweighting functions between detector-level and particle-level distributions, as implemented in OmniFold.

Despite their successes, the use of discriminative models can pose a series of challenges. Neural network outputs often require calibration to provide accurate probability estimates for downstream statistical analysis. The performance of discriminative models critically depends on simulation quality, with domain shifts between simulation and data requiring specialized approaches. These models are typically considerably more complex than their generative counterparts, and may identify features that lack clear physical interpretation. Quantifying model uncertainties remains challenging, particularly for ensemble approaches, when bootstrapping is computationally unfeasible.

Support Vector Machines

SVMs find the maximum–margin hyperplane separating classes by solving the primary optimization problem

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to } y_i(w^T x_i + b) \geq 1 \tag{3.56}$$

The dual formulation of this problem leads to the kernel trick, enabling nonlinear decision boundaries:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad \text{subject to } \alpha_i \geq 0 \text{and } \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.57}$$

where$K(x_i, x_j)$ is a kernel function.

The choice of kernel dramatically affects SVM performance. The most common choices in particle physics problems are the

- Linear Kernel, $K(x_i, x_j) = x_i^T x_j$, which is efficient for high–dimensional but linearly separable data,
- Radial Basis Function (RBF), $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$, which is capable of capturing nonlinear detector responses, and
- Polynomial Kernel $K(x_i, x_j) = (x_i^T x_j + c)^d$, which models polynomial relationships in feature space.

SVMs have been applied across various particle physics processes. Early applications in the Higgs discovery process demonstrated competitive performance and propelled these models forward in the discourse.**[cite –KD]** Around the same time, in analyses involving quark–gluon discrimination, SVMs with RBF kernels achieved discrimination power comparable to dedicated physics–inspired variables.**[cite –KD]** Attempts to construct SVMs for unfolding focus on their potential to estimate the density ratio between detector–level and particle–level distributions

A persistent roadblock to their more widespread adoption has been the fact that traditional SVM implementations scale poorly with dataset size, $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$. Unlike deep networks, SVMs also require careful feature selection, and standard SVMs don't provide natural probability estimates, limiting their usefulness.

Boosted Decision Trees

Historically one of the most popular models employed in jet physics analyses, Boosted Decision Trees (BDTs) combine weak learners (shallow decision trees) into a powerful ensemble.

$$F(x) = \sum_{m=1}^{M} \beta_m h_m(x) \tag{3.58}$$

where h_m is the $m-$th tree and β_m is its corresponding weight. Each tree recursively partitions the feature space according to information gain criteria,

$$\text{Gain} = \text{Impurity(parent)} - \sum_{j \in \{\text{children}\}} \frac{N_j}{N} \text{Impurity}(j) \tag{3.59}$$

Several boosting algorithms have either been developed specifically for HEP or been employed in HEP analyses.

- **AdaBoost** iteratively reweights misclassified samples.

$$w_i^{(t+1)} = w_i^{(t)} \cdot e^{\alpha_t \cdot \mathbf{1}(y_i \neq h_t(x_i))} \tag{3.60}$$

  where $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ and ϵ_t is the weighted error rate.

- **Gradient Boosting** fits each tree to the negative gradient of the loss function with respect to the current prediction.

$$h_m = \arg\min_h \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + h(x_i)) \tag{3.61}$$

- **XGBoost** incorporates second–order derivatives and regularization into the above schema.

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{3.62}$$

  where g_i and h_i are the first and second derivatives of the loss.

BDTs have been extensively deployed in particle physics analyses. The Toolkit for Multivariate Analysis (TVMA) integrated BDTs into the ROOT framework, cementing their place as a staple in HEP analyses. BDTs were instrumental in the Higgs boson discovery by improving signal–to–background discrimination. They can also be used in unfolding tasks, since they can model complex response matrices and estimate reweighting functions

Despite their success, BDTs present several challenges, causing them to be eclipsed more recently by deep learning methods. They struggle with highly correlated features common in detector data, and performance typically degrades in very high-dimensional spaces. Tree–based models also introduce discontinuities in the mapping function that can complicate uncertainty propagation.

Graph Neural Networks

GNNs operate on graph–structured data, represented as$G = (V, E)$ with nodes $v_i \in V$ and edges $e_{ij} \in E$. The message–passing framework updates node representations through the rule

$$y_i^{(l+1)} = \phi \left(y_i^{(l)}, \bigoplus_{j \in \mathcal{N}(i)} \psi(y_i^{(l)}, y_j^{(l)}, e_{ij}) \right) \tag{3.63}$$

where ϕ, ψ are learnable functions, $\mathcal{N}(i)$ denotes neighbors of node$i$, and $\bigoplus$ is the permutation–invariant aggregation operator.

A few different specialized GNN variants have been developed with an eye towards specific physics applications. For example, Interaction Networks model physical interactions between particles with explicit edge functions. Dynamic Graph CNNs construct graphs dynamically based on spatial proximity in detector space, and Particle Flow Networks incorporate physics-informed constraints into the message-passing mechanism. Similarly, HEP specific objectives functions have also been developed to train GNNs for HEP applications. These include node–level classification for identifying individual particles, graph–level classification for categorizing entire events, edge prediction for reconstructing particle interaction vertices, and energy regression for estimating particle energies and momenta.

Through these specialized architectures and objective functions, GNNs have shown remarkable performance across a range physics use cases.

- Jet Tagging: Representing jets as graphs of constituents for improved identification performance
- Event Reconstruction: Modeling entire collision events as interaction graphs
- Tracking: Reconstructing particle trajectories from detector hits
- Unfolding: Learning mappings between detector–level and particle–level observables while preserving physical symmetries

The most significant constraint on the use of GNNs in HEP is their computational complexity. Message–passing operations scale with graph size and connectivity. While their ability to naturally handle variable–sized inputs can be incredibly useful, implementing them requires careful batching strategies to maintain the differentiable structure of the network that backpropagation relies on.

Transformer Models

Transformers process sequences through self–attention mechanisms,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.64}$$

where Q, K, V are query, key, and value matrices derived from input embeddings. In HEP applications, self–attention enables modeling various relationships between particles, such as permutation invariance for the natural handling of varying particle multiplicities, long–range dependencies, for capturing correlations between distant detector regions, and interpretable attention maps for visualizing learned physical interactions.

Transformer models, which revolutionized the field of natural language processing, are increasingly being adopted for complex HEP tasks. They have been used for jet classification, because they naturally allow the processing of jets as sequences of constituents. Entire collision events have been modeled with particle transformers, aiding event–level analysis.

However, despite their potential, some of the characteristics of transformer models have hindered their widespread adoption in HEP. Self-attention operations scale quadratically with sequence length, constraining the size of attention modules. Transformer models typically require large training datasets, as a consequence of which, training runs can rarely exceed one epoch. Hence, while transformers can learn surface level relationships extremely effectively, they are limited in their ability to learn deeper structure that would require multiple passes over each data point to discover.

Adversarial Models

Adversarial models leverage the competitive dynamics between two neural networks to achieve powerful generative and discriminative capabilities. These models have revolutionized high energy physics applications by enabling novel approaches to simulation, unfolding, and domain adaptation. Fundamentally different from traditional architectures, adversarial models employ a game–theoretic framework that can capture complex, high–dimensional distributions and transformations relevant to particle physics.

Adversarial models are typically constructed using two competing neural networks, a generator g and a discriminator d. The generator attempts to produce outputs that the discriminator cannot distinguish from real data, while the discriminator attempts to correctly classify inputs as either real or generated. This minimax game is formalized as

$$\max \min L(d, g) = -\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log d(x)] - \mathbb{E}_{z\sim p_z(z)}[\log(1 - d(d(z)))] \tag{3.65}$$

where $p_{\text{data}}(x)$ is the true data distribution, $p_Z(z)$ is a latent space distribution (typically Gaussian noise), $g(z)$ maps the latent space to the data space, and $d(x)$ outputs the probability that x came from the real data rather than the generator.

In the context of particle physics, this framework can be extended to incorporate domain–specific constraints and physical laws, leading to specialized variants such as

$$\max_g \min_d L(d, g) = -\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log d(x)] - \mathbb{E}_{z\sim p_z(z)}[\log(1 - d(g(z)))] + \lambda\mathcal{R}(g) \tag{3.66}$$

where $\mathcal{R}(g)$ represents physics–informed regularization terms that enforce conservation laws, symmetries, or other physical constraints, weighted by hyperparameter λ.

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) represent a fundamentally different approach to generative modeling, employing an adversarial training framework rather than explicit density estimation **[cite –KD]**. First introduced by Goodfellow et al. in 2014 **[cite –KD]**, GANs have revolutionized generative modeling through their ability to produce remarkably realistic samples, particularly for high–dimensional data like images. By training the two networks simulteneously in the aforementioned minimax game, the generator learns to produce increasingly realistic samples that can fool the discriminator, while the discriminator improves its ability to distinguish real from generated samples.

From a theoretical perspective, the GAN objective can be interpreted as minimizing the Jensen–Shannon divergence between the data distribution and the generator distribution **[cite –KD]**. Training proceeds by alternating between optimizing the two networks using opposing gradient information, *viz.* d is updated using gradient descent,

$$\theta_d \leftarrow \theta_d - \eta\nabla_{\theta_d} L(d, g), \tag{3.67}$$

and g is updated using gradient ascent,

$$\theta_g \leftarrow \theta_g + \eta \nabla_{\theta_g} L(d, g). \tag{3.68}$$

In equilibrium, the generator produces samples indistinguishable from the real data distribution, and the discriminator outputs a probability of 0.5 for all inputs. Formally the optimal discriminator computes

$$d(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \quad \text{for all } x \tag{3.69}$$

and the optimal generator perfectly captures the data distribution

$$p_g(x) = p_{\text{data}}(x) \tag{3.70}$$

In practice, reaching this optimum can be challenging due to the min–max nature of the objective, leading to training instabilities like mode collapse and oscillations. For this reason numerous GAN variants have been proposed to address stability issues and enhance performance. The Wasserstein GAN (W-GAN) **[cite –KD]** replaces JS divergence with Wasserstein distance, providing more stable gradients.

$$\max_g \min_{d \in D} \mathbb{E}_{z \sim p_Z(z)}[d(g(z))] - \mathbb{E}_{x \sim p_{\text{data}}(x)}[d(x)], \tag{3.71}$$

where D is the set of 1-Lipschitz functions. Conditional GANs **[cite –KD]** enable conditioning the generation process on auxiliary information, and Progressive GANs **[cite –KD]** gradually increases model complexity during training, improving stability and quality.

Methods such a *feature matching,* which encourages the generator to match statistics of intermediate discriminator activations,

$$\mathcal{L}_{\text{feature}} = \|\mathbb{E}_{x \sim p_{\text{data}}}[f(x)] - \mathbb{E}_{z \sim p_z}[f(G(z))]\|_2^2, \tag{3.72}$$

where $f(x)$ represents activations from an intermediate layer of the discriminator, and *spectral normalization,* which controls the Lipschitz constant of the discriminator by normalizing its weights,

$$W_{\text{SN}} = \frac{W}{\sigma(W)}, \tag{3.73}$$

where $\sigma(W)$ is the spectral norm of weight matrix W, provide additional tools to avoid training instabilities. These advances have made GANs more practical for scientific applications like those in particle physics.

In high-energy physics, GANs have found diverse applications.

- **Fast detector simulation:** GANs can approximate the mapping from particle-level to detector-level observables, significantly accelerating the simulation process **[cite –KD]**
- **Event generation:** GANs can generate collision events, potentially replacing or augmenting traditional Monte Carlo approaches **[cite –KD]**
- **Anomaly detection:** Identifying rare or anomalous collision events that deviate from Standard Model predictions **[cite –KD]**

For unfolding specifically, GANs offer a unique perspective through their ability to learn complex mappings between distributions.

GANs are however infamous for their training instability. The adversarial objective can lead to convergence issues, particularly for the sparse, high–dimensional distributions common in particle physics. Even when the two networks are well balanced, the minmax optimization can lead to oscillatory behavior rather than convergence. Another particularly well studied failure mechanism is mode collapse, in which GANs may fail to capture the full diversity of the target distribution, focusing instead on a limited subset of modes. This occurs when the generator network is too powerful, and discriminator is unable to provide receive gradient information from it. Conversely, when the discriminator is too powerful and only provides minimal gradient information to the generation, the training stalls due to the vanishing gradients.

These challenges have motivated hybrid approaches that combine the strengths of GANs with other novel architectural design choices to use them more effectively for unfolding**[cite –KD]**.

Conditional Adversarial Networks

Conditional GANs (cGANs) extend the standard GAN by incorporating conditioning information y into both generator and discriminator,

$$\max_g \min_d L(d,g) = -\mathbb{E}_{x,y\sim p_{\text{data}}(x,y)}[\log d(x|y)] - \mathbb{E}_{z\sim p_Z(z),y\sim p_{\text{data}}(y)}[\log(1-d(g(z|y)|y))] \tag{3.74}$$

This formulation enables the generation of samples conditioned on specific physical parameters, detector configurations, or particle properties.

Conditioning information can be in GANs in a few different ways. Most directly, conditioning information can be concatenated as additional input features

$$g(z,y) = g_\theta([z,y]) \quad d(x,y) = d_\phi([x,y]) \tag{3.75}$$

Conditional normalization is an approach that involves modulating normalization parameters based on conditioning

$$\gamma(y) = \text{NN}_\gamma(y), \quad \beta(y) = \text{NN}_\beta(y), \tag{3.76}$$

$$\text{CN}(h, y) = \gamma(y) \cdot \frac{h - \mu(h)}{\sigma(h)} + \beta(y). \tag{3.77}$$

A more recent approach is to incorporate conditioning information instead through Feature-wise Linear Modulation (FiLM) layers, that scale and shift feature maps.

$$\text{FiLM}(h_i, y) = \gamma_i(y) \cdot h_i + \beta_i(y) \tag{3.78}$$

Conditional GANs provide powerful tools for physics–constrained generation. In HEP studies, they have been used for

- **Parameterized Simulation:** Generating detector responses across different operating conditions,
- **Conditional Unfolding:** Correcting detector effects with explicit energy dependence,
- **Systematic Variation Generation:** Creating samples with varied systematic uncertainties,
- **Model Parameterization:** Generating samples across theoretical model parameter spaces.

Adversarial Autoencoders

Adversarial Autoencoders (AAEs) combine autoencoder reconstruction with adversarial training.

$$\mathcal{L}_{\text{AAE}} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{adv}} \tag{3.79}$$

where $\mathcal{L}_{\text{recon}} = \|x - \text{Dec}(\text{Enc}(x))\|^2$ measures reconstruction quality, and $\mathcal{L}_{\text{adv}}$ is the adversarial loss that forces the encoder's latent space distribution to match the sample distribution,

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{z \sim p(z)}[\log d(z)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - d(\text{Enc}(x)))]. \tag{3.80}$$

AAEs are trained in two phases per iteration.

1. **Reconstruction Phase:** Update encoder and decoder to minimize reconstruction error,
2. **Regularization Phase:**

- Update discriminator to distinguish between samples from the prior and encoded data,
- Update encoder to fool the discriminator.

AAEs have been used especially in applications involving dimensionality deduction, to learn physics–preserving low–dimensional representations; anomaly detection, to identify unusual events through reconstruction errors or latent space deviations; and simulation enhancement, to improving the fidelity of simplified simulations.

Cycle–Consistent Adversarial Networks

Cycle–GANs enable unpaired domain translation through cycle consistency.

$$\mathcal{L}_{\text{CycleGAN}} = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F) \tag{3.81}$$

where $G : X \to Y, F : Y \to X$ are mapping functions between domains, and the cycle consistency loss is

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1] \tag{3.82}$$

Cycle–GANs employ two generators and two discriminators. Generator G: Maps domain X to domain Y, generator F: maps domain Y back to domain X, discriminator D_X distinguishes real X from generated $F(Y)$, and discriminator D_Y distinguishes real Y from generated $G(X)$.

In HEP, Cycle-GANs enable several important applications, such as

- **Sim2Real Translation:** Mapping between simulated and real detector data without paired examples,
- **Cross-Experiment Translation:** Translating measurements between different experimental setups,
- **Systematic Variation Generation:** Creating physically plausible variations for systematics studies.

Research addressing the challenges adversarial models pose continues to expand the applicability of adversarial models in particle physics, with recent advances incorporating invariant representations, equivariant architectures, and physics-informed losses to enhance both performance and reliability.

3.6 ML–based Unfolding: Advantages and Challenges

Machine learning approaches to unfolding offer several compelling advantages over traditional methods while introducing new challenges that must be carefully addressed. This section examines both perspectives to provide a comprehensive view of ML–based unfolding techniques.

Advantages

ML-based approaches excel at capturing patterns in high–dimensional spaces, thereby addressing a fundamental limitation of traditional binned methods. As the dimensionality of the measurement increases, traditional binning methods suffer from the “curse of dimensionality"—the number of bins grows exponentially with the number of dimensions, leading to sparsely populated bins and unstable unfolding results. Neural networks can effectively extract patterns from high–dimensional data with limited sampling, enabling unfolding in previously intractable phase spaces. By operating directly on event–level data without the need for binning, ML methods also eliminate binning artifacts such as bias and resolution loss. This is particularly advantageous for observables with complex structures such as resonance peaks or threshold effects, where binning might obscure important features. Unbinned approaches preserve the full information content of the data and enable more flexible downstream analyses that are not tied to specific binning schemes.

Neural networks provide inherent regularization through their architecture, training procedure, and hyperparameters. Unlike traditional methods that require explicit regularization terms (such as Tikhonov regularization) or stopping criteria (as in IBU), the complexity of neural networks can be controlled through architectural choices such as layer width, depth, and activation functions. This implicit regularization can be more adaptive to the data than explicitly imposed constraints.

Traditional unfolding pipelines often involve multiple discrete steps that propagate errors and may introduce biases. ML approaches allow for end–to–end optimization, where the entire unfolding process can be optimized jointly. This integrated approach can reduce error propagation and potentially improve overall performance by directly optimizing for the quantities of interest rather than intermediate objectives.

Once trained, many ML models enable rapid inference on new data without retraining. This “amortized" inference is particularly valuable in experimental settings where multiple unfolding procedures may need to be performed with different systematic variations. Methods like Neural Posterior Unfolding provide a function approximation of the likelihood that can be efficiently evaluated for different data configurations, reducing the computational burden compared to traditional methods.

Challenges

ML models, particularly deep neural networks, often function as “black boxes," making it difficult to interpret their internal workings. This lack of transparency can complicate validation and uncertainty quantification, which are crucial for scientific measurements. Establishing robust validation procedures and developing more interpretable ML architectures remains an active area of research.

While ML models provide implicit regularization, selecting appropriate hyperparameters and architectural elements introduces its own form of regularization tuning. Unlike traditional methods where regularization parameters have clear physical interpretations, the relationship between neural network hyperparameters and the resulting regularization strength is often less direct. This can make it challenging to select optimal configurations and compare results across different studies.

ML–based unfolding methods, like their traditional counterparts, are sensitive to the distribution modeled by the simulation. Mismodeling in the simulation can lead to biases in the unfolded results. Some methods, such as OmniFold, employ iterative procedures to mitigate prior dependence, on the individual simulated distributions, but the dependence on the simulated response kernel is unavoidable.

Training complex neural networks can require substantial computational resources, especially for high–dimensional problems and large datasets. The need for multiple training runs to evaluate uncertainties, such as in ensemble methods, further increases the computational burden. Methods like bootstrapping or Monte Carlo dropout can help estimate uncertainties, but they are still be computationally intensive for at–scale applications. Propagating systematic uncertainties themselves through ML-based unfolding pipelines while maintaining interpretability presents unique challenges. Traditional approaches often involve repeating the unfolding procedure with varied inputs to assess systematic effects. For ML methods, this can mean retraining models multiple times, which besides being computationally expensive, rarely provides any physical insight into the nature of the errors. Developing efficient methods for systematic uncertainty estimation in ML–based unfolding remains an important research direction.

A significant challenge in ML–based unfolding is the proper treatment of correlations in the unfolded data. Traditional covariance matrices for binned methods have clear statistical interpretations, but quantifying correlations in unbinned or high-dimensional unfolded distributions is more complex. There is no universally accepted unbinned analogue of the correlation matrix. Nevertheless, these correlations can significantly impact downstream analyses and uncertainty estimation, and therefore cannot be ignored. Chapter **??** will show that event correlations in unfolded data may lead to misestimation of uncertainties when these correlations are ignored, emphasizing the need for careful statistical treatment of unfolded results.

The field of ML-based unfolding is rapidly evolving, with new methods continuously being developed. While these innovations offer exciting possibilities for more precise and comprehensive measurements, they must be balanced with the rigorous validation and uncertainty quantification required for scientific results. As these methods mature, establishing best practices for validation, uncertainty estimation, and systematic assessment will be crucial for their wider adoption in experimental analyses. The combination of traditional insights with modern ML capabilities presents a promising path forward for addressing the unfolding problem in increasingly complex measurement contexts.

3.7 Case Studies: ML Based Unfolding in HEP Analyses

This section examines how machine learning–based unfolding methods have been successfully applied in recent high energy physics analyses, demonstrating their practical utility and impact on physics measurements.

OmniFold

The OmniFold algorithm has become one of the most widely applied ML–based unfolding methods in experimental particle physics. The H1 Collaboration at HERA pioneered its experimental use, applying OmniFold to measure multivariate jet substructure observables in deep inelastic electron-proton scattering **[cite –KD]**. This landmark analysis demonstrated for the first time that unbinned, ML–based unfolding could be successfully implemented in a high profile experimental measurement The H1 results showed that OmniFold could handle correlations between multiple observables while maintaining precision comparable to traditional methods but with reduced model dependence.

Building on this success, the LHCb experiment applied OmniFold to unfold charged particle multiplicity distributions in proton-proton collisions **[cite –KD]**. This analysis showcased the method's ability to handle measurements with significant statistical and systematic uncertainties while providing model–independent results across a wide kinematic range. More recently, ATLAS has employed OmniFold to perform measurements of jet substructure in large–radius jets **[cite –KD]**. These measurements benefit particularly from the unbinned nature of OmniFold, as they involve complex multi–dimensional phase spaces where traditional binned methods would either lose information or become computationally intractable. Notably, this analysis demonstrated OmniFold's ability to unfold multiple observables simultaneously, preserving correlations that would be lost in separate one-dimensional unfoldings.

Neural Posterior Unfolding in Practice

Neural Posterior Unfolding (NPU) represents an emerging direction in ML-based unfolding, combining normalizing flows with Bayesian inference to provide principled uncertainty quantification. While newer than OmniFold, NPU has shown promise in preliminary studies of jet mass distributions on simulated LHC data**[cite –KD]**. These applications highlight NPU's ability to capture complex posterior distributions and provide natural regularization through its neural network architecture. A key advantage demonstrated in these studies is NPU's amortized inference capability, which significantly reduces computational costs when performing repeated unfolding procedures for systematic uncertainty evaluation, a crucial consideration for practical experimental analyses. NPU is discussed in greater detail in Chapter **??**.

Invertible Neural Networks for Unfolding

Invertible Neural Networks (INNs) and their conditional variants (cINNs) represent another significant direction in ML–based unfolding. Unlike discriminative approaches like OmniFold, these models learn bijective mappings between data spaces, offering natural probabilistic interpretations. The ATLAS collaboration has explored cINNs for unfolding jet measurements, demonstrating their ability to model complex, non-linear detector response functions while maintaining computational efficiency **[cite –KD]**.

Bellagente et al. pioneered the application of cINNs to high energy physics unfolding, showing that these models could effectively handle the inverse problem of reconstructing particle–level distributions from detector measurements **[cite –KD]**. Their approach leveraged normalizing flows to model the conditional probability of particle–level quantities given detector measurements, providing both point estimates and uncertainty quantification through the learned probability distribution.

Recent extensions have incorporated physical constraints directly into INN architectures, ensuring that conservation laws and symmetries are preserved in the unfolded results **[cite –KD]**. This physics–informed approach has shown particular promise in jet physics applications where momentum conservation and Lorentz invariance provide strong constraints on the unfolded distributions.

Moment Unfolding

Moment Unfolding is a recently proposed method, focusing specifically on unfolding statistical moments of distributions rather than entire spectra **[cite –KD]**. This method is particularly valuable when theoretical predictions are at the level of moments rather than differential distributions, as is often the case in QCD calculations.**[cite –KD]**

Moment Unfolding employs a GAN–inspired structure with a generator that implements Boltzmann weighting factors to constrain the moments of the unfolded distribution. By focusing only on a small number of moments, this method provides natural regularization and has demonstrated excellent precision in jet substructure applications where scaling behaviors and moments are of primary interest **[cite –KD]**.

Moment Unfolding unfolds statistical moments as a function of another observable (e.g., jet transverse momentum), enabling detailed studies of energy–scale dependence without requiring full spectral unfolding **[cite –KD]**. Feature specific methods like Moment Unfolding have the potential to offer precision and computation cost improvements over generalized unfolding methods for comparisons focused on individual features rather than the entire density. Moment Unfolding is discussed in further detail in **??**.

Reweighting Adversarial Networks (RAN)

Building on the conceptual framework introduced in Moment Unfolding, Reweighting Adversarial Networks (RAN) extend the methodology to unfold full spectra through an adversarial learning framework **[cite –KD]**. RAN implements particle–level reweighting functions guided by detector–level classifiers, enabling non–iterative full spectral unfolding.

Unlike OmniFold, which requires multiple iterations and separate classifiers at each step, RAN employs a single adversarial training process to determine optimal weights. This approach offers computational advantages while maintaining performance comparable to iterative methods **[cite –KD]**.

RAN has been used to successfully unfolding jet substructure observables in multi-dimensional spaces, on simulated data where its non-iterative nature provides significant computational advantages **[cite –KD]**. Studies on real collider examples are left to future work. Implemented specifically as a Wasserstein GAN, RANs are effective in scenarios with limited detector–level overlap between simulation and data, a challenging regime for other reweighting-based methods **[cite –KD]**. Chapter **??** delves into more detail about RANs.

Schrödinger Bridge Unfolding

The application of Schrödinger Bridge processes to unfolding represents a theoretically elegant approach based on optimal transport theory. Schrödinger Bridge Unfolding (SBU) **[cite –KD]** frames the unfolding problem as finding the most likely path between probability distributions under entropy constraints.

This approach has shown particular promise in handling multi–modal distributions and cases with significant detector non-linearity **[cite –KD]**. SBU provides theoretical guarantees on the transport map between detector and particle spaces, offering a principled approach to regularization through its entropic formulation.

Recent applications have demonstrated SBU's effectiveness in unfolding complex jet structures and rare decay topologies **[cite –KD]**, with particular strength in preserving multi–modal features that might be smoothed over by other regularization approaches.

3.8 ML Upstream and Downstream of Unfolding

Uncertainty Quantification in ML Unfolding

Robust uncertainty quantification remains a central challenge in ML–based unfolding. Several recent developments in the field have addressed this challenge specifically. Bayesian Neural Networks (BNNs) have been applied to unfolding to provide natural uncertainty estimates through posterior sampling **[cite –KD]**. These methods capture both statistical and model uncertainties in a unified framework, though at increased computational cost.

Ensembling techniques have proven effective in practice, with the CMS collaboration implementing ML unfolding with multiple network initializations to estimate modeling uncertainties **[cite –KD]**. This approach balances computational feasibility with comprehensive uncertainty estimation. For unbinned unfolding results, it is important to highlight the importance of accounting for event correlations in downstream analyses **[cite –KD]**. Standard statistical procedures that assume independence can significantly misestimate uncertainties when applied to correlated unfolded events, necessitating specialized approaches for uncertainty propagation. Results demonstrating this, alongside a discussion on uncertainty quantification methods in general are provided in Chapter **??**.

Hybrid Approaches and Integration with Simulation

Hybrid approaches that combine traditional and ML–based unfolding have emerged as popular, pragmatic solutions in experimental settings. The ATLAS collaboration has explored methods that use ML techniques for response modeling while employing traditional matrix inversion for the actual unfolding step **[cite –KD]**, leveraging the strengths of both approaches. Integration of differentiable simulation into the unfolding pipeline represents another promising direction. By making detector simulations differentiable, these approaches enable end–to–end optimization of the entire measurement chain **[cite –KD]**. Early results have demonstrated improved precision and reduced systematic uncertainties, particularly for complex final states.

Recent work has also explored direct integration of theory predictions into ML-based unfolding **[cite –KD]**, enabling simultaneous unfolding and parameter fitting. This unified approach can provide more direct constraints on physical parameters while properly accounting for all experimental effects.

Domain Adaptation Techniques for Unfolding

Domain adaptation methods have been applied to address cases where simulation and data distributions differ significantly. These techniques, adapted from computer vision research, aim to learn domain–invariant features that facilitate unfolding despite simulation–data discrepancies.

The ALICE collaboration has employed domain adversarial neural networks to perform unfolding in heavy–ion collisions where detector effects are particularly complex and challenging to simulate accurately **[cite –KD]**. These methods have shown robustness to modeling uncertainties that would significantly impact traditional approaches. Similarly, CMS has explored gradient reversal techniques to mitigate simulation biases in unfolding jet measurements **[cite –KD]**, demonstrating improved robustness to mismodeling effects while maintaining good statistical precision.

Equivariant Networks for Physics-Informed Unfolding

Equivariant neural networks, which preserve symmetry transformations by design, have recently been applied to unfolding problems in particle physics. These architectures incorporate physical symmetries as inductive biases, naturally handling rotational, translational, or permutation symmetries common in physics applications.

Particle Flow Networks with equivariance properties have shown promise for unfolding jet constituent distributions **[cite –KD]**, preserving Lorentz symmetry while capturing complex detector effects. These approaches leverage the rich structure of jets while respecting fundamental physical constraints. The LHCb experiment has explored equivariant graph neural networks for unfolding decay topologies **[cite –KD]**, where the underlying physical process exhibits various symmetries that can be exploited to improve unfolding precision and physical consistency.

These physics–informed approaches highlight the broader trend toward integrating explicit physical knowledge into ML–based unfolding methods, combining the flexibility of neural networks with the strong constraints provided by fundamental physical principles. Chapter **??** presents SymmetryGAN, a method that has been used to discover the symmetries which has subsequently been used to inform physics inspired networks.**[cite –KD][cite –KD][cite –KD]**

Machine Learning for Detector Response Modeling

Several recent analyses have employed ML techniques not just for the unfolding procedure itself, but also for modeling the detector response. The CMS collaboration has explored the use of Generative Adversarial Networks (GANs) to model detector effects in jet mea-

surements **[cite –KD]**, providing a more flexible and potentially more accurate alternative to traditional Monte Carlo-based detector simulations. These approaches integrate naturally with ML–based unfolding methods, forming end–to–end pipelines that can be jointly optimized. Studies have shown that such integrated approaches can reduce systematic uncertainties related to detector modeling, particularly in complex final states with multiple jets or in regimes where traditional simulations are less reliable **[cite –KD]**.

Comparison with Traditional Methods

An important aspect of ML for HEP research is the direct comparison of proposed ML–based methods with traditional unfolding techniques in controlled experimental settings. The ALICE collaboration performed a systematic comparison of ML–based and traditional unfolding methods in charged particle multiplicity measurements **[cite –KD]**, finding that ML methods achieved comparable or better precision while handling higher–dimensional observables that would be challenging for traditional approaches. Similarly, the STAR collaboration applied both OmniFold and traditional Singular Value Decomposition (SVD) methods to unfold jet measurements in heavy-ion collisions **[cite –KD]**. Their results demonstrated that ML methods could achieve similar central values but with improved handling of boundary effects and systematic uncertainties.

ML Unfolding for Beyond Standard Model Searches

Beyond measuring Standard Model processes, ML–based unfolding has found applications in searches for new physics. Several analyses have employed these techniques to provide model–independent measurements that can subsequently be interpreted in various theoretical frameworks. For instance, unbinned unfolding has been applied to searches for resonances in dijet mass distributions **[cite –KD]**, where traditional binned methods might obscure narrow features.

The flexibility of ML approaches is particularly valuable in these contexts, as they can accommodate a wide range of systematic variations without requiring extensive recalculation This adaptability has made ML–based unfolding increasingly attractive for analyses where model independence and systematic robustness are paramount.

Implementation Considerations from Real Experiments

These case studies have revealed important practical considerations for implementing ML–based unfolding in experimental analyses. One key lesson has been the importance of careful validation procedures. Experiments typically employ closure tests, linearity tests,

and stress tests with modified simulation samples to ensure the robustness of ML unfolding methods **[cite –KD]**.

Another practical consideration is computational efficiency. While ML methods typically require significant resources for training, the optimized implementations that leverage high–performance computing resources make these methods practical for large–scale analyses **[cite –KD]**.

Systematic uncertainty evaluation remains a central challenge. Recent analyses have developed methods for efficiently propagating systematic uncertainties through ML unfolding pipelines, often employing ensemble techniques or incorporating uncertainty sources directly into the training procedure **[cite –KD]**.

Impact on Physics Results

The application of ML–based unfolding has had measurable impacts on physics results across multiple experiments. In jet substructure measurements, these methods have enabled more precise constraints on Monte Carlo tuning parameters **[cite –KD]**, improving our understanding of fundamental QCD processes. In heavy–ion physics, they have contributed to more detailed characterizations of the quark–gluon plasma through multi–dimensional particle correlation measurements **[cite –KD]**.

Perhaps most significantly, ML–based unfolding has expanded the dimensionality and scope of experimental measurements, enabling analyses that would be impractical with traditional methods. This has opened new avenues for physics exploration, particularly in areas where correlations between multiple observables carry important physical information making it essential to unfold their joint distributions. As these methods continue to mature and gain wider acceptance within the experimental community, their impact on the precision and scope of physics measurements at current and future colliders is likely to grow substantially, potentially enabling new insights into fundamental physics processes.

Chapter 4

Neural Posterior Unfolding

- Physics motivation for improved binned unfolding - degeneracy, null spaces
- Normalizing flows and applications to inverse problems
- Neural posterior estimation and degeneracy + Implicit regularization
- Substructure example
- Inference advantages and computational efficiency: compare FBU
- Set the stage for unbinned methods

4.1 Motivation for Improved Binned Unfolding

Differential cross section measurements represent the fundamental currency of scientific exchange in particle and nuclear physics. They quantify the probability density of specific particle interactions as a function of kinematic variables, providing the essential link between theoretical predictions and experimental observations. However, a critical challenge in these measurements arises from detector effects that distort the underlying physics distributions. Unfolding, or deconvolution, is the process of statistically removing these distortions to recover the true particle–level distributions from the measured detector–level data.

When measuring a physical observable, the detector response can be mathematically expressed as the forward mapping,

$$p_{\text{detector}}(x) = \int R(x|z) \cdot p_{\text{particle}}(z)\,\mathrm{d}z \tag{4.1}$$

Here, $p_{\text{detector}}(x)$ represents the detector–level distribution, $p_{\text{particle}}(z)$ is the true particle-level distribution, and $R(x|z)$ is the response kernel that encodes the detector effects **[cite**

–KD]. Unfolding aims to invert this relationship to reconstruct $p_{\text{particle}}(z)$ from the observed $p_{\text{detector}}(y)$. A binned method is one in which both the particle–level and detector–level distributions are discretized into histograms, transforming the integral equation into a matrix equation:

$$\boldsymbol{\mu} = \mathbf{R}\boldsymbol{\nu} + \boldsymbol{\epsilon} \tag{4.2}$$

Where $\boldsymbol{\mu}$ represents the detector–level bin counts, $\boldsymbol{\nu}$ represents the particle–level bin counts to be estimated, $\mathbf{R}$ is the discretized response matrix, and $\boldsymbol{\epsilon}$ accounts for measurement uncertainties **[cite –KD]**.

Degeneracy and Null Spaces in Unfolding

One of the fundamental challenges in unfolding collider data comes from degeneracies in the response matrix. Consider a case where two particle–level bins α and β are indistinguishable at the detector level. Formally, there exists a detector-level bin κ such that for all detector-level bins ι:

$$R_{\iota\alpha} = R_{\iota\beta} = \delta_{\iota\kappa} \tag{4.3}$$

where δ is the Kronecker delta **[cite –KD]**. In such scenarios, traditional methods like Iterative Bayesian Unfolding (IBU) face a critical limitation. The unfolded result becomes entirely dependent on the simulation, and learns no information from the data. The update rule for IBU takes the form

$$t_\alpha^{(n)} = \left(\frac{t_\alpha^{(0)}}{t_\alpha^{(0)} + t_\beta^{(0)}}\right) \cdot m_\kappa \qquad t_\beta^{(n)} = \left(\frac{t_\beta^{(0)}}{t_\alpha^{(0)} + t_\beta^{(0)}}\right) \cdot m_\kappa \tag{4.4}$$

This means the relative contributions of bins α and β to the unfolded result are entirely determined by the MC, regardless of the observed data **[cite –KD]**. Even worse, traditional methods will report zero uncertainty on the ratio $\frac{t_\alpha}{t_\alpha + t_\beta}$ when in reality this ratio should have maximal uncertainty since the detector cannot distinguish between these contributions.

Regularization Limitations

To address the ill-posedness detailed in Sec. **??**, traditional methods employ various forms of regularization, examples of which are provided in Sec. **??**. However, all these approaches require a careful balance between fidelity to the data and conformity to the regularization constraints. Furthermore, these methods typically do not provide a natural framework for propagating uncertainties and can significantly underestimate the true uncertainty in degenerate regions. These limitations motivate the development of new unfolding methods that can

1. Naturally handle degeneracies and null spaces in the response matrix
2. Provide proper uncertainty quantification, especially in poorly constrained regions
3. Reduce dependence on arbitrary regularization parameters
4. Offer computational efficiency for large-scale problems

Neural Posterior Unfolding addresses these challenges head on, by leveraging normalizing flows for neural posterior estimation. Unlike traditional methods that provide point estimates, NPU yields a full posterior distribution over the unfolded parameters, naturally capturing uncertainties in poorly constrained regions **[cite –KD]**. Furthermore, the regularization in NPU emerges implicitly from the neural network architecture and training protocol, reducing the need for manual tuning of regularization parameters.

In the following sections, we introduce normalizing flows and neural posterior estimation as the foundation for the NPU method, demonstrating how these modern machine learning techniques address the fundamental challenges of binned unfolding. You're right - since we've already covered normalizing flows in detail in Chapter 3, we should focus this section specifically on how they apply to inverse problems like unfolding rather than repeating information. Here's what I suggest we cover in this section:

4.2 Normalizing Flows for Inverse Problems

A Bayesian Perspective on Inverse Problems

The unfolding problem can be naturally framed in Bayesian terms. Given detector–level measurements x, we aim to reconstruct the posterior distribution over particle–level quantities z,

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{4.5}$$

Where $p(x|z)$ is the detector response, $r(x|z)$ (likelihood), $p(z)$ is the prior for particle–level quantities, and $p(x)$ is the evidence. Traditional methods typically focus on finding a point estimate that maximizes this posterior, but a full Bayesian treatment would characterize the entire posterior distribution to properly account for uncertainties and degeneracies.

Normalizing flows offer several key advantages for modeling the posterior distribution in inverse problems.

- **Complex Distribution Modeling** Unlike standard parametric approaches, normalizing flows can represent multimodal, asymmetric posteriors that often arise in inverse

problems with degeneracies where multiple particle–level configurations can lead to similar detector-level observations.

- **Amortized Inference**: Once trained, normalizing flows enable rapid posterior sampling without requiring new optimization runs for each new observation. This is particularly valuable when processing large datasets or when rerunning statistical analyses with different systematic variations.
- **End-to-End Differentiability**: The entire posterior estimation process can be optimized end–to–end, enabling gradient–based learning approaches that are more efficient than traditional MCMC methods for high–dimensional unfolding.
- **Implicit Regularization**: Neural network architectures naturally introduce an inductive bias that serves as implicit regularization, potentially reducing overfitting to statistical fluctuations compared to direct matrix inversion approaches without requiring explicit regularization parameters.

Conditional Normalizing Flows for Inverse Problems

In the context of unfolding, we're particularly interested in modeling the conditional distribution $p(z|x)$, the particle–level distribution given detector–level observations. Conditional normalizing flows are specifically designed for this task. A flow f_θ with learnable parameters θ can be trained to model the particle–level distribution z,

$$z = f_\theta(u|y), \tag{4.6}$$

where u is drawn from a simple base distribution (typically a standard normal), and x is the conditioning variable (detector–level data). This conditional structure allows the flow to learn the posterior distribution specifically tailored to each detector–level observation, capturing the uncertainty and potential multimodality of the solution even in regions where the detector response is non–injective.

While Markov Chain Monte Carlo (MCMC) methods have traditionally been used for Bayesian inference in inverse problems (as in Fully Bayesian Unfolding)**[cite –KD]**, normalizing flows have some unique properties that make them especially well suited to this task. No burn–in period is required once the flow is trained, and samples are generated independently rather than in a correlated chain. The computational cost of generating samples does not increase with the dimensionality of the parameter space, and perhaps most importantly, due to the fully differentiable structure of the network, training can leverage GPU acceleration and parallelization. These advantages make normalizing flows particularly attractive for inverse problems where MCMC methods may face mixing and convergence ch allenges or become computationally prohibitive.

4.3 The Neural Posterior Unfolding Algorithm

Neural Posterior Unfolding (NPU) is a novel approach to unfolding that combines the rigor of Bayesian statistics with the flexibility and computational efficiency of modern machine learning. Rather than focusing solely on point estimates, NPU aims to characterize the full posterior distribution over particle–level observables, providing comprehensive uncertainty quantification while addressing the key limitations of traditional unfolding methods.

Statistical foundation

At its core, NPU leverages normalizing flows for neural posterior estimation in the context of binned unfolding. The fundamental insight is to directly learn the conditional posterior distribution $p(t|m)$, the probability density of the true particle–level histogram counts, t given the measured detector–level histogram counts, m.

The relevant statistical quantities are

- A detector–level histogram $m = \{m_j\}_{j=1}^{N_D}$, where m_j is the number of events in detector-level bin j
- A particle–level histogram $t = \{t_i\}_{i=1}^{N_T}$, where t_i is the number of events in particle-level bin i
- A response matrix $\mathbf{R}$ with elements $R_{ji} = P(m_j|t_i)$, the probability of an event in particle–level bin i being measured in detector–level bin j

The forward model relating these quantities is

$$m_j = \sum_{i=1}^{N_T} R_{ji} t_i + \epsilon_j, \tag{4.7}$$

where ϵ_j represents statistical fluctuations, typically modeled as Poisson noise. In a Bayesian framework, the posterior distribution is given by

$$p(t|m) = \frac{p(m|t)p(t)}{p(m)} \tag{4.8}$$

where $p(m|t)$ is the likelihood, $p(t)$ is the prior, and $p(m)$ is the evidence. For Poisson–distributed measurements, the likelihood takes the form:

$$p(m|t) = \prod_{j=1}^{N_D} \frac{(\sum_i R_{ji} t_i)^{m_j} e^{-\sum_i R_{ji} t_i}}{m_j!} \tag{4.9}$$

NPU is conceptually similar to Fully Bayesian Unfolding (FBU), but replaces the traditional Markov Chain Monte Carlo (MCMC) sampling with normalizing flows for neural posterior estimation, providing computational advantages and increased flexibility while maintaining the same full Bayesian treatment of the problem. **[cite –KD]**.

Machine Learning Architecture

NPU builds upon the neural posterior estimation framework, which uses conditional normalizing flows to directly model the posterior distribution. The NPE framework comprises a base distribution, a simple distribution (typically a multivariate standard normal) $p_u(u)$ that is easy to sample from, an invertible transformation, a learnable, invertible function $f_\theta(u|m)$ that transforms samples from the base distribution into samples from the approximate posterior, conditioned on the measured data m, and a density transformation, encoding the the change of variables formula,

$$p_\theta(t|m) = p_u(f_\theta^{-1}(t|m)) \left| \det \left(\frac{\partial f_\theta^{-1}(t|m)}{\partial t} \right) \right| \tag{4.10}$$

The neural network parameters θ are optimized to maximize the likelihood of the true posterior, using pairs of simulated (t, m) examples. The training objective is to minimize the loss function

$$\mathcal{L}(\theta) = \mathbb{E}_{(t,m)\sim p_{\text{sim}}(t,m)} \left[\log p_\theta(t|m) \right] \tag{4.11}$$

This approach allows the model to learn complex posterior distributions, including multimodal and highly correlated structures that often arise in unfolding problems **[cite –KD]**.

In practical implementations, uses a normalizing flow based on the MADE (Masked Autoencoder for Distribution Estimation) architecture. The network is made up of an invertible transformation network with three fully connected layers, each containing $50 - 100$ nodes with Swish activation functions. Conditional inputs incorporated through an auxiliary fully connected layer. The network is then trained for $1000 - 1500$ epochs using the Adam optimizer with a learning rate of 10^{-4} and a batch size of 10^4. The loss function is the negative log likelihood described above.

This architecture provides sufficient flexibility to model complex posterior distributions while remaining computationally tractable **[cite –KD]**. After training the normalizing flow, the unfolded response is obtained through a Maximum Likelihood Estimation (MLE) step, where the negative log-likelihood of the observed data conditioned on the learned model is minimized. This optimization process uses the same Adam optimizer.

Addressing Degeneracy with NPU

A key advantage of NPU over traditional unfolding methods is its ability to naturally handle degeneracies in the response matrix. Consider the two-bin degenerate example presented earlier, where two particle-level bins α and β are indistinguishable at the detector level. In traditional methods like IBU, the unfolded result becomes entirely dependent on the prior and incorrectly report zero uncertainty on the ratio $t_\alpha/(t_\alpha + t_\beta)$.

In contrast, NPU naturally returns a broad posterior distribution that properly reflects the uncertainty. This is demonstrated in **[NPU –KD]** with a two–bin example where the response becomes increasingly degenerate (with correlation coefficient $\rho \to 0$). As the detector loses its ability to differentiate between truth bins, NPU produces credible intervals that encompass all values consistent with the total counts, appropriately reflecting the inherent uncertainty. Simultaenously, while allowing for uncertainty in degenerate directions, NPU maintains constraints in well-determined directions, such as the total sum $t_\alpha + t_\beta$. This behavior emerges naturally from the Bayesian formulation, without requiring explicit identification of the degenerate subspaces. By modeling the full posterior, NPU provides a principled way to represent uncertainty in precisely those directions where the data provides little or no constraint **[cite –KD]**.

Implicit Regularization in NPU

A significant advantage of NPU is its implicit regularization, which arises naturally from the neural network architecture and training procedure rather than requiring explicit regularization terms or early stopping rules. Several mechanisms contribute to this implicit regularization.

- **Network Capacity**: The finite capacity of the neural network limits the complexity of the posterior approximation, preventing overfitting to statistical noise in a manner similar to how traditional regularization constrains solution complexity.
- **Smooth Transformations**: Normalizing flows use smooth transformation functions, which naturally bias the solution toward smooth posterior distributions rather than ones with sharp, unphysical features.
- **Amortized Inference**: By learning a conditional distribution that applies across many possible measurements, NPU averages over many training examples, reducing sensitivity to outliers or statistical fluctuations in any single measurement.
- **Architectural Inductive Bias**: The specific architecture of the normalizing flow introduces an inductive bias. For instance, MADE architectures model dependen-

cies between variables in a structured way that can align with physical correlations between bins.

- **Optimization Dynamics**: The stochastic gradient descent training process itself provides regularization through early stopping based on validation performance, preventing overfitting to the training data.

This implicit regularization offers some advantages over the explicit regularization used in traditional methods. It adapts automatically to the complexity of the problem rather than requiring manual tuning of regularization parameters, it can handle different degrees of regularization for different regions of the solution space, providing stronger regularization where constraints are weak and less regularization where data provides strong constraints, and it emerges from general principles rather than specific assumptions about the solution (such as smoothness or proximity to a prior), potentially leading to less biased results. The effectiveness of this implicit regularization can be observed empirically in cases where NPU produces unfolded distributions with excellent agreement with the truth, despite the ill–posed nature of the problem **[cite –KD]**.

In the following section, we will demonstrate the application of NPU to concrete examples, including both controlled Gaussian distributions and realistic jet substructure measurements, to illustrate its performance in practice.

4.4 Numerical Results

This section presents comprehensive empirical validation of Neural Posterior Unfolding (NPU) through a series of increasingly complex examples. We first consider a simple two–bin degenerate case to illustrate NPU's advantages in handling unconstrained regions of phase space, then move to a Gaussian example to systematically assess statistical performance, and finally demonstrate the method's efficacy on realistic jet substructure simulation from the Large Hadron Collider (LHC).

2-Bin Degenerate Response Example

Our first example illustrates NPU's behavior by honing in degenerate scenarios where traditional methods struggle. We construct a two–bin setup where $t, m \in \mathbb{R}^2$ with a response matrix $\mathbf{R} \in \mathbb{R}^{2\times 2}$. We fix the truth values at $t_0 = t_1 = 5\times 10^4$ and parameterize the response matrix using a correlation coefficient ρ and diagonal elements $\sigma_0 = \sigma_1 \equiv \sigma = 0.8$.

When $\rho = 1$, the response matrix is well-conditioned with small off–diagonal terms, meaning each detector bin primarily receives readout from a single truth bin. In this case,

both IBU (with uncertainty quantification via bootstrapping) and NPU yield unfolded distributions that align well with the truth, with statistically consistent confidence regions. These results are demonstrated in **[fig:2bin:a –KD]**. However, as ρ approaches zero, the detector loses its ability to differentiate between the two truth bins. This creates a degeneracy where multiple truth configurations produce identical detector–level observations. Under these conditions, IBU provides only a point estimate based on the prior, with incorrectly estimated zero uncertainty on the ratio $t_0/(t_0 + t_1)$. In contrast, NPU returns a full posterior distribution with credible intervals that appropriately span all values consistent with the total counts, accurately reflecting the inherent degeneracy of the problem as seen in **[fig:2bin:b –KD]**.

This example highlights a fundamental advantage of NPU and other Bayesian methods; they naturally capture uncertainty in unconstrained regions of phase space, while traditional methods like IBU can dramatically underestimate uncertainties when degeneracies are present.

Gaussian Example

Our second test employs a Gaussian distribution, allowing us to systematically evaluate NPU's performance with varying levels of detector smearing.

Experimental Setup

We create two pairs of datasets drawn from one–dimensional Gaussian distributions, an generated "MC" dataset (D_{MC}) with 10^6 events drawn from a Gaussian with mean $\mu = 0$ and standard deviation $\sigma = 1$ at particle level, and a "natural" dataset (D_{nature}) with 10^5 events drawn from the same Gaussian

For detector effects, we apply Gaussian smearing to the generated data with parameter $\epsilon = 0.5$, resulting in detector–level distributions with the same mean but increased width, $\sigma_{\text{detector}} = \sqrt{1^2 + 0.5^2} \approx 1.12$. The response matrix is constructed from D_{MC}. We then use this response matrix to unfold the detector–level distribution of D_{nature} and compare the results with the known particle–level truth. The initial setup is illustrated in **[fig:gaus-init –KD]**.

Results and Comparison

The unfolded distributions from NPU (Maximum Likelihood Estimate), IBU, and FBU all recover the truth distribution accurately **[fig:gaus:a –KD]**. The NPU posterior provides a full characterization of the uncertainty, including bin–to–bin correlations visible in the corner plot **[fig:gaus:b –KD]**, which shows pairwise relationships between bins. The corner

plot demonstrates strong agreement between the posterior distributions from NPU and FBU, with both encompassing the true distribution.

To assess the statistical properties of NPU more rigorously, we evaluate pull distributions across 100 pseudo–experiments. For each bin i in pseudo-experiment j, we compute

$$\text{Pull}_{ij} = \frac{\mu_{ij}^{\text{method}} - t_i}{\sigma_{ij}^{\text{method}}} \tag{4.12}$$

where μ_{ij}^{method} and $\sigma_{ij}^{\text{method}}$ are the mean and standard deviation of the posterior in that bin. For properly calibrated uncertainty estimates, these pulls should follow a standard normal distribution with mean zero and unit variance.

The pull distributions for both NPU and FBU with moderate smearing ($\epsilon = 0.5$) are indeed centered at zero with unit width, as demonstraed in **[fig:pulls:a –KD]**, confirming proper calibration. We also tested the robustness of this calibration by varying the smearing parameter across the range $\epsilon \in [0.3, 0.6]$, finding that both methods maintain proper coverage throughout this range. These results are presented in **[fig:pulls:b –KD]**.

An important practical advantage of NPU is its computational efficiency for repeated unfolding tasks. For 100 pseudo-experiments, FBU with 10,000 MCMC draws requires approximately 40 seconds per experiment, totaling around 67 minutes. In contrast, NPU requires approximately 280 seconds for initial training, after which new datasets can be processed in just a few seconds each, resulting in a total time of approximately 5 minutes for all 100 experiments. This efficiency gain stems from NPU's amortized inference approach, which eliminates the need for repeated MCMC sampling when processing additional datasets **[cite –KD]**.

Particle Physics Example

Our final evaluation demonstrates NPU's application to a realistic high–energy physics scenario, focusing on jet substructure measurements at the Large Hadron Collider (LHC).

Dataset and Observables

We analyze simulated proton–proton collision data at $\sqrt{s} = 14$ TeV, following the setup in Andreassen et al. **[cite –KD]**. Two simulation generators are employed,

- Herwig 7.1.5 **[cite –KD]** serves as our "natural" data and truth distributions
- Pythia 8.243 with Tune 21 **[cite –KD]** is used to construct the response matrices.

To emulate detector effects, we use Delphes 3.4.2 **[cite –KD]** fast simulation of the CMS detector with particle flow reconstruction. Jets are clustered using the anti$-k_T$ algorithm

[cite –KD] with radius parameter $R = 0.4$, as implemented in FastJet 3.3.2 **[cite –KD]**. To minimize acceptance effects, we analyze only the leading jets in events containing a Z boson with transverse momentum $p_T^Z > 200$ GeV. After selection, approximately 1.6 million events from each simulation are retained.

We focus on four key jet substructure observables that are widely used in LHC analyses.

1. Jet width (ω): the transverse-momentum-weighted first radial moment of radiation within a jet,
2. Jet constituent multiplicity (M): the number of constituents in a jet
3. $N-$subjettiness ratio ($\tau_{21} = \tau_2^{\beta=1}/\tau_1^{\beta=1}$): quantifies the compatibility of a jet with a two–prong substructure hypothesis relative to a one–prong hypothesis **[cite –KD]**
4. Groomed momentum fraction (z_g): the momentum sharing between subjets after soft drop grooming **[cite –KD]**

These observables span a diverse range of physical characteristics and detector sensitivities, providing a comprehensive test of NPU's capabilities.

Results

Figure **[fig:substructure –KD]** presents the unfolded distributions for each observable, comparing results from NPU, FBU, and IBU against the known truth. For the FBU implementation in this more complex scenario, we needed to increase the number of MCMC steps ten-fold compared to the Gaussian example, using 100,000 tuning steps and 500,000 draws.

For all observables, NPU accurately recovers the truth distributions, with uncertainty bands that properly account for statistical uncertainties. The results demonstrate a few different characteristics of NPU. First, we can see that NPU effectively handles the complex detector effects present in realistic LHC measurements. Second, the method remains robust despite differences between the simulation used for response matrix construction (Pythia) and the "truth" distribution (Herwig). Finally, full posterior information enables rigorous uncertainty quantification across the entire distribution. The corner plots for these distributions **[fig:phys-corner –KD]** reveal the complex correlation structure between bins, information that is typically unavailable with traditional unfolding methods unless explicitly computed via bootstrapping or similar approaches.

These results highlight NPU's value for practical cross–section measurements at the LHC, where detector effects are complex and true distributions may differ significantly from simulation. In a full experimental analysis, uncertainties in the response matrix itself could also be incorporated, either by repeating the procedure with varied response matrices or by including these uncertainties directly in the likelihood.

Summary of Numerical Results

These numerical studies demonstrate that NPU provides appropriate uncertainty quantification in degenerate scenarios where traditional methods fail, while maintaining proper statistical coverage across varying degrees of detector smearing. NPU efficiently processes multiple datasets through amortized inference, and accurately recovers truth distributions in both targeted degenerate toy examples and realistic high–energy physics scenarios. The method combines the Bayesian foundations of FBU with several architectural and computational advantages. The normalizing flow architecture can represent a wide range of posterior distributions, including multimodal, asymmetric, and strongly correlated distributions that may be challenging for traditional MCMC methods. By providing differentiable access to the posterior density, NPU enables gradient–based optimization for finding maximum likelihood estimates or other derived quantities.

While FBU requires MCMC sampling for each new measurement, NPU's amortized inference approach front–loads computational cost into the training phase. Once trained, inference with NPU requires only forward passes through the neural network, making it particularly valuable for analyzing large datasets, performing multiple analyses with different systematic variations, bootstrapping for uncertainty estimation, and studies requiring many unfolding runs NPU also scales more favorably to high–dimensional problems compared to MCMC–based methods, which often suffer from the curse of dimensionality and mixing problems in complex posterior landscapes.

However, several limitations should be kept in mind. The flexibility of normalizing flows comes with increased model complexity, requiring careful architecture design and hyperparameter tuning. While less explicit than in traditional methods, NPU's results still depend on the distribution of the generated data, which implicitly defines a prior over the particle–level histograms. As with any Bayesian method, it's important to validate that the posterior credible intervals have the correct frequentist coverage properties for critical analyses.

Despite these considerations, NPU represents a significant advancement in unfolding methodology, combining the statistical rigor of Bayesian inference with the flexibility and computational efficiency of modern deep learning approaches. The method combines the statistical rigor of fully Bayesian approaches with the computational efficiency of neural network-based inference, making it a promising tool for cross–section measurements in particle and nuclear physics.

4.5 Beyond Binning: The Path Forward

The Neural Posterior Unfolding method presented in this chapter represents a significant advancement in binned unfolding approaches. By incorporating normalizing flows for posterior estimation, NPU addresses several limitations of traditional methods while maintaining the established binning paradigm that has served particle physics for decades. However, binning itself introduces fundamental constraints that motivate the development of alternative, unbinned methodologies.

First, binning inherently discards information about the precise location of events within each bin, reducing statistical precision. As Cowan notes, "The choice of binning is subjective and can introduce biases in the unfolded results"**[cite –KD]** a sentiment that is oft repeateed in widely-cited statistical treatments of the unfolding problem.**[cite –KD]** This discretization is particularly problematic for observables with sharp features or resonances that may be obscured by bin boundaries.

Second, the curse of dimensionality severely restricts binned methods' applicability to multivariate problems. The number of bins grows exponentially with the dimensionality of the observable space, quickly becoming computationally intractable and statistically limited by available data. For a typical analysis with 20 bins per dimension, a modest six–dimensional measurement would require 6.4×10^7 bins. Even with the large datasets available at modern colliders, this invariably leads to untenably sparsely populated bins, creating statistical challenges for traditional unfolding methods.

Third, binned methods require separate unfolding procedures for each differential distribution of interest. This approach is inefficient when multiple observables derived from the same dataset need to be unfolded, as is common in comprehensive physics analyses.

The Promise of Unbinned Methods

Unbinned unfolding approaches aim to overcome these limitations by operating directly on event–level data, preserving the full information content of the measurement. Recent advances in machine learning have enabled significant progress in this direction as reviewed in comprehensive surveys of machine learning applications for unfolding. The key conceptual shift that unbinned methods introduce is to reframe unfolding as a density reweighting problem rather than a histogram correction procedure. Instead of inverting a binned response matrix, unbinned methods typically aim to learn a transformation or reweighting function that maps the simulated particle-level distribution to the true one.

Several advantages emerge from an unbinned approach. By avoiding binning altogether, unbinned methods retain the full resolution of the data across the entire phase space. Unbinned techniques can be better suited to handle high-dimensional inputs, enabling simultaneous unfolding of multiple observables without combinatorial explosion. Unbinned

methods also provide more flexibility for downstream analyses. Once an unbinned unfolding model is trained, it can be used to derive any distribution or summary statistic from the unfolded data without requiring repeated unfolding procedures. Although machine learning methods are typically a few orders of magnitude slower than traditional methods like IBU, modern ML techniques can leverage efficient data representations and parallelised computation for faster inference, particularly for high-dimensional problems.

Emerging Approaches

Recent years have seen the rapid development of unbinned unfolding methods powered by advances in deep learning. We have already discussed classifier–based methods such as OmniFold, and generative models that use architectures such as Normalising Flows, VAEs, and GANs. Another direction that has shown great promise recently is *optimal transport* Optimal transport–based formulations recast unfolding as finding the minimum cost mapping between detector–level and particle–level distributions providing a theoretically grounded framework for the problem. These approaches have connections to both classical methods and modern machine learning techniques.

The next chapters will explore some unbinned methodologies in detail, examining their theoretical foundations, practical implementations, and performance characteristics on realistic physics examples. We will see how these methods build upon the insights from binned approaches like NPU while transcending their fundamental limitations to enable new capabilities for precision measurements in high-energy physics. As experimental datasets grow larger and theoretical predictions become more precise, unbinned methods will play an increasingly important role in extracting the full physics potential from collider experiments. The transition from binned to unbinned unfolding represents not just a technical evolution but a paradigm shift in how we approach the measurement and interpretation of differential cross sections.

Chapter 5

Moment Unfolding: Direct Deconvolution of Distribution Moments

5.1 Why Moments: Physics Context and QCD Calculations

Statistical moments of probability distributions serve as powerful tools in physics, providing a concise and theoretically meaningful way to characterise complex phenomena. In the context of particle physics, and particularly in quantum chromodynamics (QCD), working at the level of moments rather than distributions offers significant advantages for both theoretical calculations and experimental measurements. This section explores the fundamental importance of moments in physics, their theoretical foundations, and their specific applications in QCD.

The Theoretical Significance of Moments in Physics

Statistical moments represent a systematic way to characterize probability distributions, with each successive moment providing additional information about the shape and properties of the distribution. The k-th moment of a probability density $p(z)$ is defined as

$$\langle Z^k \rangle = \int_{-\infty}^{\infty} z^k p(z) dz \tag{5.1}$$

While the complete probability distribution contains all available information, working with moment is often much more tractable. In many physical theories, moments can be calculated analytically even when full distributions cannot. Each moment has a direct physical interpretation–the first moment corresponds to the mean, the second moment to the variance, and higher moments characterize asymmetry (skewness) and peakedness

(kurtosis). Moments therefore provide much more concrete insight into the properties of an observable than distributions.

In many physical theories, including QCD, there exist relatively simple evolution equations for how moments change as a function of energy scale. If certain moments exhibit universal behaviour across different physical systems, this too reveals universality in the fundamental underlying principles. For many physical systems, including those governed by QCD, moments provide a natural language for connecting theoretical predictions with experimental measurements, particularly when examining how observables scale with energy.

Moments in QCD Calculations

In quantum chromodynamics, moments play a particularly crucial role in understanding the scaling behaviour of parton distributions and fragmentation functions. The DGLAP equations form the theoretical framework for parton distribution function evolution.

DGLAP Evolution Equations

The Dokshitzer–Gribov–Lipatov–Altarelli–Parisi (DGLAP) evolution equations**[cite –KD]** govern how parton distribution functions evolve with energy scale. While these equations are integro–differential equations in their most general form, they transform into ordinary differential equations when expressed in terms of moments, significantly simplifying their solution **[cite –KD]**.

For a parton distribution function $f(x, Q^2)$, where x is the momentum fraction and Q^2 is the energy scale, the $n-$th moment is defined as

$$M_n(Q^2) = \int_0^1 x^{n-1} \, f(x, Q^2) \mathrm{d}x. \tag{5.2}$$

The DGLAP equations for these moments take the form

$$\frac{\mathrm{d}M_n(Q^2)}{\mathrm{d}\ln Q^2} = \frac{\alpha_s(Q^2)}{2\pi} \, P_n \, M_n(Q^2). \tag{5.3}$$

where P_n are the moments of the splitting functions. This formulation converts the complex integro–differential evolution equations into a set of simple ordinary differential equations, one for each moment independently.

Operator Product Expansion

The Operator Product Expansion (OPE) in QCD naturally expresses deep inelastic scattering structure functions in terms of moments **[cite –KD]**. In this formalism, the moments

of structure functions are related to matrix elements of local operators, which have clear physical interpretations and scaling properties. For example, the moments of the structure function $F_2(x, Q^2)$ can be expressed as

$$M_n(Q^2) = \int_0^1 x^{n-2}\, F_2(x, Q^2)\, \mathrm{d}x = \sum_i C_n^i(Q^2)\, \langle O_n^i \rangle. \tag{5.4}$$

where C_n^i are Wilson coefficients calculable in perturbation theory, and $\langle O_n^i \rangle$ are matrix elements of local operators.

Event Shape Moments

Event shape observables, which characterize the geometrical properties of energy flow in collisions, are often studied through their moments. Moments of event shapes like thrust, broadening, and jet mass distributions provide sensitive tests of perturbative QCD and allow precise extractions of the strong coupling constant α_s **[cite –KD]**. The moments of these event shapes can be calculated as

$$\langle e^n \rangle = \int_0^{e_{\max}} e^n\, \frac{1}{\sigma}\, \frac{\mathrm{d}\sigma}{\mathrm{d}e}\, \mathrm{d}e, \tag{5.5}$$

where e is the event shape variable, and $\frac{1}{\sigma}\frac{\mathrm{d}\sigma}{\mathrm{d}e}$ is its normalized differential cross section. These moments can be predicted in perturbative QCD, typically expressed as power series in α_s:

$$\langle e^n \rangle = A_n \left(\frac{\alpha_s}{2\pi}\right) + B_n \left(\frac{\alpha_s}{2\pi}\right)^2 + \mathcal{O}(\alpha_s^3), \tag{5.6}$$

plus non–perturbative corrections that scale as inverse powers of the centre–of–mass energy.

Experimental Significance of Moments in QCD

From an experimental perspective, moments offer several advantages that make them particularly valuable for QCD studies.

Some of the most precise determinations of the strong coupling constant α_s come from measurements of moments. For example, moments of event shapes measured at LEP have provided determinations of α_s with uncertainties at the few-percent level **[cite –KD]**. Studies by AMY, TASSO, OPAL, DELPHI, L3, and ALEPH collaborations have all used moment–based analyses to extract α_s values that contribute significantly to the world average **[cite –KD]**.

The scaling behavior of moments with energy provides a direct and robust window into the running of QCD couplings. By measuring how moments of distributions change with

energy scale, experiments can observe the logarithmic scaling violations that are a hallmark of asymptotic freedom in QCD **[cite –KD]**.

Many theoretical QCD calculations are more readily available (and more precise) for moments than for full distributions. For certain observables, perturbative calculations may exist to next–to–next–to–leading order (NNLO) or beyond for moments, while full distribution calculations may only be available at lower orders **[cite –KD]**.

In practice, measuring moments directly can reduce certain experimental uncertainties, particularly those related to detector resolution and acceptance effects in regions of phase space with sparse statistics.

Traditionally, there was no straightforward method to directly unfold moments. Instead, moments were computed after first measuring and unfolding the full distribution. This typically involved binning the data, unfolding the binned distribution, and then computing moments from the unfolded histogram. The limitation of this approach though, is the binning itself, not the moments qua moments.

The use of discrete bins introduces a bias in moment calculations. For a variable z binned into n_{bins} with centers $z_{\text{bin},i}$ and counts N_i, the binned approximation of the $k-$th moment is

$$\langle z^k \rangle_{\text{bin}} = \frac{1}{N} \sum_{i=1}^{n_{\text{bins}}} N_i z_{\text{bin},i}^k \tag{5.7}$$

This approximation introduces a bias relative to the true moment, which grows with the order of the moment computed, and is particularly large in regions where the distribution varies rapidly within bins **[cite –KD]**.

These limitations motivate the development of unbinned approaches that can directly unfold moments without first discretizing the data, which is precisely the focus of the Moment Unfolding method described in this chapter.

Applications in Jet Physics

In the context of jet physics specifically—a central arena for testing QCD—moments of jet substructure observables provide particularly powerful probes of QCD dynamics **[cite –KD]**>. Several notable applications deserve special mention.

The energy scale dependence of jet properties is best studied through their moments. The moments of jet substructure observables such as jet mass, width, and multiplicity exhibit characteristic scaling with jet energy that can be predicted by perturbative QCD. For example, the first moment of the jet mass scales approximately as **[cite –KD]**>

$$\langle m^2 \rangle \propto \alpha_s(p_T)\ p_T^2\ \ln(R/R_0) \tag{5.8}$$

where p_T is the jet transverse momentum, R is the jet radius, and R_0 is the reference scale. Higher moments have different scaling behaviours, providing multiple handles on the underlying QCD dynamics.

The moments of jet substructure variables are also used in the classification of jets. This is because jet moments differ characteristically between quark–initiated and gluon–initiated jets, making them valuable for jet flavour tagging. For instance, gluon jets typically have higher multiplicity and broader mass distributions than quark jets, reflected in the moments of these distributions **[cite –KD]**.

Jet substructure moments also provide a window to study non–perturbative effects and hadronization. The interplay between perturbative and non-perturbative effects in QCD is particularly evident in moments of jet observables. Lower moments often capture the perturbative, large–angle radiation physics, while higher moments become increasingly sensitive to non–perturbative effects like hadronization **[cite –KD]**.

In the context of Soft-Collinear Effective Theory (SCET), moments of jet substructure observables provide direct constraints on the parameters of the effective theory, offering a bridge between first–principles QCD and phenomenological models of jet formation **[cite –KD]**.

Moments in Physics Beyond the Standard Model

While moments are crucial for precision QCD studies, their utility extends to searches for BSM physics. Certain moments of distributions can exhibit enhanced sensitivity to new physics scales. For example, higher moments of mass distributions can be more sensitive to heavy particle contributions than the full distribution average **[cite –KD]**.

Moments can therefore provide a model–independent way to parametrize deviations from Standard Model predictions, similar to the role of effective field theory coefficients. By measuring a series of moments, experiments can place constraints on a wide class of possible new physics scenarios without committing to specific models **[cite –KD]**. Hence, changes in the moment structure of distributions can serve as an early indicator of anomalous physics, even before the full nature of the anomaly is understood <kdcite>.

The Case for Direct Moment Unfolding

Given the theoretical significance of moments in QCD and their experimental advantages, there is a strong motivation for developing methods that can directly unfold moments from detector–level data without first unfolding the entire distribution. Such methods would eliminate binning biases inherent in traditional approaches, while potentially improving precision and reducing computational cost by focusing directly on the quantities of interest. They would enable moment measurements in higher–dimensional phase spaces where

binned approaches become impractical to provide more direct comparisons with theoretical predictions that are formulated at the level of moments.

The remainder of this chapter presents the Moment Unfolding method, a novel approach that addresses precisely this need by directly deconvolving moments of distributions using machine learning techniques inspired by the Boltzmann distribution and implemented through a GAN–like architecture. This method offers a significant advance in precision moment measurements, enabling more stringent tests of QCD predictions and potentially uncovering subtle effects that might be obscured in traditional analyses.

5.2 A GAN–like Method to Unfold Moments

The direct extraction of distribution moments without first reconstructing the entire distribution presents a unique challenge in unfolding methodology. Traditional approaches typically reconstruct the full differential cross–section before computing moments, introducing unnecessary computational complexity. The Moment Unfolding technique presented in this chapter takes an entirely different approach by directly targeting the moments themselves.

The basic idea underlying Moment Unfolding is the connection between statistical moments and the Boltzmann distribution from statistical mechanics. In thermodynamics, the Boltzmann distribution represents the probability distribution that maximizes entropy while satisfying certain constraints on average quantities (internal energy). This principle can be adapted for our unfolding problem, where we seek to find a reweighting of simulated events that accurately reproduces the moments of the true particle–level distribution.

Boltzmann inspired reweighting

The central mathematical construction of Moment Unfolding is a reweighting function inspired by the Boltzmann factor,

$$g(z) = \frac{1}{P(\boldsymbol{\beta})} \exp\left(-\sum_{a=1}^{n} \beta_a \, z^a\right) \tag{5.9}$$

where z is the particle–level observable whose moments we wish to unfold, β_a are parameters to be determined (analogous to Lagrange multipliers in statistical mechanics), P is a normalization factor similar to the partition function, and n is the number of moments we aim to unfold simultaneously.

This form is particularly powerful because it directly connects to the maximum entropy principle in statistical mechanics. Just as the Boltzmann distribution maximizes entropy

subject to constraints on average energy, this reweighting function maximizes the binary cross entropy loss between the discriminator's classification output between the reweighted distribution and the prior distribution while enforcing constraints on the moments. When applied to a simulated particle–level distribution with probability density $q_{\text{Gen.}}(z)$, this reweighting function produces a modified distribution

$$\widetilde{q}(z) = g(z) \cdot q_{\text{Gen.}}(z) \tag{5.10}$$

The training objective then is to determine the optimal values of the parameters β_a such that the moments of $\widetilde{q}(z)$ match those of the true underlying distribution $p(z)$.

Adversarial Training Framework

Given the theoretical foundation provided by the Boltzmann form, we now need a practical method to determine the optimal parameters. The task can very naturally be framed as a two–level optimization problem:

1. The parameters β_a control the reweighting at the particle level
2. The quality of the reweighting must be assessed at the detector level

This task is highly non–trivial because the detector response function—which maps particle-level to detector–level observables—-is typically stochastic and known only implicitly through simulation.

To address this, Moment Unfolding employs an adversarial training architecture inspired by Generative Adversarial Networks (GANs), but with significant modifications tailored to the unfolding problem. Unlike traditional GANs that generate new samples, our "generator" $g(z)$ reweights existing particle-level simulated events according to the Boltzmann form. The discriminator is a neural network classifier $d(x)$ that attempts to distinguish between real detector–level data and the reweighted simulation at detector level.

The system is then trained adversarially, with the generator attempting to fool the discriminator while the discriminator attempts to correctly classify events. This setup is illustrated in Figure **[fig:moment-unfolding-architecture –KD]**, showing the flow of information and the adversarial training process. In this framework, the optimal parameters β_a are those that make the reweighted detector–level simulation indistinguishable from the real detector–level data. Once these parameters are found, the moments of the particle–level reweighting are the unfolded moments.

Mathematical Formalism

We can formalize the Moment Unfolding approach as follows. Let $p(x)$ be the detector–level data distribution and f$q(z, x)$ be the joint distribution of particle–level and detector–level variables in the MC, Generation and Simulation. Then $q(z)$ and $q(x)$ are the marginal distributions of Generation and Simulation respectively. Let $r(x|z)$ be the detector response function. The assumption of a universal detector response can be formalized as

$$p(x|z) = q(x|z) = r(x|z). \tag{5.11}$$

The goal is to find parameters β_a such that the moments of the reweighted distribution $g(z) \cdot q(z)$ match the true moments of the underlying particle–level distribution, $p(z)$.

The detector–level distribution after reweighting is

$$\widetilde{q}(x) = \int g(z) \ \cdot q(z, x) \, \mathrm{d}z. \tag{5.12}$$

The adversarial training process optimizes a weighted binary cross entropy loss function,

$$\mathcal{L}[g, d] = -\mathbb{E}_{X\sim p(x)}[\log d(X)] - \mathbb{E}_{X\sim \widetilde{q}(X)}[(1 - d(x))] \tag{5.13}$$

This loss function encourages the discriminator to output high values for real data and low values for reweighted simulation, while the generator tries to make the reweighted simulation indistinguishable from real data.

Once the optimal parameters β_a are determined, the unfolded moments can be computed directly from the reweighted Generation

$$\langle Z^k \rangle_{\text{unfolded}} = \frac{\sum_{z\in\text{Gen.}} g(z) \cdot z^k}{\sum_{z\in\text{Gen.}} g(z)} \tag{5.14}$$

This approach differs fundamentally from traditional unfolding methods that first reconstruct the full distribution and then compute moments. By directly targeting the moments themselves, we avoid both the binning artifacts and dimensionality challenges associated with binned methods, and the computational complexity associated with unbinned full distribution unfolding.

Theoretical Properties

The Moment Unfolding method possesses several important theoretical properties that justify its application to physics problems. Under certain conditions (detailed in Appendix**[ref –KD]**), the Moment Unfolding method provably converges to the correct moments.

The parameter n (the number of moments being unfolded) serves as an implicit regularization parameter. By limiting the number of moments, we constrain the complexity of the reweighting function, helping to stabilize the unfolding process against statistical fluctuations. This creates a natural bias-variance tradeoff: including more moments allows for a more flexible reweighting function but increases sensitivity to statistical fluctuations. In practice, the optimal choice of n depends on the physics application and available statistics.

Connection to the Maximum Entropy Principle

The Boltzmann form of the reweighting function has a deep connection to the maximum entropy principle in statistical mechanics. The reweighting function maximizes the entropy of the reweighted distribution relative to the prior (simulation) distribution, subject to constraints on the moments. This provides a principled physical and informational theoretic basis for the method: among all possible reweighting functions that reproduce the target moments, the Boltzmann form is the one that introduces the least additional information beyond what is required by the moment constraints.

Comparison with Traditional GAN Architectures

While Moment Unfolding draws inspiration from GANs, it differs from traditional GAN architectures, in that the so–called “generator" does not generate data; it generates weights. Traditional GANs generate new samples by mapping random noise through a neural network. In contrast, Moment Unfolding reweights existing simulated events, preserving the physics correlations already present in the simulation and focusing only on correcting the distribution’s moments.

Furthermore, unlike conventional GANs where the generator can represent any function within the capacity of the neural network, our generator has an extremely specific parametric form constrained by the Boltzmann expression. This constraint massively reduces the effective degrees of freedom in the optimization process and encodes our prior knowledge about the problem structure. The parameters β_a in the Moment Unfolding approach have clear physical interpretations as Lagrange multipliers enforcing moment constraints. This interpretability contrasts with the typically opaque nature of neural network weights in traditional GANs.

Finally, traditional GANs operate at a single level, with the generator and discriminator acting on the same “type" of data. Moment Unfolding operates across two levels, particle–level and detector–level, which might not even have the same dimensionality, which is the source of the additional complexity but is essential for addressing the detector response effect.

Practical Considerations

While theoretically elegant, the Moment Unfolding method has practical considerations and limitations that must be understood in order to apply the method correctly and effectively.

Like all reweighting–based methods, Moment Unfolding requires that the Generation distribution has overlapping support with the Truth distribution. Regions of phase space not covered by the MC cannot be properly unfolded. With finite statistics, extremely large weights can arise, in regions of phase space with sparse MC coverage as the generator tries and fails to reweight a zero density into a non–zero density. Various regularization techniques must be employed to mitigate these effects if they arise in a given analysis.

The optimal number of moments to unfold depends on the physics application, available statistics, and the complexity of the true distribution. Too few moments may miss important features of the distribution, while too many may lead to overfitting and numerical instabilities.

Extension to Differential Measurements

An extension of the Moment Unfolding approach is to study moments as a function of another observable, such as transverse momentum. This is achieved by making the parameters β_a functions of the conditional variable,

$$g(z; p_T) = \frac{1}{P(\boldsymbol{\beta}(p_T))} \exp\left(- \sum_{a=1}^{n} \beta_a(p_T)\ z^a \right) \tag{5.15}$$

This allows the study of how moments evolve with energy scale or other experimental conditions, providing direct comparisons with theoretical predictions of scaling behaviors in QCD.

The parameters $\beta_a(p_T)$ can be modeled as polynomials, splines, or neural networks, depending on the expected complexity of their dependence on p_T. This extension significantly enhances the physics applications of the method, enabling studies of how observable moments scale with energy.

Theoretical Foundations in Statistical Mechanics

The mathematical form at the heart of Moment Unfolding has deep connections to statistical mechanics and information theory, providing a solid theoretical foundation for the method. In statistical mechanics, the Boltzmann distribution arises as the solution to the maximum entropy problem. The maximum entropy problem is the problem of finding

the probability distribution that maximizes entropy subject to constraints on expectation values (typically energy). The solution takes the form

$$p(s) = \frac{1}{Z} e^{-\beta E(s)} \tag{5.16}$$

where $E(s)$ is the energy of state s, β is the inverse temperature, and Z is the partition function.

The Moment Unfolding algorithm borrows this idea to include multiple constraints on different moments, with each β_a serving as a Lagrange multiplier for the constraint on the a-th moment. This connection to maximum entropy principles provides a theoretical justification for the specific form of the reweighting function and suggests that it is, in some sense, the most natural choice for the moment unfolding problem. The success of this approach in practical applications, as will be demonstrated in subsequent sections, confirms the value of this theoretical grounding and highlights the power of cross-disciplinary approaches in developing new data analysis techniques for particle physics.

5.3 Machine Learning Implementation

The theoretical framework introduced in the previous section provides the foundation for Moment Unfolding, but implementing this approach in practice requires careful architectural design and optimization strategies. This section details the machine learning implementation that enables robust moment extraction through our GAN–like approach.

The generator in Moment Unfolding differs fundamentally from traditional GAN generators. Instead of producing new samples through a complex neural network, our generator implements the Boltzmann–inspired weighting function, Eq. **??**. This function is applied to existing simulated particle–level events to reweight them, producing a weighted distribution that matches the moments of the true distribution. The parameters β_a are the trainable parameters of the generator, and P is a normalization constant estimated at the batch level,

$$\hat{P} = \sum_{z \in \text{batch}} \exp\left(- \sum_{a=1}^{n} \beta_a z^a \right). \tag{5.17}$$

For differential moment measurements, where we study the dependence of moments on another variable like transverse momentum (p_T), the β_a parameters become functions of p_T, as described in Eq. **??**. In our implementation, we parameterize $\beta_a(p_T)$ as linear functions,

$$\beta_a(p_T) = \beta_a^{(0)} + \beta_a^{(1)} p_T \tag{5.18}$$

This linear parameterization is motivated by empirical observations that the ratios of spectra between different simulations often exhibit approximately linear dependence on kinematic variables **[cite –KD]**.

The discriminator network $d(x)$ takes detector–level features as input and outputs a probability that a given event came from the real data rather than the reweighted simulation. For our implementation, we use a dense neural network with three fully–connected hidden layers with 50 nodes per layer. ReLU activation functions are applied to intermediate layers, and a sigmoid activation function is applied to the output layer.

This relatively simple architecture provides sufficient capacity to distinguish between data and simulation distributions.

The training objective is formulated as a weighted version of the Maximum Likelihood Classifier (MLC) loss **[cite –KD]**,

$$L[g,d] = -\sum_{x\in\text{Data}} \log d(x) - \sum_{(z,x)\in(\text{Gen., Sim.})} g(z)(1-d(x)) \tag{5.19}$$

where the first sum is over detector-level events from real data, and the second sum is over matched pairs of particle–level and detector–level events from simulation, weighted by the generator function $g(z)$.

This loss function differs from the standard Binary Cross Entropy (BCE) loss often used in GANs, providing smoother gradients and better convergence properties for our specific application. While the BCE loss function could also be used (and indeed gives similar empirical performance), the MLC loss function provides certain theoretical guarantees about the convergence of the method to the correct moments **[cite –KD]**.

Training Procedure

The training procedure for Moment Unfolding follows the standard adversarial approach, but with specific adaptations for the reweighting framework.

The generator parameters β_a are initialized to small random values near zero, corresponding to minimal reweighting (i.e., starting close to the original simulation). The discriminator weights are initialized using standard neural network initialization techniques**[cite HeNormal initialization –KD]**.

During each training step, we sample batches of events from Data, Generation, and Simulation. The training proceeds by alternating between optimizing the discriminator and the generator. First, with the generator parameters fixed, the discriminator weights are updated to minimize the loss function. Then with the discriminator weights fixed, the generator parameters are updated to maximize the loss function. This alternating optimization continues for a fixed number of epochs, or until convergence criteria are met.

We employ the Adam optimizer **[cite –KD]** with an initial learning rate of 5×10^{-4} for both the discriminator and generator The learning rate is reduced by a factor of 0.5 every 20 epochs to ensure stable convergence.

To ensure stable training and prevent overfitting, a few regularization techniques are employed. Gradient values are clipped to the range $[-1, 1]$ to prevent large updates that could destabilize training, particularly important for the generator parameters which can be sensitive to statistical fluctuations. Batch normalization **[cite –KD]** is applied after each hidden layer in the discriminator to stabilize training and reduce sensitivity to weight initialization and learning rates.

Gradient Updates

For the discriminator, gradient computation follows standard backpropagation through the neural network. For the generator, gradients are computed with respect to the parameters β_a by differentiating the loss function.

$$\frac{\partial L}{\partial \beta_a} = - \sum_{(z,x)\in(\text{Gen., Sim.})} \frac{\partial g(z)}{\partial \beta_a}(1 - d(x)), \tag{5.20}$$

where

$$\frac{\partial g(z)}{\partial \beta_a} = -g(z)\left(z^a - \frac{1}{P}\frac{\partial P}{\partial \beta_a}\right) = -g(z)(z^a - \langle z^a \rangle_g) \tag{5.21}$$

Here, $\langle z^a \rangle_g$ represents the $a-$th moment computed from the reweighted distribution. This formulation provides a direct connection between the gradient updates and the moment matching objective of the method.

Implementation Details

he method is implemented using Keras **[cite –KD]** with the TensorFlow2 backend **[cite –KD]**, leveraging their automatic differentiation capabilities for efficient gradient computation. NumPy **[cite –KD]** is used for data manipulation, and Matplotlib **[cite –KD]** is employed for visualization.

The computational requirements of Moment Unfolding are remarkably light compared to many deep learning applications. Training typically takes less than five minutes per observable on an NVIDIA RTX6000 GPU for the case studies presented in this dissertation. Even the potentially computationally intensive calculation of the partition function P, which requires summation over all events in a batch is implemented efficiently using vectorized operations in TensorFlow.

Hyperparameter	**Value**	**Description**
Discriminator layers	3	Number of hidden layers in the discriminator
Discriminator nodes	50	Number of nodes per hidden layer
Activation function	ReLU	Activation function for hidden layers
Output activation	Sigmoid	Activation function for the output layer
Optimizer	Adam	Optimization algorithm used for training
Learning rate	5×10^{-4}	Initial learning rate for the optimizer
Batch size	10,000	Number of training events in each batch
Training epochs	50–100	Number of full passes through the training dataset
Gradient clip value	1.0	Maximum allowed gradient norm for clipping

Table 5.1: Summary of training hyperparameters used in the model. These values control the architecture, optimization behaviour, and regularization of the training process.

Tab. **??** summarizes the key hyperparameters used in our implementation. These hyperparameter values were determined through systematic experimentation and provide a good balance between training stability and model performance across a range of physics scenarios.

Extensions to Multiple Observables

While the basic formulation of Moment Unfolding applies to moments of a single observable, the method can be extended to handle multiple observables simultaneously. For multiple observables $z_1, z_2, ..., z_d$, we can define joint moments as $\langle z_1^{k_1} z_2^{k_2} \cdots z_d^{k_d} \rangle$. The reweighting function then takes the form,

$$g(z_1, z_2, ..., z_d) = \frac{1}{P} \exp\left(- \sum_{k_1,k_2,...,k_d} \beta_{k_1,k_2,...,k_d} z_1^{k_1} z_2^{k_2} \cdots z_d^{k_d} \right) \tag{5.22}$$

In practice, including all possible joint moments can quickly become intractable as the number of observables increases. To address this, one can employ strategies such as

- Limiting the total order of moments (e.g., $k_1 + k_2 + ... + k_d \leq K_{max}$),
- Including only specific joint moments known to be physically relevant, and
- Using factorized forms that capture the most important correlations while maintaining computational feasibility.

Uncertainty Estimation

Accurate uncertainty estimation is crucial for meaningful physics measurements. In Moment Unfolding, uncertainties on the extracted moments are obtained through bootstrap resampling. Statistical uncertainties are estimated by generating N bootstrap replicas of the data by resampling with replacement, applying Moment Unfolding to each replica, and computing the standard deviation of the resulting moment values across all replicas. This procedure accounts for statistical uncertainties in both the data and the simulation.

Systematic uncertainties related to theoretical modeling, detector effects, and other sources are evaluated by varying the underlying simulation according to systematic uncertainty sources, applying Moment Unfolding to each variation, and taking the envelope of the resulting moment values as the systematic uncertainty.

This approach allows for comprehensive uncertainty quantification that accounts for all relevant sources of uncertainty in the measurement.

Validation Procedures

Several validation procedures are employed to ensure the reliability of the Moment Unfolding method. Closure tests verify that the method can correctly recover known moments from simulated data. The method's internal consistency is verified by checking that the moments computed from the reweighted Generation match the target moments, the reweighted Simulation reproduces the features of the Data, and that the discriminator is unable to distinguish between Simulation and Data.

Results from Moment Unfolding are also compared with those from traditional unfolding methods like Iterative Bayesian Unfolding followed by moment calculation. Agreement between different methods provides additional confidence in the results, while disagreement may indicate methodological issues that require further investigation.

Code Availability and Reproducibility

The implementation of Moment Unfolding is available as open–source Python code, facilitating reproducibility and adoption by the scientific community. This open–source approach promotes transparency, enables independent verification of results, and facilitates improvements and extensions to the method by the broader scientific community. The full code repository can be found at [GitHub URL] **[cite –KD]**, and the specific datasets used for the studies in this dissertation are archived on Zenodo **[cite –KD]**.

By combining a theoretically motivated approach with practical machine learning implementation details, Moment Unfolding provides a robust and efficient method for extracting moments directly from data. The next sections will demonstrate the application of this

method to jet substructure measurements, showcasing its performance on realistic physics problems.

5.4 Case studies

Gaussian experiments

Before applying Moment Unfolding to jet physics scenarios, it is essential to validate the method in a controlled environment where the ground truth is precisely known. To this end, the approach is first demonstrated using a one–dimensional Gaussian example, providing a clear illustration of the method's capabilities while establishing a benchmark for its performance.

Experimental Setup

For this controlled study, we generate datasets from Gaussian distributions with known parameters. The Truth dataset comprises $10,000$ events drawn from a Gaussian distribution with mean $\mu_{\text{Truth}} = 0.0$ and variance $\sigma^2_{\text{Truth}} = 1$. The Generation dataset comprises $100,000$ events drawn from a Gaussian distribution with mean $\mu_{\text{Gen.}} = -0.5$ and variance $\sigma^2_{\text{Gen.}} = 1.0$. Both distributions are then subjected to detector effects modelled as additive Gaussian noise with resolution parameter $\sigma_{\text{det}} = 0.5$. This setup mimics the typical scenario in particle physics where the MC differs from the true distribution, and both are observed through an imperfect detector.

Figure **[fig:gaussian-setup –KD]** illustrates the particle–level and detector–level distributions, showing how the detector resolution affects both the truth and simulation datasets.

Results

Figure **[fig:gaussian-results –KD]** shows the results of applying Moment Unfolding to the Gaussian example. The left panel displays the particle–level distributions, the Truth distribution (blue), the Generation (orange), and the reweighted Generation (black dashed line). Even visually, the close overlap between the Truth histogram and the reweighted Generation demonstrates the success of the Moment Unfolding procedure.

The right panel of Figure **[fig:gaussian-results –KD]** provides a more quantitative perspective by showing the discriminator–optimized loss landscape as a function of the Boltzmann parameters β_1 and β_2. This landscape visualization allows us to verify that the maximum of the loss function (red star) aligns with the parameter values that produce the correct moments, providing empirical confirmation of the method's theoretical properties.

These results confirm that Moment Unfolding successfully recovers the true moments of normal distributions. Since normal distributions are fully characterized by their first two moments, the reweighted distribution closely matches the entire truth distribution in this Gaussian case even though the method explicitly targets only the first two moments. We should not expect this to be the case for more complex distributions with non–trivial higher moments.

This controlled study confirms that Moment Unfolding performs as expected in a well–understood scenario, providing confidence to extend the method to more complex physics environments.

Jet Substructure in Collider Physics

Having validated Moment Unfolding in a controlled environment, we now apply the method to a realistic high–energy physics scenario: measuring moments of jet substructure observables in proton–proton collisions at the Large Hadron Collider (LHC).

Datasets

The simulated samples used for this study follow the setup from Andreassen et al. **[cite –KD]**. Proton–proton collisions at $\sqrt{s} = 14$ TeV are simulated using Pythia 8.243 with Tune 26 **[cite –KD]** (standing in for MC) and Herwig 7.1.5 **[cite –KD]** (standing in for nature). The Delphes 3.4.2 **[cite –KD]** fast simulation of the CMS detector is used to model detector effects, configured with particle flow reconstruction **[cite –KD]**. Jets are clustered using the anti-k_T algorithm **[cite –KD]** with radius parameter $R = 0.4$, implemented in FastJet 3.3.2**[cite –KD]**. To reduce acceptance effects, only leading jets in events with a Z boson having transverse momentum $p_T^Z > 200$ GeV are studied. This setup allows us to test Moment Unfolding in a realistic environment where the “Truth" (Herwig particle–level) and “Generation" (Pythia particle–level) are different generators with distinct physics models, and both are observed through an imperfect detector to produce “Data", Herwig + Delphes and “Simulation", Pythia + Delphes.

Observables

Four key jet substructure observables are unfolded, each providing different insights into the internal structure of jets.

1. **Jet Mass** (m): The invariant mass of the jet, calculated as

$$m = \sqrt{\sum_k E_k^2 - \sum_k \mathbf{p}_k^2}, \tag{5.23}$$

where the sum runs over all constituents of the jet, and E_k and $\mathbf{p}_k$ are their energies and three–momenta.

2. **Jet Charge** (q): An infrared–safe but not collinear–safe observable that measures the jet's electric charge **[cite –KD]**.

$$q = \frac{1}{\sum_k \sqrt{p_{T,k}}} \sum_k q_k \sqrt{p_{T,k}}, \tag{5.24}$$

where q_k is the electric charge of constituent k.

3. **Jet Width** (w): A measure of the radial distribution of radiation within the jet.

$$w = \frac{1}{p_{T,\text{jet}}} \sum_k p_{T,k} \Delta R_k, \tag{5.25}$$

where ΔR_k is the angular distance between constituent k and the jet axis.

4. **Groomed Momentum Fraction** (z_g): The momentum sharing between subjets after Soft Drop grooming **[cite –KD]** with $z_{\text{cut}} = 0.1$ and $\beta = 0$.

$$z_g = \frac{p_{T,\text{subleading}}}{p_{T,\text{leading}} + p_{T,\text{subleading}}} \tag{5.26}$$

These observables span a diverse range of jet properties, including both infrared–and–collinear safe observables (m and w), a Sudakov safe observable, z_g**[cite –KD]**, and q, which is only infrared safe.

Results

Table **??** presents the first and second moments of each jet observable extracted using Moment Unfolding, compared with the true values (from Herwig) and those from the original generation (Pythia). The uncertainties in the Truth and Generation columns are computed by bootstrapping the respective datasets. For the Moment Unfolding column, the uncertainty includes the bootstrap uncertainty from the Generation and the empirical uncertainty from the unfolding process added in quadrature.

These results demonstrate that Moment Unfolding successfully recovers the true moments from the data for all observables within statistical uncertainties. The method effectively corrects for both the differences between the Pythia and Herwig physics models and the detector distortions introduced by Delphes.

Figure **[fig:jet-substructure-distributions –KD]** shows the particle–level distributions for each observable, comparing the truth, generation, and reweighted generation after

Table 5.2: Moments of jet observables at particle level. First and second moments of m, q, w and z_g are shown for truth (Herwig), generation (Pythia) and Moment Unfolding. Uncertainties in the truth and generation columns are estimated via bootstrap resampling; uncertainties in the unfolding column combine in quadrature the generation bootstrap uncertainty with the empirical 1σ spread from repeated unfolding on the same dataset.

	Truth (Herwig)	Generation (Pythia)	Moment Unfolding
$\langle m \rangle$	$(2.182 \pm 0.030) \times 10^{1}$	$(2.064 \pm 0.043) \times 10^{1}$	$(2.173 \pm 0.047) \times 10^{1}$
$\langle m^2 \rangle$	$(6.049 \pm 0.222) \times 10^{2}$	$(5.360 \pm 0.350) \times 10^{2}$	$(6.115 \pm 0.364) \times 10^{2}$
$\langle q \rangle$	$(1.006 \pm 0.037) \times 10^{-2}$	$(1.582 \pm 0.038) \times 10^{-2}$	$(1.090 \pm 0.040) \times 10^{-2}$
$\langle q^2 \rangle$	$(1.216 \pm 0.082) \times 10^{-2}$	$(1.508 \pm 0.074) \times 10^{-2}$	$(1.207 \pm 0.074) \times 10^{-2}$
$\langle w \rangle$	$(1.498 \pm 0.025) \times 10^{-1}$	$(1.231 \pm 0.029) \times 10^{-1}$	$(1.499 \pm 0.029) \times 10^{-1}$
$\langle w^2 \rangle$	$(3.370 \pm 0.113) \times 10^{-2}$	$(2.421 \pm 0.128) \times 10^{-2}$	$(3.374 \pm 0.128) \times 10^{-2}$
$\langle z_g \rangle$	$(2.334 \pm 0.029) \times 10^{-1}$	$(2.457 \pm 0.030) \times 10^{-1}$	$(2.353 \pm 0.059) \times 10^{-1}$
$\langle z_g^2 \rangle$	$(6.789 \pm 0.166) \times 10^{-2}$	$(7.425 \pm 0.165) \times 10^{-2}$	$(6.767 \pm 0.330) \times 10^{-2}$

applying Moment Unfolding While the method explicitly targets only the first two moments, it produces reasonable agreement across the entire distribution, particularly for the regions that contribute most significantly to the moments. Figure **[fig:jet-loss-landscapes –KD]** displays the discriminator–optimized loss landscapes for each observable, confirming that the maximum of the loss function aligns with the parameter values that produce the correct moments. The blue ellipses represent the 1σ confidence intervals for the Moment Unfolding parameters, and they all contain the true parameter values (red dots), indicating proper uncertainty estimation.

Momentum–Dependent Unfolding

One of the most powerful features of Moment Unfolding is its ability to extract moments as a function of another observable, such as jet transverse momentum. When implemented with momentum–dependent parameters in the Boltzmann weight function,

$$g(z; p_T) = \frac{1}{P(p_T)} \exp\left(-\sum_{a=1}^{n} \beta_a(p_T) z^a \right), \tag{5.27}$$

the generator unfolds moments conditional on the jet p_T.

This ability to unfold moments as functions of p_T is a necessary feature for any modern unfolding method. Many QCD phenomena exhibit strong scale dependence, including parton shower evolution, hadronization effects, and detector response. Understanding the

momentum–dependence is necessary to account for these variations. Jets at different p_T values populate different regions of the substructure phase space. A momentum–dependent reweighting is therefore important for better coverage of the phase space. As we will see, the momentum–dependent approach effectively leverages the statistical power of the entire dataset while allowing for local adaptations in different p_T regions, resulting in more precise measurements overall.

The implementation of momentum–dependent parameters, $\beta_a(p_T)$ introduces additional complexity that must be carefully managed. We found that linear parametrization of $\beta_a(p_T)$ provides a good balance between flexibility and stability for the jet substructure observables studied.[1] More complex parametrizations might be warranted for observables with a strongly non–linear p_T dependence. However, increasing the number of parameters in the momentum–dependent approach must be approached with caution, since that precisely undoes the regularisation imposed by the Moment Unfolding method. If the training remains unstable despite appropriately constraining the functional form of $\beta_a(p_T)$ regularization in the neural network training may become necessary.

Inclusive Distributions

Figure **[fig –KD]** presents the inclusive distributions of jet mass, jet charge, jet width, and groomed momentum fraction after applying momentum–dependent Moment Unfolding. Compared to the non-momentum-dependent results shown in Figure **[fig –KD]**, these distributions demonstrate markedly better agreement with the truth. This improvement occurs because the momentum-dependent approach effectively implements a more refined correction, allowing the reweighting function to adapt to the changing detector response and physics differences across the p_T) spectrum. By parametrising β_a as a function of p_T, we account for the fact that both detector effects and physics modelling discrepancies can vary significantly with jet energy.

The inclusive distributions obtained through momentum–conditioned unfolding also reveal subtle physical features that might be obscured in the non–conditioned approach. For example, in the jet mass distribution, the shape of the low–mass region shows better agreement with truth when using momentum conditioning, correctly capturing the interplay between soft radiation and resolution effects that varies with jet p_T.

Differential analysis

When the momentum–dependent unfolding approach is used for a differential analysis of moments as a function of p_T, it provides a comprehensive understanding of scaling

[1] Moment Unfolding can be run as a binned method by parametrizing $\beta_a(p_T)$ as a piecewise constant function.

behaviours. This dual perspective offers particularly powerful constraints on theoretical models, as it simultaneously tests the overall distribution shapes and their evolution with energy scale. Figure **[fig:pt-dependent-moments –KD]** shows the first and second moments of each observable as a function of p_T.

The results demonstrate that Moment Unfolding successfully recovers the p_T–dependent moments of all the observables, capturing the variations in these quantities with jet p_T. This capability is particularly valuable for testing theoretical predictions of how jet properties scale with energy, a fundamental aspect of QCD phenomenology.

These momentum–dependent results allow us to analyze the scaling behaviour of each observable. The jet mass shows expected direct scaling with p_T, consistent with theoretical expectations from perturbative QCD **[cite –KD]**, the mean jet charge decreases slightly with increasing p_T, reflecting the increased role of gluon radiation at higher energies **[cite –KD]**, the jet width decreases with p_T, consistent with the collimation of high–energy jets due to the Lorentz boost **[cite –KD]**, and the groomed momentum fraction shows minimal p_T dependence, a feature expected from the scale–invariant nature of the splitting functions in QCD **[cite –KD]**.

These results not only validate the Moment Unfolding method but also demonstrate its utility for extracting physically meaningful insights about jet formation and evolution. These scaling behaviours, when applied to real collider data, serve as stringent tests of perturbative QCD and provide constraints on Monte Carlo tuning parameters.

5.5 Comparison with Alternative Methods

To assess the performance of Moment Unfolding relative to existing approaches, we compare our results with those obtained from two alternative methods,

1. **OmniFold**: A state–of–the–art unbinned unfolding method that reconstructs the full distribution **[cite –KD]**, and

2. **Iterative Bayesian Unfolding (IBU)**: A traditional binned unfolding method followed by moment calculation **[cite –KD]**.

For a comprehensive comparison, we also include a variation on IBU that applies bin–by–bin corrections to account for binning effects in the moment calculation. Figure **[fig:method-comparison –KD]** shows the moments of each observable as a function of jet p_T for all methods, comparing them to the true values. The lower panels show the ratio to truth, highlighting differences between methods.

This comparison highlights some of Moment Unfolding's most important properties. The Moment Unfolding algorithm achieves comparable or better precision than OmniFold

for most observables, despite, or perhaps due to, its more targeted approach. This demonstrates the advantage of directly targeting the moments without having to unfold the entire distribution. Both unbinned methods (Moment Unfolding and OmniFold) significantly outperform the binned approaches (IBU and IBU+Correction) in precision, highlighting the benefits of avoiding binning artifacts. The bin–by–bin correction improves the performance of IBU but still does not match the precision of the unbinned methods, indicating that the fundamental limitations of binning cannot be fully mitigated by post–hoc corrections.

However, Moment Unfolding requires significantly less computational resources than OmniFold due to its non–iterative nature. On average, Moment Unfolding can be run in about 1% of the time required for OmniFold, while achieving comparable or better precision. IBU is about 10^4 times faster than Moment Unfolding. This comparison highlights the practical advantages of Moment Unfolding in scenarios where computational resources are limited or where multiple unfolding tasks need to be performed, such as in systematic uncertainty studies. That said, in analyses where computation is at a high premium, and the binning artifacts introduced by IBU are acceptable, IBU continues to offer a low–tech, light unfolding procedure.

These results establish Moment Unfolding as a powerful tool for precision measurements of statistical moments in high-energy physics, offering advantages in both statistical precision and computational efficiency compared to existing methods.

5.6 Conclusion

Moment Unfolding represents a significant advancement in the statistical toolkit for particle physics measurements, providing a novel approach to directly extract moments of distributions without the intermediate step of reconstructing the full spectrum. This methodology addresses several challenges in traditional unfolding techniques while offering unique advantages for theoretical comparison and interpretation. An intrinsically unbinned method, Moment Unfolding eliminates binning artifacts and dimensionality limitations that affect traditional methods. By using a physically–motivated reweighting function based on the Boltzmann distribution, it provides a principled approach to the unfolding problem, with clear connections to the maximum entropy principle from statistical mechanics.

The adversarial training framework effectively addresses the two–level nature of the unfolding problem, allowing particle–level corrections to be optimized based on detector–level comparisons. By extending this basic architecture to unfold moments conditional on momentum, Moment Unfolding can be used for detailed studies of how moments scale with energy, providing powerful constraints on theoretical models.

As a non-iterative method that is designed specifically to unfold moments rather than full spectra, Moment Unfolding offers significant computational advantages over iterative

approaches, particularly for systematic uncertainty studies that require multiple unfolding runs. The application to jet substructure measurements in Sec. **??** demonstrates the practical utility of Moment Unfolding for real physics analyses. The method successfully recovers both inclusive and differential moments of diverse jet observables, providing enhanced precision compared to traditional binned approaches and comparable or better accuracy than full-distribution unbinned methods. These capabilities make Moment Unfolding particularly valuable for precision tests of QCD, where theoretical predictions are often most precise at the level of moments rather than full distributions.

Although Moment Unfolding excels at extracting moments, it does not guarantee optimal reconstruction of the full distribution, particularly when higher moments are significant. The choice of how many moments to unfold represents a regularization parameter that must be carefully selected based on the physics application and available statistics. The basic formulation discussed above is designed for moments of a single observable, though extensions to joint moments are possible.

Towards Unfolding Distributions

While Moment Unfolding offers significant advantages for moment–specific measurements, many analyses require the full unfolded distribution rather than just its moments. This raises the question: can the Boltzmann–inspired approach be extended to unfold entire distributions? In a sense, this should be relatively straightforward. If the degree of the polynomial in the exponent of the Boltzmann factor governs the number of moments unfolded, by simply replacing the polynomial with a Taylor series,

$$g(z) = \frac{1}{P} \exp\left(- \sum_{a=1}^{\infty} \beta_a z^a \right), \tag{5.28}$$

we could, in principle, constrain all moments simultaneously, thereby unfolding the entire distribution. However, since the degree of the polynomial was the regularizing parameter for the inverse problem, by setting it to infinity this approach removes the regularisation central to the algorithm. The resulting ill-posed inverse problem requires significant regularization.

These challenges motivate the development of the Reweighting Adversarial Network (RAN) method, which will be introduced in the next chapter. RAN builds upon the foundations laid by Moment Unfolding, extending the approach to full distribution unfolding while maintaining its unbinned, non–iterative, adversarial nature. By incorporating more flexible neural network parameterizations and Wasserstein metrics for distribution comparison, RAN addresses the limitations of Moment Unfolding while preserving its core strengths. This evolution from moment–specific to full–distribution unfolding represents a natural progression in developing a comprehensive framework for unbinned cross-section measurements in particle physics.

Chapter 6

RAN: Reweighting Adversarial Networks

6.1 The Need for Full Spectral Measurements

Motivation: Beyond Moments and Binned Spectra

Measurements in high energy physics traditionally report differential cross sections in a binned format, or even just a few summary statistics (moments) of a distribution. This approach has produced numerous important results in the past**[cite –KD]**, but it fundamentally limits the information available for theoretical interpretation. A small set of moments (e.g. mean, variance) offers only a coarse summary of a probability distribution and can hide critical features. In fact, infinitely many different distributions can share the same first few moments. Two observables with identical mean and variance may have drastically different tails or multi-modal structures that only a full distribution reveal. Thus, relying solely on low–order moments risks missing new physics signals or subtle QCD effects that manifest as shape differences rather than overall normalization changes.

Binning an observable into histograms is a more detailed approach than just quoting moments, but it too imposes significant limitations. Finite bin widths smear out fine structures and impose an arbitrary discretization on inherently continuous spectra. Moreover, once data are binned, it becomes impossible to reconstruct or analyze the distribution for any arbitrary transformation of that observable without returning to the original unbinned data. For example, if a cross section is unfolded in bins of an angle θ, one cannot later obtain the distribution in $\cos\theta$ or $\ln\theta$ except by repeating the unfolding from scratch. In contrast, an unbinned (or "full spectral") measurement preserves maximum information, allowing *a posteriori* reprocessing such as deriving moments, re-binning in different intervals, or studying functions of the measured observable. This flexibility is especially valuable when comparing with various theoretical models, each of which may suggest different variables or summary statistics to highlight.

Another key motivation for capturing the *full differential spectrum* is that many theoretical predictions in quantum chromodynamics (QCD) and beyond are sensitive to the detailed shape of distributions. New physics might appear as excess events in the tails of kinematic distributions, or subtle distortions across a spectrum rather than an overall rate change. Precision QCD studies often rely on the *scaling patterns* of entire distributions with energy or other parameters, not just their averages. By measuring the full spectrum, experimental results can be directly fed into global fits or theory calculations that integrate over the entire phase space. In summary, to fully exploit the data and enable the broadest possible comparisons to theory, it is imperative to go beyond a few moments or fixed–bin histograms and aim to unfold the continuous differential cross section itself.

Let us now discuss in more detail some of the central limitations of binned approaches and approaches focused on unfolding a few moments summarised above in greater detail.

- **Information Loss:** Low–order moments (such as mean or variance) condense an entire distribution into one or two numbers. This sacrifices information about higher–order fluctuations and tail behavior. Many distinct distributions can reproduce the same set of moments, so important differences (e.g. a long tail vs. a sharp cutoff) remain hidden if only moments are reported. Even in cases where theory predicts moments more readily than spectra (as in some QCD calculations of energy scaling of moments**[cite –KD]**), relying exclusively on moments means discarding data that could otherwise constrain models. Binned histograms similarly integrate the underlying distribution over each bin, blurring details smaller than the bin width. Fine binning could, in principle, mitigate this, but the detector resolution and statistical limits cause too–fine bins to lead to instability due to bin migration effects **[cite –KD]** and large uncertainties.

- **Binning Artifacts and Biases:** Any choice of bin boundaries is inherently arbitrary. Two experiments measuring the same observable might choose different binning schemes, complicating direct comparisons. Small shifts in bin edges can redistribute events and lead to apparent differences that are purely due to binning choices rather than physics. Furthermore, when binning multivariate data (or examining an observable's moments in bins of another quantity), the necessary discretization in each dimension can introduce artificial discontinuities. These artifacts hinder comparing unfolded results with continuous theoretical predictions or with results from other experiments that used different bin definitions**[cite –KD]**. Additionally, binning can bias downstream analysis; for instance, extracting moments from a binned distribution requires assuming a shape within each bin (often a constant or linear interpolation). If the true distribution varies non–linearly inside the bin, the extracted moment is biased by the bin size and shape assumption such that even the sign of the error

Table 6.1: Comparison of measurement approaches. Reporting only low–order moments loses most distribution information. Binned differential cross sections retain shape information but suffer from discretization and limited dimensionality. Full spectral measurements preserve the complete distribution, enabling maximal reusability and detailed theory comparisons, but require regularizing a much more ill–posed training.

Aspect	**Moments only**	**Binned spectrum**	**Full spectrum**
Information	Low	Moderate	High
Reuse	High	Limited	High
Dimensions	Many	1 to 2	Many
Stability	Easy	Moderate	Hard

cannot be known *a priori*.**[cite Moment Unfolding –KD]** An unbinned measurement eliminates this intermediate step and potential bias.

- **Limited Dimensionality:** Perhaps most importantly, binned unfolding severely constrains the number of observables one can unfold simultaneously. Each additional dimension (feature) requires exponentially more bins to maintain a given resolution. In practice, traditional unfolding is often performed in one dimension at a time (or at most two) because higher–dimensional histograms would be too sparsely populated. This precludes measuring cross sections differential in many variables at once. It also means that if one is interested in an observable O that is a complicated function of several kinematic quantities, one either must unfold those quantities jointly with fine binning (which is infeasible) or settle for unfolding a projection of O in one dimension (which loses information). In contrast, unbinned approaches can handle high-dimensional data naturally, since they do not require constructing an $d-$dimensional grid of bins. In principle, unfolding the full phase space (i.e. a fully differential cross section in all relevant kinematic variables) is only realistic with an unbinned strategy.

Table **??** summarizes these differences between unbinned moment–based unfolding, binned density unfolding, and unbinned full–spectrum unfolding methods. By construction, a full spectral measurement retains maximal information and avoids discretization issues, at the cost of a more challenging unfolding procedure. These challenges, as discussed next, were a major deterrent historically, explaining why experiments long favoured simplified (binned or moment) results despite their limitations.

Use Cases

Some of the most compelling motivations for unbinned unfolding unfolding of probability density functions come from specific classes of observables and analyses in high–energy physics that demand detailed distributions rather than summary measures. A few examples are

- **Jet substructure observables:** Modern studies of jet physics often examine intricate internal properties of jets, for example, the distribution of jet mass, angularity, $N-$subjettiness ratios like τ_{21}, energy correlation functions, and so on. These observables have rich distributions shaped by QCD radiation and hadronization inside the jet. For instance, the jet mass spectrum in hadronic collisions exhibits a steeply falling shape with resonance peaks or grooming–induced features in certain regions; capturing these details is essential for validating parton shower models and grooming techniques.

  If only the average jet mass or a few quantiles were reported, one would miss the full story of how often jets are heavy vs. light, or how substructure techniques sculpt the distribution. Similarly, distributions of τ_{21} (a ratio used to tag two–prong substructure) contain information about the fraction of jets with two subjets vs. one, which is crucial for signal (e.g. boosted W) vs background (QCD jet) discrimination. Capturing the entire τ_{21} spectrum allows experimentalists and theorists to identify where in the distribution their models agree or fail, rather than just comparing an efficiency at a fixed cut. Unfolding these jet substructure distributions in an unbinned way provides a high–resolution view of QCD dynamics and is increasingly necessary as theory tools (like analytic resummation or first–principles simulation) improve to the point of predicting differential shapes**[cite –KD]**. In fact, recent measurements have demonstrated the power of full–phase–space unfolding for jets, using multivariate ML techniques to correct detector effects and obtain particle–level jet observable spectra without binning**[cite –KD]**. These cases underscore that jet physics benefits enormously from preserving the full shape information.

- **Charge and flavour sensitive observables:** Some observables aim to distinguish particle charge or flavour inside complex final states, and their distributions can be especially telling. An example is the jet charge distribution, the electric charge of a jet computed from the momentum–weighted charges of its constituent particles. This quantity is used to tag whether a jet originated from a quark of a given electric charge (as opposed to a gluon, which has none). The jet charge is a continuous-valued observable that can take positive and negative values, and experiments measure its probability distribution for jets of various momenta. The full shape of the jet

charge distribution contains information about the underlying quark/gluon mixture and fragmentation processes; for instance, a broader distribution indicates a mix of high–charge (quark–origin) and zero–charge (gluon–origin) jets, whereas a narrow distribution peaked near zero suggests predominantly gluon jets. Only by unfolding the entire jet charge spectrum (including its dependence on jet p_T) can one provide data precise enough to tune fragmentation models and compare to theoretical calculations of charge transport in jets**[cite –KD]**.

Another example is the distribution of identified particle multiplicities (e.g. number of charged hadrons in an event or in a jet). The multiplicity distribution is discrete but often measured as a histogram. It is a particularly challenging observable to unfold because it is strongly sensitive to soft QCD and hadronization effects.**[cite –KD]** Two Monte Carlo generators might predict the same average multiplicity but differ in the width or tail of the multiplicity distribution, which affects extreme cases (like very high multiplicity events). Such observables are often *infrared–unsafe* theoretically (because soft particle emission has no cutoff, perturbative predictions diverge for the distribution shape), meaning theory must rely on phenomenological models. The only way to constrain and improve those models is for experiments to provide the unfolded full distributions of these IR–unsafe observables at particle level. In summary, any analysis where internal structure, charge assignments, or other detailed event properties matter will benefit from (or even require) full spectral unfolding rather than a few summary numbers.

- **Multi–differential and high–dimensional measurements:** The ultimate form of "full" spectral measurement is unfolding in multiple kinematic dimensions simultaneously, effectively measuring a multi–differential cross section over a high–dimensional phase space. While this is extremely challenging, certain physics questions demand correlating several observables. For example, consider measuring an observable O as a function of another variable Q (say, the distribution of jet substructure variable O in bins of jet p_T or event energy Q). With traditional methods, one would perform a two–dimensional unfolding (bins in O vs bins in Q) to then extract moments or other features as a function of Q.**[cite Moment Unfolding –KD]** Binning in two (or more) dimensions quickly suffers from sparse data in many bins and complicated systematic uncertainties. An unbinned approach, by contrast, could in principle unfold the joint (O, Q) distribution without requiring an explicit grid, allowing arbitrary slicing and analysis after the fact. This is especially relevant in the era of high–luminosity colliders, where huge datasets invite more differential measurements. Rather than publish dozens of separate one–dimensional spectra (each in a single kinematic region), one could publish a multi–dimensional unfolded distribution that coherently captures all correlations. Such a result would be far more powerful for global interpretations,

albeit significantly more complex to obtain. Unbinned density unfolding techniques are a step toward this ambitious goal, enabling higher dimensional unfolding than previously feasible with manageable uncertainties.

Challenges in Unfolding Full Distributions

The push for unbinned, high–resolution unfolded spectra comes with substantial challenges. Unfolding a full distribution (especially in multiple dimensions) is a markedly more ill–posed problem than unfolding a small set of summary statistics or coarse bins. These challenges are statistical, physical, and computational.

At its core, unfolding requires inverting the detector response—a many–to–many mapping where a given particle–level distribution can produce a range of detector–level outcomes due to resolution and inefficiencies. This inversion is ill–posed, that is to say, infinitely many particle–level spectra are, in principle, consistent (within uncertainties) with a given set of detector–level data, especially if the data are treated in fine detail. Small fluctuations or statistical noise in the detector–level histogram can cause huge oscillations in the naive unfolded solution if one attempts a direct inversion of the response matrix. In classical unfolding methods, this is well known. Directly inverting the response matrix $\mathbf{R}$ leads to amplified noise and unstable solutions.**[cite –KD]** The problem is exacerbated when the "matrix" is essentially continuous (unbinned) since here one is trying to reconstruct a function rather than a finite vector, which has infinitely many degrees of freedom. Without any constraints or regularization, unfolding is mathematically underdetermined; one must introduce additional information to obtain a physical solution. This additional information can be statistical regularization (penalizing roughness in the solution), or a strong prior (initial guess of the spectrum) to guide the result. Either way, the unfolded distribution will depend to some degree on these assumptions, which is a point of concern.

All unfolding methods require some initial model or ansatz for the true distribution, explicitly or implicitly. Traditional regularized unfolding (e.g. Bayesian iterative methods or Tikhonov regularization) starts from a prior distribution and updates it in light of the data. If the prior is significantly wrong in a region where data have low sensitivity, the unfolded result may inherit that wrong shape (bias) because there isn't enough information in the data to correct it. For binned methods with strong regularization, the unfolded spectrum can end up looking very much like the prior except in regions where the data clearly indicate otherwise. When unfolding a full spectrum, prior dependence can be even trickier: with many bins or continuous degrees of freedom, there are more opportunities for the prior assumptions to creep in unless the method explicitly works to mitigate this. Modern ML based unbinned unfolding techniques like iterative reweighting (used by OmniFold) attempt to reduce prior bias by gradually adjusting the prior to fit the data**[cite –KD]**, but they still require that the initial simulation populate all regions of phase space that data

might cover.**[cite –KD]** In other words, if the true distribution has support outside the domain of the starting simulation, no unfolding can recover that—a condition that applies to all known methods. This need for sufficient overlap in the support of the prior and true distributions is a fundamental mathematical limitation: full spectral unfolding demands that our simulation model is flexible and broad enough to encompass the truth, at least roughly, or else certain features will be missed entirely.

Unfolding more finely (or in more dimensions) inherently means extracting more parameters (or effectively, more bins worth of information) from the same finite dataset. This trade–off means individual elements of the unfolded spectrum will have larger statistical uncertainties than if the data were aggregated into a few bins or moments. For example, unfolding 100 bins will yield each bin count with larger uncertainty than if one had combined them into 10 bins. If not carefully handled, an unbinned unfolding will overfit statistical fluctuations in the observed data (referred to as *sculpting* in the machine learning literature**[cite –KD]**.) Effective regularization is essential to prevent noise amplification. Additionally, the detector response kernel must be well modelled. Any mismodeling (systematic error) can imprint itself on the unfolded result in complex and unpredictable ways. When only a few numbers are extracted, one can sometimes correct for known detector biases by simple scale factors; but when unfolding an entire distribution, any mismodeling of the response shape can distort the unfolded spectrum non–uniformly. This places high demands on detector simulation fidelity and on methods to incorporate systematic uncertainties (e.g. variations of the response model) into the unfolding procedure. Fully Bayesian approaches or profile likelihood methods can propagate uncertainties to the unfolded spectrum, but doing so in high dimensions is computationally intensive. In short, the richer the information we unfold, the more careful we must be to quantify the reliability of each feature of the spectrum.

Traditional unfolding algorithms (like solving $\mathbf{R}^{-1}\boldsymbol{\mu}$ for binned histograms) scale poorly as the number of bins grows. These traditional algorithms like iterative Bayesian unfolding (IBU, or D'Agostini method) are relatively fast for tens of bins but become slow for hundreds of bins, and practically unusable for thousands of bins or continuous data, as each iteration must refine a fine–grained distribution. Unbinned algorithms, on the other hand, typically rely on machine learning or Monte Carlo sampling and are computationally expensive: they may involve training complex models on large datasets or performing high–dimensional optimizations. For instance, iterative reweighting methods train a classifier multiple times (each iteration is a full supervised learning task on the dataset)**[cite –KD]**, and generative approaches might require training a high–capacity generative model flow on millions of events. These computations require substantial computing resources (CPU/GPU), careful hyperparameter tuning, and sometimes suffer from convergence issues. Ensuring that an unbinned unfolding converges to a stable solution without excessive computation is a non–trivial challenge. In summary, the practical feasibility of unfolding full spectra depends

on advances in algorithms and computing, precisely the advances that modern machine learning–based methods aim to provide.

Despite these challenges, the drive to measure full spectra is strong because of the scientific payoff. The next subsection discusses how recent machine learning methods, including the RAN architecture, tackle these issues.

Unbinned Unfolding Approaches

Recent years have seen rapid development of machine learning techniques for unfolding, which seek to overcome the limitations of traditional methods and make full spectral measurements possible in practice. Broadly, these approaches fall into two categories, those based on *reweighting* an existing simulated sample to better agree with data, and those based on *generating* new events (typically with generative models) to reproduce the data distribution. Both categories strive to avoid fixed histogram binning and instead work with unbinned data, using the power of ML to handle high–dimensional inputs and complex detector responses.

One of the pioneering ML-based methods is **OmniFold[cite –KD]**, which introduced an unbinned, multivariate unfolding technique using iterative reweighting. OmniFold uses classifier neural networks to reweight a Monte Carlo sample. Essentially, it trains a classifier to distinguish between data and simulation, and then uses the classifier output to assign weights to simulation events such that the weighted simulation better matches the data. This procedure is done in stages (iterations) and at both detector and particle level in alternation.**[cite –KD]** After several iterations, the method produces a set of weights for generation events that yield an unfolded distribution. OmniFold demonstrated, for the first time, that one can simultaneously unfold many observables (even the entire event record) without binning, given a sufficiently flexible classifier and a robust iterative scheme.**[–KD]** It has been successfully applied in multiple experimental analyses,**[–KD]**. The key advantages of OmniFold are that it naturally accounts for high–dimensional correlations (since the classifier can use any or all features) and it actively mitigates prior dependence by iterative refinement, effectively performing a form of expectation–maximization to find a self consistent unfolded result that does not overly rely on the initial generation distribution. However, a noted downside is its computational cost: each iteration requires training two classifiers to convergence, and in realistic cases one might need $5 - 10$ iterations, amounting to training a large number of networks. This iterative nature can also complicate uncertainty evaluation and hyperparameter tuning (e.g. one must subjectively choose when to stop iterating to avoid instability). Still, OmniFold set the stage for practical unbinned unfolding and proved the feasibility of full spectral measurements on real collider data.

In parallel, other methods have explored using explicit generative models to unfold distributions. For example, VAEs have been employed to learn a mapping from random

noise to particle–level distributions such that, when those events are passed through a detector simulation, the output matches the observed data distribution**[cite –KD]**. Similarly, normalizing flows and other invertible neural networks have been used to model the probability density function of true observables and deform it until its convolution with the detector response matches data**[cite –KD]**. These approaches are extremely powerful in principle: a sufficiently flexible generator could capture the full true distribution without needing a starting simulation. In practice, however, training such generative models is challenging Pure generative unfolding would require vast amounts of data to constrain the high–dimensional space and can suffer from mode dropping (the generator might fail to produce some less common features of the distribution if not properly incentivized). Additionally, learning a full generative model from scratch means the method must learn *both* the underlying physics distribution *and* compensate for detector effects at the same time. This is a high–dimensional optimization that can be unstable or require careful conditioning. Some recent works have attempted to combine generative modelling with explicit usage of the known detector response to guide the training (for instance, using differentiable detectors or gradient-based deconvolution**[cite –KD]**), but these are still at the experimental stage.

Between pure reweighting (which uses an existing simulation as a starting point) and pure generation (which starts from random noise) lies an attractive compromise: use machine learning to *reweight or recalibrate* an initial simulation in a single pass, by directly comparing its predictions to data. The **Reweighting Adversarial Networks (RAN)** method falls into this category. Other examples include optimal transport inspired methods that train a single neural network to learn a transport map by minimizing some distance between weighted simulation and data distributions**[cite –KD]**. Such approaches leverage the fact that modern simulations are a good but imperfect approximation of reality; rather than throw them away and learn from scratch, it may be easier to learn small corrections (reweightings) to the simulation. In this sense, RAN and similar methods treat unfolding as a density ratio estimation problem: find a function $g(z)$ such that the weighted MC distribution $g(z)\,q(z)$ matches the true distribution $p(z)$. If Z denotes particle–level kinematics, this $g(z)$ effectively encapsulates how the Monte Carlo needs to be modified to agree with nature. The challenge is that we cannot observe $p(z)$ directly—we only have its smeared version at detector level. Therefore, these algorithms set up a two-level training objective to adjust $g(z)$ based on how well the weighted events, after going through the detector simulation, agree with the detector–level data. This two-level problem is naturally addressed with adversarial setups (a simulator or reweight function competing against a discriminator) or with optimal transport objectives that connect particle and detector level distributions.

In summary, the landscape of modern unfolding methods includes iterative classifiers (OmniFold) **[cite –KD]**, single–shot adversarial reweighting (e.g. RAN), and generative approaches (cINNs, VAEs)**[cite –KD]**, each with their own strengths. Table **??** provides a high-level comparison. All these methods aim to enable unbinned, high-dimensional

Table 6.2: Comparison of unfolding methods. "Unbinned" indicates the method can use continuous data without fixed histograms. "Iterative" indicates whether multiple training iterations are required (not counting training epochs, which all neural methods require internally). "Perturbative" indicates that the method reweights a MC sample (as opposed to generating events from scratch). RAN (this work) occupies a middle ground, using a prior simulation like OmniFold but aiming to avoid costly iterations by employing a stable adversarial training objective.

Method	**Unbinned**	**Iterative**	**Uses sim.**
Traditional (IBU, TUnfold)	No	Sometimes	Yes
Discriminative (OmniFold)	Yes	Yes	Yes
Generative (cINNs/VAEs)	Yes	No	No (from noise)
Adversarial reweight (RAN)	Yes	No	Yes

unfolding; they chiefly differ in how they incorporate prior knowledge, whether they iterate, and how computationally intensive they are. Notably, most current ML–based methods (except full generative ones) still require that the starting simulation has support in the regions of interest (the issue of support overlap). They also share common challenges of training stability and overfitting prevention, which are addressed through techniques like regularization or specific loss functions (for instance, OmniFold uses early stopping of iterations, while adversarial methods use regularized discriminators).

Addressing Key Challenges

The Reweighting Adversarial Networks (RAN) approach described in this chapter is designed to tackle the aforementioned challenges head–on, enabling full spectral unfolding with improved stability and efficiency. Here I outline how the methodology of RAN addresses the needs and difficulties detailed above (a complete description of the method follows in later sections, so the focus here is on concepts rather than implementation details).

RAN explicitly targets the entire distribution rather than a fixed set of moments. In fact, it can be viewed as an extension of the Moment Unfolding method**[cite –KD]** described in Chapter **??** to infinitely many moments. By using a flexible neural network to parametrize the weighting function, RAN does not impose a rigid functional form limited to a few moment constraints. This means it has the capacity to adjust the simulation such that not only the first few moments, but *all* features of the distribution (in principle, every differential element) are brought into agreement with the observed data. Framing the problem this way ensures that when RAN converges, the result is a faithful unfolded spectrum, from which

moments or any other summary statistic can be derived. Importantly, because it handles the full spectrum at once, RAN avoids the bias of selecting certain observables upfront—it lets the data inform the entire shape. Hence RAN is aligned with the basic motivation of full spectral measurements: nothing gets thrown away or averaged out prematurely.

By construction, RAN is an unbinned method. It operates on individual events, using distance measures defined on samples rather than comparing binned counts. The adversarial training framework means that a discriminator network looks at distributions of features (at detector level) and tries to tell apart weighted simulation from real data. If it finds any discrepancy, no matter how localized, the reweighting network is encouraged to adjust weights to eliminate that discrepancy. This is effectively a fine–grained comparison across the full phase space, without ever projecting data into predetermined bins. As a result, RAN does not suffer from the bin alignment issues or interpolation biases that plague binned unfolding. Different experiments using RAN on the same observable should, in principle, get comparable results without worrying that one used 50% purity bins and another used 70.7% purity bins, since neither uses bins at all. The output of RAN can be presented as a smooth distribution or as a weighted event sample at particle level, which can then binned or analysed downstream as needed. This flexibility maximizes the utility of the measurement.

One of the core design goals of RAN is to eliminate the need for multiple iterative reweighting cycles (as in OmniFold). Instead of training sequential classifiers for each iteration, RAN uses a single coupled training procedure where the particle–level reweighting function and the detector–level discriminator are learned together (analogous to a generator and critic in a Wasserstein GAN). This yields a one–shot solution for the weights after convergence. In practice, this means RAN trains one neural network (the reweighter) with feedback from another (the discriminator) in one integrated run. The computational cost is roughly equivalent to training a single GAN rather than a dozen separate classifiers. As demonstrated in Sec. **[ref RAN results –KD]**, this can lead to a significant reduction in runtime and resource usage for large–scale unfolding tasks**[cite –KD]**. Furthermore, by using an optimal transport–based loss (inspired by the Wasserstein GAN) and techniques like spectral normalization in the discriminator, RAN achieves stable training dynamics.**[cite –KD]** The Wasserstein metric provides a smooth loss function that correlates well with distribution difference, avoiding the chaotic or oscillatory behaviour that vanilla GANs can exhibit, described in Sec **??**. Spectral normalization bounds the discriminator's gradient, effectively regularizing the learning and preventing mode–collapse or instability.**[cite –KD]** These choices were crucial technical innovations needed to extend the moment–matching method to a full-spectrum matching method.**[cite –KD]** The end result is that RAN converges reliably to a solution where the weighted generation agrees with truth, all in a single training loop. This addresses the computational challenge by trading an iterative series of simpler trainings for one more complex adversarial training, which has been shown to be tractable and efficient.

Like any reweighting–based approach, RAN does require a MCMC samples and will fail if the true distribution lies entirely the support of the generation. However, RAN upon convergence, provably does not rely on the exact prior shape within the supported region. The use of an adversarial loss means RAN is effectively solving a constrained optimization to find the weight function that makes the simulation statistically indistinguishable from data by reweighting only at particle level. RAN's connection to the Boltzmann entropy–inspired moment unfolding provides a theoretical understanding of how it approaches the "maximum entropy" solution consistent with the constraints (the data)**[cite –KD]**. This is desirable because the maximum entropy solution is the least biased one given the information at hand. In essence, RAN inherits the moment method's stability from having a limited functional form, and compensates for the larger freedom of full spectra by adding proper regularization in training, thereby taming the ill–posedness.

An important practical aspect is that RAN yields physically reasonable unfolded distributions, such as non–negative weights and normalized total cross sections. By parametrizing the weight function as $g(z) = \frac{1}{P}\exp(-NN(z))$ (one convenient choice), we ensure $g(z) \geq 0$ for all events, so the unfolded cross section is positive–definite. The training objective can be set up to include a normalization constraint or simply allow the weights to float the total normalization to match the data yield. This avoids unphysical outcomes like negative weight factors or mismatched totals that can occur in some matrix inversion methods without extra constraints. Moreover, because RAN operates at the level of reweighting actual events, the resulting weighted event sample can be validated easily. One can always forward–fold the weighted sample through the detector simulation to check that it indeed reproduces all features of the observed data (closure test). This built–in consistency check is a powerful advantage of the reweighting paradigm in general. It is straightforward to verify that the full spectral measurement is successful, by confirming that no residual differences remain between data and weighted simulation across all distributions of interest.

It's worth noting that the RAN framework could potentially be extended to perform background subtraction simultaneously with unfolding by modifying the weight parameterization to allow negative weights. This might be particularly valuable for heavy ion physics, where background contamination presents significant challenges.[1] In conclusion, RAN is conceived to meet the need for unfolding multivariate probability densities by addressing

[1]While this extension has not been implemented or tested in the current work, allowing negative weights could potentially enable joint background subtraction and unfolding. This approach might be especially valuable for heavy ion collisions where background separation is notoriously challenging due to the high multiplicity environment and complex underlying events **[arXiv:2402.10945 –KD]**. Several studies have explored related joint unfolding and background estimation approaches, including work on machine learning based background subtraction methods that can reduce fluctuations below the statistical limit **[Machine Learning based jet momentum reconstruction in heavy-ion collisions, arXiv:2305.16826 , arXiv:2402.10945 –KD]**. The difficult interplay between background subtraction and unfolding has been examined in detail by Apolinário et al.**[An analysis of the influence of background subtraction and quenching on jet ob-**

the key challenges that have historically prevented such measurements. It leverages modern ML (specifically adversarial training and neural network flexibility) to unfold complete distributions without binning, while incorporating solutions to the statistical and computational pitfalls (regularization via WGAN, one–pass training, etc.). The following sections will describe the RAN methodology in detail and demonstrate its performance. Here, we have established *why* a method like RAN is needed—because it enables us to unfold the maximal information from experimental data, the full differential cross section, in a way that is both theoretically and practically sound, thus empowering deeper insights into particle physics phenomena.

6.2 From Moments to Complete Differential Cross Section Spectra

In Sec. **??** we discussed the motivation for unfolding complete differential cross sections—retrieving the full distribution of an observable at particle–level from distorted detector–level data. We now build a rigorous framework linking distribution moments to the full probability density, and review how leveraging moment constraints can facilitate unbinned unfolding. By treating moments as fundamental constraints (as opposed to bin–by–bin values), we establish a pathway from a limited set of summary statistics to a complete unfolded spectrum. This section first develops the theoretical foundation connecting moments and probability distributions (including moment–generating functions and maximum entropy arguments), then surveys classical and modern unfolding methods that rely on moment constraints (such as regularization and entropy–based techniques). This is followed by a discussion of the advantages and pitfalls of moment–based unfolding, notably issues of ill–posedness and non–uniqueness, and the regularization strategies to mitigate them. Finally, I explain how the Moment Unfolding method introduced in Chapter **??** naturally extends toward unfolding the entire distribution (all moments) via a GAN–like approach inspired by the Boltzmann distribution, setting the stage for a full differential cross section measurement.

---

servables in heavy-ion collisions –KD], who analyzed how different background subtraction methods affect jet observables in heavy ion collisions. Recent research has also improved background subtraction techniques specifically for jet substructure measurements**[Improved background subtraction and a fresh look at jet sub-structure in JEWEL –KD]**, demonstrating that proper handling of medium response is crucial for meaningful comparison with experimental data The theoretical framework of transport theory deconvolution with background contributions **[cite –KD]** could provide mathematical guidance for such extensions. Development of these capabilities would require careful validation and is left as future work.

Moments and the Full Probability Distribution

Statistical moments provide a powerful, coarse–grained description of a probability distribution. The nth moment (about zero) of a random variable Z is $\mathbb{E}[Z^n]$, and moments about the mean (central moments) characterize the shape (variance, skewness, kurtosis, etc.) of the distribution. In principle, an infinite sequence of moments can completely characterize a distribution. This is referred to as the moment representation of a distribution. If all moments $\mu_1, \mu_2, \ldots$ are known and certain technical conditions hold (e.g. the moment-generating function exists in a neighbourhood of 0), then there is a unique probability density consistent with those moments.**[DOI:10.11647/obp.0333.07 –KD][cite one more –KD]** The moment generating function (MGF), $M_Z(t) = \mathbb{E}[e^{tZ}]$, encapsulates the entire moment sequence via its Taylor expansion,

$$M_Z(t) \;=\; 1 + \sum_{n=1}^{\infty} \frac{t^n}{n!}\,\mathbb{E}[Z^n]\,, \tag{6.1}$$

and serves as a proxy for the full distribution. In fact, knowledge of $M_Z(t)$ (or the characteristic function) allows one to reconstruct the probability density $p_Z(z)$ by inverse transformation. In short, an infinite set of moment constraints is mathematically equivalent to knowing the complete distribution.

In practice, however, we can only estimate a finite number of moments from data with finite precision. A finite moment set underdetermines the distribution, because infinitely many distinct densities can share the same first n moments. This non-uniqueness under finite information is closely related to the ambiguity of solutions when attempting to invert detector effects: the data usually provide limited "moments" of the true distribution (for example, binned event counts are themselves integrals of the true spectrum over bin ranges). As n increases, the moment constraints become more informative and the space of consistent solutions shrinks, but noise and uncertainties also grow. In the limit of $n \to \infty$, the true distribution would be recovered, but this limit cannot be reached exactly with finite and noisy data.

The Maximum entropy principle and exponential families

Given a few known moments, a common approach to approximate the underlying distribution is to apply the principle of maximum entropy.**[ISBN 3-89429-543-0 –KD]** This principle dictates that, among all distributions satisfying the known moment constraints, we should prefer the one with the largest entropy (i.e. the least additional assumptions or information). Imposing constraints on expectations $\mathbb{E}[f_a(Z)] \;=\; c_a$ (for some set of functions $f_a(Z)$ defining the moments of interest) leads to a unique maximum-entropy

solution in the exponential family. In particular, one finds a probability density of the form

$$p^*(z) \;=\; \frac{1}{Z(\boldsymbol{\beta})}\,\exp\Big(-\sum_a \beta_a\, f_a(z)\Big)\,, \tag{6.2}$$

where β_a are Lagrange multipliers adjusted to enforce the desired $\mathbb{E}_{p^*}[f_a(Z)] = c_a$, and $Z(\boldsymbol{\beta}) = \int \exp(-\sum_a \beta_a f_a(z))\,\mathrm{d}z$ is the normalization factor (partition function). This is analogous to the Boltzmann distribution in statistical mechanics, which maximizes entropy given a fixed average energy. Indeed, if we choose $f_a(z) = z^a$ (the monomials), the above $p^*(z)$ is a Boltzmann–like ansatz with coefficients β_a related to the distribution's moments. As we will see, this exponential–family form is central to our moment–based unfolding method. It provides a flexible yet principled parameterization of the true distribution in terms of a finite set of parameters λ_a or equivalently a finite set of moments. Crucially, if the list of moment constraints is extended and refined, $p^*(z)$ can approximate the true distribution arbitrarily well (approaching the actual $p_Z(z)$ as the number of moments grows). Hence the full differential cross section can be reached in the limit of sufficiently many moment constraints.

Unfolding with Moment Constraints: Classical and Modern Approaches

Many unfolding methods, both classical and modern, can be interpreted as using moment constraints or related regularization assumptions to tackle the ill–posed inversion of detector effects. In a binned setting, each bin count can be seen as a moment (an integral of the continuous distribution against a top–hat basis function). Unfolding those bin counts with minimal noise amplification often requires additional constraints such as limiting the number of iterations. Although not usually thought of in this fashion, this can be equivalently reformulated as effectively limiting the space of possible moment values of these distributions convolved with a top–hat function.

Linear Regularized Unfolding (Tikhonov and SVD)

One class of unfolding methods formulates the problem as a linear system $mu_i = \sum_j R_{ij}\nu_j$, where mu_i are observed counts in detector bin i and ν_j are the true distribution values (e.g. cross section in true bin j). Solving for ν_j directly (e.g. matrix inversion or unregularized maximum likelihood) is notoriously unstable. Although we have discussed this in **??**, we are now equipped to reformulate the problem in the language of moments. High–frequency fluctuations in ν, equivalent to variations in high–order "moments", can fit statistical noise in μ. Tikhonov regularization addresses this by adding a penalty on undesirable solutions, usually favouring smoothness.**[A.N. Tikhonov, On the solution of improperly posed problems and the method of regularization, Sov. Math. 5 (1963) 1035. –KD]** For

example, one penalizes the squared second derivative of the unfolded spectrum or deviations from a prior guess. This effectively constrains the higher–order moments of the solution (suppressing oscillatory components that are poorly determined by data). The result is a bias toward smooth moment behaviour, trading a bit of bias for a dramatic reduction in variance.**[https://www.ippp.dur.ac.uk/Workshops/02/statistics/proceedings/cowan.pdf –KD][see references –KD]**

Similarly, the singular value decomposition (SVD) unfolding method truncates small singular values of the response matrix.**[Michael Schmelling, The method of reduced cross-entropy. A general approach to unfold probabil- ity distributions, Nucl. Instrum. Methods A340 (1994) 400. –KD]** This is equivalent to discarding combinations of moments that cannot be determined well (those along directions of the solution space corresponding to tiny eigenvalues would otherwise blow up with noise). By keeping only the dominant modes, essentially the lowest–frequency or largest–size moments, SVD unfolding ensures stability at the cost of not fully utilizing high–frequency information.

Both Tikhonov and SVD thereby regularize the moment space, either explicitly or implicitly limiting the effective number of moments that contribute to the unfolded solution.**[Cowan –KD]**

Iterative Bayesian Unfolding (D'Agostini)

The iterative method by D'Agostini **[cite –KD]** (known by various names; in HEP, Iterative Bayesian Unfolding) approaches unfolding as a successive moment matching procedure. It starts with an initial guess for the true distribution (the prior Monte Carlo prediction call 'generation', which corresponds to some initial moment estimates) and alternately updates the expectations to better agree with the observed data. At each iteration, the method reweights the Monte Carlo events by the ratio of data to simulation in each detector bin (effectively adjusting the candidate true distribution's moments to reduce the discrepancy in those bins). This procedure can be interpreted as ensuring that the predicted detector–level counts (moments of the true distribution under the response) match the observed counts, one step at a time. The algorithm converges to a solution that maximizes the likelihood (in the limit of many iterations), but in practice one stops after a finite number of iterations to avoid over-fitting statistical fluctuations.**[cite cowan –KD]** Stopping early or adding a prior is a form of regularization: it limits the effective degrees of freedom (moments) that are fitted, similar to Tikhonov or SVD albeit via a different mechanism. This iterative method has the advantage of intuitively incorporating a prior (the initial guess acts as a prior shape, providing stability if data are sparse) and is widely used due to its simplicity and ability to include prior knowledge.

Entropy-Based Methods

Another class of unfolding techniques explicitly incorporates entropy or information criteria to regularize the solution. The maximum entropy (MaxEnt) unfolding approach chooses the unfolded distribution that maximizes entropy subject to reproducing the observed detector data (usually via a likelihood term).**[cite cowan references –KD]** In practice this might mean maximizing $S = -\sum_j \mu_j \ln(\mu_j)$ minus a term for agreement with data. The solution tends to be as smooth and featureless as possible (high entropy) unless the data significantly demand a structure. Entropy regularization thus penalizes any extraneous moment fluctuations that are not required by the data, biasing toward a flat distribution in the absence of strong evidence for shape.**[cite C. Pruneau Data Analysis Techniques for Physical Scientists –KD][Marshall:2001ax –KD]** Schmelling's method of reduced cross-entropy**[cowan [8] –KD]** is a related technique, effectively combining likelihood maximization with an entropy prior. By treating the deviation from a prior distribution in terms of Kullback–Leibler divergence (relative entropy), one can incorporate prior knowledge while still preferring the least structured solution beyond that prior These methods make the connection to moment constraints explicit in that they treat the unfolding problem as one of satisfying certain expectation values (the data constraints) while maximizing uncertainty elsewhere. As discussed, this yields an exponential–family solution. In fact, the MaxEnt solution can be written in the form $p(z) \propto \exp(-\sum_a \beta_a f_a(z))$ where $f_a(z)$ are the functions defining the data constraints (for example, indicator functions for each detector bin to ensure those predicted counts match the observed).**[cowan –KD]** This is formally identical to the moment–constrained maximum entropy distribution discussed above; the only difference is the nature of the constraints (data–driven constraints rather than actual physical moments of the distribution).

Unbinned and Machine Learning–Based Methods

With advances in computation, unbinned unfolding techniques have emerged that avoid histogramming data altogether. These methods often use machine learning to compare distributions without bins. One prominent example is OmniFold,**[–KD]** an unbinned iterative unfolding approach based on modern machine learning classifiers. OmniFold uses a classifier (often a neural network) to distinguish weighted simulation from data; the classifier's output is used to reweight simulation events such that, after reweighting, the simulation is more similar to the data. This is done in an iterative fashion (multiple rounds, including both detector–level and particle–level weights) to converge to a set of per–event weights that yield an unfolded distribution matching the observed data.**[–KD]** While OmniFold does not explicitly constrain low–order moments or use an analytic parametrization, it implicitly matches all features of the distribution that the classifier

can discern—effectively attempting to equate the full set of moments by the end of the procedure. In each iteration, the classifier focuses on the current differences between data and weighted simulation, which are often in some "mode" of the distribution; iterating allows successively finer differences (higher–order moments or localized shape features) to be corrected. This is conceptually similar to performing an increasingly detailed moment matching, guided by the ML classifier as a flexible test statistic. Other approaches employ adversarial neural networks or optimal transport metrics to directly learn the unfolded distribution in one go, rather than iteratively.**[–KD]** These can be seen as the next step in unbinned unfolding: using a generator model for the true distribution and a discriminator to enforce that the generator's folded output looks like the data. Such setups, inspired by Generative Adversarial Networks (GANs),**[–KD]** take advantage of the same idea that ultimately all moments need to match for the generated distribution to be indistinguishable from the data. I will delve into one such adversarial approach, Reweighting Adversarial Networks (RAN), shortly. For now, suffice it to note that modern ML–based methods are moving toward treating unfolding as a fitting problem, using flexible function approximators and powerful statistical distances (e.g. Wasserstein distance**[–KD]**) to overcome binning and to handle large dimensional data. These methods still must address ill–posedness (e.g. through network architecture choices, training tricks, or implicit regularization like early stopping), but they offer a way to directly target the entire set of distributional degrees of freedom rather than pre–selecting a fixed set of basis functions.

Benefits and Challenges of Moment Unfolding

Focusing on moments as the quantities to unfold offers several clear benefits. First, moments are often directly related to physics predictions. Many theoretical models provide predictions for means, variances, or other moment–like observables (especially in QCD where sum rules and scaling laws involve moments of distributions). Unfolding at the level of moments thus yields results that can be compared to theory with minimal further processing.**[–KD]** Second, by compressing the data into a few numbers, moment–based unfolding can greatly reduce statistical fluctuations and noise sensitivity. A small set of global features (e.g. the first few moments) can usually be determined more precisely than an entire binned spectrum. The variance of estimators is lower since we effectively integrate over many events to get each moment. This was one motivation for the development of the Moment Unfolding method; measuring moments directly can be more precise and less sensitive to binning choices.**[–KD]** Third, moment unfolding can simplify high–dimensional problems. In multi–differential measurements (with many kinematic variables), fully binning the data becomes impractical due to sparsity ("curse of dimensionality"). But one might still meaningfully measure lower–dimensional moments, e.g. the average of some observable as a function of another variable, avoiding the need to populate multi–dimensional histograms.

In summary, moment-based approaches concentrate on the most salient features of the distribution, potentially yielding robust, computationally efficient unfolding results when only those features are of interest.

However, there are important limitations and challenges inherent to moment–based unfolding. By construction, focusing on a limited set of moments discards information contained in the distribution beyond those moments. Two very different underlying distributions can share the same few moments. Thus unfolding only those moments provides an incomplete picture. This non–uniqueness means that moment–based results must be interpreted carefully. They answer only the questions they explicitly ask. For instance, if only the first two moments (mean and variance) are unfolded, any differences residing in the non–Gaussian shape (skewness, tails, multimodal structure) will go undetected. Moment unfolding shifts the ill–posedness to the choice of moments. One must assume or hope that the chosen moment set is sufficient to capture the physically relevant differences. If an unexpected feature lies outside this span, it will be missed. This is connected to the concept of model dependence – choosing an insufficient set of moments imposes a bias (a kind of model) on the unfolding result.

A few different options exist to regularize the problem. For example, if unfolding a large number of moments, one might impose a smooth fal–off in the moment values (since very high–order moments tend to be increasingly sensitive to rare tails). In our adversarial Moment Unfolding approach, the number of moments n we choose to unfold acts as a regularization knob: a small n strongly regularizes (since it ignores any structure beyond the nth moment), while a larger n allows more detailed structure at the cost of higher variance and potential instability.**[–KD]** This is analogous to the bias—–variance trade–off in classic unfolding: using fewer moments yields a biased but low–variance estimate; using more approaches an unbiased full result but with higher variance (and risk of over–fitting data fluctuations). One limitation of any reweighting-based unfolding (moment-based or otherwise) is the requirement of support overlap. The method can only correct distributions within the support of the MC (prior) distribution. If the true distribution has events in regions where the simulation has zero or negligible events, no amount of reweighting or moment adjustment can cover that gap. One would need to generate new events (a different approach entirely) or rely on extrapolation. Thus, moment unfolding, like other reweighting methods (OmniFold, RAN, etc.), assumes that the generation's phase space is broad enough to contain the truth (or that any deficiencies are corrected by separate means). This is usually ensured by using a sufficiently generic simulation or augmenting it with additional samples if needed. It is worth noting that as we increase the number of moments or attempt to unfold the full distribution, the demand on simulation support becomes stricter: essentially all features of the data distribution must be present in the starting sample to be recovered by reweighting.

Extending Moment Unfolding

This Ansatz can be viewed as a minimal deformation of the simulation needed to match data. $\beta_a = 0$ for all a corresponds to $g(z) = 1$ (no reweighting, i.e. using the raw generation as is), and as we turn on β_a's, we gently adjust the weight of each MC event based on its z value. The exponential form guarantees that no arbitrary structure is introduced beyond that needed to fulfil the moment constraints i.e. it is the maximum entropy solution consistent with those constraints.**[–KD]** Another perspective is to see $g(z)$ as the Radon-–Nikodym derivative between the true distribution $p(z)$ and the generation $q(z)$, restricted to the family of functions parametrized by β_a.**[–KD]** If the true distribution $p(z)$ lies within this family (for the chosen T_a set), then there exists some $\boldsymbol{\beta}$true such that $p(z) = g(z; \boldsymbol{\beta}_{\text{true}}), q(z)$. Even if $p(z)$ is outside this family, we expect that with a sufficiently rich set of basis functions T_a, one can approximate $p(z)$ in the moment sense. In summary, Eq. **??** translates our physics question ("what are the true moments?") into a set of parameters β_a to be determined.

Once the optimal β_a parameters are found, the unfolded moments are obtained immediately by computing the weighted averages in the reweighted particle–level sample. The unfolded kth moment of Z is simply

$$\langle Z^k \rangle_{\text{unfolded}} = \frac{\int z^k \, g(z) \, q(z) \, \mathrm{d}z}{\int g(z) \, q(z) \, \mathrm{d}z} , \tag{6.3}$$

or in practice, the ratio of weighted sums over all generation events. By construction, these unfolded moments will agree with the true moments of $p(z)$ if the procedure succeeds in finding β_a such that the reweighted simulation matches the data atdetector-level. Incorporating the normalization factor $P(\boldsymbol{\beta})$ (the denominator above) is important. It ensures that $g(z)$ does not arbitrarily change the total number of events. In statistical mechanics language, $P(\boldsymbol{\beta})$ is the partition function ensuring probability conservation. In the unfolding context, it guarantees that the reweighted distribution properly normalizes to the total cross section (or total event count) expected. There is no explicit formula linking β_a to a simple goodness–of–fit measure at detector level, because the detector response $r(x|z)$ is typically a complex stochastic mapping. Instead, a machine learning approach evaluates and drives this optimization.

The Moment Unfolding method was demonstrated to yield accurate and statistically stable moment estimates in realistic scenarios. By avoiding any intermediate binning, it circumvents bin-size biases and fully utilizes the fine–grained shape information in the data. Moreover, it is remarkably efficient. Focusing on a handful of parameters β_a is much simpler than attempting to learn a full function (as a generative model would). In essence, it reduces the unfolding task to a parameter fitting problem under the hood, albeit a high–dimensional, simulation–informed, adversarially–solved one. This focus provided a built–in regularization (as we emphasized, choosing small n limits complexity). The results

in Section **[sec:results –KD]** showed that even with n as small as 2 one can capture key shape characteristics of distributions. However, ultimately one may wish to unfold the entire distribution without having to pre–select moments. This is where we transition to extending the Moment Unfolding concept to all moments, which leads us to the Reweighting Adversarial Networks (RAN) framework.

While unfolding a fixed set of moments is valuable, the ultimate goal in many analyses is to recover the complete differential cross section, effectively, to determine the true distribution $p(z)$ itself (within resolution limits). The Moment Unfolding framework provides a natural stepping stone to this goal. Conceptually, one can imagine increasing the number of moments n in the weight function Eq. **??** to capture finer and finer detail of the distribution. In the limit $n \to \infty$ (or including an appropriate functional basis for $T_a(z)$), the weight function $g(z)$ could represent an arbitrarily complex reweighting, capable of morphing the simulation into the true distribution. In practice, however, trying to include a very large number of moment parameters directly poses challenges. The normalization factor $P(\boldsymbol{\beta})$ (partition function) becomes increasingly complicated to compute or differentiate when $\boldsymbol{\beta}$ is high–dimensional, and the adversarial training may become unstable or inefficient when there are so many degrees of freedom to adjust. In the initial Moment Unfolding studies, keeping n small was important for stable training—it acted as a strong regularizer and simplified the learning problem.**[–KD]**

To extend the method toward full distributions, the Reweighting Adversarial Networks (RAN) approach was developed.**[–KD]** RAN can be seen as Moment Unfolding taken to the continuum limit, where instead of a few predetermined moments, all aspects of the distribution are learned. Practically, RAN forgoes the explicit parametrization of $g(z)$ with a fixed set of β_a multiplying known basis functions. Instead, it employs a more general function approximator (such as a neural network or a high-capacity ansatz) to represent $g(z)$, which can adjust weights flexibly for different regions of phase space. The adversarial setup remains, but now the generator's parameters are not directly interpretable as specific moments; they are a more granular description of the weight function. The discriminator in RAN is tasked with comparing the fully continuous shapes of $\tilde{q}(x)$ and $p(x)$, effectively enforcing an infinite number of constraints (since matching two distributions means matching an infinite set of moments or test statistics). In this way, RAN builds on the moment-based approach by removing the artificial limit on the number of moments: the method 'learns to unfold all the moments of a distribution'. **[–KD]**

One important innovation in RAN was the use of the Wasserstein GAN framework,**[–KD]** which provides a stable way to train the adversarial system even when the generator distribution and true distribution initially differ significantly. The Wasserstein distance (or Earth Mover's distance) offers a continuous and meaningful loss metric that correlates with distribution differences, which helped guide the training when $g(z)$ has high flexibility. This addresses a subtle issue. When extending to complete distributions, one must contend

with the fact that $g(z)$ estimates the partition function from batch data. In a naive GAN, if $g(z)$ significantly changes normalization, the discriminator could easily detect a total rate difference, and the generator would then simply rescale everything to match the event counts, potentially neglecting shape differences. By using an optimal transport loss and carefully constraining $g(z)$ to keep the total weight near unity, RAN avoids trivial solutions and focuses on shape adjustments. In effect, RAN had to incorporate the partition function normalization into the training. Although this was also true of Moment Unfolding, there is a quantitive difference. When n was small, a few β primarily tweak shape within a mostly normalized scheme, but misestimations of the partition function become very relevant when $g(z)$ becomes a free-form function.**[–KD]** Section **[sec:infiniteunfolding –KD]** is devoted to a detailed description of the RAN methodology; this section emphasizes the conceptual link: RAN is the natural extension of Moment Unfolding to the case where the number of moment–like features is unlimited.

By allowing a more expressive reweighting function, RAN is able to unfold the entire distribution in an unbinned, non–iterative manner. This stands in contrast to OmniFold, which, while also ultimately recovering the full distribution, does so via multiple iterative reweighting steps. RAN achieves a similar end point in one training loop by leveraging the powerful GAN paradigm. However, this power comes with technical challenges: the loss landscape is more complex and the risk of overfitting noise is higher when we essentially have as many “parameters" as there are events. Our implementation of RAN thus required additional care in training (e.g. gradient penalty terms for WGAN, spectral normalization, etc.) to ensure convergence to a physically sensible solution (one that generalizes and doesn’t chase statistical fluctuations). The success of RAN in toy examples and real data applications demonstrates that it is indeed possible to go from moments to complete spectra.,

In summary, the progression from Moment Unfolding to full distribution unfolding can be seen as a continuum. At one end, we have a highly regularized method focusing on a few moments (stable but incomplete); at the other end, we have methods like RAN that aim to extract the entire distribution (complete but needing stronger techniques to control fluctuations). Moment Unfolding paves the way to the more ambitious goal of unbinned differential cross section measurement via its extension to RAN. This connection between moments and spectra ensures that the insights and constraints from one domain (e.g. theoretical expectations for certain moments) can be seamlessly integrated into the process of obtaining the full spectrum. By casting the unfolding problem in the language of moment constraints and exponential families, one can gain both intuitive and quantitative control over the unfolding, whether one stops at a few moments or pushes onward to recover the complete distribution. This framework not only bridges classical and modern techniques but also provides a clear rationale for why unfolding even a handful of moments is a stepping stone to unfolding everything: the moments are the fingerprints of the distribution, and once

one learns to reliably recover those fingerprints (even individually), one is well–equipped to tackle the entire handprint that is the full differential cross section.

6.3 Methodology and Regularization

A variety of regularization strategies have been developed historically and in contemporary machine learning (ML) approaches to ensure stable, physically meaningful unfolding solutions. This section begins by reviewing these strategies, from classical techniques to modern ML–based methods, before detailing how the Reweighting Adversarial Networks (RAN) methodology is designed to address regularization. The focus will be on the theoretical and conceptual aspects of the RAN approach, rather than low–level implementation details.

Historical Approaches to Regularization

Classical unfolding methods introduce explicit regularization to tame the ill–posed nature of the problem. A prominent example is Tikhonov regularization, which adds a penalty on the curvature or norm of the solution to the fitting objective.**[–KD]** In a typical binned unfolding scenario with response matrix $\mathbf{R}$ relating true binned spectrum $\boldsymbol{\nu}$ to measured data $\boldsymbol{\mu}$, Tikhonov regularization finds the unfolded estimate by minimizing a modified χ^2:

$$\hat{\boldsymbol{\nu}} = \arg\min_{\boldsymbol{\nu}} \left[|\mathbf{R}\boldsymbol{\nu} - \boldsymbol{\mu}|^2 + \lambda \, |\mathbf{L}(\boldsymbol{\nu} - \boldsymbol{\nu}_0)|^2 \right], \tag{6.4}$$

where $\mathbf{L}$ is typically a discrete derivative (smoothness) operator and $\boldsymbol{\nu}_0$ is a prior estimate.**[–KD]** The regularization parameter λ controls the bias–variance trade-off**[–KD]**. $\lambda \to 0$ recovers an unbiased but high–variance solution, while large λ biases $\hat{\boldsymbol{\nu}}$. Techniques such as L–curve scans or cross–validation are used to choose λ optimally.**[–KD]** Tikhonov–style regularization (implemented for instance in TUnfold**[–KD]**) is widely used in precision measurements for its ability to suppress statistical noise when the true distribution is expected to be reasonably smooth**[cite –KD]**. Related linear regularization schemes include singular value decomposition (SVD) truncation, which diagonalizes the response matrix and discards or down-weights contributions along directions with small singular values (which are dominated by noise)**[cite –KD]**. This effectively regularizes the solution similarly to a Tikhonov cutoff, by eliminating high-frequency oscillatory components that are weakly constrained by data**[cite –KD]**. Another classical strategy is iterative Bayesian unfolding (often called D'Agostini or Richardson-–Lucy iterative method)**[cite –KD]**. In this approach, an initial guess (prior) for the truth distribution is repeatedly updated using Bayes' theorem and the observed data. The regularization is introduced by not iterating to full convergence;

stopping after a finite number of iterations (usually a few) serves as an implicit regularization that prevents overfitting to statistical fluctuations.**[–KD]** The number of iterations thus plays a role analogous to λ in controlling the smoothness of the outcome, with early stopping yielding a more biased (closer to the prior) but stabler result.**[–KD]** Finally, maximum entropy methods introduce an entropic prior, favouring solutions that maximize information entropy subject to data constraints**[cite –KD]**. By penalizing deviations from a featureless (flat or known prior) distribution, maximum-entropy unfolding produces a smooth unfolded spectrum unless the data strongly indicate otherwise. These "semi-classical" methods (e.g. SVD truncation, MaxEnt) were developed to reduce ad–hoc tuning. For instance, maximizing entropy automatically regularizes by pushing the solution toward the least–informative shape consistent with the measurements**[cite –KD]**. All of these classical techniques acknowledge the need for external constraints to obtain a stable inversion of the detector response.

In recent years, unbinned and high-dimensional unfolding methods based on machine learning have emerged, necessitating new ways to incorporate regularization.**[–KD]** Many ML-based approaches avoid explicit binning (thus eliminating bin–size artifacts) but must still combat the same amplification of statistical noise. Often, the regularization in ML approaches is *implicit*, coming from network architectures and training protocols.**[–KD]** For example, deep neural networks have a finite capacity and tend to learn simpler patterns first, a form of Occam's razor that can act as a prior. Choices of network depth, width, and activation functions impose smoothness biases on the learned mapping.**[–KD]** In convolutional networks, weight sharing and locality encode translational invariance, effectively constraining the space of possible unfoldings to those that respect certain symmetries.**[–KD]** Many ML unfolding methods also apply standard regularization techniques from predictive modelling, such as weight decay (L^2 penalties)**[cite –KD]**, dropout**[cite –KD]**, or batch normalization, to prevent overfitting. These generic measures help ensure the learned model does not simply reproduce statistical fluctuations in the training sample.

In adaptive or iterative ML unfolding algorithms, an explicit analogue of early stopping is used. The OmniFold method, for instance, performs successive reweighting iterations with neural classifiers.**[–KD]** By limiting the number of iterations (often on the order of $4-6$) and monitoring when the updates become consistent with statistical uncertainty, OmniFold effectively regularizes the solution.**[–KD]** If iterated to completion, an algorithm like OmniFold would overfit the data (analogous to iterative Bayesian unfolding); halting earlier mitigates this risk.**[–KD]** Neural network–based unfolding methods also permit incorporation of physics–informed constraints as a form of regularization. For example, one can modify the loss function to penalize unphysical deviations or enforce known conservation laws and symmetries**[cite –KD]**. Recent studies have integrated exact symmetry constraints into normalizing flow or invertible network architectures used for unfolding, so that the learned particle–level distribution automatically preserves quantities like momen-

tum or charge**[cite –KD]**. This reduces the solution space a priori to physically plausible ones, which is a powerful regularization aligning with domain knowledge.

Beyond reweighting approaches, generative models learn to generate events from scratch so that, when passed through a detector simulation, they reproduce the observed data.**[–KD]** Examples include normalizing flows and variational autoencoders trained to invert detector effects**[cite –KD]**. These models face even greater risk of instability due to their high flexibility. Regularization here can take the form of strong prior distributions (for instance, seeding a flow with a known prior and only allowing limited deviation)**[cite –KD]**, or carefully tuned network constraints. In practice, the success of ML generative unfolding has been limited by the need for large training samples and the difficulty of covering the full phase space without introducing spurious modes**[cite –KD]**. Indeed, most existing ML unfolding methods (OmniFold and variants**[cite –KD]**, flow-based methods**[cite –KD]**, etc.) still assume that the simulated data sample has sufficient overlap with the true distribution in all regions of interest.**[–KD]** If the simulation has zero or very little support in some region that the real data populate, no amount of reweighting or network fitting can compensate; this is the well-known curse of extrapolation in unfolding.

Regularization in RAN

The Reweighting Adversarial Network approach is expressly designed to address regularization challenges while performing unbinned unfolding in one stage. RAN can be viewed conceptually as an extension of the *Moment Unfolding* method**[cite –KD]**, which unfolded only a fixed set of distribution moments and thereby enjoyed a strong built–in regularization (by restricting to a few summary statistics). RAN lifts that restriction by attempting to unfold the entire distribution (the "infinite–moment" limit), requiring new regularization techniques to keep the problem well–defined.**[–KD]** Unlike OmniFold, which iteratively trains many separate classifiers, RAN solves for all weights in a single training round by jointly optimizing g and d. This provides a significant computational advantage, but it demands a very stable training procedure because the full flexibility of $g(z)$ is unleashed at once.

To ensure stable and meaningful solutions, RAN incorporates several regularization driven design choices in its methodology.

Wasserstein GAN

First and foremost, RAN employs a Wasserstein GAN (WGAN) framework for the adversarial training, in place of a standard (Jensen–Shannon divergence-based) GAN. In a vanilla GAN, the discriminator $d(x)$ is trained to output a binary classification (real vs fake) and the generator is trained to fool it, typically using a binary cross–entropy or similar loss. Such

losses correspond to minimizing a divergence like Kullback-–Leibler or Jensen—Shannon; if the real and model distributions do not overlap, the gradient of these losses vanishes, leading to unstable or mode–collapsed solutions. By contrast, RAN leverages the Earth Mover (Wasserstein-1) distance as the measure of difference between distributions.**[–KD]** The WGAN critic $d(x)$ does not output a class label but rather a real–valued score, which should be high for real data and low for simulated (reweighted) data.**[–KD]** The generator seeks to maximize this critic score on the simulated data, thereby minimizing the Wasserstein distance between the weighted simulation and the true data distribution. The objective function adopted in RAN's training can be written as

$$L_{\text{WGAN}}[g, d] = \mathbb{E}_{x\sim\text{Data}}\big[d(x)\big] - \mathbb{E}_{(z,x)\sim(\text{Gen., Sim.})}\big[g(z)\, d(x)\big] + \lambda\, \mathbb{E}\Big[\big(|\nabla_{\hat{x}} d(\hat{x})| - 1\big)^2\Big]\,, \tag{6.5}$$

where x denotes a detector-level event from the real data or simulation, and $(z, x) \sim$ (Gen., Sim.) indicates a MC event with truth features z and corresponding detector–level observation x (the MC is generative, so $z \to x$ via the detector model). The first two terms estimate the Wasserstein-1 distance between the true and reweighted simulated distributions. The last term is a *gradient penalty***[cite –KD]** (with strength λ) enforcing the requirement that $d(x)$ be $1-$Lipschitz continuous, as required for the Wasserstein distance theory.**[–KD]** This penalty drives $|\nabla d|$ toward unity on random points $\hat{x}$ interpolated between real and simulated samples, a technique introduced in Ref. **[cite –KD]** to stabilize WGAN training. The minimax game on the loss (**??**) with respect to g and d corresponds to the following adversarial logic:

1. d tries to minimize L_{WGAN} (making the data–sim discrepancy as large as possible by assigning higher scores to real data than to any weighted sim sample)

2. g tries to maximize it (reducing discrepancy by adjusting weights $g(z)$ to make the sim look as "real" as possible).

At equilibrium, the weighted simulation cannot be distinguished from data by any $1-$Lipschitz function, meaning the distributions are empirically matched in the optimal transport sense.

The choice of a WGAN loss is fundamentally motivated by regularization considerations. Because the Wasserstein distance evaluates the "distance" between distributions in terms of an optimal transport cost, it provides smooth, meaningful gradients even when the support of the model and data distributions do not overlap perfectly.**[–KD]** In other words, if the simulation is deficient in some region of phase space, a standard GAN discriminator would drive $g(z)$ to extreme values (or simply saturate, providing no gradient) in an attempt to cover the deficit, often leading to unstable oscillations in $g(z)$ (very large weights assigned to a few events). The WGAN critic, however, will assign a large positive $d(x)$ to real data

regions with no simulated equivalent, but the gradient of its linear output will encourage nearby simulated events to increase in weight in a more controlled fashion, effectively telling the generator *which direction* to move probability mass to reduce the discrepancy. This greatly mitigates mode collapse and high-variance weight solutions.**[–KD]** In our context of an "infinite–moment" unfolding (unconstrained by moment selection), using the WGAN framework is a key to avoiding the generator thrashing that occurred with a naive GAN approach. Empirically, we find that replacing the binary cross–entropy GAN loss with the WGAN loss stabilizes the training and yields a smoother weight distribution $g(z)$, especially in sparse regions of the detector feature space. An additional benefit is that RAN can handle cases with minimal detector-level overlap between simulation and data. Since the Wasserstein metric remains finite as long as the support of the weighted simulation can be transported to cover data (even if initially disjoint), RAN does not strictly require the simulation and data distributions to overlap bin–by–bin. Of course, overlap at particle level is still required. RAN cannot invent truth–level events outside the generation's support. Nonetheless, the ability to tolerate detector–level mismatches is a major advantage over methods that rely on probability ratios (which diverge when supports do not overlap). In essence, the use of WGAN in RAN provides a strong form of regularization–by–design: it selects a smoother notion of distribution difference that avoids infinite gradients and excessive weight variance.

Regularizing the critic: Lipschitz Constraints

The critic network $d(x)$ in RAN is subject to the $1-$Lipschitz constraint required by the Wasserstein theory. We enforce this through two complementary regularization techniques. The first, already noted, is the **gradient penalty** term in Eq. (**??**)**[cite –KD]**. Rather than clipping weights (the original WGAN approach**[cite –KD]**, which can impede learning by reducing capacity), we adopt the improved strategy of penalizing the norm of the critic's gradient on random interpolations to be close to 1.**[–KD]** This encourages $d(x)$ to remain within the space of $1-$Lipschitz functions without harsh constraints on its parameters. The second technique is **spectral normalization** applied to the critic's layers.**[–KD]** Spectral normalization rescales the weight matrix of each layer such that its largest singular value is 1**[cite –KD]**. By doing so at every training step, we effectively bound the Lipschitz constant of each linear component of $d(x)$, ensuring that $d(x)$ cannot change faster than a certain rate with respect to its input.**[–KD]** Even small violations of the Lipschitz condition can lead to training instabilities (the critic might exploit them to push L_{WGAN} lower, causing g to react with wildly large weights). Spectral normalization prevents the critic from assigning arbitrarily large scores to individual samples.**[–KD]** This has an important regularizing effect on the generator's behaviour. If $d(x)$ is bounded and smooth, the incentives for $g(z)$ to produce extremely large weight factors are reduced, because no single event can ever

receive a disproportionately large score advantage. In effect, spectral normalization caps the influence of any one data—simulation discrepancy on the loss. The combined use of gradient penalties and spectral normalization in RAN was found to be effective for training stability—they keep the critic in check so that the adversarial game remains in a regime where gradients are informative and the generator's updates remain moderate. With these in place, we substantially avoid the divergence and mode–collapse issues that plague naive adversarial training.**[–KD]**

Regularizing the Generator: Initialization and Activation Constraints

On the generator side (the reweighting function $g(z)$), RAN incorporates two important measures to regularize the solution. First, we initialize $g(z)$ to close to the identity function before beginning adversarial training. In the context of unfolding, the "identity" reweighting corresponds to assigning weight 1 to every MC event (i.e. initially assuming the Monte Carlo truth distribution is correct and no reweighting is needed). We realize this by initializing the neural network representing $g(z)$ such that its output ≈ 1 for all z. This initialization reflects our prior belief that the generator only needs to make relatively small, smooth adjustments to the starting simulation, which is a reasonable assumption in many cases where the Monte Carlo is tuned to approximate reality. By starting close to the identity mapping, we avoid a situation where early in training the critic finds large differences and drives g to extreme compensations. Large fluctuations in weights early on can kick off a feedback loop of instability (the critic then sees a very irregular simulated distribution and reacts pathologically, etc.). The initialization acts as a strong regularizer by biasing g toward the null reweighting unless the data indicate otherwise. This idea is analogous to using a Bayesian prior equal to the simulation and regularizing toward it at the start; only genuine discrepancies will pull g away from 1. Indeed, from the perspective of classical regularization, our weight initialization is equivalent to choosing the simulated distribution as a prior and initially penalizing deviations from it. We find that this leads to a much smoother evolution of the adversarial game—initially the critic cannot easily tell data from reweighted simulation because $g \approx 1$ still leaves them broadly similar, so d learns gradually to distinguish the two, and g then adapts gradually in turn. This procedure significantly reduces the risk of the training falling into bad local minima or diverging in the first epochs. In summary, initializing close to the identity provides a "cold start" regularization that keeps RAN in the perturbative regime where it performs best, rather than having to learn an unpredictable transformation from scratch.**[–KD]**

The second generator–side regularization is controling the asymptotic behaviour of the generator output parameterization for the weights. It is essential that $g(z)$ produce positive weights (events can only be reweighted by positive factors) and have the capacity to yield a wide range of values (some events might need weight $\gg 1$ if data are more abundant

there, others weight < 1 if data are scarce). A naive choice is to define $g(z) = \exp[f_\theta(z)]$ where $f_\theta(z)$ is the output of a neural network with no constraints, and the exponential ensures positivity and unboundedness. This choice is in a sense very well motivated by the Boltzmann distribution that inspired Moment Unfolding. In practice however, one finds that this choice, while mathematically valid, leads to numerical instability. The exponential is a very rapidly growing function, so any relatively large output of the network $f_\theta(z)$ would translate into an astronomically large weight, drastically skewing the training. Moreover, the gradient of $\exp[f_\theta(z)]$ is proportional to the output itself, so once a weight becomes huge, its gradient is huge too, often causing large and unstable oscillations. To remedy this, a custom activation was designed, that *tames* the asymptotic growth of $g(z)$ at large network outputs while preserving the desired mathematical properties. In the final layer of the $g(z)$ network, we replace the $\exp$ with a composite function $\log(1 + \mathrm{softplus}(x))$. $\mathrm{softplus}(x) = \log(1 + e^x)$ is a smooth approximation to a ReLU (it grows linearly for large x, but is smooth everywhere)**[cite –KD]**. Wrapping it in $\log(1 + \cdot)$ further slows the growth for large inputs: as $x \to \infty$, $\mathrm{softplus}(x) \approx x$, so $\log(1 + \mathrm{softplus}(x)) \approx \log(1 + x) \sim \log x$. Thus for very large raw network output x, this activation grows only logarithmically, instead of exponentially or even linearly. This adjusted activation guarantees that

1. $g(z) > 0$ for any input (by construction of softplus and log),
2. $g(z)$ can still produce arbitrarily large values in principle (no finite upper bound, unlike a sigmoid which saturates) so we do not limit the solution space unduly, but
3. Extremely large weights are disfavored because pushing $f_\theta(z)$ to $\gg 0$ yields only diminishing returns in $g(z)$.

In practice, this means the network would have to invest a lot of capacity to achieve a huge weight on a single event, which is only worth it if that truly reduces the Wasserstein distance significantly. Typically, it will be more effective to increase weights more evenly on a group of events covering a region of phase space than to make one weight enormous. This activation function thus serves as an *internal regularizer*, curbing the tendency of the solution to form spikes or outlier weights. It is important to emphasize that this is a soft constraint. Large weights are not forbidden, but the system must pay a price to realize them, much like a physical prior that discourages sharp discontinuities in the solution unless the data demand them. The use of this modified activation proves to be crucial for achieving stable training in RAN; with the standard exponential, one observes frequent instances of a single event's weight blowing up and derailing the fit, whereas the log–softplus activation yields more balanced weight distributions.

Regularizing via MC Prior

Finally, it is worth noting an intrinsic form of regularization in RAN that comes "for free": by construction, the unfolded result is expressed as a reweighted version of the initial generated truth sample. This means the unfolded distribution cannot introduce structures that were not present (even in latent form) in the MC. In effect, the MC provides a prior support and shape for the solution. If the true underlying distribution has features outside the MC's support, RAN (like any reweighting method) cannot recover them. But conversely, it will not produce spurious artifacts that violate known physics encoded in the simulation. This is similar in spirit to a Bayesian prior or a template fit: one starts with a template (the generation) and only deform it as necessary to fit the data. The closer the generation is to reality, the less adjustment is needed (and the smaller the risk of overfitting). RAN's whole design of "tweaking a known density rather than learning a new one from scratch" is motivated by the desire to leverage this built–in regularization. This allows RAN's solutions to be smoother and more statistically robust than those from unconstrained generative models, precisely because $g(z)$ operates within the scaffolding of the Monte Carlo sample. This comes at the cost of some bias if the Monte Carlo is poor (no method can escape that without additional input), but it is a conscious regularization choice favouring stability over unrestricted flexibility.

In summary, the RAN methodology interweaves modern ML techniques with classical regularization principles to achieve a stable unfolding. The use of the Wasserstein distance (WGAN) provides a gentle, physics–aligned way of comparing distributions that avoids the brittleness of binary classification metrics. Imposing Lipschitz continuity on the critic via gradient penalties and spectral normalization keeps the adversarial game well–behaved and prevents the discriminator from overemphasizing statistical noise. On the generator side, starting from the physical prior (simulation) and limiting the capacity for extreme weights (through initialization and activation choices) anchor the solution close to expected physics and discourage overfitting. Together, these innovations allow RAN to unfold full distributions in one shot (non–iteratively) without the severe instabilities that one might fear from an unconstrained GAN approach. In effect, RAN attains a balance: it is flexible enough to accurately fit complex, high–dimensional data, yet sufficiently regularized to suppress unphysical oscillations and variance. The result is an unfolding method that achieves competitive or superior performance to iterative methods like OmniFold, while operating in a single training pass and maintaining controlled behaviour even in challenging regimes (such as limited detector overlap or low–statistics bins)**[cite –KD]**. The following sections will demonstrate these points quantitatively with examples, but the methodological foundation laid out here is key to understanding why RAN performs as well as it does. It marries the strengths of adversarial learning with the hard–earned lessons of regularization

from decades of unfolding research, yielding a novel and powerful approach to this classic inverse problem.

6.4 ML Implementation

Neural Network Architecture

A Reweighting Adversarial Network (RAN) consists of two components: a *generator* network that assigns event-wise weights, and a *critic* network that evaluates the discrepancy between weighted simulation and data. We implemented both as small fully-connected (dense) neural networks using the TensorFlow 2/Keras framework for prototyping **[cite –KD]**. [2] The architecture was kept intentionally simple to ensure stable training and avoid over–fitting, with identical layer widths for the generator and critic.

The generator $g(z;\boldsymbol{\beta})$ takes as input the *particle–level* feature vector $z \in \mathbb{R}^{N_T}$ of a generated event (for example, $N_T = 1$ for one–dimensional toy data or $N_T = 6$ for the multi–observable jet dataset described below). It outputs a scalar weight $w = g(z)$ that reweights that event in order to correct the simulation towards data. The generator is a feed–forward multilayer perceptron with three hidden layers of 50 nodes each. Each hidden layer uses a Leaky Rectified Linear Unit (Leaky ReLU) activation **[cite maas2013rectifier –KD]**. We apply batch normalization and a dropout of 20% after each dense layer to promote stable training and reduce overfitting. The final layer of the generator is a single linear neuron (no activation) producing a raw scalar t. This output is then transformed to a positive weight. Instead of a direct exponential mapping (which can be prone to numerical instability for large positive or negative t), we employ a stabilized transformation,

$$w \;=\; \ln\Big(1 + \mathrm{softplus}(t)\Big)\,. \tag{6.6}$$

Finally, the set of weights w_i in each mini–batch is normalized so that their average is unity. This batch–level normalization means the total weight of the simulated sample remains consistent (preserving overall event counts) and prevents trivial solutions where the network could simply scale all weights up or down without improving the relative distribution shape. The generator network thus defines a differentiable reweighting function $g(z)$ that can smoothly adjust the contribution of each simulated event during training.

The critic $d(x;\boldsymbol{\theta})$ is a discriminator–like network that takes as input the detector–level feature vector $x \in \mathbb{R}^{N_D}$ of an event and outputs a real–valued score. This network is tasked with distinguishing the *weighted* simulated data from the true data in the detector space,

[2] The design is straightforward to reproduce in PyTorch using analogous layers and normalization techniques.

and its output is used as an approximate distance measure between the two distributions. Like the generator, the critic is implemented as a fully–connected network with three hidden layers of 50 nodes each, using Leaky ReLU activations and interleaved batch normalization and 20% dropout. The output layer is a single linear node producing the critic score $d(x)$. To satisfy the requirements of the Wasserstein GAN framework, the critic must be a $1-$Lipschitz function. We enforce this constraint via spectral normalization on each dense layer's weights **[cite –KD]**DBLP:journals/corr/abs-2009-02773. As discussed, spectral normalization scales the layer weights such that their largest singular value is 1, effectively controlling the Lipschitz constant of d without the need for weight clipping. This technique, combined with dropout, greatly improved training stability in the unbounded–weight setup by preventing the critic from becoming overly sensitive to outlier events. Additionally, we apply a mild output clipping on the critic scores **[cite –KD]**, capping $d(x)$ within a reasonable range to eliminate spurious large values that could arise from statistical fluctuations (which in our context would correspond to unphysical negative probabilities or overly large separation scores). A schematic of the RAN architecture is shown in Fig. **??** **[(fig: model architecture) –KD]**. The balanced capacity of the generator and critic (each with $\mathcal{O}(10^4)$ trainable parameters) was found sufficient to learn the required reweighting functions without overfitting, given the complexity of the tasks considered.

Adversarial Training Procedure

Training a RAN involves a minimax game between the generator and critic, similar in spirit to a Generative Adversarial Network (GAN). However, unlike a standard GAN that generates new events, our generator only reweights existing events, and we adopt the Wasserstein GAN (WGAN) approach **[cite –KD]**pmlr-v70-arjovsky17a for improved stability. The critic's objective is to maximize the statistical distance (in the Wasserstein-1 sense) between the weighted simulation and the true data distributions at the detector level, while the generator's objective is to produce weights that minimize this distance. Formally, let $p(x)$ be the true data distribution in detector space and $q(z, x)$ be the joint distribution of generation and simulation. At each training step, the critic d is trained to minimize the loss function

$$L_d = -\mathbb{E}_{x\sim p}[\, d(x)\,] \;+\; \mathbb{E}_{(z,x)\sim q}\big[\, g(z)\, d(x)\,\big]\,, \tag{6.7}$$

where (z, x) denotes corresponding generation and simulation event pairs. $-L_d$ is an empirical estimate of the Wasserstein-1 distance between the true distribution and the reweighted simulated distribution. The generator, on the other hand, is trained to maximize this same quantity (equivalently, to minimize the negative critic loss $-L_d$), thereby pushing the weighted simulation to resemble the data. Unlike the binary cross–entropy

loss in a traditional GAN discriminator, the Wasserstein formulation provides a smooth, continuous loss landscape even when the two distributions do not overlap perfectly **[cite arjovsky2017wassersteingan –KD]**. This is helpful for unfolding problems because even if detector effects cause $p(x)$ and the original $q(x)$ to have disjoint support in x, the WGAN critic can still provide informative gradients to the generator. This choice eliminates the severe mode collapse or large weight oscillations that occur when using the BCE loss in extant classifier based methods (the WGAN's stronger theoretical guarantees, combined with the property that RAN only classifies at detector–level, and only reweights at particle–level, ensure the training remains well-behaved even with minimal overlap) **[cite gulrajani2017improvedtrainingwassersteingans –KD]**. To further enforce the $1-$Lipschitz condition required by the Wasserstein theory, one can include a gradient penalty term **[cite gulrajani2017improvedtrainingwassersteingans –KD]** in the critic's loss. This penalty,

$$L_{GP} = \lambda \, \mathbb{E}_{\hat{x}} \left[(\|\nabla_{\hat{x}} d(\hat{x})\|_2 - 1)^2 \right], \tag{6.8}$$

is computed on random points $\hat{x}$ interpolated between real and generated samples. λ usually set between five and ten. The gradient penalty, together with spectral normalization, acts as a robust regularizer against critic over–training.

Training proceeds by alternating between critic and generator updates in each iteration, following the typical WGAN–GP strategy. We used an update ratio of 3:2 (critic:generator), meaning the critic is updated slightly more frequently to keep it near optimality as the generator evolves **[cite –KD]**arjovsky2017wassersteingan. Pseudocode for one training cycle is outlined in Algorithm **??**.

These steps constitute one training iteration, and they are repeated until convergence criteria are met (typically tens of thousands of mini–batch iterations, depending on dataset size). We optimize both networks using the RMSProp optimizer **[cite –KD]**, which we found to outperform Adam in our WGAN setting (consistent with previous studies **[cite DBLP:journals/corr/abs-1810-02525 –KD]**). The learning rate for both generator and critic was set in the range $\eta \sim 1 \times 10^{-4}$ to 5×10^{-4} (with the exact value tuned per dataset; for the toy example we used the lower end, while the more complex jet data benefited from a slightly higher rate around 2×10^{-4}). We did not observe the need for explicit learning rate decay schedules beyond those automatically implemented by Keras,**[–KD]** as training typically converged to a stable solution before any signs of stalling. A mini-batch size of $B = 256$ was used for all experiments, providing a good balance between gradient estimate stability and computational efficiency. We note that we did not perform an exhaustive hyperparameter search; instead, we relied on standard GAN/WGAN settings from the literature and minor tuning. In practice, we found the performance of RAN to be robust against moderate changes in these settings: for instance, altering the learning rate by a factor of two or using 40 or 60 nodes per layer (instead of 50) had no significant impact on

Algorithm 1: WGAN-GP Training Step

Input: Data distribution $p(x)$, generation–simulation pairs $q(z, x)$, critic d_θ, generator weights $g(z; \beta)$,
gradient-penalty weight λ, critic steps n_c, generator steps n_g, batch size B
Output: Trained parameters θ and β
for *iteration* $= 1, 2, \ldots$ **do**
  `// Critic updates`
  for i *in* 1 **to** n_c **do**
    Sample $\{x_i^{data}\}_{i=1}^{B} \sim p(x)$;
    Sample $\{(z_j, x_j\}_{j=1}^{B} \sim q(z, x)$;
    Compute $w_j = g(z_j; \beta)$;
    Compute

$$L_d = \frac{1}{B}\sum_{i=1}^{B} d_\theta(x_i^{data}) - \frac{1}{B}\sum_{j=1}^{B} w_j \, d_\theta\big(x_j^{sim}\big) + \lambda \, L_{GP}(\theta)\,; \tag{6.9}$$

    $\theta \leftarrow \theta + \eta_d \nabla_\theta L_d$;
  `// Generator updates`
  for j *in* 1 **to** n_g **do**
    Sample fresh $\{x_i^{\text{data}}\}_{i=1}^{B} \sim p(x)$;
    Sample $\{(z_j, x_j\}_{j=1}^{B} \sim q(z, x)$;
    Compute $w_j = g(z_j; \beta)$;
    Compute

$$L_g = -\frac{1}{B}\sum_{j=1}^{B} w_j \, d_\theta\big(x_j^{\text{sim}}\big)\big) + \frac{1}{B}\sum_{i=1}^{B} d_\theta(x_i^{\text{data}})\,; \tag{6.10}$$

    $\beta \leftarrow \beta - \eta_g \nabla_\beta L_g$;

the final unfolded distributions. This insensitivity suggests that the RAN method does not require fine–tuned hyperparameters to achieve reliable results, an important consideration for reproducibility.

Throughout training, we continuously monitor key loss terms to ensure the optimization remains stable. In particular, we track the Wasserstein critic loss L_d and the gradient penalty magnitude. A sharp increase in the gradient penalty or a sudden large oscillation in L_d can indicate that the critic is violating the Lipschitz constraint or that the generator is producing pathological weights (a potential onset of mode collapse). If such behaviour is

observed, one can intervene by pausing training and lowering the learning rate or increasing regularization. In our experiments, the use of spectral normalization and gradient penalty together largely prevented such divergences. We emphasize that RAN's one–shot training (as opposed to an iterative approach) simplifies the overall optimization schedule: once the adversarial equilibrium is reached, we obtain the final reweighting function without needing multiple retraining cycles.

Datasets and Preprocessing

We applied the above architecture and training procedure to two representative unfolding scenarios: a controlled $1-$dimensional Gaussian example and a realistic high–dimensional jet physics example. These two cases allow us to validate the RAN implementation on both simple and complex distributions.

Gaussian Example

The first dataset is a synthetic scenario with known analytical truth, designed to illustrate RAN's behavior under varying degrees of detector smearing. We generate two sets of events from normal distributions at the particle level, a "truth" distribution $Z_{\text{T}} \sim \mathcal{N}(\mu_{\text{true}}, , 1)$ and a "generator" (simulation) distribution $Z_{\text{G}} \sim \mathcal{N}(\mu_{\text{gen}}, , 1)$, with means chosen to be $\mu_{\text{true}} = 0$ and $\mu_{\text{gen}} = -1$. We sample 10^4 events from the truth distribution and 10^5 events from the generation distribution. To emulate detector effects, each particle-level value z is transformed by a simple deterministic response: $x = S \cdot z$, where S is a scalar "smearing" factor. For example, with $S = 1$ the detector–level measurement equals the true value (no smearing), while $S > 1$ stretches the distribution, reducing the overlap between the "data" ($x = S \cdot z_{\text{T}}$) and "simulated" ($x = S \cdot z_{\text{G}}$) detector–level observations. We consider a range of S values to progressively degrade the overlap. For this one–dimensional setup, $N_T = N_D = 1$ (each event is described by a single number). Apart from the deterministic scaling, no additional noise or inefficiency is introduced, so the challenge lies purely in the shift of distributions and the lack of detector–level support overlap when S is large. We do not apply any additional preprocessing to the input features since they are already standardized (mean shifts aside) and bounded. During training, at each iteration we draw a random batch of $B = 256$ values from the 10^4 available smeared data points and another batch of 256 from the 10^5 smeared simulation points. Because the simulation sample is larger, we cycle through it (randomly shuffled) such that multiple data epochs correspond to one pass through the simulation. It is beneficial to shuffle and reshuffle both samples frequently during training to avoid any periodic artifacts given the fixed deterministic smearing.

The RAN is tasked with learning event weights $g(z)$ that correct the generation distribution (centered at -1) to match the true distribution (centered at 0), using only classification information from the x domain where in extreme cases the distributions barely overlap. This toy example is especially useful for validating the implementation because the underlying true reweighting function is known (in the limit of infinite statistics, the optimal weight would be proportional to the ratio of target to sim densities at z, which in this case is $\exp[-(z-\mu_{\text{true}})^2/2 + (z-\mu_{\text{gen}})^2/2]$, a Gaussian weight). It also allows us to verify that RAN's adversarial training indeed converges to the correct solution, by comparing the unfolded distribution to the analytic expectation.

Jet Substructure Dataset

The second dataset studied is a realistic example taken from high–energy particle physics, involving the unfolding of jet substructure observables. We follow the setup of the OmniFold study **[cite Andreassen:2019cjw –KD]** to enable direct comparisons. Proton-proton collision events at $\sqrt{s} = 14$ TeV were generated using two different Monte Carlo simulators, Pythia8.243 **[cite Sjostrand:2014zea,Sjostrand:2007gs –KD]** (with CMS Tune26 **[cite ATL-PHYS-PUB-2014-021 –KD]**) provided the initial *generation* sample, and Herwig 7.1.5 **[cite Bahr:2008pv,Bellm:2017bvx –KD]** was used to generate an independent sample treated as the *truth* target distribution. Each event in both samples contains a high-p_T Z boson (with $p_T^Z > 200$ GeV) decaying leptonically, produced back–to–back with a hadronic jet. The presence of a Z boson trigger ensures a well–defined event sample and reduces acceptance differences. The final–state particles in each event were passed through a fast detector simulation using Delphes 3.4.2 **[cite deFavereau:2013fsa –KD]** with a CMS detector configuration (including particle–flow reconstruction **[cite Mertens:2015kba,CMS:2017yfk –KD]**). Jets are then reconstructed in both the particle–level and detector–level event records using the anti$-k_T$ clustering algorithm **[cite Cacciari:2008gp –KD]** (radius parameter $R = 0.4$) as implemented in FastJet3.3.2 **[cite Cacciari:2011ma –KD]**. We focus on the hardest jet in each event (the jet that balances the Z boson's recoil) to define our observables. For each jet, we consider six different substructure observables that characterize its internal pattern of particles. These include classic measures like the jet mass m and jet breadth or width w, as well as more specialized variables such as the $N-$subjettiness ratio τ_{21}, the groomed momentum fraction z_g from the Soft Drop algorithm, and a logarithmic version of the dimensionless jet mass $\ln \rho$. Precise definitions of these observables follow those in Ref.**[cite Andreassen:2019cjw –KD]**, and we refer the reader there or to Appendix **[ref –KD]** for the formulae. In total, each jet is described by a feature vector of $N_T = N_D = 6$ numbers (since the same set of observables is defined at particle–level and, after Delphes simulation, at detector–level). These six features form the input to the RAN networks. It is helpful to whiten (z–score)**[–KD]** these inputs to minimize finite precision and overflow errors. We

prepared the data such that the truth sample (Herwig, particle–level) contained on the order of 10^6 jets, and the simulation sample (Pythia, particle-level) a similar order of magnitude. The detector–level representations for each sample are obtained via the same Delphes procedure, yielding paired (z, x) examples for Pythia and independent x examples for Herwig. During training, we treat the Delphes+Herwig jets as our "real data" distribution $p(x)$ and the Delphes+Pythia jets as the initial $q(x)$. At each iteration, mini-batches of $B = 256$ jets are drawn from each distribution. Given the relatively large size of the samples, we did not need to worry about cycling through the data too many times; we simply train for a fixed number of iterations (determined by early stopping as described next). We ensured that each jet in the training sample is used many times over the course of training, which is important for the generator to refine the weights on all regions of phase space. Notably, because RAN's generator applies weights at the particle level (prior to detector simulation), it effectively learns to reweight the Pythia particle-level jets such that, after Delphes, their distribution of observables matches that of the Delphes-processed Herwig jets. Once training is complete, those learned weights $w = g(z)$ can be applied directly to the Pythia particle–level events to produce an unfolded distribution for each observable, which we can then compare to the true Herwig particle–level distribution. We emphasize that in this example the "data" is actually fully simulated (Herwig+Delphes) rather than collider observations; this is a common practice in validation studies, allowing us to quantitatively evaluate the accuracy of the unfolding by comparing to known truth. The RAN method, of course, does not rely on knowing the truth distribution during training—it only uses the detector–level information, so it is directly applicable to real data once validated.

Training Monitoring and Validation

To ensure reliable performance and prevent overtraining, one should employ a rigorous monitoring and validation regimen during RAN training. First, one should partitioned each dataset into a training set (used to update network parameters), a smaller validation set (used only for performance monitoring), and a testing set. The ratios used were training:validation:testing $70 : 15 : 15$ The validation data is not seen by the networks during gradient updates, but one computes the same losses on it to check how well the model generalizes. Specifically, after every fixed number of training iterations (for instance, every 100 mini-batch updates), one evaluates the critic loss L_d on the validation batch and tracks its evolution. If one observes the validation loss starting to increase while the training loss continues to decrease, this is a clear sign of overfitting i.e. the critic is learning spurious differences that do not generalize, or the generator is over–adjusting to peculiarities of the training sample. In such cases, one should invoke an *early stopping* criterion. In our implementation, early stopping was triggered if the validation Wasserstein distance (approximated by $-L_f$ on the validation set) had not improved for 10 consecutive evaluation

intervals. Once triggered, one stops training and rolls back to the generator state that had the best validation performance. This strategy ensures that one did not over–train the critic to exploit statistical noise, which would manifest as unstable weights when applied to new data.

In addition to the WGAN loss, one can track several high–level divergence metrics between the reweighted generation and truth distributions. While these metrics are not used to train the model (since truth is not known in real applications), they are extremely useful to diagnose convergence and compare to alternative methods. One such metric is the **Wasserstein-1 distance** itself between the weighted and true distributions. Although our critic provides an estimate of this distance via the loss, one can also compute it independently for simpler cases. For the 1D Gaussian example, the Wasserstein distance has a closed-form (it reduces to the difference in means when both distributions are Gaussian with equal width), and for the jet observables one can calculate one-dimensional Wasserstein distances for each observable's distribution individually. One can also monitored the **Vincze–Le Cam (VLC) divergence** **[cite DBLP:journals/corr/abs-2009-10838, nishiyama2022relationstightboundssymmetric –KD]** between the unfolded (reweighted) generation and the truth. The VLC divergence, sometimes referred to as the triangular discrimination, is a symmetric divergence measure defined for two probability density functions p and q as

$$\Delta_{VLC}(p,q) \;=\; \frac{1}{2}\int_{-\infty}^{\infty} \frac{\big(p(x)-q(x)\big)^2}{p(x)+q(x)}\,\mathrm{d}x\,. \tag{6.11}$$

Lower values indicate closer agreement between distributions ($\Delta_{VLC}=0$ if and only if $p=q$). This metric is particularly sensitive to differences in the tails of distributions and provides complementary check to the Wasserstein distance. During training, one could periodically apply the current set of weights $g(z)$ to a large sample of simulation events and then evaluate Δ_{VLC} and the Wasserstein distance between this weighted sample and the truth sample (using the known truth in validation studies). A decreasing trend in both metrics as training progresses signals that RAN is converging towards the correct reweighting.

For instance, in the Gaussian example with moderate smearing, we observed the VLC divergence between the weighted simulation and truth drop rapidly in the first few dozen iterations and flatten out at a low value, indicating that the generator had found an effective weighting scheme. Similar behavior was seen in the jet case: the combined six-dimensional weight optimization gradually improved the agreement in all observables. By the end of training, the unfolded distributions for all six jet observables were nearly indistinguishable from the truth, as quantified by final Wasserstein distances on the order of a few times 10^{-2} and VLC divergence on the order of 10^{-3} (see **[cite Andreassen:2019cjw –KD]** for

comparable benchmarks with iterative methods). We include summary tables (Table **??** and Table **??** in Section **??**) comparing these divergence metrics for RAN versus other methods; the RAN implementation achieves competitive or superior scores, validating our training procedure.

Crucially, the monitoring process also guards against the possibility of divergence or weight blow–up. If at any point the training appears to destabilize (e.g. the critic loss oscillates wildly or the weights $\{g(z)\}$ start to become extremely large for a small subset of events), we can intervene by either stopping early or introducing additional regularization. In practice, thanks to the built-in regularizers (spectral norm, gradient penalty, dropout), such pathological behaviour was rare. One symptom we might encounter in some trials was the appearance of a few very large weights in sparsely populated regions of phase space. This is a known issue in unfolding problems. When there are regions that have little simulated density but non–negligible data density, any method will assign large weights to the few simulation events in that region. RAN is no exception; nonetheless, the spectral normalization on the critic help limit how fast and how large those weights could grow. If needed, one could impose an explicit cap on weights or an additional L_p norm term in the loss to penalize overly large weights, but we did not find this necessary for our benchmarks.

In summary, the ML implementation of RAN described above proved to be robust and reproducible across the different datasets. By carefully specifying the architecture (with sufficient complexity to learn the needed reweighting, but regularized to avoid overfitting), using a stable training objective (Wasserstein GAN with gradient penalty) and optimizer (RMSProp), and employing diligent monitoring/early–stopping criteria, we ensured that the training converges reliably to a good solution. The design choices were informed by both theoretical considerations (e.g. the advantage of WGAN for non–overlapping supports) and practical experimentation (e.g. the choice of 3 critic updates per 2 generator updates for efficiency). The resulting RAN model can be straightforwardly re–implemented in common deep learning frameworks, and our reported hyperparameters can serve as a baseline for future applications. All training scripts and data have been made available open–source to facilitate verification and reuse. Thus, this implementation satisfies the dual goals of performance and transparency, which are essential for trustable unfolding in high–energy physics.

6.5 Results

We evaluate the Reweighting Adversarial Network (RAN) on two case studies: a targetting one–dimensional Gaussian unfolding task and a realistic jet substructure unfolding problem. In each case, RAN's performance is compared to two baseline methods, the OmniFold algorithm **[Andreassen:2019cjw,Andreassen:2021zzk –KD]** and Iterative Bayesian

Unfolding (IBU) **[DAgostini:1994fjx –KD]**. We report quantitative divergence metrics for each method and analyse their ability to reproduce the true (particle–level) distributions (closure tests), as well as the methods' stability under statistical fluctuations.

Gaussian Model with Smearing

We first consider a targeted unfolding scenario to stress-test RAN in a regime of limited detector–level overlap. In this model, the truth particle–level distribution is a Gaussian $T \sim \mathcal{N}(0, , 1)$, while the initial simulation (prior) distribution is $G \sim \mathcal{N}(-1, , 1)$. This introduces a modest discrepancy at particle level (a mean offset of 1 unit). We then simulate a detector response that scales each observable by a constant "smearing" factor. As the smearing factor increases above unity, the detector–level distributions of the data (smeared truth) and simulation (smeared prior) become increasingly separated (with less overlapping support), even though the underlying particle–level distributions remain fixed. This controlled setup allows us to examine how each unfolding method copes with a progressively worsening mismatch between the observed (detector–level) data and the simulation. Notably, methods like OmniFold rely reweighting at the detector level, which can fail if the detector–level supports do not overlap well. In contrast, RAN performs reweighting before the detector simulation (at particle level) and uses an optimal transport–based loss to compare detector–level outputs. We therefore expect RAN to be more robust in scenarios with poor detector–level overlap, provided the particle–level distributions still cover each other.

To quantify performance, we use the Wasserstein-1 distance (W_1) and the Vincze—Le Cam (VLC) divergence Δ_{VLC} between the unfolded outcome and the true distribution as metrics of accuracy. Figure **[fig:omnifold-comp –KD]** summarizes the unfolding accuracy as a function of the smearing factor, with Δ_{VLC} plotted for RAN and OmniFold. At minimal smearing (factor 1.0, essentially an identity detector response), both methods perform excellently, and OmniFold achieves a slightly smaller divergence to truth than RAN (for example, at smearing 1, Δ_{VLC} for OmniFold is 0.057×10^{-3}, compared to RAN's 0.122×10^{-3}). This confirms that when the detector effects are negligible and the problem reduces to a straightforward reweighting, OmniFold's direct classifier–based approach is very effective—indeed, it finds a nearly perfect weighting in this easy regime.

However, as the smearing increases and detector–level overlap deteriorates, OmniFold's performance degrades sharply. Beyond a smearing factor of about 1.5, the OmniFold solution's divergence from truth grows rapidly. By the highest smearing values tested (where the detector–level simulation and data distributions are almost disjoint), OmniFold's unfolded result significantly deviates from the true distribution (nearly an order of magnitude worse Δ_{VLC} compared to the no–smear case). This is consistent with expectations. Omnifold tried to reweight at detector level, and when it cannot find common phase space between

data and simulation, it struggles to assign meaningful weights. In stark contrast, RAN maintains a stable performance across the entire range of smearings. As shown in Figure **[fig:omnifold-comp –KD]**, RAN's Δ_{VLC} remains approximately constant (with only slight variation within statistical error) as smearing increases. Even at the largest smearing factors, RAN's divergence to truth is essentially the same as it was at zero smearing, indicating that RAN successfully unfolds the distribution despite scalar multiplication. This behaviour can be attributed to RAN's methodological property of only reweighting at particle level, combined with its use of an optimal transport loss at detector level, which, unlike a classifier, does not require overlapping support to provide a meaningful gradient—the adversarial training can adjust weights to minimize the Wasserstein distance between the smeared simulation and data, even if those distributions are partially disjoint. The error bars in Figure **[fig:omnifold-comp –KD]** (representing 1σ uncertainties from bootstrapping the experiment) confirm the statistical stability of each method. RAN's results are consistent across bootstrap resamples, and while OmniFold's uncertainty also grows with smearing, the trend of degradation is clear beyond the error bars. In summary, this Gaussian study demonstrates that RAN is robust to limited detector–level support, recovering the true distribution reliably in a single pass, whereas OmniFold requires sufficient detector–level overlap to perform optimally. This advantage of RAN becomes critical in more complex, realistic scenarios where detector effects can be large.

Jet Substructure Unfolding Results

We next assess RAN on a realistic high-energy physics unfolding task: correcting jet substructure observables from detector–level "measured" data to particle–level truth. This example serves as a stringent test of RAN's ability to handle multi–dimensional, non–Gaussian distributions with sharp features and long tails. The observables chosen span a range of distribution shapes—from broad distributions to sharply peaked ones and those with kinematic cutoffs—providing a comprehensive testbed. The unfolding task is to reweight the Pythia generation such that, after detector simulation, it matches the "data" distribution (here, the Herwig detector-level output), and then to compare the reweighted Pythia particle-level jets to the Herwig particle-level truth. RAN is applied to learn a single weighting function $g(z)$ that assigns a weight to each Pythia event (based on its features) to achieve this alignment, as described in Secs.**??**–**??**. For comparison, we also apply OmniFold **[Andreassen:2019cjw,Andreassen:2021zzk –KD]** and the classical IBU **[DAgostini:1994fjx –KD]** to the same task. Since IBU is a binned algorithm, we perform separate one–dimensional IBU unfolds for each observable using a fine histogram binning, to benchmark its performance on each projection of the data.

Overall performance

Figure **[fig:particle-level-distribution –KD]** presents the unfolded particle–level distributions for each of the six jet observables, for RAN, IBU, and OmniFold, compared to the truth distributions. Each panel shows the truth spectrum (Herwig, treated as "true" data) in solid blue, the generation in orange, the unfolded result from RAN as a black dashed line, and the unfolded result from OmniFold as a red dashed line, and the result from IBU as a green dashed line. The lower ratio sub–panels display the unfolded/Truth ratio for each method, indicating how well each method closes the gap to truth across the range of each observable. We observe that both RAN and OmniFold achieve good agreement with truth across all observables, a non–trivial accomplishment given the complexity of the distributions, but RAN consistently provides a closer match, particularly in regions that are challenging to unfold. For example, in the jet mass distribution (Fig.**[fig:particle-level-distribution –KD]**, top-left panel), the Pythia simulation initially undershoots the probability in the high-mass tail and misplaces the peak. After unfolding, RAN's distribution closely follows the truth curve: it accurately reproduces the peak around $m \approx 20$ GeV and the long tail up to high masses. OmniFold also significantly improves the agreement, but its ratio plot reveals a slight residual bias in the tail (deviations of order 5–10% from unity in the highest mass bins), whereas RAN's ratio is within a few percent of unity throughout. A similar pattern is seen for the N-subjettiness ratio τ_{21}. The generator (Pythia) initially produces a broader τ_{21} distribution than truth (Herwig), indicating an overestimation of two–prong substructure. RAN's unfolding narrows this distribution to align almost exactly with the truth shape, correcting both the peak and the tail of the τ_{21} spectrum; OmniFold moves in the right direction but leaves a noticeable difference (the OmniFold unfolded τ_{21} distribution remains slightly too broad, with the ratio to truth dipping below 1.0 in the peak region and rising above 1.0 at higher τ_{21}). For observables like the jet width w and logarithmic groomed mass $\ln \rho$, which have sharply peaked distributions near zero and long tails, RAN again excels: it captures the steep drop-off and the tail behavior with high fidelity, whereas OmniFold shows small mismodeling in the intermediate region (for w) or tail (for $\ln \rho$). The groomed momentum fraction z_g is a particularly challenging variable because of its hard cutoff at $z_g = 0.5$. Generative or reweighting methods often struggle to reproduce such cutoff behavior precisely. We find that RAN handles this boundary reasonably well The unfolded z_g distribution from RAN matches the truth both in the low–z_g region and near the cutoff, correctly recovering the falling slope as $z_g \to 0.5^-$. OmniFold's result for z_g is also reasonable but tends to slightly undercorrect near the cutoff (its ratio is a bit below 1.0 approaching 0.5, indicating a remaining deficit in events just below the cutoff). Finally, the constituent multiplicity M (number of particles in the jet) is an interesting case. It is a discrete distribution with a long tail. RAN succeeds in reducing the discrepancy at both the low-M and high-M extremes, yielding an unfolded distribution that tracks the truth

across the full range. OmniFold improves the multiplicity distribution in the bulk region but exhibits larger fluctuations in the very high-multiplicity tail (partly due to limited statistics there and the difficulty for the classifier to learn in sparse regions). In summary, qualitatively, RAN's unfolded spectra are virtually indistinguishable from the truth for all six observables, within statistical uncertainties, whereas OmniFold shows minor but noticeable deviations in certain challenging regions. The ratio (closure) plots in Fig.**[fig:particle-level-distribution –KD]** highlight RAN's better closure: the RAN/Truth ratio is closer to 1.0 (often within a few percent) across the phase space, while the OmniFold/Truth ratio deviates by up to 5–15% in some bins (especially in the tails of m, M, τ_{21}, and $\ln \rho$).

To make these comparisons quantitative, we compute the Wasserstein-1 distances (W_1) and VLC divergences between each unfolded distribution and the truth distribution for all methods. Table **[tab:comp-w –KD]** and Table **[tab:comp-vlc –KD]** summarize these metrics for RAN, OmniFold, and IBU on each jet observable (lower values indicate better agreement with truth; the tables also include, for reference, the baseline distances for the Generation vs. truth and Data vs. truth distributions with no unfolding). RAN achieves the smallest divergence from truth in every one of the six observables under both metrics. In particular, RAN outperforms OmniFold by a substantial margin in those observables identified as challenging: for the jet mass m, W_1(RAN) is 3.35×10^{-2} (in units of the table's scale) compared to OmniFold's 8.02×10^{-2}, a roughly factor–of–2 improvement. For the logarithmic groomed mass $\ln \rho$, RAN's W_1 distance is 0.20×10^{-3}, dramatically smaller than OmniFold's 4.15×10^{-3} – an order-of-magnitude reduction, indicating RAN has far better accuracy in the extreme tail of this distribution. Similarly, for z_g, RAN's W_1 is about 0.80×10^{-3} vs. OmniFold's 6.32×10^{-3}, reflecting how well RAN handled the cutoff region. Even in observables where OmniFold performed strongly, such as jet width w, RAN still edges out a win (0.59×10^{-3} vs. 0.81×10^{-3}). The VLC divergence results (Table **[tab:comp-vlc –KD]**) tell a consistent story: RAN yields the lowest Δ_{VLC} for all observables, with OmniFold typically second–best. For example, $\Delta_{VLC}(m)$ is 2.31 (in the table's scaled units) for RAN, versus 5.06 for OmniFold and 5.11 for IBU. In some cases the gap between RAN and OmniFold is negligible (jet width w: 1.29 vs 1.33), while in others it is significant (multiplicity M: 5.83 vs 6.32 for OmniFold and a much larger 16.07 for IBU, indicating IBU struggled with M; or τ_{21}: 0.92 vs 7.91 for OmniFold, an order of magnitude difference). We emphasize that IBU, being a binned, per–observable method, generally underperforms the unbinned methods here. This is likely due to binning and statistical issues: the need to choose finite bin widths leads to information loss and large uncertainties in sparse regions (for M, the tail probabilities in high bins were not well–estimated, leading to an inflated divergence). In contrast, RAN and OmniFold operate on unbinned data and can leverage the full event information, yielding superior precision. It is noteworthy that RAN's advantage is achieved while producing a single set of event–level weights that simultaneously correct all six distributions. In other words, RAN (like OmniFold) inherently unfolds the joint multi–dimensional distribution of

these observables without splitting the problem into one dimension at a time. The fact that a single model can balance all observables and still attain better per-observable accuracy is a strong testament to the effectiveness of the adversarial reweighting approach.

Beyond accuracy, stability and efficiency are important considerations for unfolding methods. We examined the stability of the RAN training by performing multiple independent training runs and by bootstrapping subsets of the jet dataset. The variations in the resulting unfolded distributions (and in the W_1/VLC metrics) were found to be small, on the order of a few percent, indicating that the RAN solution is robust to statistical fluctuations and the stochastic nature of training. Likewise, OmniFold's iterative procedure, when run on the same data with different initializations, converged to comparable results (though minor differences in the weights per iteration can occur, the final distributions were consistent within uncertainties). IBU, being an analytic iterative method, showed negligible run-to-run variation for fixed binning. However, all iterative methods raise the question of convergence and tuning. OmniFold requires choosing a stopping criterion or number of iterations (too few iterations can under–correct, too many can overfit the statistical noise), and IBU similarly depends on the number of iterations (or an implicit regularization). In our OmniFold implementation we used five iterations (the standard approach advocated in **[Andreassen:2019cjw –KD]**) which was sufficient for good performance; using more iterations does not notably improve the agreement and can introduce instability. RAN avoids this ambiguity entirely, since it is a one–shot training. In practice, we found RAN's training to converge steadily without signs of overtraining (monitored via validation loss) and to reach a stable solution within a few dozen epochs. In terms of computational cost, RAN's single–pass adversarial training (using a Wasserstein GAN-like setup) was comparable to the cost of a single OmniFold iteration. But since OmniFold required two classifier trainings per iteration (one at detector level, one at particle level) and we performed two iterations (four trainings total), the overall runtime for RAN was roughly a fifth of that for OmniFold on this problem. IBU is extremely fast for one–dimensional histograms, but extending IBU to many dimensions would be combinatorially expensive (and practically impossible for high-dimensional continuous observables), whereas RAN and OmniFold scale gracefully to high–dimensional data. Thus, RAN offers an attractive trade–off: it achieves equal or better accuracy than OmniFold on these benchmark tasks, while being simpler to use (no iterative loop to manage) and potentially more efficient.

Generality of results

Combined with the findings from the Gaussian experiments, these jet experiments suggest that RAN's advantages are not specific to a particular distribution but rather reflect general features of the algorithm. RAN's ability to handle non-overlapping detector effects (demonstrated in the Gaussian toy study) implies it could be deployed in experimental

scenarios with poor detector resolutions or acceptance gaps, where traditional unfolding might struggle. The jet study confirms that RAN scales to complex, realistic tasks, providing high–fidelity unfolding for multiple correlated observables in a single training. We expect that these properties would carry over to other unfolding applications. For example, measurements of other final states or higher–dimensional distributions (such as multi–variate phase–space distributions) should similarly benefit from RAN's unbinned, non–iterative approach. Of course, some caution is warranted when extrapolating beyond the tested cases: real experimental data may involve additional complexities like non–identical simulation vs. reality in ways not captured by generator differences alone, or require careful treatment of systematic uncertainties in the unfolding procedure. However, the closure tests performed here (using one simulator's data as "pseudo-data" and another's as truth) are a stringent validation, and RAN passes them with flying colours. In particular, RAN's strong performance on observables with sharp features (z_g cutoff) and heavy tails (m, $\ln \rho$) bodes well for its application to other distributions where similar challenges arise (e.g. distributions with kinematic edges or long perturbative tails). Moreover, RAN's inherently multivariate nature means that, unlike IBU, it can preserve and unfold correlations between observables, an important aspect for modern high–dimensional analyses in particle physics. In summary, the results presented in this section demonstrate that RAN achieves state–of–the–art unfolding performance on both simple and complex tasks. It matches or exceeds the accuracy of established methods like OmniFold and IBU, while offering improved stability in difficult scenarios and simplifying the unfolding workflow. These characteristics make RAN a promising tool for future precision measurements and exploratory studies in which unfolding high–dimensional data without bias is critical.

Chapter 7

Unbinned Inference on Correlated Data

In the preceding chapters we developed a hierarchy of unfolding techniques, from classical binned approaches (Chapter **??**) through Neural Posterior Unfolding (Chapter **??**) and the Moment–GAN strategy (Chapter **??**), culminating in a fully unbinned unfolding algorithm that operates directly on event-–level information. Once the distributions are unfolded, parameter inference is performed on the unfolded data to summarize the the event information into a small number of observables whose differential cross–sections are compared with theory by computing best fit parameters and confidence intervals. While performing inference, an implicit assumption of *statistical independence* of individual events is often made. This independence guarantees that the joint likelihood factorises and allows the log–likelihood in **??** to be written as a *sum* over events.

In practice, however, independence is broken whenever events are processed through an unfolding procedure. Deconvolution couples phase-–space regions and induces highly non—trivial (often long—range) correlations between formerly independent entries. Ignoring those correlations can bias parameter estimates and/or lead to misestimation of confidence intervals.

7.1 Statistical Independence in HEP

In high energy physics (HEP) cross section measurements, standard statistical treatments for unbinned inference rely on a set of core assumptions of *event independence*. These assumptions posit that collision events are generated and observed as independent trials of a stochastic process, which greatly simplifies the construction of likelihood functions and inference procedures. Below I outline the key independence assumptions commonly made in unbinned analyses, and subsequently examine their applicability or limitations in experimental contexts.

Poisson Point Process for Event Counts

A fundamental premise is that the total number of events N observed in an experiment follows a Poisson distribution. If $\mu(\theta)$ denotes the expected event yield for parameters θ (e.g. proportional to the integrated luminosity times the cross section), the probability of observing N events is

$$P(N|\theta) = e^{-\mu(\theta)}\,\mu(\theta)^N/N!. \tag{7.1}$$

This reflects the physical picture of collisions occurring as a Poisson point process in time, with each event occurrence independent of the last **[cite –KD]**. The Poisson assumption is built into the *extended* likelihood formalism **[cite –KD]**, ensuring that normalization (total event count) is appropriately handled in parameter inference for cross sections. It provides a way to incorporate both the shape of distributions and the overall event yield into the likelihood. In practice, this means that in repeated identical experiments one would expect the observed N to fluctuate about $\mu(\theta)$ according to Poisson statistics, and it justifies including a factor $e^{-\mu}\mu^N/N!$ in the likelihood function of a single dataset.

Independent and Identically Distributed (i.i.d.) Events

It is assumed that each event can be treated as an independent draw from the *same* underlying probability density function (pdf) $p(x|\theta)$, where x denotes the measured observables (kinematic variables, detector signals, etc.) for that event. In other words, conditional on the physics model parameters θ, all events are statistically independent and governed by an identical distribution **[cite –KD]**. This implies that one event's occurrence or properties do not influence any other event. Equivalently, there are no intrinsic correlations or memory between events. The identical distribution assumption further requires that the experimental conditions remain stable so that each collision is sampled from the same pdf $p(x|\theta)$. This condition typically enforced by construction, by dividing data–taking into consistent periods with fixed detector configuration and calibrations. Combined with the Poisson law for N, the i.i.d. assumption forms the basis of the usual HEP data generation model, *vidilicet,* the data are viewed as a Poisson sample of independent events from $p(x|\theta)$. This assumption underlies virtually all unbinned analysis techniques in HEP, from simple maximum likelihood fits to modern machine learning based inference methods **[cite –KD]**. It also justifies resampling techniques like bootstrapping (sampling events with replacement to create pseudo–experiments), since events are treated as exchangeable independent samples.

Likelihood Factorization and Conditional Independence

Because events are modelled as independent draws, the joint probability density for N events factorizes into a product of single–event densities. For a given dataset $\{x_1, x_2, \ldots, x_N\}$, one can write

$$P(x_1, \ldots, x_N \,|\, \theta) \;=\; \prod_{i=1}^{N} p(x_i \,|\, \theta)\,, \tag{7.2}$$

which in turn leads to a factorized likelihood function. In the common case where N itself is treated as Poisson-distributed, the full (extended) likelihood is

$$\mathcal{L}(\theta) \;=\; e^{-\mu(\theta)} \frac{\mu(\theta)^N}{N!} \prod_{i=1}^{N} p(x_i \,|\, \theta)\,, \tag{7.3}$$

assuming all events are independent of each other **[cite –KD]**. The independence assumption allows the log–likelihood to be written as a sum over events,

$$\ln \mathcal{L}(\theta) \;=\; -\mu(\theta) + \sum_{i=1}^{N} \ln p(x_i|\theta)\,, \tag{7.4}$$

(up to the constant term $\ln(N!)$). In cases where N is fixed and not of interest, the $-\mu + \frac{\mu^N}{N!}$ portion is often dropped, yielding the simplified unbinned likelihood $\prod_i p(x_i|\theta)$. Crucially, the factorization in Eq. **??** holds only under the assumption that events are statistically independent. This property of likelihood factorizability enormously simplifies inference: it enables the use of well–known asymptotic statistical results (such as Wilks' theorem) on the sum of per–event log–likelihood contributions **[cite –KD]**, and it permits analytical derivations of estimators and uncertainties. Indeed, widely–used formulae for the variance of estimators and for test statistics (e.g. those in Ref. **[cite –KD]**) presume an underlying model where $\ln \mathcal{L}$ sums over independent events. We also note that independence is typically understood as *conditional* independence given the model parameters θ and any nuisance parameters. For example, if events come from multiple sources (signal and background processes), one usually models each source as an independent Poisson process; all events are then independent samples overall, with an additional mixture component in $p(x|\theta)$. Likewise, any uncertainty in calibration constants or other global parameters is introduced via nuisance parameters rather than treating events as correlated. Under fixed values of all such parameters, events remain independent. This conditional independence perspective justifies how multiple contributions are combined in likelihoods and how one accounts for systematic effects without introducing inter–event correlations (the correlations induced by shared nuisance parameters are handled at the parameter level rather than by coupling events in the probability model).

7.2 Violation of statistical independence

These statistical independence assumptions have served as the backbone of cross section measurements and other inference tasks in HEP for decades. They are reasonably well–motivated by the physics of particle collisions. Each proton–proton collision (or other fundamental interaction) is localized in space–time and causally independent from other collisions, and detectors are typically designed and operated to measure each collision event in isolation. Moreover, by treating events as independent, analysts can take full advantage of powerful likelihood–based inference techniques and modern machine learning methods that operate on event–by–event data **[cite –KD]**. Unbinned machine learning approaches to parameter estimation, for instance, explicitly rely on the factorized likelihood in Eq. (**??**) to avoid information loss from binning **[cite –KD]**. The independence assumption is also what allows one to rigorously define an asymptotic χ^2 or log–likelihood ratio statistic that sums over events and, in the large$-N$ limit, follows known distributions for hypothesis testing **[cite –KD]**. In short, the usual paradigm treats each recorded event as an independent piece of evidence about the underlying physics parameters.

However, these assumptions are *idealizations* that can be violated in a variety of scenarios, thereby challenging the simplistic i.i.d. picture. It is crucial to recognize and mitigate these issues because blindly applying independent–event methods in their presence can lead to biased estimates or misestimated uncertainties.

Detector Effects

While the i.i.d. assumption requires a stable underlying distribution $p(x|\theta)$ for all events, in reality the detector and experimental conditions can evolve or fluctuate, effectively making events drawn from slightly different distributions. For example, changes in detector response (due to calibration drifts, aging of instrumentation, or varying operational settings) can cause the probability distribution of observables to shift over time. If unaccounted, this means an event recorded early in the run is not identically distributed to an event recorded later. In addition, the finite resolution and inefficiencies of the detector are typically incorporated into $p(x|\theta)$ via a detector–response model; if this model is imperfect, the residuals can introduce correlations. Another detector–induced effect is the presence of noise bursts or environmental backgrounds (such as cosmic rays or electronics noise) that can affect multiple events in a correlated way. For instance, a temporary malfunction in a subdetector could bias a whole set of consecutive events in a similar manner. Experimentalists usually handle these issues by segmenting data–taking periods and calibrating each segment separately, or by introducing nuisance parameters to account for time–dependent efficiency changes. When treated properly, one can restore the approximation of identical distributions; but if such variations are neglected, the assumption of identical (and independent) events breaks

down. In essence, the detector can introduce a slight correlation structure or at least a heterogeneity among events, violating the ideal independence assumption. These effects are often manifested as systematic uncertainties in the analysis (e.g. a calibration uncertainty correlates the predicted rates of all events), and are handled by propagating those uncertainties rather than by treating events as correlated. Still, it is a reminder that truly independent, identical event samples exist only in the limit of a perfectly stable detector and complete modelling of its response.

Pileup

At modern hadron colliders (notably the LHC), multiple proton-proton collisions can occur in the same beam crossing and be recorded together. This phenomenon, known as *pileup*, means that what is recorded as a single "event" may actually be a composite of several independent collisions superimposed in the detector readout. High–luminosity running conditions can lead to dozens of overlapping collisions per event. In an ideal scenario, these would be uncorrelated collisions, and indeed they are physically independent interactions; however, the merging of their signals confounds the independence at the level of reconstructed events. Sophisticated event reconstruction algorithms attempt to separate pileup interactions (for example, by identifying multiple distinct collision vertices and attributing detector hits to different vertices). Despite these efforts, residual entanglement remains: additional tracks or energy deposits from pileup interactions can slip into the reconstruction of the hard-scatter event of interest, affecting measured quantities like jets, missing energy, or lepton isolation. As a result, the observables x_i for a given recorded event can include contributions from other simultaneous events, blurring the notion that each x_i is drawn from the single-event distribution $p(x|\theta)$. In extreme cases, one collision's presence can influence whether another collision is recorded at all (due to trigger bandwidth or detector occupancy limits), introducing a form of inter-event dependence. Pileup therefore violates the assumption that each event is an independent trial of the same distribution: instead we have conglomerate events with extra particles, and the probability of certain features (e.g. multiple soft jets, high detector occupancy) is higher than what a single-collision $p(x|\theta)$ would predict. To mitigate pileup, experiments apply corrections or weights to event observables (such as subtracting average energy from soft interactions) **[cite –KD]**, or they incorporate pileup explicitly into the modeling by extending $p(x|\theta)$ to include pileup contributions. When pileup is included in the Monte Carlo model and properly tuned, the effective independence can be partially restored by enlarging the event description (i.e. treat the hard collision and pileup as parts of one compound event). Nonetheless, any imperfections in pileup mitigation mean that events in a high-pileup dataset are not as independent as assumed. This is especially pertinent for precision measurements at the High-Luminosity LHC, where average pileup multiplicities will be extremely high **[cite**

–KD]. Overlapping events can also occur in other contexts (e.g. neutrino experiments might have cosmic ray overlays, astrophysical observations might have coincident signals) — all such scenarios require care if one is to maintain the statistical interpretation of independent events.

Unfolding and Data Processing

An increasingly common workflow in HEP is to first *unfold* the measured data (correcting for detector effects to infer particle–level distributions) and then perform inference or fits on these unfolded results. Unfolding algorithms, including modern unbinned ones, assign weights or probabilities to events in order to match detector–level data to a particle–level prediction. However, the unfolding procedure itself induces statistical correlations among the resulting events. In an unbinned unfolding, the output "events" (often weighted events) are no longer independent: they are coupled by the global constraints of the unfolding (such as conserving overall normalization and matching distributions). For example, if one unfolded event's weight fluctuates upward, typically some other event's weight must fluctuate downward to compensate and preserve the total yield or certain distributions **[cite –KD]**. Thus, the set of unfolded events has an empirical covariance structure i.e. knowledge of one event's outcome gives information about others. If one were to naïvely apply an unbinned likelihood analysis assuming those unfolded events are independent, the likelihood factorization would be incorrect, and one might misestimate uncertainties or bias the fit. This situation has been explicitly observed: studies have shown that treating unfolded events as independent can misestimate the confidence intervals, with asymptotic error formulas significantly underestimating the true uncertainty **[cite –KD]**. In our context, this is a prominent example where the conditional independence assumption of events (conditional on θ) does not hold: the data points (unfolded events) carry correlations from the statistical inversion procedure. In practice, experimental analyses that use unfolded data eitehr avoid unbinned fits on the raw unfolded events (they aggregate unfolded results into bins with an associated covariance matrix that captures the induced correlations **[cite –KD]**) or they simply ignore the covariance structure of unbinned unfolded data.**[–KD]** For instance, if an unfolded spectrum is used to extract a physics parameter, the unfolding team will provide a covariance matrix Σ such that a binned χ^2 fit can be performed,

$$\chi^2_{\text{full}}(\theta) = (D - P(\theta))^T \Sigma^{-1} (D - P(\theta)) \,, \tag{7.5}$$

where D is the vector of unfolded data (binned) and $P(\theta)$ the corresponding prediction **[cite –KD]**. This χ^2 (or the equivalent binned likelihood) correctly accounts for correlations via Σ. By contrast, a fully unbinned likelihood $\prod_i p(x_i|\theta)$ on the unfolded events x_i would ignore the Σ information and treat events as independent, which is formally unjustified.

The advent of machine learning has enabled high–dimensional unbinned unfolding **[cite –KD]**, increasing the urgency to address these correlations. Our current recommendation is to avoid analytic uncertainty formulas in such cases and rely on bootstraps to evaluate uncertainties **[cite –KD]**. Future research efforts could attempt to incorporate event–level correlations into the unbinned likelihood formalism—for example, by parametrising weight fluctuations or using augmented likelihoods that include correlation terms. Until such methods mature, the safest approach for unfolded data is to introduce correlations either at the binned level, which effectively relinquishes some of the benefits of unbinned methods in exchange for correct coverage of uncertainties, or by bootstrapping, which significantly increases the computational cost of the analysis.

Global Constraints

Even when individual collisions are truly independent, certain *global* aspects of an analysis can couple what were otherwise independent events. A prime example is the presence of common normalization factors or theoretical parameters that influence all events. For instance, the integrated luminosity of a dataset (used to convert event counts to cross sections) is usually known with some uncertainty. If treated as a nuisance parameter, that single parameter induces correlations between the rates of events in different signal regions: effectively, an upward fluctuation in luminosity would scale up the expected counts for all events. In a frequentist formulation, events remain independent conditional on a fixed luminosity, but when one considers the uncertainty on luminosity, the joint distribution of all events together has an additional covariance (because they all scale together). Similarly, theoretical model uncertainties (like parton distribution functions in a hadron collision) can correlate the kinematic distributions of all events. For example, if a PDF parameter shifts, it will concurrently affect the probability distributions of many events, creating a correlation in their fluctuations. Usually these effects are handled by introducing correlated systematic shifts in the expectation rather than by modifying the event independence in the likelihood; nonetheless, they highlight that the true data–generating process has an interdependence through shared parameters.

In Bayesian terms, if one marginalizes over an uncertain global parameter, the remaining distribution of events is no longer a simple product of independent PDFs. Beyond nuisance parameters, there are also physics scenarios that produce genuine correlations between events: for instance, certain new physics might produce clustered events (e.g. a decay of a heavy state that yields two separate collision–like signals in the detector, or cosmic ray air showers producing multiple spatially separated detector hits treated as separate events). These are exotic cases but illustrate that nature can produce data that do not align with the one–event–at–a–time paradigm. Generally, whenever a theoretical or methodological constraint ties the outcomes of different events, the independence assumption is violated.

In the context of cross section measurements, one common manifestation is in combined or multi-dimensional fits: if one fits multiple distributions simultaneously with shared parameters, the events populating those distributions are effectively analysed in one joint likelihood. The independence assumption still might hold event–by–event, but events in different categories become statistically coupled through the shared parameter constraints. The customary solution is again to incorporate those constraints at the likelihood level (via nuisance parameters, profile likelihood techniques, or covariance matrices across distributions) **[cite –KD]**. This ensures that while the calculation may still treat events as independent given the parameters, the final results correctly reflect the induced correlations.

In summary, the assumption that events are generated independently and identically is a powerful simplification that underlies many HEP analysis techniques, from classical maximum likelihood fits to cutting-edge machine learning inference frameworks. These assumptions are approximately valid for raw collision data under well–controlled conditions and have enabled analysts to exploit the full statistical power of unbinned data **[cite –KD]**. Nonetheless, real–world complexities such as detector imperfections, pileup, processing–induced correlations, and global effect correlations demand a careful treatment beyond the idealized i.i.d. model. In contemporary binned analyses, it is standard to incorporate such effects via covariance matrices or nuisance parameters rather than to abandon the independence assumption entirely. The challenge for the field moving forward is to extend our statistical formalisms (and tools) to unbinned data, so that we can continue to perform unbiased, efficient inference even as we perform complex global fits of unbinned unfolded data, where events are no longer perfectly independent. Ultimately, recognizing the limits of the independence assumptions and quantifying any resulting biases or miscoverage is essential for robust cross section measurements. By confronting these limitations, either by correcting them or explicitly incorporating them into the statistical model, HEP analyses can ensure that improvements in methodology such as unbinned approaches remain scientifically reliable even in the face of complex data dependencies.

7.3 Consequences for Inference

In high energy particle collider experiments, it is usually assumed that individual collision events are statistically independent. This assumption holds true at the detector level for collisions because each interaction is separate from the next, and with appropriate event selection one can treat the observed events as independent samples of an underlying probability distribution. Traditional likelihood–based inference methods heavily rely on this independence. he joint likelihood for N independent events factorizes into a product of single-event likelihoods, making the log-likelihood a sum over events, as shown in Eq. **??**.

Equation**??** underpins many inference techniques and allows the use of well–known asymptotic statistical results (e.g. Wilks' theorem) to efficiently compute confidence intervals and $p-$values in high energy physics analyses **[cite –KD]**. In the context of collider measurements, this means one can often streamline parameter extraction by summing likelihood contributions from each collision event independently, an approach that has been validated and applied extensively for detector–level analyses where events are indeed uncorrelated **[cite –KD]**.

However, correlated data arise inevitably in the process of reconstructing physical (particle–level or parton–level) quantities from raw detector observations. In modern collider analyses of differential cross sections, a number of effects introduce non–negligible correlations among data points (or even among reconstructed "events"). In addition to the general sources of correlations discussed in Sec. **??**, some HEP specific effects are

- **Detector resolution and acceptance:** Imperfect detector measurements cause migrations of events between bins of a distribution. For example, a true particle with high momentum might be reconstructed with a lower measured momentum. Correcting for these migrations via unfolding or calibration couples the statistical content of neighbouring bins. As a result, the final differential cross section values in adjacent kinematic bins are often statistically anti–correlated. If one bin's yield fluctuates high, the unfolding procedure tends to pull down the neighbouring bin's yield to conserve the overall event count and vice versa **[cite –KD]**. Likewise, global efficiency or acceptance corrections (e.g. accounting for detector coverage) can introduce positive correlations across all bins by scaling yields coherently.

- **Pileup (multiple collisions per bunch crossing):** [1] In high–luminosity running, each recorded event may contain overlapping contributions from several simultaneous collisions. This overlap complicates event reconstruction and usually requires algorithmic subtraction or weighting to estimate the true single–collision observables. Residual pileup contributions act like a correlated background noise affecting many events in a similar way. For instance, an increased number of simultaneous interactions tends to add extra low-p_T particles across the event, causing all jet energies in that event to shift slightly. When such effects are corrected statistically (e.g. subtracting an average pileup contribution), the correction uncertainty is correlated across all events in a given run. Moreover, pileup mitigation often relies on average profiles, meaning that any mismodeling of pileup leads to systematic shifts in distributions that are common to many events (correlations between events' weights). Thus, while individual events remain physically independent, their inferred particle–level properties carry a common uncertainty component from pileup treatment **[cite –KD]**.

[1] Referred to in heavy ion collider experiments as the "underlying event".

- **Unfolding and deconvolution:** A consequence of regularization during unfolding is that the resulting unfolded data points are no longer independent. Instead, they come associated with a covariance matrix that encodes both the variance of each bin and the bin–to–bin correlations. In binned unfolding (such as iterative Bayesian or matrix inversion methods), adjacent bins often develop substantial anti–correlations due to the smoothing constraints, and all bins can share common normalisation systematics. In unbinned unfolding approaches, one obtains a reweighted set of events at particle level rather than fixed bins. These reweighted events carry event–by–event weight factors that are determined collectively to reproduce the measured detector data. Consequently, the unfolded events are statistically correlated with each other. If one unfolded event's weight fluctuates upward to fit a fluctuation in the data, another event's weight must adjust downward to compensate. The assumption of independent events is violated for such unfolded particle–level samples **[cite –KD]**. In other words, the deconvolution procedure entangles events, breaking the direct applicability of Eq. **??**.

- **Common systematic uncertainties:** Many experimental systematic effects (detector calibration, energy scale, efficiency corrections, luminosity uncertainty, etc.) induce fully or partially correlated uncertainties across multiple data points. For example, the uncertainty in the integrated luminosity (used to normalize cross sections) is a single multiplicative factor affecting the normalization of the cross section in every bin simultaneously. This manifests as a 100% bin–to–bin correlation for that component of uncertainty since all bins could shift up or down together by the same relative factor. Similarly, a jet energy scale uncertainty might cause a shape distortion that correlates a rise in high$-p_T$ bin yields with a fall in low$-p_T$ bin yields (introducing anti–correlations among those bins). These systematic effects mean that the total uncertainty covariance on the final results has significant off–diagonal terms. Even if the statistical fluctuations of events were independent, the presence of correlated systematics requires careful treatment in any inferential test or fit.

These sources of uncertainty and their effects are summarised in Table **??**.

Denoting by D_j the unfolded, particle-level yield (or cross section) in bin j (with $j = 1, \ldots, M$ bins) and by $P_j(\theta)$ the theory prediction (or fit template) for that bin given parameters θ, the standard measure of agreement is a correlated χ^2 statistic described in Eq. **??**, where Σ is the $M \times M$ covariance matrix of the measurement**[cite –KD]**. The covariance matrix Σ encodes the variance of each bin along the diagonal ($\Sigma_{jj} = \mathrm{Var}(D_j)$) and the correlations in the off–diagonal entries ($\Sigma_{ij} = \mathrm{Cov}(D_i, D_j)$ for $i \neq j$). In the limit of many events and Gaussian uncertainties, $\chi^2_{\text{full}}(\theta)$ is expected to follow a χ^2 distribution with $M - n_\theta$ degrees of freedom (n_θ being the number of fitted parameters), provided Σ

is properly included. This forms the basis of parameter fits and theory tests using binned data. One finds the best–fit parameters by minimizing χ^2_{full}, and evaluates goodness–of–fit or confidence intervals by referring to the χ^2 distribution **[cite –KD]**. Crucially, all modern precision measurements must incorporate such correlations in their inferential procedure. Analyses that fit parton distribution functions or theoretical model parameters to data from multiple bins (or even multiple experiments) must propagate the full covariance; otherwise the fit would erroneously constrain parameters using spurious information.

To illustrate the impact of correlated versus uncorrelated treatments, consider a simple example. Let a measurement report two bins with identical central values and uncertainties, and let there be a 100% positive correlation between their uncertainties (for example, a fully correlated normalization error). If a theory predicts both bins' values higher than observed (say by $+1\sigma$ of the reported uncertainty in each bin), a proper correlated χ^2 calculation will find *no significant tension* because the two data points moved together could be explained by a single upward fluctuation or a slight systematic shift. In fact, the χ^2_{full} in this case might be of order 1, well within expectations.

In contrast, if one had naïvely ignored the correlation and treated the points as independent, the same $+1\sigma$ deviations in each bin would add in quadrature, giving $\chi^2_{\text{diag}} = ((+1\sigma)^2 + (+1\sigma)^2) = 2$ for two degrees of freedom. The corresponding $p-$value would be significantly smaller, and one might incorrectly suspect a poor fit to the theory. In this way, neglecting correlations can inflate test statistics and lead to overly aggressive claims of discrepancy between data and theory. This simple example underscores how a common systematic uncertainty (like an overall normalization) should not be counted as independent evidence of mismatch in each bin. Only by using the full covariance (which, in this case, has identical off–diagonal entries equal to the variance) do we obtain the correct statistical interpretation.

More generally, the effect of data correlations on parameter extraction can be understood in terms of effective sample size and information content. If one has N independent data points, uncertainties on fitted parameters typically scale as $\sim 1/\sqrt{N}$ (all else being equal). But if the N points have positive correlations, the effective amount of independent information is smaller than N. For instance, in the idealized case that each pair of points has a uniform correlation coefficient $\rho \in [0, 1]$, the variance of the mean (or any common–strength parameter estimator) is

$$\text{Var}(\bar{X}) \;=\; \frac{\sigma^2}{N}\Big[\,1 + (N-1)\rho\,\Big], \tag{7.6}$$

where σ^2 is the variance of each individual point. The factor $1 + (N-1)\rho$ quantifies the inflation in variance due to correlation. One can define an effective number of independent points $N_{\text{eff}} = N/[1 + (N-1)\rho]$**[cite –KD]**. For $\rho > 0$, $N_{\text{eff}} < N$; in the extreme case of $\rho \to 1$ (all points completely correlated), $N_{\text{eff}} \to 1$ no matter how large N is. This

is precisely what would happen, for example, if a common normalization uncertainty dominated. No matter how many bins are measured, if they all share the same normalization shift, the overall normalization is just one degree of freedom of uncertainty rather than N independent ones. Ignoring correlations amounts to pretending $N_{\text{eff}} = N$ underestimates the true variance of combined measurements or fitted parameters. Consequently, the fit uncertainties can be grossly underestimated if correlations are neglected. Confidence intervals derived under the false assumption of independent data will miscover—the actual probability that the true value lies within the reported interval will be lower than the stated confidence level. It is important to note that the magnitude and direction of the error induced by ignoring correlations cannot be predicted *a priori* in the general case. In some cases, treating correlated uncertainties as if they were independent can also overestimate the total uncertainty or degrade fit precision. For example, if unfolding induces anti–correlations between neighbouring bins, a fit that neglects those anti–correlations will effectively treat upward fluctuations in one bin and simultaneous downward fluctuations in the adjacent bin as if they were statistically independent deviations of opposite sign. The fit might then struggle to accommodate both, leading to a larger fit uncertainty. The concrete study of a Gaussian unfolding problem demonstrated in Sec. **??** shows that using only the diagonal elements of the covariance matrix (i.e. assuming no bin–to–bin correlation) yields larger asymptotic errors on the fitted parameters compared to using the full covariance, particularly when the detector smearing (and thus induced correlations) is large. In this study, the "diagonal-only" error bands are overly conservative because they double–count fluctuations that in reality were constrained by correlation. In contrast, the full–covariance analysis and a pseudo–experiment (meant to represent bootstrapping) estimate of uncertainty are in good agreement and show smaller uncertainties. These findings reinforce that incorporating the full covariance structure is critical for obtaining accurate uncertainty estimates, both to avoid underestimation in some scenarios and excessive conservatism in others.

From the perspective of bias in parameter extraction, the presence of correlated data does not inherently introduce bias in a fit provided the fitting procedure properly accounts for those correlations. An unbiased estimator remains unbiased under linear transformations (such as combining bins with weights given by Σ^{-1}). That said, the processes that create correlated data can themselves introduce bias. For instance, unfolding methods that regularize strongly can produce a biased estimate of the true distribution (often biasing towards some smooth default model). If one then fits theory parameters to such unfolded data, the results might inherit this unfolding bias. Moreover, an inappropriate handling of correlations can indirectly lead to bias if the fitter effectively misweights parts of the data. In extreme cases, ignoring correlation could cause the fit to chase statistical fluctuations in one bin without realizing that a correlated fluctuation in another bin is providing a counteracting influence. The binned study in Sec **??** found that the best–fit values of parameters (such as the mean and width of a Gaussian distribution) were very similar whether correlations

were accounted for or not, and any small biases as a function of detector smearing were attributable to the overall analysis procedure rather than the correlation handling. This suggests that for reasonably well–behaved problems, the central values of fits may not shift dramatically by ignoring correlations (i.e. the maximum likelihood estimator for θ can remain approximately unbiased). The bigger impact is on the estimated uncertainties and the validity of statistical tests, not necessarily the point estimate itself. Still, in more complex or poorly modelled situations, neglecting correlations could conceivably pull the fit off the true value if the fitter is effectively solving the wrong optimization problem.

One frontier where the correlated data problem has become especially salient is in the advent of unbinned unfolding and machine learning–driven differential measurements. New algorithms allow experiments to publish results in an unbinned form i.e. instead of histogramming the unfolded cross section in fixed bins, the result may be a reweighted set of events or a learned probability density function representing the particle-level distribution **[cite –KD]**. This is extremely powerful for downstream physics interpretation because it preserves maximum information (no binning or averaging) and, in principle, enables multi–differential or high–dimensional comparisons that would be infeasible with coarse bins. However, a serious caveat emerges. All unfolding techniques (binned or unbinned) introduce correlations among the events or data points that make up the unfolded result. These techniques output a set of weights or density function parameters that are fitted to the entire dataset at once, effectively entangling the contributions of individual events. When we attempt to perform likelihood–based inference directly on such an unbinned result, we confront the fact that we do not have N independent draws from the true distribution, but rather a single, composite object that was sculpted by the full dataset. In absence of a known analytical likelihood for correlated events, one tempting but naïve approach is to ignore the correlations and plug the unfolded events (with their weights) into Eq. **??**, treating them as if they were independent samples of $p(x|\theta)$. This is precisely the scenario in which the independence assumption is violated and can lead to pathological statistical conclusions.

The studies in Sec. **??** quantitatively demonstrate the pitfalls of using unbinned unfolded data naïvely in likelihood fits. A Gaussian example is used to compare three approaches,

1. A traditional binned unfolding followed by a binned likelihood fit (using Eq.**??**),
2. An unbinned unfolding with the results binned post hoc for a covariance–based fit, and
3. An unbinned unfolding with a fully unbinned fit ignoring event correlations (i.e. treating the unfolded events as independent in Eq.**??**).

This study also found that the central values of the fitted parameters (e.g. the Gaussian mean and width) did not significantly differ between these approaches. For this experiment, the

naïve unbinned fit still found roughly the correct answer for the best–fit θ. This indicates that, at least in simple cases, the bias on the estimator from ignoring correlations might be small. However, when using the naïve unbinned likelihood (ignoring correlations), the experiment found that the asymptotic formulas for uncertainty (based on the second-derivative of the log–likelihood or $\Delta \ln \mathcal{L} = 0.5$ for 1σ) no longer held true. In fact, for very large smearing, the usual relationship between the curvature of the log–likelihood and the confidence interval broke down because the likelihood function was misspecified by assuming independence. The study confirmed that the coverage of nominal 95% confidence intervals was increasingly misspecified as the smearing grew in the naïve unbinned approach, meaning one would frequently get confidence intervals that fail to contain the true parameter at the promised rate. This is a direct manifestation of miscoverage due to neglecting the induced correlations. From another angle, one can say that the test statistic distribution deviates from the expected χ^2 or Z-score calibration: for example, a likelihood ratio that should follow a χ^2_1 distribution for 1 parameter might have a much sharper distribution, so the actual p-values are not what one would infer under the independent assumption**[cite –KD]**.

An important observation from these investigations is that using numerical methods for inference is one possible way to work around this problem. If, instead of relying on asymptotic error formulas or nominal χ^2 distributions, one employs Monte Carlo techniques (such as bootstrapping or toy pseudo-experiments) to directly compute the distribution of estimators or test statistics, then it is possible to obtain correct uncertainties when event–level correlations are present **[cite –KD]**. In the Gaussian toy study, when they performed many bootstrap replicas of the unfolding + fit process, the empirical spread of the fitted parameters ("numerical uncertainty") matched well with the full–covariance analytic result and revealed the the naïve method's error. In fact, the naïve unbinned inference (ignoring correlations) yielded uncertainty estimates that were only reliable when those uncertainties were derived by bootstrap rather than by an analytic approximation. In other words, one can still use unbinned unfolded data for parameter extraction, but one must forgo simplistic formulas and instead rely on computationally intensive resampling to get trustworthy error bars. Practically, this is a strong motivation either to re–bin the data (recovering a standard χ^2 approach with covariance) or to develop new statistical formalism that explicitly includes the correlation structure in unbinned likelihoods. At present, there is no known closed–form likelihood analogue to Eq. **??** that accommodates event–to–event correlations in an unbinned dataset **[cite –KD]**. Developing such a formalism (for example, an approach using copulas to model the joint PDF of all events, or incorporating the covariance matrix into a generalized likelihood) is left to future research. Until then, the safest course for analyses is either to stick to the well–validated binned methods or to use brute–force numerical inference procedures for unbinned results.

It is worth highlighting that various HEP experiments are actively exploring these modern methods. For instance, the H1 collaboration and others have released unfolded

results using machine learning that are effectively unbinned density estimates **[cite –KD]**. So far, no official parameter fits or new physics tests have been performed directly on such unbinned outputs. The field is cognizant that while unbinned unfolded data contain rich information, any naïve use could yield misleadingly tight constraints or false signals if correlations are ignored. Hence correlations must be treated with the same rigour as any other aspect of the measurement. In practical terms for theory testing, this means that if one is given a published covariance matrix along with a set of data (whether binned or unbinned), one should incorporate it in the likelihood or χ^2; if one only has an unbinned sample of weighted events, one should either bin them (with the provided weights) or obtain the equivalent covariance before fitting, or otherwise use provided ensemble replicas to assess uncertainty. For the moment, the recommendation for robust inference is to avoid relying on analytical approximations (like Wilks' theorem) on unbinned unfolded data unless the correlations are somehow accounted for. Instead, one can use techniques like the bootstrap to calibrate the confidence intervals and ensure correct coverage **[cite –KD]**. This naturally comes at a computational cost, but it is necessary to trust the results. In summary, the problem of correlated data in collider analyses is a critical consideration for precision measurements and ignoring these correlations can lead to bias in fitted values (if subtle), under– or over–estimated uncertainties, and confidence intervals or $p-$values that do not mean what we think they mean. As experiments push toward ever more differential and high–dimensional measurements (often aided by machine learning), developing sound statistical tools to handle event correlations will be essential to fully realize the potential of these data for testing the Standard Model and searching for new physics.

In conclusion, correlated data have profound impacts on both parameter extraction and theory tests in collider experiments. Any analysis that ignores the correlation structure in its data is at risk of drawing incorrect conclusions, whether that be an unrealistically precise measurement (uncertainties too small), a failure to detect a true deviation (uncertainties too large or biases introduced), or a claimed discrepancy with theory that is actually just a mis–modeled common uncertainty. The challenges posed by correlated data become increasingly acute as we move toward more granular unbinned measurements. Ensuring statistical procedures remain robust in the face of these correlations is a key task for the interplay of machine learning techniques and rigorous uncertainty quantification in high energy physics.

7.4 Uncertainty Quantification

In high energy physics analyses, quantifying uncertainties in measurements is as important as the central values themselves. As outlined in **??????**, real–world measurements often violate these independence assumptions. This section will establish a rigorous framework for

Table 7.1: Common sources of correlated data in proton-proton cross section measurements and their effects on parameter extraction. Proper modeling of these correlations is essential for unbiased fits, accurate uncertainty estimates, and correct confidence interval coverage.

Source	**Impact**
Resolution	True events migrate between bins, inducing statistical correlations in unfolded spectra. Inference must account for migration–induced covariance to avoid biasing shape–sensitive fits.
Pileup	Extra particles from overlapping collisions affect all events in a run. Residual uncertainties from pileup subtraction act as event–common noise. Yields across events (or bins) carry a correlated uncertainty, influencing global normalisation and shape.
Regularization	Couples bin estimates to each other. The unfolded bin counts are not independent random variables but linked by smoothing constraints, leading to significant off–diagonal covariance.
Systematics	Shared uncertainties (luminosity, calibration, efficiencies) move all data points coherently. They produce positive correlations across bins. Neglecting them can underestimate the uncertainty on overall normalization or shape shifts, causing miscoverage.

uncertainty quantification in the presence of such correlations, contrasting analytic asymptotic techniques with numerical resampling approaches. We will then discuss concrete guidance on implementing these methods, including the choice of replicas, computational cost trade–offs, and diagnostics to ensure correct coverage.

Analytic Approaches: Asymptotic Theory and Its Limitations

Traditional uncertainty estimates in HEP rely heavily on analytic results from classical asymptotic theory. Under the independent, identically distributed (i.i.d.) data assumption (see **??**), the maximum likelihood estimator (MLE) of a parameter θ is consistent and asymptotically normal. Formally, as the sample size $N \to \infty$, $\hat{\theta} \sim \mathcal{N}\big(\theta_{\text{true}},\ I^{-1}(\theta_{\text{true}})\big)$, where $I(\theta)$ is the Fisher information matrix**[cite –KD]**. In practical terms, one often obtains the covariance matrix of estimators by inverting the Hessian of the log likelihood at the optimum or via the observed information matrix. Likewise, Wilks' theorem guarantees that $-2\Delta \ln \mathcal{L}$ follows a χ^2 distribution in the $N \to \infty$ limit (with degrees of freedom equal to the difference in fit parameters) under regular conditions**[cite –KD]**. This underpins the common practice of deriving confidence intervals from profile likelihood scans. For instance,

the 1σ interval on a single parameter is given by the range $\Delta(-2\ln\mathcal{L}) = 1$, assuming the asymptotic χ^2_1 distribution**[cite –KD]**. These asymptotic formulas are analytically convenient and computationally cheap, requiring only the final fit result and local curvature information rather than additional dataset replicas.

However, the accuracy of asymptotic methods can deteriorate when their assumptions are violated, such as in scenarios with correlated or weighted events, model mis–specification, or parameter boundaries. In our context, if events are correlated (or effectively so, due to shared systematic effects), the simple χ^2 approximations may no longer hold. For example, a composite likelihood or mis–specified likelihood (where correlations among data points are ignored in the model) can yield a test statistic whose distribution deviates from χ^2, often requiring a mixture of χ^2 distributions or an effective scale factor**[cite –KD]**. In such cases, naive use of $\Delta\chi^2 = 1$ for 1σ intervals can significantly misestimate the true coverage.

Godambe Information (Sandwich Estimator)

When the model used for inference does not perfectly describe the data's correlation structure, one can resort to the sandwich covariance estimator to obtain valid uncertainties. The sandwich (or Godambe) information matrix**[cite –KD]** is given by

$$G(\hat{\theta}) \;=\; H(\hat{\theta})\, J(\hat{\theta})^{-1}\, H(\hat{\theta})\,, \tag{7.7}$$

where $H = -\mathbb{E}[\nabla^2 \ln\mathcal{L}]$ is the information function and $J = \mathrm{Var}(\nabla \ln\mathcal{L})$ is the variance of the score function. Intuitively, H^{-1} is the optimistic covariance estimate assuming the model is correct, while J captures the actual observed fluctuations of the score (which increase if data are correlated or the model is incorrect). The "sandwich" $J^{-1} = H^{-1}GH^{-1}$ then provides a robust covariance for $\hat{\theta}$ that remains consistent even if the likelihood is mis-specified (e.g. ignoring correlations)**[cite –KD]**. In the limit of i.i.d. data with correct model, $H = J$ and the sandwich reduces to H^{-1} as expected. In the presence of event–to–event correlations, $J > H$ (in a matrix sense), meaning the sandwich variance $H^{-1}JH^{-1}$ is larger than the naive H^{-1}, reflecting the loss of information from the unmodeled dependencies. This approach effectively generalizes the notion of "effective degrees of freedom" due to correlations. While the Godambe information is rarely computed explicitly in everyday HEP analyses, it underlies the validity of techniques employed in some composite likelihood fits**[cite –KD]**. It provides a formal path to include correlations analytically. J can be computed by evaluating the empirical covariance of score contributions event–by–event (accounting for any known shared systematics), then use it to adjust the reported errors. The challenge is that J is increasingly non–trivial to compute when the full joint distribution of data is complicated.

Wilks' Theorem Violations and Bartlett Corrections

Even when one can form a profile likelihood that accounts for some correlations via nuisance parameters, subtle deviations from asymptotic assumptions can persist. For instance, finite–sample bias in the likelihood ratio statistic can lead to systematic undercoverage (reported intervals are too narrow) or overcoverage (too wide). A classic remedy in statistical theory is the Bartlett correction, which rescales the test statistic to better match the χ^2 distribution**[cite –KD]**. In essence, one multiplies $-2\Delta \ln \mathcal{L}$ by a factor $c < 1$ (determined from lower–order asymptotic expansions or auxiliary simulation) such that $c \cdot (-2\Delta \ln \mathcal{L})$ has expectation equal to the nominal χ^2 degrees of freedom under H_0**[cite –KD]**. Bartlett corrections have been studied for improving likelihood ratio tests in small–sample and correlated–data situations in certain mixture models**[cite –KD]**). In practice, applying a Bartlett factor in HEP would require either analytical derivation for the specific model or an empirical calibration. One would have to fit a scale factor so that the distribution of $-2\Delta \ln \mathcal{L}$ from pseudo–experiments matches χ^2. While not routinely done in most collider measurements, this is one possible correction that can be applied when asymptotic results are suspect.

Another scenario of interest is the profile likelihood under model misspecification. If the true data generating process lies outside the assumed model family, the MLE will converge to the "closest" point in parameter space (the pseudo–true value**[cite –KD]**) but the usual confidence intervals might be misleading. The profile likelihood curve may be too steep or too shallow relative to the actual sampling distribution of $\hat{\theta}$. In such cases, one can again use the sandwich variance to adjust uncertainties, or else rely on a numerical calibration of the likelihood ratio. For example, one could generate pseudo–datasets from a more complete model (or from the data itself via resampling) and directly measure the distribution of the profile $\Delta \ln \mathcal{L}$ statistic**[cite –KD]**, using that to set confidence intervals rather than assuming a χ^2. This is essentially a hybrid approach. One is still using the likelihood ratio as the test statistic, but determining its cut-off values by simulation. This technique often employed in searches for new physics when parameters are near physical boundaries or when the approximate independence of data does not hold.

In summary, analytic asymptotic methods provide a powerful toolkit for uncertainty quantification under ideal conditions (large samples, independent data, correctly specified models). We have analytic handles like the Fisher information, which are used to propagate uncertainties, and the profile likelihood method, which has been a workhorse for collider measurements**[cite –KD]**. Yet, in the presence of strong correlations or non–standard conditions, these methods can lead to errors in inference parameters. As we discussed in **????**, ignoring an unfolding-induced covariance or a shared systematic can violate the regularity conditions of Wilks' theorem. The net effect can be either an underestimate or overestimate of uncertainties, depending on the nature of the correlation. In other words,

the actual uncertainty in the fitted parameters is different from what the naïve formula (which ignores those induced correlations) would predict. This is a clear indication that Wilks' theorem in its standard form is failing here. The take–home message is that one must either incorporate the correlation structure into the analytic calculation (via approaches like the sandwich estimator) or fall back on numerical methods to assess uncertainties. In the next subsection, we turn to the latter: bootstrapping and related resampling techniques that offer a practical, if computationally intensive, route to uncertainty quantification when analytic formulas become unreliable.

Numerical Resampling Approaches: Bootstrapping & Toy Monte Carlo

Numerical uncertainty estimation techniques generate many "fake" realizations of the experiment to directly empirically measure the spread of an estimator or unfolded distribution. These methods do not rely on an explicit formula for the variance; instead, they approximate the sampling distribution by Monte Carlo simulation or resampling. Two broad classes are widely used: non–parametric bootstrapping (resampling from the observed data itself) and parametric bootstrapping (also known as toy Monte Carlo pseudo–experiments, sampling from a fitted model). Both have become indispensable in modern HEP analyses, especially when dealing with complex datasets and high–dimensional outputs where analytic propagation is intractable. This section will review these approaches in turn, including variations relevant to unbinned inference with correlations, and then discuss practical guidelines for their use.

Non-Parametric Bootstrap Resampling

The classical (non-parametric) bootstrap**[cite –KD]** is a generic tool to estimate the uncertainty of virtually any statistic by resampling the data. Given an original dataset of N events $\{x_1, \ldots, x_N\}$, one generates a bootstrap replica by sampling N points with replacement from the original set. This procedure randomly selects events such that some original events may appear multiple times in the replica while others may be omitted, effectively drawing from the empirical distribution of the data. By repeating this many times (say B replicas), one can simulate the variability of any statistic: the distribution of the statistic across the B replicas serves as an approximation to its true sampling distribution. In context of an unfolding or cross–section measurement, the "estimator" might be the unfolded spectrum itself or, more often, a physics parameter extracted in a subsequent fit. For example, **??** employed bootstrap resampling to propagate statistical uncertainty in the Moment Unfolding method. We generated N bootstrap datasets by resampling events and ran the full unfolding on each, then took the standard deviation of the unfolded moments

across replicas as the uncertainty. This ensured that fluctuations in both the measured data and the simulation–based correction were reflected in the final error bands.

One advantage of the non–parametric bootstrap is that it makes minimal assumptions about the underlying distribution. One does not need an analytic model of how data are distributed, only the empirically observed sample. This is particularly useful in high–dimensional or highly non–Gaussian situations common in HEP (for instance, when the outcome of an analysis is a complicated function of many event properties and detector effects). The bootstrap will faithfully capture the variance as long as the original dataset is representative of the true distribution. It also naturally incorporates correlations among observables. If one is interested in the covariance between two measured quantities (say, two bins of an unfolded histogram or two separate measurements using the same data), the bootstrap ensemble can estimate that. Each bootstrap replica is like a pseudo–experiment drawn under the null hypothesis that "the observed dataset is the true underlying distribution". By analysing all replicas identically, one can extract not just variances for each observable but the full covariance matrix $\mathrm{Cov}(O_i, O_j)$ via the sample covariance of the replica results.**[cite https://cds.cern.ch/record/2759945 –KD]** For example, the ATLAS Collaboration demonstrated that bootstrapping provides a straightforward way to estimate statistical correlations between bins of an unfolded cross–section, or even between two separate analyses that share data.**[cite ATL-PHYS-PUB-2021-011 –KD]** Because each event is resampled with a deterministic seed, the same random fluctuations can be propagated coherently into multiple analysis outcomes, allowing an estimate of their correlation**[cite –KD]**. This is a significant benefit over analytic approaches, where deriving the covariance between two complex observables might be extremely cumbersome.

Despite its generality, the non–parametric bootstrap must be applied with care in the presence of correlated events. The standard bootstrap assumes each observation is an independent draw from some distribution. If the true data have correlations (e.g. an event consists of multiple particles' measurements, or there are clusters of events that fluctuate together due to a common systematic), a naïve bootstrap that samples individual data points independently will destroy the correlation structure. In such cases, one should resample at the level of the correlated unit. For instance, if each collision event produces multiple jets whose properties are analysed (hence jets from the same event are correlated), the bootstrap should treat the entire event as the fundamental unit to resample. This is known as a cluster bootstrap or block bootstrap**[cite –KD]**. With sufficient computational resources available, one could resample whole events (retaining all their associated jets) with replacement to build each replica dataset, thus preserving intra–event correlations. Similarly, if data have an inherent time or spatial correlation, one might sample temporal or spatial blocks of consecutive events. By doing so, the bootstrap replicas reflect the dependency structure present in the original data. Failing to do this can lead to underestimated uncertainties because the

resampled replicas would appear overly variable or overly independent compared to reality, giving a misleading impression of more information than actually available.

Another subtlety to be aware of is that the non–parametric bootstrap conditions on the observed sample size N. Each replica has exactly N events (though not N unique events). However, in many HEP measurements, N itself is a Poissonian quantity (e.g., the number of events passing selection follows Poisson statistics with mean proportional to the true cross section and integrated luminosity). By fixing N, the bootstrap neglects the contribution of counting uncertainty. For very large N, this distinction is negligible (Poisson(λ) fluctuations are $\sqrt{\lambda}$, which is small relative to λ). But for small datasets or when quoting total rate uncertainties, this matters. A simple remedy is the so–called Poisson bootstrap. Instead of drawing exactly N samples each time, each original event is assigned a Poisson(1) random weight in each replica**[cite –KD]**. This means we effectively decide for each event how many times it gets included $(0, 1, 2, \dots)$ by a Poisson draw with mean 1. The expected total count in a bootstrap is then also N (since $\sum_{i=1}^{N} \text{Poisson}_i(1) \sim \text{Poisson}(N)$), but it now fluctuates around N. This method has been recommended in HEP applications**[cite –KD]** because it mirrors the way a real experiment could have produced a few more or fewer events. The Poisson bootstrap has the additional advantage of simplifying code. One does not need to sample events one by one; instead, once can generate N Poisson random numbers and replicate event i that many times. The ATLAS analysis note**[cite ATL-PHYS-PUB-2021-011 –KD]** implements exactly this approach (with a fixed random seed per event to ensure reproducibility), producing thousands of Poisson–weighted replicas to evaluate both per bin uncertainties and bin–to–bin correlations for unfolded spectra. Importantly, this approach implicitly treats the entire collection of events as coming from a Poisson process, which is appropriate for counts of collisions.

For unbinned unfolded data, where the result of an unfolding procedure is a continuous distribution or a set of weighted events at “truth level", the bootstrap can be applied by resampling at the detector level before unfolding. In other words, one can generate bootstrap variations of the raw data (and even of the simulation used in the unfolding, if that has statistical uncertainty) and run the unfolding on each variation. This yields an ensemble of unfolded results, from which uncertainties can be derived. This is the strategy we followed in this work for evaluating unfolding uncertainties. we took the same detector–level data and fluctuated it many times (via resampling counts in each bin), running the unfolding algorithm each time and observing the spread in the unfolded solution. For unbinned methods, output could be, for example, a weight for each simulated truth event. Then, applying the procedure to bootstrapped detector samples yields many sets of weights, whose variance indicates the uncertainty on the underlying truth distribution. Because these modern methods lack simple closed–form error propagation, such a bootstrap approach is often the only viable way to quantify uncertainties on the unfolded distribution. One must ensure, of course, that the computational cost is manageable. Unbinned unfolding

(especially with neural networks) can be expensive, so the number of bootstrap replicas B may be limited by available compute. We will address this balance shortly in the practical guidelines.

Parametric Bootstrap (Toy Monte Carlo Simulations)

The parametric bootstrap differs in that it assumes one has a generative model for the data. Instead of resampling the observed events, one simulates new datasets from a known or fitted probability distribution. In particle physics, this is synonymous with conducting many toy Monte Carlo pseudo–experiments. The procedure is

1. Take the known model as an estimate of reality
2. Repeatedly draw “fake" experiments from this model (using Monte Carlo event generators or analytical distributions as applicable)
3. For each fake dataset, run the full analysis chain (unfolding, fitting, etc.) to obtain an outcome
4. Use the spread of outcomes to infer uncertainty.

This approach is inherently model dependent—it relies on the assumption that our fitted model is a good representation of the underlying truth. However, in many cases (especially when the goal is to test consistency with that model or to quote uncertainties assuming the model), this is perfectly acceptable. In fact, it is often the only way to incorporate known physics processes and detector effects exactly as they enter the analysis.

One benefit of the parametric bootstrap is the ease of incorporating known systematic and correlation effects. Because we are generating data, we can build in whatever correlations we expect. For example, if two variables in each event have a known correlation, our simulation can sample them jointly rather than independently. If there is a nuisance parameter (like an overall efficiency or energy scale) with some uncertainty, we can randomly vary that parameter for each pseudo–experiment (drawn from its uncertainty prior) so that the ensemble of pseudo–datasets reflects our lack of knowledge of it. This way, the resulting spread of outcomes includes the effect of that systematic.[2] Similarly, parametric generation naturally handles Poisson fluctuations in event counts. Each pseudo–experiment can draw $N_{\text{events}} \sim \text{Poisson}(\langle N \rangle)$ where $\langle N \rangle$ is the expected number given the cross section and luminosity. Thus, unlike the fixed–size non–parametric bootstrap, the parametric approach will inherently include the statistical uncertainty from the finite event count as well.

[2]This is analogous to the commonly used “toy Monte Carlo" approach of varying nuisance parameters within their errors to see the impact on a fit result. Here we integrate it into the generation of each bootstrap.

In HEP, the parametric bootstrap underlies many standard practices. For instance, the CL_s method for setting limits on new physics involves generating many pseudo–datasets under the background–only and signal+background hypotheses to determine the distribution of a test statistic**[cite –KD]**. Similarly, when experiments quote an unfolding covariance matrix from "toy MC," they have typically taken the central unfolded result as truth, re-simulated many fake datasets through detector simulation, unfolded each, and computed the covariance of the ensemble. This can be interpreted as a parametric bootstrap of the unfolding result. In **??**, we discussed how ignoring certain correlations can undermine analytic formulas; the parametric bootstrap gives a robust alternative because it does not assume those formulas. Instead it measures the uncertainty by brute force. For example, if one is unsure about the validity of Wilks' theorem in a complex fit, one can generate toy experiments at the best–fit parameters and validate empirically what fraction of the time $\Delta \ln \mathcal{L}$ exceeds a certain value.

The downside of bootstrapping is the heavy computational load and potential model bias. Generating and analysing B pseudo–experiments can be extremely compute intensive. In practice, one often uses a fast approximation (such as a parameterized smearing or a fast detector simulation) to make this feasible. The ATLAS bootstrap note**[cite ATL-PHYS-PUB-2021-011 –KD]** suggests that on the order of $B = 1000$ replicas is a reasonable compromise in many cases, and even used $B = 10000$ for certain jet cross–section covariance evaluations. Those numbers are only intended to be illustrative. The exact choice of B should be chosen by monitoring when the statistical error on the uncertainty estimate itself becomes negligible (see **??**). As for model bias, if the true data–generating process differs from the model used for toys, the bootstrap will estimate the variance around the wrong central value. This is usually acceptable when we are interested in relative uncertainties or when the model has been tuned to data (e.g., using the unfolded result itself as the truth baseline). Nonetheless, it is wise to cross check. One can, for example, perform both a non–parametric and parametric bootstrap and ensure they yield compatible uncertainty estimates. Discrepancies might indicate sensitivity to modelling.

This section would be incomplete without noting a subtle variant: the hybrid bootstrap. In some cases, one might resample certain aspects from data and others from a model. For instance, one could bootstrap the residuals of a fit rather than raw data, which is common in regression contexts (to mimic new noise samples), though less common in HEP. Or one might resample events but also randomly fluctuate a global parameter for each replica. These approaches blur the line between non–parametric and parametric and are tailored to specific correlation structures.

Practical Considerations and Guidelines

While bootstrapping and toy MC methods are conceptually straightforward, their successful deployment in a HEP analysis requires careful planning. Here are some high level practical guidelines and considerations for using these techniques to quantify uncertainties.

Number of Replicas (B) The accuracy of bootstrap estimates improves with the number of replicas, but computational cost grows linearly. In practice, $B \sim \mathcal{O}(10^2 - 10^3)$ is common. A few hundred replicas may suffice for stable estimates of variances, but for reliable estimation of the full covariance matrix (especially in high dimensions) or tails of distributions, one might need even an order of magnitude more. It is good practice to check convergence of the uncertainty estimate: e.g. run $B = 200, 400, 800$ and monitor if the reported uncertainty (or key figure of merit) changes appreciably. If it stabilizes, B is sufficient. The standard error on a bootstrap-derived standard deviation is $\frac{\sigma}{\sqrt{2(B-1)}}$, so diminishing returns kick in as B grows.

Computational Cost and Parallelization Each bootstrap replica is an independent analysis of a fake dataset, and so parallelization can offer overwhelming computational gains. One should exploit parallel computing [3] to run many bootstraps concurrently. In a modern analysis framework, it is often feasible to distribute $B = 1000$ jobs to a cluster and retrieve results in a few hours, whereas running them serially would be prohibitive. If the analysis involves training a machine learning model (e.g. a neural network for unbinned unfolding) for each replica, one might reduce B or use a pre–trained model across replicas if appropriate[4]. An alternative to reduce cost is the bootstrapping of approximate models. For example, one could fit a faster surrogate model, like a parametric function, to each replica instead of rerunning a full simulation or complex inference. This sacrifices some fidelity but can massively speed up the procedure.

Diagnostics for Coverage and Reliability After obtaining uncertainty estimates (whether analytic or bootstrap), it is crucial to validate that they have the correct coverage. Coverage means that a nominal 68% confidence interval indeed contains the true value about 68% of the time in repeated experiments. Often we can leverage simulation to perform a coverage study. This involves generating many mock datasets from a known truth and seeing how often our method's interval would capture that truth. Such studies can be done with the parametric bootstrap (treating the truth as known). If the intervals under–cover

[3] e.g. multithreading

[4] bearing in mind that if the model training itself is subject to statistical fluctuation, that adds another layer of variance

or over–cover significantly, one might adjust the method[5]. Additionally, within a single dataset, one diagnostic is to compare different methods. If bootstrap intervals are significantly different from asymptotic ones, it flags a potential issue with the latter. Conversely, agreement between them builds confidence that uncertainties are well understood; discrepancies require investigation[6]. In the next few sections, we will see examples of this where the asymptotic formulae give systematically different errors than the bootstrap spread, indicating mis–coverage. The recommendation then, echoing recent studies**[INSPIRE VanDenBroeck:2006qi –KD]**, is to trust the numerical approach or augmented analytic approach that accounts for correlations.

Incorporating Systematic Uncertainties So far, we discussed statistical uncertainty from limited data samples. In a full measurement, one must also account for systematic uncertainties [7]. Bootstrapping can be extended to some systematic effects. One can, in principle, treat a systematic variation as producing an alternate "truth" and then bootstrap around that. However, more common in HEP is to evaluate systematics separately (e.g., by shifting calibrations and re–running the analysis a few times) and then combine statistical and systematic errors. One must be careful not to mix these in the same replicas unless the systematic is itself statistical in nature (for example, the statistical error on a background estimation can be propagated by bootstrapping the background sample generation**[cite –KD]**). In **??**, we saw an approach where statistical and systematic uncertainties were handled in parallel; statistical via bootstrap, systematics via repeated trials with varied inputs. The overarching principle is to match the uncertainty evaluation method to the source of uncertainty, i.e. to use bootstraps or toys for random fluctuations, and to treat systematic shifts by exploring the parameter space of uncertainties (possibly with their own pseudo–experiments if they have uncertainty distributions).

Reporting and Using Bootstrap Results When using bootstrap ensembles, one often obtains a covariance matrix for multiple observables or parameters as the primary output. This covariance can be used in downstream fits. E.g., treating the unfolded spectrum and its bootstrap covariance as "data" for a theory fit. It is important to regularize or smooth the covariance if B is not extremely large, because being an empirical estimate it can be noisy, and potentially not even positive definite if B is too low relative to number of bins. Simple techniques include increasing B or applying shrinkage to the covariance matrix. Showing consistency between the diagonal errors from bootstrap vs. analytic, and quoting the correlation coefficients if relevant, can be very informative in publications. Tables or

[5] e.g., use a different statistic, or apply a Bartlett–like scale factor

[6] they could indicate anything from a bug in analysis to a breakdown of assumptions

[7] from detector calibration, theory model choices, etc.

matrices of bootstrap correlations might be provided in an Appendix or auxiliary material if they are of interest.

Common Pitfalls Finally, we should discuss a few potential pitfalls.

- **Bootstrap bias:** The bootstrap can sometimes reveal bias in an estimator, e.g., the average of bootstrap estimates differs from the original estimate. If significant, one might use the bootstrap to correct the bias (bias-corrected estimates) or employ the BCa (bias-corrected and accelerated) percentile method for confidence intervals**[cite –KD]**.
- **Small sample issues:** In regions with extremely few events, bootstrap samples may not be representative because one keeps drawing the same few events in different orders. In such regimes, exact methods or simple conservative analytic bounds might be more reliable.
- **Randomness:** One should ensure reproducibility by fixing the random number seeds for bootstrapping procedures. This allows others to regenerate the same replicas and verify results.
- **Interpreting bootstrap outcomes:** One should remember that the bootstrap gives an estimate of uncertainty, not a guarantee. If the data are very sparse or the model very wrong, the bootstrap will dutifully reflect those. Results should always be interpreted in the physical context and cross–checked with sanity checks.

In conclusion, bootstrapping versus asymptotics is not an either–or choice but a complementary set of tools. Asymptotic techniques are fast and illuminating; they tell us how uncertainty scales with data size and often give insight into which parameters or features dominate the error. Numerical techniques, on the other hand, are robust and account for real–world complexities that elude analytic treatment. For cutting–edge analyses a balanced approach might be most appropriate, where one uses analytic formulae as a first pass and consistency check, but relies on bootstrapping or pseudo–experiments for the final uncertainty quantification when data correlations or algorithmic complexities are significant. This strategy ensures that confidence intervals are trustworthy. Indeed, using numerical techniques can turn up hidden uncertainty contributions that asymptotic formulae would have glossed over. In the next sections, we will apply this framework to concrete case studies, comparing binned vs. unbinned approaches and demonstrating how the choice of uncertainty quantification technique impacts the physics conclusions.

7.5 Case Studies

Setup

We define a “truth" particle–level distribution as a Gaussian with mean $\mu_{true} = 0.2$ and variance $\sigma^2_{true} = 0.81$. We draw $N_{\text{true}} = 10^4$ events from this distribution to serve as our toy dataset at the particle–level. In addition, a larger Monte Carlo (MC) sample of 10^5 events is generated from a prior Gaussian with mean 0.0 and variance 1.0. This MC sample plays two roles—it provides an initial guess of the truth distribution for unfolding, and it is used to derive the detector response. Notably, the MC's particle–level distribution differs from the true distribution, mimicking the realistic situation in which the nominal simulation does not perfectly match reality.

Both the true dataset and the MC sample are passed through a detector response model. We simulate detector-level measurements by smearing each particle–level event with an independent Gaussian error of mean 0 and a certain resolution σ_{det} (representing finite detector resolution). We will examine multiple smearing levels, with σ_{det} ranging from 0, no smearing, i.e. a perfect detector, up to 0.75, substantial resolution degradation. The smeared true sample represents the detector–level data that an experiment would observe, while the smeared MC sample represents the detector–level simulation.

Fully Binned Baseline

To establish a baseline, we first consider this simple one–dimensional unfolding scenario using a known Gaussian distribution and the traditional binned unfolding approach. In this fully binned context, we will generate Gaussian particle–level and detector–level data, apply Iterative Bayesian Unfolding (IBU)**[cite –KD]**, and perform a standard binned parameter fit to the unfolded histogram. The goal is to characterize the bias, variance, and confidence interval coverage of the inferred Gaussian parameters under ideal conditions, before proceeding to more advanced unbinned methods in subsequent sections. We intentionally use a Gaussian example with known truth parameters so that any bias can be directly evaluated and so that many pseudo–experiments can be quickly generated to study statistical fluctuations. This demonstration will highlight the importance of properly accounting for the covariance matrix of the unfolded result when making inferences on physics parameters. We generate 500 independent pseudo–datasets from the truth/data model, each with a Poisson fluctuation in total event count (mean of 10^4 events per dataset) to simulate realistic statistical scatter.

Methodology

We bin all datasets using a fixed histogram binning (for this example, 15 uniform bins spanning the range $[-3, 4]$). The choice of binning is kept constant for the unfolding procedure and subsequent analysis. An illustration of the distributions at truth level and detector level is shown in Fig. **??**, including a visualization of the smearing kernel (resolution function).

Using the large MC sample, we construct a detector response matrix mapping particle–level bins to detector–level bins. Each element R_{ij} of this matrix gives the probability for an event originating in truth bin i to be reconstructed in detector bin j. In practice, R_{ij} is obtained by binning the MC events by their true and smeared values. The response matrix, along with the smeared data histogram, serves as input to the unfolding algorithm.[8]

We then apply the IBU algorithm**[cite –KD]** to the binned detector–level data in order to unfold the effects of smearing. Starting from the prior MC truth distribution, IBU iteratively updates the estimate of the truth histogram by comparing the MC and data in the detector space and applying Bayes' theorem to reweight contributions back to truth space. We iterate until convergence, obtaining an unfolded truth distribution for the data. This unfolded result consists of bin counts (or densities) for the truth histogram, along with an estimated covariance matrix for those bin counts. The bin-to-bin covariances arise from the finite statistics of the data and from the smearing correlations introduced by the unfolding procedure. In this study, we estimate the covariance matrix of the unfolded histogram by repeating the unfolding on many statistically independent toy datasets, described below, and via bootstrapping; both methods were checked to give consistent results for the uncertainties.

After unfolding, we perform a binned fit to extract the parameters of the underlying distribution (in this case, the Gaussian's mean μ and variance σ^2). For each unfolded histogram, a χ^2 minimization is used to fit a Gaussian model to the unfolded data. Importantly, the theoretical Gaussian model is integrated over each bin to yield the expected content in that bin for given values of μ and σ^2, so that the comparison between the model and the unfolded histogram is exact with no interpolation or bin-centre approximation. The fit's χ^2 is computed using the full covariance matrix of the unfolded bins, thereby incorporating all statistical correlations between bins. The outcome of the fit is a pair of best–fit parameters $(\hat{\mu}, \hat{\sigma}^2)$ along with their asymptotic uncertainty estimates (one standard deviation errors) derived from the curvature of the χ^2 at the minimum. For comparison, we also consider a *diagonal-covariance* fit, repeating the same procedure but using only the diagonal elements of the covariance, functionally treating unfolded bin counts as if they were independent. This allows us to see the impact of neglecting inter–bin correlations on the inferred uncertainties.

To evaluate the bias, variance, and confidence interval coverage of this procedure, we repeat the entire analysis for each pseudo–experiment.

[8] In Bayesian terms, the unsmeared MC histogram also provides a prior estimate for the truth distribution.

This ensemble provides a distribution of fitted parameter values $\hat{\mu}$ and $\hat{\sigma}^2$ from which we can quantify the bias, which is the difference between the average fitted value and the true value, the variance or spread of the estimates (related to the expected statistical uncertainty), and the empirical coverage of confidence intervals. In practice, one could alternatively use bootstrapping on a single dataset to assess these metrics; indeed, we have checked that bootstrapped replicas yield consistent uncertainty estimates as the independent toys, confirming that our ensemble size is sufficient.

Using the above procedure, we obtain an unfolded result and fit for each pseudo–experiment, and we can now examine the bias and variance of the estimates. The analysis is repeated for each value of the detector resolution $\sigma_{\text{det}} \in [0, 0.75]$ to see how worsening detector effects impact the inference. Figure **??** and Fig.**??** summarize the accuracy or bias of the method, showing the mean fitted μ and σ^2 across 500 trials as a function of the detector smearing. Figure**??** focuses on the precision and uncertainty coverage, comparing the nominal 1σ errors from the fits to the actual spread of the results. These results demonstrate that the IBU unfolding followed by a proper binned fit recovers the true distribution parameters without significant bias. For all tested smearing levels, the average fitted mean $\langle\hat{\mu}\rangle$ remains within statistical uncertainty of the true value, 0.2, and the average fitted variance $\langle\hat{\sigma}^2\rangle$ within statistical uncertainty of 0.81. In Fig.**??** and Fig.**??**, the data points (averaged best-fit values) lie on or around the horizontal red dashed lines marking the true parameters. The deviations are well within the statistical error bars[9], indicating no significant bias in the unfolding or fitting procedure. Notably, the choice of using the full covariance versus only diagonal uncertainties in the fit has no effect on the central values obtained. Both approaches yield correct $\hat{\mu}$ and $\hat{\sigma}^2$, which might be explained by the bias in this context being dominated by any unfolding imperfections and IBU, with sufficient iterations, being an asymptotically unbiased estimator of the truth distribution.

When using the full covariance matrix in the χ^2 fit, the asymptotic uncertainty estimates for $\hat{\mu}$ and $\hat{\sigma}^2$ are found to be in excellent agreement with the actual distribution of fit results across the ensemble. In Fig. **??**, the green circle markers show the average 1σ uncertainties from the fits (i.e. the fit errors from the covariance matrix of each fit) as a function of detector resolution. These are virtually indistinguishable from the green star markers, which indicate the empirical RMS standard deviation of the 500 fitted values at each resolution. In other words, the fit's error estimates accurately predict the trial–to–trial fluctuations of the outcomes. This agreement implies that the reported 68% confidence intervals have the correct coverage: approximately 68% of the pseudo–experiments' $\hat{\mu}$ (or $\hat{\sigma}^2$) results lie within $\pm 1\sigma$ of the true value, as expected for well–calibrated uncertainties. As the detector resolution worsens, the uncertainty on the inferred parameters grows, reflecting the loss of

[9] the standard error of the mean over the 500 trials

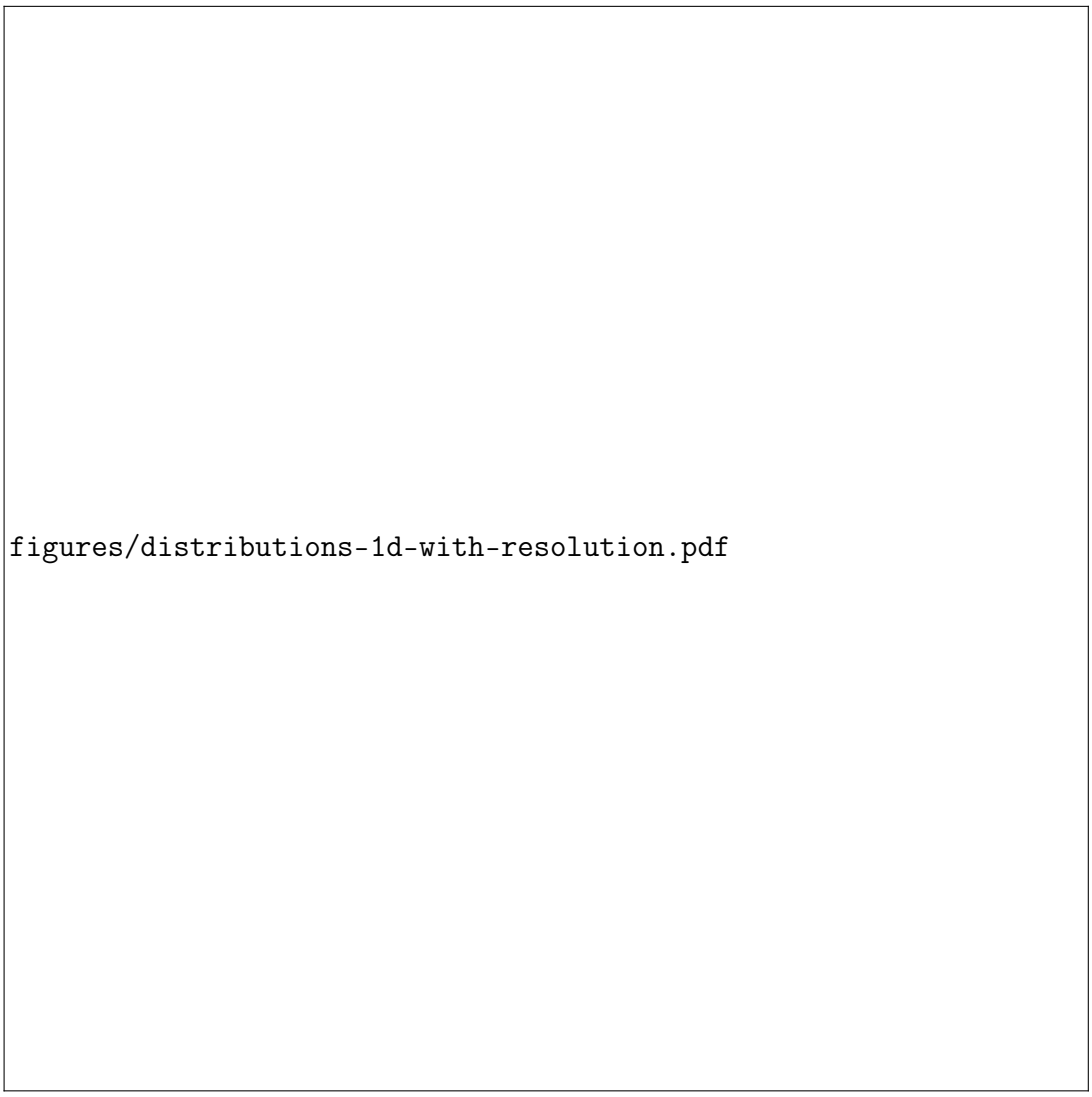


Figure 7.1: Histograms showing the datasets from the one–dimensional Gaussian study. The distributions on the left show the true distribution of the observable compared with the generation distribution. The centre panel shows the data distribution of the observable compared with the simulation. The right histogram shows the resolution function (Gaussian smearing kernel). For visualization, 31 bins are used in the range $[-5, 5]$.

information, but at each point, the full–covariance χ^2 fit's uncertainty remains an accurate representation of the actual variance in the results.

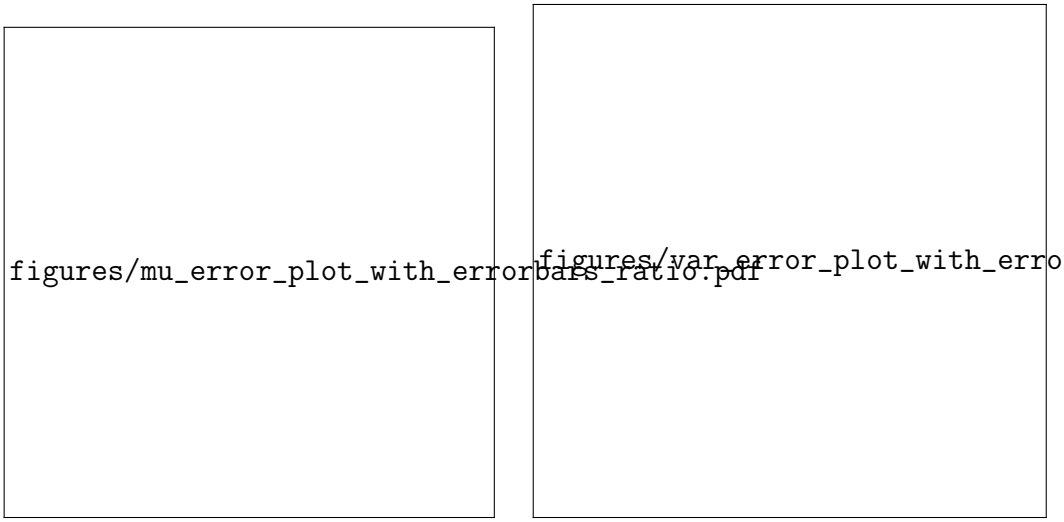
If one ignores the off–diagonal elements of the unfolded covariance matrix and assumes the bins are independent, the inferred uncertainties become misestimated. In this example, neglecting the negative bin–to–bin correlations from unfolding leads to an overestimation of the parameter errors. The pink square markers in Fig. **??** show the average fit uncertainty on μ and σ^2 obtained when using only the diagonal uncertainties. These are significantly larger than both the true ensemble spread (green stars) and the full–covariance errors (green circles) once any non-zero smearing is present. At the highest smearing tested, $\sigma_{\text{det}} = 0.75$, the diagonal–fit uncertainty is roughly $\sim 20\%$ higher than the actual RMS of the results. This overly conservative error estimate would lead to over coverage of confidence intervals. The nominal 68% interval contains the true value substantially more than 68% of the time). This indicates an inefficient use of information. The fit is effectively double–counting fluctuations in each bin that in reality are anti–correlated with fluctuations in other bins. These findings reinforce that incorporating the full covariance matrix from the unfolding is essential to obtaining reliable and correctly sized confidence intervals in the fully binned approach, and suggests that this may be the case in unbinned approaches too.

In summary, this study in the fully binned regime demonstrates that a conventional unfolding approach (IBU) combined with rigorous statistical treatment produces unbiased and well–calibrated inference of physics parameters. The unfolded results, when analysed with their full covariance matrix, give parameter estimates whose uncertainties accurately reflect the true spread across experiments, ensuring correct confidence interval coverage. We also see that simplifying assumptions like treating unfolded bins as independent can skew the uncertainty estimates, in this case, making them overly conservative, even though the point estimates remain correct. These findings provide a critical baseline for comparison with the unbinned methods discussed in the next sections. In the following section, we will investigate how unbinned unfolding techniques handle event correlations and whether they can achieve a similar level of statistical reliability as this fully binned approach. The lessons learned here, particularly the necessity of accounting for induced correlations, will carry forward as we transition to unbinned inference on correlated data.

Correlation Diagnostics after Unbinned Unfolding

Before confronting any inference procedure with an unbinned output, we must first quantify the statistical dependencies that the unfolding induces. This subsection introduces two complementary diagnostics that make those correlations both visible and quantifiable.

1. the **pairwise weight—distance correlation** $\rho(|x_i - x_j|)$ between all pairs unfolded events i and j, and



(a) (b)

Figure 7.2: Mean asymptotic 1σ error versus detector smearing for the Gaussian (a) mean μ and (b) variance σ^2. Results are shown for fits using the full covariance matrix (green circle markers) and using only the diagonal elements of the covariance (pink square markers). The green stars represent the "true" uncertainty determined by the standard deviation of the 500 fitted values from pseudo-experiments. Observe that the asymptotic errors from the full–covariance fits agree very well with the ensemble spread (green circles overlapping green stars), while the diagonal–only approximation consistently overestimates the uncertainties, especially at larger smearing values (pink squares lie above the green stars).

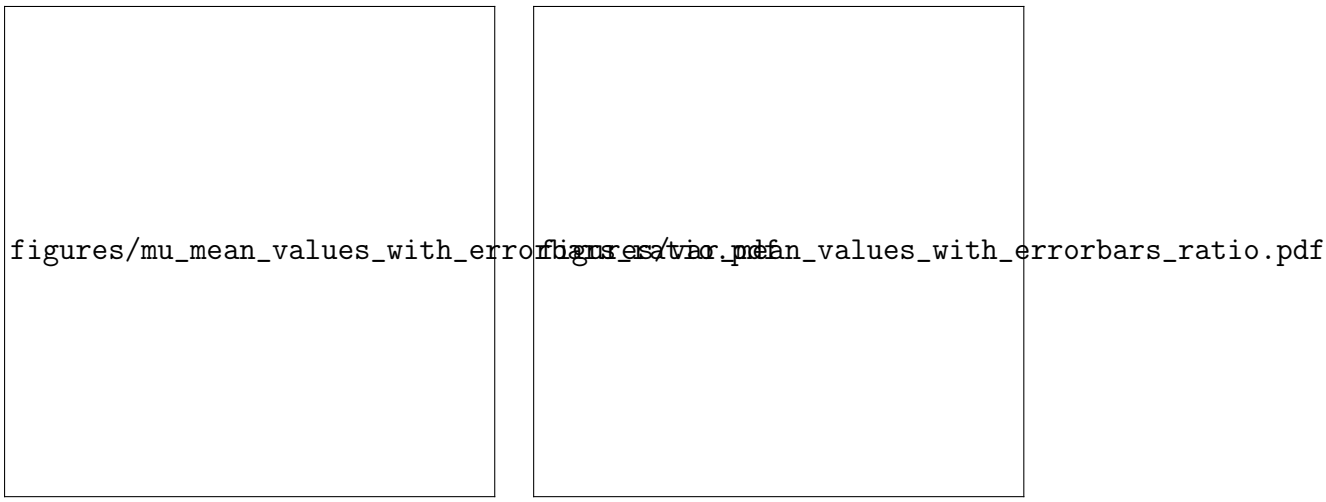
2. the **bin–bin covariance matrix** $C_{ab} = \mathrm{Cov}[\,\boldsymbol{\nu}_a, \boldsymbol{\nu}_b\,]$ of a fine histogram binned after unfolding.

Both are evaluated for four detector resolutions $\sigma_{\mathrm{det}} \in \{0, 0.25, 0.50, 0.75\}$ and for two distinct estimators, KDE and neural network (NN) conditional density estimators. The results are displayed in **??** and **??**, respectively, and form the empirical basis for the coverage study in **??**.

Pairwise weight—distance correlations

For every event, by considering the weight distributions across pseudo—experiments one can compute the correlation coefficient, defined as

$$\rho_{ij} = \frac{\mathrm{Cov}(w_i, w_j)}{\sigma_{w_i}\,\sigma_{w_j}}] \tag{7.8}$$



(a) (b)

Figure 7.3: The mean best-fit value of the Gaussian (a) μ and (b) σ^2 as a function of the detector smearing. Each point represents the average fitted value over 500 pseudo-experiments, with error bars showing the standard error on the mean (SEM). The horizontal dashed red lines indicate the true parameter values ($\mu_{true} = 0.2$ and $\sigma^2_{true} = 0.81$). Both fitting methods, using the full unfolded covariance (green circles) and using only diagonal uncertainties (pink squares), yield valid central values, demonstrating that the unfolding procedure is unbiased.

One can then average ρ_{ij} in bins of $\Delta x = |x_i - x_j|$, where x is the unfolded observable.[10] A few different features of the resulting curves (upper panels for KDE, lower for NN) are worth nothing.

- *Perfect detector ($\sigma_{\text{det}} = 0$):* One should expect the average pair-–wise correlation to be $\rho(|\Delta x|) \simeq 0$ for all separations $\Delta x > 0$, confirming that when no smearing is applied, OmniFold reproduces the factorised likelihood limit in which event weights are statistically independent**[cite:Andreassen:2019cjw –KD]**. At $\Delta x = 0$ we trivially should expect $\rho(0) = 1$ because every event is, of course, perfectly correlated with itself, and this is indeed what **??** demonstrates too. In the truly ideal i.i.d. case, one would expect the fall off from $\rho(0) = 1$ to $\rho(|\Delta x| > 0) = 0$ to resemble a Dirac δ at the origin. The KDE estimator indeed approaches this limit. Its kernel bandwidth

[10] The average is taken after combining results from 500 pseudo–experiments; error bars in **??** denote the ensemble RMS of ρ_{ij} in each Δx bin, thereby incorporating both the statistical fluctuation correlations and any run–to–run variability of the unfolding.

determines a correlation length $\ell \lesssim 0.15$.**[cite:Wasserman2006, WandJones1995 –KD]**. By contrast, even with extensive experimentation with architecture and hyperparameter tuning we find that NN priors leave a residual plateau $\rho \approx 0.75$ around $\ell \approx 1$, which then leads to the damped oscillatory structure of the correlation curve. One could speculate about reasons why unfolding with neural networks even in the perfect detector resolution case should lead to correlated weights. I propose two potential explanations.

1. *Network smoothness:* the shared weights in the neural network classifier impose a finite "receptive field, **[cite:Minderer2021SmoothNN,WeightConditioning2024 –KD]** causing nearby events to receive similar gradient updates and hence correlated weights; and
2. *Global normalisation:* OmniFold enforces $\sum_i w_i = N_{\text{MC}}$ at every iteration. This normalization might be introducing a weak positive correlation of size $1/N$ even in the absence of detector effects**[cite:Cowan2011Stats –KD]**.

Although numerically tiny, this effect, whatever its underlying explanation sets the optimistic lower bound on variance reduction when working with neural networks. Thus even the idealised zero–smearing case does not yield perfectly independent weights in practice, a useful caution when quoting statistical uncertainties from inference using ML processed data.

- *Mild smearing ($\sigma_{\text{det}} = 0.25$):* Short–range correlations of order $\rho \gtrsim 0.5$ appear for $\Delta x \lesssim 0.4$. These are already sufficient to reduce the effective sample size

$$N_{\text{eff}} = N / \big[\, 1 + (N-1)\rho \big] \tag{7.9}$$

[cite –KD] considerably. $\rho < 0$ for $\Delta \in [1, 2]$, as expected. For both the NN and the KDE, ρ exhibits the same damped oscillatory structure, as would be expected by the normalization imposed by the unfolding. ℓ_{KDE} is noticeably smaller that ℓ_{NN}. The contrast is less pronounced than in the perfect-resolution case, yet still visible.

- *Moderate and large smearing ($\sigma_{\text{det}} \geq 0.5$):* Correlations develop a broad plateau independent of distance. In that regime N_{eff} collapses explaining why a naïve fit ignoring correlations would significantly misestimate errors.

The NN generally yields more long range, but slightly weaker correlations than the KDE at the same resolution, suggesting that a higher–capacity estimator might be able to better absorb detector noise. However, the qualitative behaviour is identical; any non–zero smearing produces, long-range weight correlations that violate the i.i.d. assumption built into standard unbinned likelihoods**[cite –KD]**.

Histogram covariance matrices.

Since most downstream analyses eventually bin the unfolded events, a histogram covariance matrix can be effective to visualize the correlations. To visualise the impact on such *binned* summaries we fill a 40-bin histogram in the range $[-4, 4]$ for each pseudo–experiment and compute

$$C_{ab} = \big\langle (N_a - \bar{N}_a)(N_b - \bar{N}_b) \big\rangle. \tag{7.10}$$

The matrices, shown in Fig. **[fig:hist-cov –KD]**, reinforce the observations from the correlation curves. Even at zero smearing, while KDE produces a nearly diagonal C_{ab}, OmniFold's weights exhibit small but noticable off diagonal elements. Growing the smearing leads to pronounced *anti–correlations* along the first off—diagonal and coherent positive correlations far from the diagonal. For $\sigma_{\text{det}} = 0.75$ the largest off–diagonal coefficient reaches $|\rho_{ab}| \simeq 0.5$, echoing the plateau seen in $\rho(|\Delta x|)$. These patterns are almost identical for KDE and NN analyses.

Such strong off—diagonal structure rigorously explains why a χ^2 fit that ignores covariances by using only the diagonal of C leads to miscoverage.

Implications

The correlation diagnostics presented here empirically validate the theoretical expectation that OmniFold weights are correlated. Quantifying the strength and range of those correlations enables an *a priori* estimate of how badly an independence–based error formula will fail**[cite –KD]**. They provide input templates and motivation for developing a rigorous covariance-aware asymptotic method. With these diagnostics in hand, we are now equipped to benchmark the statistical performance of binned and unbinned inference workflows on unbinned unfolded data, which is the subject of the next subsection.

Unbinned Unfolding: Binned and Unbinned Inference

Having established a fully binned baseline in the previous section, we now examine two inference workflows that utilize unbinned unfolding on the same Gaussian data. Both approaches start by unfolding the detector–level data without binning, producing a set of weighted events at truth–level. Because the deconvolution process induces non–negligible correlations between event weights (especially for finite detector smearing, as evidenced by the covariance patterns in **??**), these workflows differ in how they handle those correlations.

When performing binned inference on unbinned unfolded data, the unfolded weighted events are aggregated into a histogram, and a binned template fit is performed using a χ^2 statistic that includes the full binned covariance matrix. In these experiments, the unfolded distribution is fit to the parametric Gaussian model (with mean μ and variance σ^2) by

integrating the model prediction over the histogram bins. The covariance matrix of the bins is estimated from an ensemble of repeated pseudo–experiments. One then obtains best fit parameters by χ^2 minimization and determines their uncertainties via the usual $\Delta\chi^2 = 1$ criterion. This approach is analogous to a standard binned analysis except that the data have been unfolded using an unbinned method. Crucially, it retains statistical rigour by propagating the full unfolding–induced covariance into the fit.

Alternatively, one could attempt to fit the parametric model directly to the weighted events using an unbinned maximum–likelihood procedure that ignores inter–event correlations. One constructs the negative log likelihood,

$$\mathrm{NLL}(\theta) = -\sum_{i=1}^{N} w_i \ln P(x_i \mid \theta), \tag{7.11}$$

where x_i and w_i are the kinematic value and weight of event i, and $P(x_i|\theta)$ is the model density for parameters $\theta = (\mu, \sigma^2)$. This unbinned likelihood sum is maximized to obtain the best fit $\hat{\theta}$. The Hessian (curvature) of the NLL at the optimum provides an asymptotic error estimate for θ. Equivalently, one finds the $\Delta \ln L = 0.5$ offset for a 1σ interval. This workflow assumes statistical independence of events.**[cite:Cowan:2002in,Blobel:2203257 –KD]**. Thus, while method the latter method yields a fit and a formal error bar, these must be interpreted with caution since the underlying likelihood model is misspecified.

Both of the above approaches are applied to the same Gaussian datasets described earlier in **??**. Unfolding is performed with the OmniFold algorithm**[cite:Andreassen:2019cjw,Andreassen:2021zzk –KD]**. In the one dimensional setting we also test a KDE based implementation. For each resolution setting, the 500 pseudo–data samples are unfolded and then subjected to both the aforementioned inference procedures. This allows us to compare the bias, uncertainty estimation, and coverage of the two workflows against the fully binned baseline.

Parameter Bias Both unbinned workflows yield fitted parameter values that are consistent with those from the binned baseline. In fact, we find no appreciable additional bias introduced by the unbinned unfolding step or the choice of inference method. **??** (bottom panels) shows the mean fitted μ and σ^2 as a function of detector smearing for each method. All approaches produce estimates of μ and σ^2 across the range of smearings within the statistical uncertainties. Notably, the bias observed in the naive unbinned ML fit is the same as that in the proper χ^2 fit that accounts for the covariance of a given dataset. This can be explained as both fits ultimately maximizing a likelihood (or minimizing a χ^2) to match the unfolded distribution to the model; if the model is correctly specified, the MLEs should coincide. Thus, unbinned unfolding does not itself induce a bias in the extracted physics parameters verifying the claim that the OmniFold procedure (with sufficient iterations and regularization) correctly reproduces the shape of the true distribution on

average.**[cite:Andreassen:2019cjw –KD]** This is demonstrated by the agreement of fitted values with the true parameter shown by the horizontal lines in Fig.**??** and the baseline results in **??**.

Uncertainty estimation and coverage In sharp contrast to the agreement in central values, the two workflows differ markedly in their reported uncertainties and the statistical coverage of those uncertainties. The binned-χ^2 approach with full covariance proves to be consistent with the baseline in its uncertainty estimates. For each smearing level, the asymptotic 1σ errors obtained from the $\Delta\chi^2 = 1$ criterion agree well with the empirical spread (RMS) of the fitted parameters over the 500 pseudo-experiments. This is illustrated in the top panels of **??**: the curve corresponding to the full-covariance χ^2 fit lies on top of the true uncertainty obtained from the pseudo-data ensemble (star markers), for both μ (left) and σ^2 (right). In other words, binned inference on unbinned data using a χ^2 fit that incorporates the full covariance matrix produces accurate confidence intervals that maintain the nominal coverage. This behaviour is as expected and desired. By incorporating the complete covariance matrix of the unfolded histogram, we correctly account for the event correlations in the statistical inference. Indeed, these results mirror the fully binned study. Recall that in the baseline binned analysis, using the full covariance matrix yielded uncertainty estimates consistent with the bootstrap pseudo–data spread, whereas using only diagonal uncertainties did not, as shown in **??**. We explicitly confirm the same result in the unbinned case too. If one performs a binned fit to the unfolded histogram but (incorrectly) ignores off–diagonal bin correlations, the uncertainties are overestimated and the χ^2 fit yields unnecessarily large error bars, analogous to the diagonal–covariance points in **??**. This over–conservative result similarly arises from double–counting the anti–correlated fluctuations in each bin and leads to overcoverage, further underscoring that the full covariance is essential for a proper statistical treatment .

By contrast, the naïve unbinned ML inference on the unbinned data fails to produce correct uncertainty estimates. As shown in **??** (top panels, orange points), the asymptotic errors reported by the unbinned likelihood fit are dramatically smaller than the true spread of the fit results, except in the trivial case of zero smearing. Intriguingly, the naive unbinned error bars hardly change with detector resolution—the orange curve in **??** is nearly flat—indicating that the fit is seemingly just as “precise" for a very smeared dataset as for a perfect detector. This unphysical result is a direct consequence of ignoring the event correlations. The ML fit treats each weighted event as independent information, thereby overestimating the effective sample size. Intuitively, when events are strongly correlated, the true number of independent degrees of freedom is smaller than N; but the naive NLL sum scales like N, yielding an misestimate of the variance of $\hat{\theta}$. In our Gaussian example, the effect is severe even at moderate smearing. For instance, at $\sigma_{det} = 0.5$ the unbinned ML formula

underestimates the uncertainty on μ by roughly a factor of two compared to the bootstrap truth, and at $\sigma_{det} = 0.75$ the discrepancy is even larger. This breakdown of the asymptotic approximation in the presence of inter–event correlations is the essential failure mode of the naive unbinned approach. It should be emphasized that when the detector resolution is perfect (no smearing), the unfolding induces no correlations and indeed the unbinned ML errors do coincide with the correct uncertainties, and all methods become equivalent in this limit. But for any non–zero smearing, the independence assumption is violated and the standard likelihood formulae no longer hold. The unbinned fit still finds the correct central values, but its error estimates must not be trusted.

The practical implication of these findings is that naively applying unbinned inference to unfolded data can produce misleadingly constraints, even though the fit may appear to converge normally. In our study, the naïve unbinned workflow would have erroneously suggested measurement insensitive to detector smearing, whereas in truth the uncertainties should grow with smearing, as correctly reflected by methods that account for correlations. This highlights the necessity of handling the event correlations in some way, potentially through the hybrid approach. By introducing a binning at the final inference stage and using a covariance matrix, one essentially restores statistical consistency, at limits the impact of binning artifacts. Such an approach largely retains the benefits of unbinned unfolding, with no information loss up to the point of inference, while yielding parameter uncertainties and coverage properties in line with a rigorous frequentist construction. Hence this "unbinned unfolding + binned fit" should be more precise than a fully binned analysis. For example, in **??** the full covariance UBU results achieve a similar or smaller uncertainty than the traditional IBU based results at each smearing value. This suggests an advantage to delaying any binning as late in the analysis chain as possible, consistent with the intuition that using unbinned distributions throughout the unfolding can preserve more information for the final fit**[cite:Andreassen:2019cjw –KD]**. One should note, however, that in the 1D case this advantage is modest. UBU and IBU approaches do not differ vastly in precision. In higher dimensions the unbinned approach is expected to drastically outperform a binned analysis (since binning in many dimensions is impractical or introduces large discretization errors)**[cite:Pan:2024rfh,Butter:2022rso2 –KD]**.

While the hybrid method provides a sound stopgap, it somewhat undermines the original motivation for unbinned methods, which aimed to avoid histogramming altogether. The fully unbinned method, on the other hand, fully actualizes the principle underlying the unbinned paradigm but lacks a statistical asymptotic formalism that accounts for correlations. The stark miscoverage we observe for the naive unbinned fit underlines the need for correlation–aware unbinned inference techniques. In other words, if one wishes to perform event–level likelihood fits on unfolded data, one must either develop a formalism to incorporate the event–to–event covariance information into the likelihood or rely on numerical methods to calibrate the uncertainties. In this study, one does finds that if one estimates

uncertainties numerically through pseudo–experiments or bootstraps, the precision of the unbinned fit can be evaluated correctly. This is evidenced by the fact that the RMS spread of the naive ML fit outcomes (the blue line in **??**) does increase with smearing and matches the covariance-fit results. However, using brute force pseudo–experiments or bootstraps to determine errors is computationally expensive and may not be feasible in a real experiment. Besides, errors computer through bootstrapping offer no analytical understanding of the uncertainty. Nonetheless, until a theoretical framework is developed to handle correlations at the likelihood level, any unbinned inference on unfolded events should be performed through bootstrapping.

In summary, the comparison of binned versus unbinned inference after unbinned unfolding reveals that both workflows yield unbiased parameter estimates, but only the approaches that account for correlations produces reliable uncertainties and coverage. A naïve approach of treating weighted unfolded events as independent leads to misestimated uncertainties and hence significant miscoverage. These results vividly demonstrate the statistical pitfalls of ignoring induced correlations, and they motivate the correlation–aware inference strategies discussed in the next section. As unbinned unfolding techniques become increasingly prevalent in extracting cross sections**[cite:ATLAS:2024xxl,ATLAS:2025qtv,CMS-PAS-SMP-23-008 –KD]**, developing a robust framework to correctly propagate uncertainties through the unbinned pipeline is essential. This study provides a first quantitative glimpse of the issue, showing that while unbinned unfolding can preserve accuracy and potentially improve precision, one must incorporate the full correlation structure to achieve valid statistical inference. This will be crucial for ensuring proper coverage and trustworthy uncertainty estimates in future high–dimensional unfolding analyses.

Extension to Higher Dimensions

The diagnostics of **????** established that even in one dimension the naïve unbinned likelihood approach seriously underestimates uncertainties once unfolding procedure induces correlated event weights. A natural next question is whether that pathology grows, diminishes, or saturates as one moves to multi—differential measurements. We therefore repeat the Gaussian study in two, four and six dimensions, keeping the experimental setup identical, except for adjusting the numerical parameters that define the Gaussians. The Gaussian parameters used in the multidimensional studies are listed in **??**. Detector resolutions are scaled coordinate–wise so that the signal–to–noise ratio in every dimension matches the 1–D baseline.

Table 7.2: Gaussian parameters used in the 2, 4, and $6-$dimensional toy studies. The column vectors $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ list the component means and standard deviations, respectively. ρ denotes the linear correlation matrix. Detector resolutions $\boldsymbol{\sigma}_{\mathrm{det}}$ are applied component-wise as additive Gaussian noise in the detector simulation.

	Gen.	**Det.**	**Truth**
		2–D	
$\boldsymbol{\mu}$	$\begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}$		$\begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix}$
$\boldsymbol{\sigma}$	$\begin{bmatrix} 1.0 \\ 1.5 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0.8 \end{bmatrix}$	$\begin{bmatrix} 0.9 \\ 1.3 \end{bmatrix}$
ρ	$\begin{bmatrix} 1.0 & -0.6 \\ -0.6 & 1.0 \end{bmatrix}$		$\begin{bmatrix} 1.0 & -0.6 \\ -0.6 & 1.0 \end{bmatrix}$
		4–D	
$\boldsymbol{\mu}$	$\begin{bmatrix} 1.0 \\ 0.0 \\ -0.5 \\ 0.5 \end{bmatrix}$		$\begin{bmatrix} 0.8 \\ 0.1 \\ -0.6 \\ 0.7 \end{bmatrix}$
$\boldsymbol{\sigma}$	$\begin{bmatrix} 1.0 \\ 0.7 \\ 1.1 \\ 0.8 \end{bmatrix}$	$\begin{bmatrix} 0.4 \\ 0.5 \\ 0.6 \\ 0.3 \end{bmatrix}$	$\begin{bmatrix} 0.8 \\ 0.6 \\ 1.0 \\ 0.6 \end{bmatrix}$
ρ	$\begin{bmatrix} 1.0 & 0.1 & -0.2 & 0.3 \\ 0.1 & 1.0 & 0.0 & 0.1 \\ -0.2 & 0.0 & 1.0 & 0.7 \\ 0.3 & 0.1 & 0.7 & 1.0 \end{bmatrix}$		$\begin{bmatrix} 1.0 & 0.0 & -0.3 & 0.4 \\ 0.0 & 1.0 & 0.2 & 0.0 \\ -0.3 & 0.2 & 1.0 & 0.5 \\ 0.4 & 0.0 & 0.5 & 1.0 \end{bmatrix}$
		6–D	
$\boldsymbol{\mu}$	$\begin{bmatrix} 1.0 \\ 0.0 \\ -0.5 \\ 0.5 \\ -1.0 \\ 0.3 \end{bmatrix}$		$\begin{bmatrix} 0.8 \\ 0.1 \\ -0.6 \\ 0.7 \\ -0.8 \\ 0.1 \end{bmatrix}$
$\boldsymbol{\sigma}$	$\begin{bmatrix} 1.0 \\ 0.7 \\ 1.1 \\ 0.8 \\ 1.2 \\ 1.4 \end{bmatrix}$	$\begin{bmatrix} 0.4 \\ 0.5 \\ 0.6 \\ 0.3 \\ 0.4 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.8 \\ 0.6 \\ 1.0 \\ 0.6 \\ 1.0 \\ 1.1 \end{bmatrix}$
ρ	$\begin{bmatrix} 1.0 & 0.1 & 0.2 & -0.3 & 0.0 & 0.0 \\ 0.1 & 1.0 & 0.0 & -0.2 & 0.3 & 0.1 \\ 0.2 & 0.0 & 1.0 & 0.1 & -0.2 & 0.3 \\ -0.3 & -0.2 & 0.1 & 1.0 & 0.1 & 0.0 \\ 0.0 & 0.3 & -0.2 & 0.1 & 1.0 & 0.7 \\ 0.0 & 0.1 & 0.3 & 0.0 & 0.7 & 1.0 \end{bmatrix}$		$\begin{bmatrix} 1.0 & 0.0 & 0.2 & -0.2 & 0.1 & 0.0 \\ 0.0 & 1.0 & 0.0 & -0.1 & 0.2 & 0.0 \\ 0.2 & 0.0 & 1.0 & 0.0 & -0.3 & 0.4 \\ -0.2 & -0.1 & 0.0 & 1.0 & 0.2 & 0.0 \\ 0.0 & 0.2 & -0.3 & 0.2 & 1.0 & 0.5 \\ 0.0 & 0.0 & 0.4 & 0.0 & 0.5 & 1.0 \end{bmatrix}$

Evaluation

For every fit parameter (all μ_k and all unique σ^2_{kl}) we compute the empirical RMS over 500 pseudo-–experiments, and the average asymptotic error reported by the naïve unbinned–likelihood Hessian. If the likelihood were well—calibrated, these two quantities should coincide. **??** plots RMS versus asymptotic error for all parameters in 1, 2, 4, and 6–D. The dashed diagonal marks perfect agreement and the solid green line has a slope fixed to the mean of the RMS/asymptotic–error ratio for that dimension.

As we can see, in each case, the RMS uncertainty is higher than the asymptotic uncertainty. The ratio of the RMS uncertainty to the asymptotic uncertainty is roughly the same for all of the model parameters with the ratio ranging from 1.18 to 1.28. The analytic error bars understate the true variance by a factor ~ 3–4, consistent with the effective sample size argument.**[https://andrewcharlesjones.github.io/journal/21-effective-sample-size.html –KD]**.

?? repeats the 6–D study with detector resolution scaled by $0, 1, 2,$ and 3 relative to the σ_{det} in **??**. The slope grows with the smearing factor, confirming that weight correlations, not intrinsic variance, drive the failure**[Eur. Phys. J. C 82 (2022) 393 –KD]**.

In high dimensions, the mapped weight vector $w(\mathbf{x})$ is a smooth function on a space where almost all pairs of points are distant. Hence the classifier must assign similar gradients to a larger neighbourhood, inflating correlations. **[https://aicompetence.org/kernel-density-estimation-non-parametric-probability/ –KD]** explains this phenomenon for KDEs, where excessively wide kernels force longer–range weight correlations. The same effect applies to NNs as well because global normalisation $\sum_i w_i = N_{\text{MC}}$ adds an $\mathcal{O}(1/N)$ positive correlation to every pair of events**[PhysRevD.90.072004 –KD]**.

In six dimensions the naïve asymptotic formula would underestimate statistical errors for several covariance elements, while the RMS spread shows true uncertainties. For collider analyses, where multi-differential analyses are the future, this degree of systematic undercoverage is unacceptable. Either a binned-covariance approach or a correlation-aware unbinned likelihood**[https://indico.cern.ch/event/671301/contributions/2745801/attachments/1557488/2449991/unfold.pdf –KD]** is necessary. Recent proposals such as Schrödinger–bridge unfolding**[arxiv 2308.12351 –KD]**, or low-rank covariance compression**[arXiv:1802.06048 –KD]** offer interesting possibilities for exploration.

7.6 Conclusions and Outlook

This chapter presented a detailed study of parameter inference performed on an unbinned unfolded dataset, using a controlled Gaussian simulation. By employing a simplified scenario where the true distribution and detector response are known analytically, we were

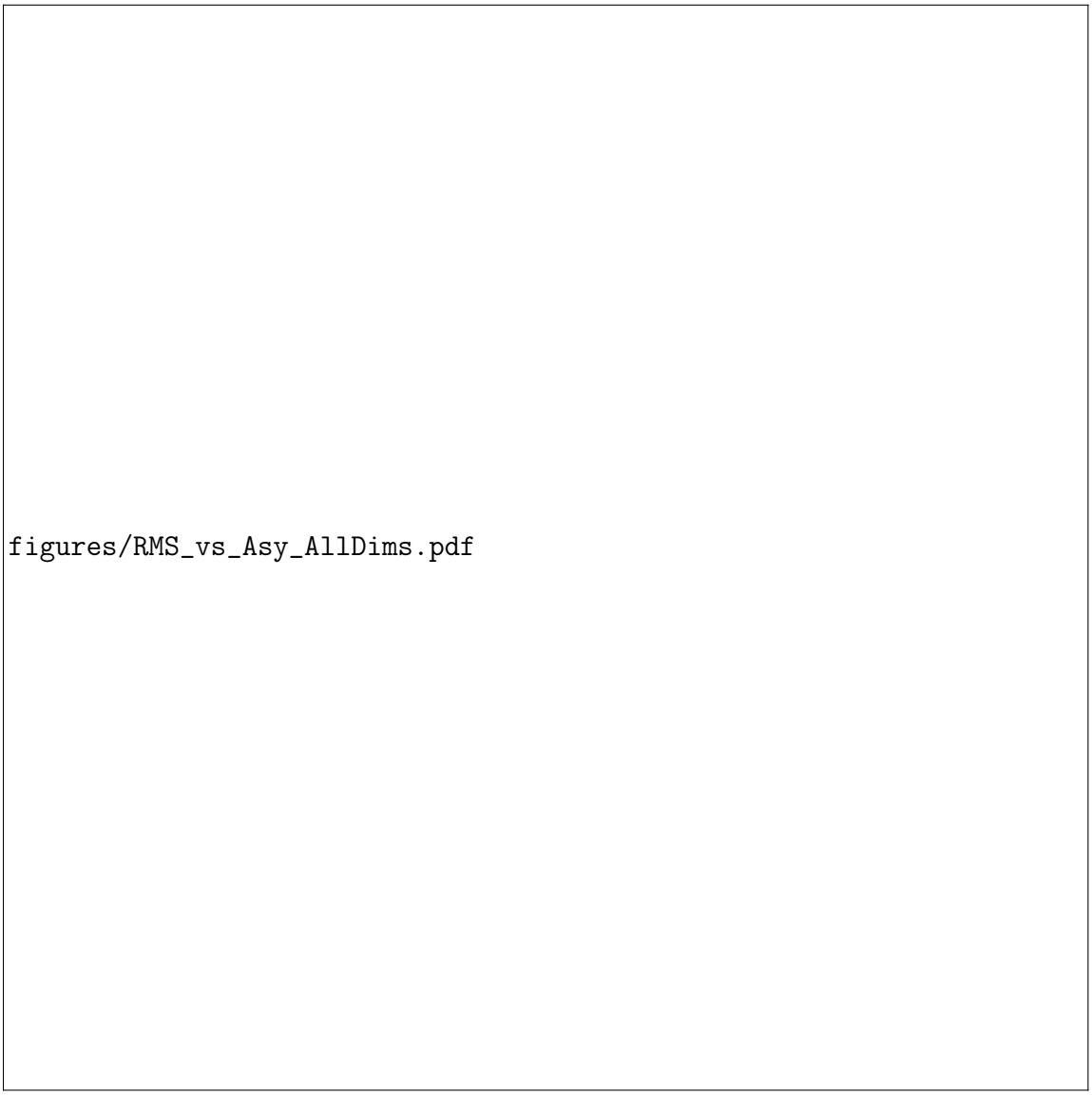


Figure 7.4: Scatter of empirical RMS vs. mean asymptotic uncertainty for every model parameter in 1, 2, 4 and 6–D unbinned unfolding studies. The grey dashed line has unit slope; green line's slope equals the mean RMS/asymptotic ratio for that dimension.

Figure 7.5: Same as Fig. **??** but confined to the 6–D study and showing four detector–smearing scale factors $(0, 1, 2, 3)$. Increased smearing increases correlations and drives the RMS/asymptotic slope ever further above unity.

able to isolate and rigorously examine the statistical subtleties introduced by unbinned unfolding. The Gaussian studies demonstrated that preserving event–level information until the final fit can indeed confer a tangible advantage: an unbinned unfolding followed by parameter estimation was observed to outperform a fully binned analysis in terms of fit precision. This validates the intuitive principle that one should delay information reduction (e.g. binning) as much as possible in the analysis chain, thereby maximizing the use of available information. Importantly, however, the findings also exposed critical caveats that must temper this optimism, especially when interpreting unbinned results with standard inference techniques.

A central observation is that the output of an unfolding procedure violates the usual assumption of statistical independence among events. In our studies, the iterative OmniFold procedure produces a weighted set of "unfolded" events that are correlated with one another. We showed that a naïve unbinned likelihood–based inference, which treats the unfolded events as if they were independent observations, fails to yield reliable uncertainty estimates. In particular, when event–to–event correlations are ignored, the standard asymptotic formulae for parameter uncertainties[11] become invalid. Concretely, the naive application of an unbinned maximum likelihood fit dramatically underestimates the true uncertainty on the fitted parameters in Gaussian examples. This underestimation was evidenced by comparing the analytic errors to the empirical spread of fit results across many pseudo-experiments. The analytic errors, assuming independence, were significantly smaller than the actual RMS of the fitted parameters. Such a discrepancy is a direct manifestation of ignoring the induced correlations and a clear warning that conventional inference tools cannot be blindly applied to unbinned unfolded data. In short, the lack of statistical independence introduced by unbinned unfolding can invalidate classical error estimates, and thus the entire inferential procedure must be handled with care.

On the positive side, the investigation also highlighted that correlation–aware approaches can effectively restore valid inference, albeit with practical limitations. When one incorporates the inter–event correlations into the analysis, one finds that the derived uncertainties align with the true performance of the fit. In the Gaussian studies, the numerically estimated uncertainties, obtained from the spread of outcomes over many trials, agree well with a covariance–informed analytical approach, confirming that the primary cause of the asymptotic formula breakdown was the neglect of correlations. This serves as an important proof of principle. If one properly accounts for the covariance between events[12] in the unbinned dataset, inference methods can yield correct confidence intervals and hypothesis tests even in the unbinned paradigm. However, it should also be emphasized that such covariance-aware treatments are computationally challenging to scale. In a binned analysis the covariance matrix is of manageable size, but in an unbinned analysis even if a suitable covariance kernel, one in principle faces an $N \times N$ covariance among N events or event weights, which is intractable for the large event samples typical of collider data. The use of bootstrap ensembles to numerically evaluate uncertainties, while effective for a toy study, would be prohibitively expensive for high–dimensional or high–multiplicity data. Thus, the practical implementation of correlation–aware unbinned inference encounters serious scaling limitations, underscoring the need for new strategies to handle or approximate these large covariance structures.

[11] derived from a Fisher information or Hessian approximation

[12] or equivalently, between event weights

Crucially, the results presented here should be viewed as a cautionary case study rather than a definitive statement on all unbinned analyses. They were obtained in an idealized Gaussian context, with complete knowledge of the generative process. This controlled setup allows one to rigorously identify potential pitfalls and verify proposed solutions, but it also means that the qualitative behaviours observed might not universally translate to real collider data. In particular, the observation that ignoring correlations had little impact on the central values and empirical precision of the fit in the Gaussian example may be a fortunate consequence of the symmetry and simplicity of that example. One should not assume this will hold in general. Therefore, while this study provides a rigorous proof of principle and important warnings, further work is required to generalize of these findings. The takeaway is that we have uncovered a set of statistical issues that could plausibly afflict unbinned cross section measurements, and we have verified their existence in a controlled setting. It now remains to investigate whether and how these issues manifest in realistic analyses. This chapter's findings should motivate vigilance so that any inference on unbinned unfolded data must be scrutinized for hidden correlations, and current methods must be extended before one can rely on them in full generality.

Looking ahead, our work motivates several important future directions for both methodology and application. An immediate next step is to extend these diagnostic studies to real collider observables and data. It will be invaluable to apply similar techniques (e.g. pseudodata experiments, bootstrap uncertainty evaluations, and covariance measurements) to a realistic experimental unfolding scenario, for example, a differential cross section measurement published with unbinned results, to evaluate the size of event–level correlations and to quantify their impact on parameter fits. Such studies on real or high fidelity simulated data would confirm whether the cautionary lessons from the Gaussian example are broadly applicable, and could reveal any additional complications arising from more complex data structure or detector effects. Another critical line of research is to develop new statistical frameworks for unbinned inference that explicitly account for event correlations. This could involve formulating modified likelihood functions or test statistics that include correlation terms, or designing hybrid approaches that retain the advantages of unbinned data while imposing an effective covariance model. The ultimate goal would be to have an unbinned inference methodology that is correlation–aware by construction, obviating the need for *ad hoc* binning or numerical uncertainty estimates. In tandem with this, there is a clear need for scalable covariance approximation techniques. Instead of attempting to handle a full covariance kernel of enormous dimension, one might seek low rank representations, clustering of events into groups with approximate independence, or other dimensionality reduction methods to capture the dominant correlation effects without the full cost. Research into such approximations (potentially informed by the structure of the machine learning algorithms used in unfolding) will be essential to make correlation-aware inference feasible for large datasets. Finally, further explorations of the interplay between machine learning

regularization and inference accuracy could be informative. A deeper understanding of how the ML aspects affect downstream inference, for instance, whether a stronger regularization might reduce variance at the cost of introducing bias or correlations, would be extremely valuable. By addressing these open questions, the community can build on the foundation laid in this chapter.

In summary, this chapter has established a foundational understanding of unbinned inference on correlated data, highlighting both the potential benefits of fully unbinned analyses and the new statistical challenges they pose. The Gaussian studies act as a proof of principle, rigorously demonstrating that unbinned unfolding methods can be combined with parameter fits, but also warning that ignoring induced correlations can invalidate conventional uncertainty estimates. These conclusions, drawn in a simplified setting, strongly motivate the development of improved tools and methods before applying unbinned inference to precision measurements. As the field moves toward ever more complex and high-dimensional data analyses, the insights gained here will guide the creation of a robust statistical framework for unbinned cross section measurements, one that maximizes information usage while properly accounting for the complex correlations introduced by the unfolding process. The lessons of this chapter therefore serve as both a caution and a call to action, laying the groundwork for more reliable unbinned inference techniques in high energy physics.

Chapter 8

[Optional] Towards Robust Unfolding with Nuisance Parameters

- The challenge of detector response uncertainties
- Profiled unfolding methodology
- Simultaneous learning of response and physics parameters
- Proof-of-concept with gaussians
- Applications and limitations
- Future development directions [Note: This chapter can be easily removed if paper is not completed]

Chapter 9

Symmetries in Data: Connections to Unfolding Challenges

- The role of symmetries in physics and measurement processes
- Statistical definition of dataset symmetries
- SymmetryGAN: Discovering symmetries with adversarial learning
- Application dijet events
- Using symmetry knowledge to constrain unfolding problems
- Symmetry-aware unfolding for improved measurement precision

9.1 Symmetries and Unfolding

The Complementary Nature of Symmetry Discovery and Unfolding

Unfolding, as discussed, refers to the inverse problem of inferring underlying truth–level distributions from observed detector–level data, accounting for distortions due to limited resolution and acceptance. Symmetry discovery, aims to identify invariant transformations of the data, that is to say, operations under which the probability distribution of the dataset remains unchanged in a statistical sense**[cite –KD]**. At first glance, these two tasks appear distinct; one concerns recovering numerical distributions, while the other uncovers structural invariances. However, symmetry discovery and unfolding are in fact complementary facets of data–driven inference, and integrating the two can yield deeper insights and improved measurements.

From a conceptual standpoint, both tasks share the common goal of revealing hidden truth from observed data. Unfolding endeavours to remove the “detector mask" and expose the true differential cross section or underlying distribution that generated the measurements. Symmetry discovery seeks to reveal underlying structures or invariances in the data-—patterns that persist under transformations, reflecting fundamental symmetries of the physical process or the measurement apparatus.

In practice, these goals are intertwined. If one discovers a symmetry in the dataset, that knowledge can constrain the unfolding procedure by reducing the effective degrees of freedom in the solution space. Conversely, a properly unfolded distribution is expected to manifest latent symmetries that may have been obscured by detector effects in the raw data. Thus, identifying a symmetry and unfolding a distribution reinforce one another. The former provides a guiding principle or constraint for the latter, while the latter provides a cleaner canvas on which the former can be observed.

Imposing a symmetry as a prior constraint in unfolding can be seen as a form of physically motivated regularization. For example, architectures that conserve four–momentum or enforce Lorentz invariance by design, as discussed in **??**, effectively impose such constraints**[cite –KD]**, narrowing the set of viable solutions. By restricting solutions to those that respect a discovered or expected invariance, one reduces the space of admissible unfolded distributions to those that are physically plausible, which leads to improved stability and fidelity of the results**[cite –KD]**.

This principle has been implicitly utilized in classical unfolding and simulation–based calibration. For example, one could assume certain symmetries such as isotropy or detector uniformity when designing the response matrix or when combining symmetric regions of phase space to reduce uncertainties. With a data–driven symmetry discovery tool in hand, one need not rely solely on presumed symmetries. Instead, one can verify them empirically or even discover unexpected invariances. In turn, these empirically verified symmetries can be fed back into the inference pipeline to sharpen measurements, such that a discovered symmetry can inform the unfolding algorithm so that the final measured distribution upholds the invariance.

It is instructive to compare symmetry discovery and unfolding side by side to appreciate their complementary roles. **??** summarizes the differences and points of contact between the two.

While unfolding typically requires an explicit model of the measurement process[1] and often relies on supervised learning or iterative inversion techniques, symmetry discovery can be pursued in an unsupervised manner, requiring only the dataset and a class of transformations to probe. The outcome of unfolding is a corrected distribution intended for direct physical interpretation. The outcome of symmetry discovery is a characterization

[1] e.g. a response matrix or a parametrized detector simulation

Table 9.1: Comparison of Unfolding and Symmetry Discovery

	Unfolding	**Symmetry Discovery**
Goal	Reconstruct true distribution from observed data	Find transformation(s) which keep the distribution is invariant.
Input	Measured data and detector response model	Measured data and a class of candidate transformations.
Output	Unfolded differential cross section or probability density,	Symmetry transformations or invariance properties.
Physics	Incorporates known physics via priors or constraints (e.g. smoothness, conservation laws) to regularize solutions.	Incorporates generic transformation families (e.g. rotations, permutations) but does not assume a particular symmetry *a priori*
Relationship	Produces clearer distribution for true symmetries should manifest, enabling validation or discovery of invariances.	Provides constraints that can regularize unfolding by restricting solution space to symmetry-respecting distributions.

of invariances, a set of transformations T such that the dataset's distribution is invariant under T within statistical uncertainties. These outcomes are different in nature, but they are mutually beneficial.

Knowledge of invariant structure can guide numerical inference, and conversely, obtaining a more accurate numerical distribution makes it easier to discern subtle invariant patterns. In essence, symmetry discovery and unfolding form a feedback loop in the broader endeavour of measurement and inference, where each can improve the other.

Symmetry–Aware Cross Sections

Differential cross section measurements lie at the core of particle physics. They quantify how often certain processes occur as a function of kinematic variables (such as angles, energies, or invariant masses), serving as detailed tests of theoretical models. Achieving high precision in these measurements is essential, as even subtle deviations between the measured spectra and theory predictions can signal new physics or the need for refined models. In this context, symmetries play a pivotal role in both the design and interpretation of cross section measurements.

Many physical processes come with known symmetry expectations. For instance, in proton—proton collisions producing particle jets, one expects azimuthal symmetry about the beam axis, i.e. the physics is invariant under rotations in the plane perpendicular to the beam. Consequently, the differential cross section should, after correcting for detector non–uniformities, be independent of the absolute azimuthal angle ϕ of a jet or dijet system**[cite –KD]**. Likewise, for processes initiated by identical colliding particles, one often anticipates a symmetry between forward and backward directions. In a symmetric proton—proton collider, this implies that the rapidity distribution of a centrally produced system [2] should be symmetric about zero rapidity**[cite –KD]**. Such symmetry means that the cross section for producing a system at rapidity $+y$ is the same as at $-y$, all else being equal. If a measured differential cross section exhibits a significant asymmetry in these variables after unfolding and acceptance corrections, it would either indicate a previously unaccounted detector bias or hint at a physical effect, both of which are of great interest to investigate.

Being symmetry–aware in a measurement can mean two things. First, verifying that expected symmetries are indeed present, within uncertainties, in the data, and second, leveraging those symmetries to improve the measurement. On the verification side, symmetry considerations provide valuable consistency checks. Experiments can test whether their unfolded distributions respect fundamental symmetries, and a failure to observe an expected symmetry is a red flag, prompting scrutiny of systematic effects or potential new physics contributions**[cite –KD]**.

On the other hand, when a symmetry is confirmed, one can exploit it to gain statistical and systematic advantages. For example, if a distribution is believed to be symmetric in a certain variable, one can "augment" or combine data from symmetric regions, effectively doubling the effective statistics for that distribution. A common practice is to report differential cross sections as a function of $|y|$ or other symmetry–reduced variables, which assumes $y \leftrightarrow -y$ symmetry and thereby reduces statistical fluctuations**[cite –KD]**. By incorporating symmetry in this manner, uncertainties can be reduced and the measurement becomes more robust against localized fluctuations.

Symmetry–aware analysis also serves to impose physically motivated constraints that guard against overfitting noise in the unfolding process. If one has discovered that a distribution must be invariant under a transformation,[3] imposing this invariance in the unfolding procedure will tie neural network parameters or bin values together that would otherwise float independently. This effectively decreases the number of free parameters describing the unfolded result, acting as a regularizer that prefers solutions consistent with the symmetry. The net effect is an improvement in the precision of the measured cross section and a reduction in spurious oscillatory features that might arise from statistical fluctuations. Moreover,

[2] e.g. dijet pair

[3] e.g. rotating the entire event by some angle, or exchanging two identical particles in the final state,

by reducing the dependence on bins with low occupancy (because they are combined with their symmetric counterparts), binned symmetry-aware unfolding can also mitigate the impact of detector acceptance edges or inefficiencies in specific regions of phase space.

It is important to note, however, that any symmetry–based constraint should be applied with careful consideration. One must ensure that the symmetry is either theoretically well-founded or empirically validated, lest one impose a false invariance and obscure a genuine asymmetry. This caution further motivates the need for data–driven symmetry discovery and validation tools. An experimenter can use methods like SymmetryGAN**[cite –KD]** to verify whether the data uphold the symmetry to a high degree of confidence. Only then would one proceed to incorporate that symmetry into the unfolding process or in the presentation of results. Thus symmetry-aware differential cross section measurements can harness known, or discovered invariances to enhance precision and reliability, while simultaneously providing a framework to detect symmetry violations that could point to new phenomena.

The research presented in this chapter builds upon the foundations laid in earlier chapters of this thesis, extending the paradigm of symmetry utilization in the context of measurement and unfolding. **??**, in its survey of existing techniques, provides a statistical foundation for incorporating known symmetries into data analyses, for example, by augmenting jet images with rotations to enforce rotational invariance. It also helps us note the challenges such approaches face, such as the need for smoothing and careful validation of assumptions. **??** includes references to how *a priori* known symmetries can be hard coded into machine learning models, introducing Lorentz group equivariant networks that guarantee Lorentz invariance in the unfolding of particle physics data**[cite –KD]**. The inclusion of symmetry constraints in unfolding models described in **??????** through preserving physical invariants like momentum or charge conservation in the generative model would reduce the solution space and lead to more physically plausible unfolded results**[cite –KD]**. All of these strategies rely on prior knowledge of the symmetry. This chapter shifts to an inference driven approach that provides a novel, flexible, and fully differentiable deep learning based method to discover symmetries directly from data using adversarial learning, which then might allow one to leverage those discovered symmetries to inform the unfolding process. This perspective emphasizes the overarching theme of the thesis, the interplay of measurement and inference, by using data–driven insights in the form of symmetry discovery to enhance the core measurement task of differential cross section unfolding.

The remainder of this chapter is organized as follows. **??** introduces the formal statistical definition of a dataset symmetry, addressing subtleties like Jacobian volume effects via the concept of an inertial reference density. It also provides a brief overview of the symmetries most relevant to HEP. **??** presents the SymmetryGAN framework, which employs a generative adversarial network to automatically learn symmetry transformations from data. **??** validates this approach on illustrative examples and then applies SymmetryGAN to

simulated dijet events, demonstrating how it can uncover physically meaningful symmetries in collider data. **??** discusses how the discovered symmetry information can be used to constrain unfolding problems: we outline methods to incorporate symmetry constraints into the unfolding procedure to reduce uncertainties and bias. In **??**, the chapter introduces a symmetry aware unfolding methodology and discusses how enforcing the symmetries identified by SymmetryGAN can improve the precision of differential cross section measurements. Finally, the chapter concludes by highlighting how the insights gained here connect back to the broader narrative of the thesis, reinforcing the benefits of combining machine learning driven discovery with principled measurement techniques.

9.2 Formalism and Importance

In physics and statistics, a symmetry refers to an invariance of a system or dataset under a well-defined transformation. Formally, let G be a group of transformations (continuous or discrete) acting on a space of states or observations X. A physical system or probability distribution is symmetric under G if applying any transformation $g \in G$ leaves the relevant observables unchanged. In group–theoretic terms, there exists an action $g : x \mapsto g(x)$ such that for all $x \in X$ and $g \in G$, the value of an function $O(x)$ remains equal to $O(g(x))$. However, if $p(x)$ denotes a probability density on X, a group G is a symmetry of p if

$$\forall g \in G \; p(x) \; \mathrm{d}x = p(g(x)) \; \mathrm{d}(g(x)) \; .\textbf{[PhysRevD.111.072002 –KD]} \tag{9.1}$$

In measure–theoretic language, a symmetry corresponds to an invariant measure. For any measurable subset $A \subseteq X$ and any transformation g, g is a symmetry of A if $\mu(A) = \mu(g \cdot A)$, meaning the measure assigned to outcomes in A is the same as that for the transformed set $g \cdot A$. This definition encompasses both continuous symmetries[4] and discrete symmetries [5]. Symmetry principles lie at the heart of modern particle physics and also strongly influence experimental measurements. At the theoretical level, fundamental symmetries constrain the form of physical laws and often correspond to conserved quantities or selection rules. At the data level, symmetries, and their breaking, shape the distributions of observed events and can be leveraged for more efficient data analysis. In the context of colliders, many observables are governed by symmetries of the underlying theory as well as symmetries introduced by the detector and measurement process. It is therefore crucial to articulate how these symmetries operate both in ideal physics scenarios and in real observations. This section provides a rigorous overview of symmetries relevant to collider physics and

[4] Lie groups, such as rotations depending on a continuous angle parameter

[5] groups constructed as Jordan–Hölder extension**[Michael Aschbacher (2004) –KD]** e.g. a mirror reflection or a permutation of identical objects

measurements. It begins with the fundamental symmetries in particle physics that underlie observable phenomena in **??**. **??** discusses symmetries in detector response functions and how the measurement apparatus can preserve or violate underlying invariances. Next, **??** examines how symmetries manifest in measured cross sections and data distributions, clarifying the translation from physical symmetry to statistical patterns in experimental histograms. Finally, **??** highlights the challenges in identifying symmetries from noisy data, setting the stage for data–driven symmetry discovery techniques. This foundation will be essential for later sections that introduce methods like SymmetryGAN for learning symmetries from data.

Fundamental Symmetries in HEP

Particle physics is built upon a framework of symmetries that determine the allowed forms of interactions and the conservation laws observed in experiments. Spacetime symmetries, in particular subgroups of the Poincaré group, are foundational. The Poincaré group includes continuous Lorentz invariance (rotations and boosts) and translations (in space and time). Lorentz invariance implies that the laws of physics take the same form in any inertial reference frame. Equivalently, physical observables can be expressed in terms of Lorentz–invariant quantities[6] that remain unchanged under boosts or rotations. For example, the Mandelstam variables s, t, u in a scattering process or the decay angle distribution in a particle's rest frame are formulated to respect Lorentz symmetry. In practice, exact Lorentz invariance means there is no preferred direction or absolute velocity in the underlying theory. A given collision process should yield identical outcomes whether the laboratory frame is, say, Earth–bound or boosted to a constant velocity. As a consequence of Lorentz symmetry and spatial isotropy, angular momentum and linear momentum are conserved in isolated systems[7]. Additionally, time translation symmetry leads to energy conservation, ensuring that system's total energy and the collision centre–of–mass energy are fixed constants of motion. These spacetime symmetries are exact symmetries of all known fundamental interactions and provide the basis for defining covariant formalisms in quantum field theory.

Beyond spacetime, the internal symmetries of the Standard Model dictate the spectrum of particles and their interactions. Chief among these is the gauge symmetry group $SU(3)_C \times SU(2)_L \times U(1)_Y$, which defines quantum chromodynamics and electroweak theory. Gauge symmetries are local symmetries that require the introduction of gauge bosons; although these are internal symmetries rather than symmetries of observable spacetime, they have observable consequences such as charge conservation, associated with $U(1)_Y$ hypercharge

[6] e.g. invariant masses, angles, and dimensionless ratios

[7] Noether's theorem associates these conservations with rotational and translational symmetry, respectively **[cite –KD]**

symmetry and electric charge, and the existence of multiple particle generations. The gauge symmetries of the Standard Model are spontaneously broken in certain cases.[8] However, even broken symmetries leave remnant effects, such as the custodial symmetry in the Higgs sector or approximate conservation of isospin in QCD. These internal symmetries set selection rules. Processes that violate gauge charge conservation are forbidden and decays proceed only via symmetric channels.

Alongside continuous symmetries, several discrete symmetries play a crucial role in particle physics. The most prominent are C (charge conjugation, exchanging particles with their antiparticles), P (parity, spatial inversion or mirror reflection), and T (time reversal). Each of these can be considered a transformation that might leave the fundamental laws invariant. In the Standard Model, CP symmetry is approximately a symmetry of electromagnetic and strong interactions, but notably broken in weak interactions. This manifests as differences in the behaviour of matter and antimatter. Like most notable instance of this is the well known CP violation in neutral kaon and B-meson decays means those processes occur at different rates or with different phase relationships than their CP-mirrored counterparts.**[–KD]** If CP were an exact symmetry of the dynamics, one would expect, for example, the angular distribution of decay products in a mirror–reflected process, swapping particles for antiparticles, to be identical to the original. The observed deviations are vital clues to physics beyond simple symmetries. Parity by itself is also violated maximally in the weak interaction.[9] On the other hand, the strong and electromagnetic interactions conserve parity, so for many processes, especially at high energies where electroweak effects are subdominant, it is a good symmetry. A collider process governed by QCD, like multijet production, should occur equally in a configuration and its mirror reflected image, unless the experimental setup selects a handedness. Charge conjugation is likewise not a symmetry of the full Standard Model, since, for example, there are no right–handed neutrinos to pair with left–handed ones under C, but for purely electromagnetic processes C–symmetry implies, that producing a negatively charged particle is as likely as producing the corresponding positively charged antiparticle under equivalent conditions. Importantly, the combination CPT is believed to be an exact symmetry of local quantum field theory. CPT symmetry implies, for instance, that particle and antiparticle masses and lifetimes are exactly equal **[cite –KD]**. While CPT is not directly tested by single distribution symmetries in colliders, it provides a fundamental consistency check on any observed CP or T violation. **??** summarizes these fundamental symmetries, their group-theoretic character, and their status in the Standard Model.

[8] One of the most notable examples is the electroweak $SU(2)_L \times U(1)Y$ breaking to $U(1)_{\text{EM}}$ via the Higgs mechanism, which introduces masses for the $W^{\pm}$ and Z bosons and differentiates the electromagnetic and weak interactions.

[9] classic examples are the left handed nature of neutrinos and the parity asymmetric angular distribution of electrons in polarized ^{60}Co beta decay **[cite –KD]**.

Table 9.2: Summary of key fundamental symmetries relevant to collider observables. "Charge" refers broadly to conserved / constrained quantities via Noether's theorem or selection rules. "Status in the SM" indicates whether the symmetry is exact, approximate, or broken at tree level.

Symmetry	**Group**	**Charge**	**Observables**	**Status in SM**
Lorentz	$SO(3,1)$	$x^\mu p^\nu - x^\nu p^\mu$	Invariant masses, angular distributions	Exact **[cite –KD]**
Translation	$\mathbb{R}^{1,3}$	p^μ	Missing–p_T	Exact **[cite –KD]**
Gauge	$SU(3) \times SU(2) \times U(1)$	hypercharges	Color flow in jets; W charge asymmetry; lepton universality	Exact locally **[cite –KD]**
Electroweak breaking	$\langle H \rangle \neq 0$	M_W, M_Z	$M_W, M_Z, \frac{M_W}{M_Z}$	Broken**[cite –KD]**
C	$q^\pm \leftrightarrow \bar{q}^\mp$	$\alpha_{q^\pm}$ vs. $\alpha_{\bar{q}^\mp}$	$\frac{e^+}{e^-}; \frac{\mu^+}{\mu^-}$	Conserved in EM/QCD; violated in weak **[cite –KD]**
P	$\boldsymbol{x} \to -\boldsymbol{x}$	handedness	lepton asymmetry in β-decay	Conserved in EM/QCD; maximally violated in weak **[cite –KD]**
CP	$C + P$	CKM matrix	B-meson asymmetry; electric dipole moments	Approx. broken **[cite –KD]**
T	$t \to -t$	QFT amplitudes	EDM searches; K meson T-violation	Broken if CP broken **[cite –KD]**
CPT	$C + P + T$	$m, \tau_{q^\pm}$ vs. $\tau_{\bar{q}^\mp}$	$m_p = m_{\bar{p}}, \tau_\mu = \tau_{\bar{\mu}}$	Exact **[cite –KD]**
Permutation	S_n / A_n	Bose/Fermi stats.	Jet ordering; boson correlation	Exact **[cite –KD]**

Beyond the Standard Model's built in symmetries, there are approximate global symmetries that often prove useful in collider physics. Examples include isospin symmetry, an $SU(2)$ symmetry treating up and down quarks as identical in the limit of equal masses, and flavour symmetries, like the $SU(3)$ of the light uds quarks, which are not exact, but underlie patterns in hadron production and decay. For instance, isospin symmetry implies that processes differing only by swapping an up quark with a down quark, such as producing a proton versus a neutron, have nearly equal cross sections, up to corrections from the up–down mass difference or electromagnetic effects. Similarly, the universality of physical laws under interchange of identical particles leads to permutation symmetry. If two particles of the same type appear in a final state, the probability distribution is invariant under exchanging them. In quantum terms this is enforced by (anti)symmetrization of identical particle states. In collider observables, permutation symmetry means that one cannot physically distinguish, say, which of two identical jets in an event is "jet 1" or "jet 2"—any labelling is arbitrary and the underlying physics treats the two jets on equal footing. When calculating cross sections, this symmetry is accounted for by dividing by the a symmetry factor to avoid over counting identical configurations. We will see that in data analysis one often has to impose an ordering, such as "leading" and "subleading" jet by momentum, for convenience, but the fundamental permutation invariance implies that any physical conclusion should not depend on this arbitrary ordering.

In summary, fundamental symmetries, Poincaré (Lorentz and translations), gauge invariances, and discrete symmetries like C, P, CP, as well as permutation invariance for identical particles, provide a set of invariance principles for particle interactions. These symmetries constrain the form of theoretical cross sections and transition rates. Many measurable quantities in colliders such as cross sections, angular distributions, etc., either reflect these symmetries, when they hold, or provide avenues to detect symmetry breaking when deviations from the expected invariant patterns are observed. However, the symmetries of nature at the fundamental level are not always manifest in what detectors actually record. We next turn to how the detector response and measurement process can modify or obscure these symmetries.

Symmetries in Detector Response Functions

A detector response function $r(x|z)$ describes the probability of observing a measurement outcome x given a true particle-level state z. This encapsulates effects of limited efficiency, acceptance, and resolution of a detector. An ideal detector with perfect coverage and resolution would preserve all physical symmetries present at the particle level. In reality, detectors often break or reduce symmetries that the underlying physics possesses. Understanding which symmetries are preserved, approximated, or lost in $r(x|z)$ is crucial

for interpreting measured data. Here we discuss several common invariances and how they are affected by realistic collider detectors in their response.

Spatial Uniformity and Rotational Symmetry

Most collider detectors are designed with a roughly cylindrical geometry around the beam axis, aiming for azimuthal symmetry. Hence they provide close to uniform coverage in the plane around the beam. In an ideal scenario, if the physical process yields a uniform distribution in the azimuthal angle ϕ (i.e. no preferred direction around the beam line), a perfectly symmetric detector would register an equal number of events in each azimuthal segment. In practice, small asymmetries creep in. For example, the detector may have support structures or cabling at certain angles, or irregular segmentation, leading to variation in efficiency with ϕ. The electromagnetic calorimeter (ECAL), as an illustration, might be segmented into modules that cover specific ϕ slices. Given this, events falling into the gap between modules could be recorded with lower efficiency or energy resolution, creating a slight $\phi-$dependence in the observed data even if the true distribution was uniform. Detectors often have periodic segmentation, meaning continuous rotational invariance is broken down to a discrete rotation symmetry, invariant only under rotations corresponding to full module spacings. As a concrete example, imagine a detector with 360 identical modules each covering $\Delta\phi = 1^\circ$. This detector is invariant under rotation by 1–degree increments, but a rotation by an arbitrary angle (say 0.5°) would lead to a different alignment of a particle's trajectory with respect to module boundaries, yielding a measurably different response.**[PhysRevD.111.072002 –KD]** Thus, the continuous symmetry is lost to a discrete one, and even that discrete symmetry may be imperfect if modules are not exactly identical or have time varying efficiency. Thus azimuthal symmetry at the physics level is usually preserved approximately by detector design, but slight non-uniformities in ϕ response are common and must be accounted for either via calibration or acceptance corrections.

Polar Coverage and Boost Invariance

Collider detectors also have limited coverage in the polar direction, along the beam axis. No real detector covers the full 4π solid angle; there is always a cut–off at some polar angle (or pseudorange η) beyond which particles escape detection. In particular, the forward regions close to the beam are notoriously difficult to instrument. This breaks the full spherical symmetry of space. A process that is symmetric under arbitrary rotations, such as a perfectly isotropic decay in its rest frame, will not appear isotropic in the laboratory measurement if a significant portion of the solid angle is unobserved. Detectors are typically symmetric under rotations about the beam axis but not under arbitrary rotations that tilt the beam axis, because the beam direction is a fixed axis of symmetry. In effect, the presence of incoming

beams singles out a preferred direction, the beam axis $\hat{z}$, and detectors are built around this axis. Consequently, the data may reflect cylindrical symmetry, invariant under $SO(2)$ rotations around $\hat{z}$, but not full $SO(3)$ rotational symmetry. This also connects to Lorentz boost invariance. While the underlying physics is Lorentz invariant, the detector is a fixed apparatus in one frame. A boost along the beam direction i.e. a change of reference frame moving with respect to the collision will generally change how events are distributed relative to the detector acceptance. For instance, consider a boost that causes particles to have higher longitudinal momentum; in the lab frame, more particles will end up at small polar angles, closer to the beam line, where detection efficiency is lower, thus the observed distribution of, say, pseudorapidity η will shift. The detector has a finite acceptance in η, so a Lorentz boost that moves events into the far forward region will result in a fraction of events being lost. Therefore, the measured distributions are not invariant under Lorentz boosts, even though the underlying parton–level kinematics can be expressed in Lorentz–invariant terms. Thus the physical construction of detectors break global translational and boost symmetry by virtue of being static and having edges. the data in the lab frame privileges the specific frame in which the detector is at rest. A high energy collision viewed in a different inertial frame is physically identical, but the detector at rest will record it differently unless one corrects for acceptance and inefficiencies.

Resolution Effects and Approximate Invariance

Even if a symmetry could hold in principle, the resolution and threshold effects of detectors often spoil exact invariance. A salient example arises with Lorentz invariance and invariant mass reconstruction. As a thought experiment, imagine a two–body decay producing a pair of muons, such as $Z^0 \to \mu^+\mu^-$. The true invariant mass of the muon pair is fixed, irrespective of the Z boson's momentum, because this is simply a Lorentz scalar. However, a detector measures muon momenta with finite precision, and that precision typically degrades at high momentum[10]. If a Z boson is produced nearly at rest in the lab, its decay muons have moderate momenta and the detector might reconstruct the invariant mass with a narrow resolution. If instead a Z is produced with a large boost, the muons each have higher lab–frame momenta, and the detector's momentum resolution broadens the reconstructed mass distribution. The result is that the distribution of reconstructed $m_{\mu\mu}$ for boosted Z events is broader (and potentially biased) compared to that for non–boosted events. Thus, a Lorentz boost, which should not matter to an ideal measurement, actually changes the statistical distribution of an observable due to detector response. This is an example of an approximately respected symmetry. At low boost the symmetry holds well, but at high boost the symmetry is effectively broken by detector effects. Similarly, thresholds

[10] tracking detectors determine momentum from curvature in a magnetic field, which becomes very small for high momentum muons, leading to larger relative uncertainty in the measurement.

in detector sensitivity (e.g. a calorimeter that only records energy above some minimum) can break symmetry under transformations that redistribute energy. A detector that is be equally efficient for electrons and positrons, suggesting C-symmetry in detection. However, if a process produces a wide spectrum of energies, a cut on low–energy particles could cause a difference. More low-energy e^+ are likely to be lost than e^- due to different interaction rates with material. In such cases, even a symmetry of the physics might not result in equal measured counts.

Detector Mirror and Charge Symmetry

Detectors are not usually built to be fully symmetric under parity inversion or charge conjugation, even though we often assume these symmetries for the relevant physics should reflect in data. A parity inversion would swap what we call "forward" and "backward" directions in the detector. If the detector has identical coverage in the forward ($+z$) and backward ($-z$) hemispheres, one could say it is parity–symmetric with respect to the interaction point. Many detectors strive for this by having symmetric endcaps on both sides of the interaction region. However, even then, subtle asymmetries can exist because it is not feasible to prevent one side from having a slightly different material distribution or a different calibration from the other. As a result, a process that is forward-backward symmetric in physics[11] might show a small forward-backward asymmetry in the raw data if. Experiments typically correct for such differences by equalizing calibrations, but the intrinsic detector response can break the symmetry. Likewise, charge conjugation symmetry in detection would mean the detector is equally sensitive to positive and negative charges. While the detector electronics and geometry generally don't prefer one charge sign, magnetic fields introduce a notable asymmetry, because charged particles bend in opposite directions, and this can lead to charge–dependent acceptance. In a magnetic spectrometer, positive particles bend outward in one direction and negatives in the opposite. If the acceptance boundaries, like the edge of the detector volume, cut off tracks in one curvature direction more than the other, one will observe a difference in detection rates for $+$ vs $-$ even if production is symmetric. Another example is that the different interaction of e^+ and e^- with matter could lead to slightly different detection efficiencies. These are second-order effects, but they illustrate that a detector is a physical object that need not respect the abstract symmetries of the theory. Careful simulations and calibrations are performed to quantify and mitigate these asymmetries in collider experiments.

[11] In pp collisions at the LHC, the two beam directions are equivalent so the distribution of particles as a function of rapidity y should be symmetric about $y = 0$

Permutation Symmetry and Identical Particles

A subtle aspect of detector response is its effect on permutation symmetry between identical particles in an event. Physically, as noted, swapping two identical particles should change nothing in an ideal measurement. Detectors, however, could introduce differences. Two identical particles (say two photons) that go into different regions of the detector can have their energies might be measured with different resolutions or one might pass quality cuts and the other fail due to region–specific noise. s a result, the joint distribution of the two–particle system in the measured data might not be symmetric under exchange, even though it was at truth level. As a simple example, consider two jets in an event where, at the particle level, the probability $P(E_1, \eta_1; E_2, \eta_2)$ is symmetric under $(1 \leftrightarrow 2)$. After detection, suppose jet 1 falls in the central barrel, with excellent energy resolution and jet 2 falls in the forward region, with poorer resolution and lower efficiency. The measured energies $E_1^{(\text{meas})}$ and $E_2^{(\text{meas})}$ will have different response smearing. If one then orders jets by measured energy and calls the highest E jet the "leading" jet, the distribution of leading and subleading jet energy will not mirror one another exactly. Effectively, the detector–induced asymmetry has assigned labels to the jets where none existed. Analysts must be wary of these effects; one option used is to "symmetrize" the analysis if possible to recover the permutation symmetry that the physics assures.

?? summarizes a few key examples of how an ideal symmetry at the particle–level can be broken or reduced by detector effects. These examples illustrate why fully accounting for detector response is essential when testing physical symmetry hypotheses with data.

Despite these challenges, experimentalists strive to design detectors with as much symmetry as feasible and to correct for known asymmetries. For instance, collider detectors often have nearly full 2π azimuthal coverage and layered symmetries, segmenting in ϕ and η uniformly, specifically to preserve rotational invariance and facilitate combining data over symmetric regions. Detector simulation and calibration are used to quantify symmetry breaking. If a $\phi-$dependence is observed in calibration data, it can be corrected so that the final analysis treats those variations as a systematic uncertainty or removes them. Nonetheless, the reality remains that physical symmetries can fail to translate into measured symmetries. In the language of probability distributions, if $p(z)$ is invariant under transformation T, but the response $r(x|z)$ is not invariant in the corresponding way, then the folded distribution $p(x) = \int r(x|z)\, p(z), \mathrm{d}z$ will not be invariant under T applied to x. Only if both p_{truth} and r share the symmetry T will p_{data} exhibit it. This conceptual understanding is vital when one interprets measured cross sections and tries to infer or discover symmetries from data. One must always ask: is an observed symmetry or asymmetry coming from the physics, or from the detector?

Table 9.3: Ideal symmetry expectations versus typical detector-induced symmetry-breaking effects in collider experiments.

Transformation	**Ideal Outcome**	**Detector Effect**
Azimuthal Rotation	Uniform event distribution in azimuthal angle ϕ; no preferred direction around the ring.	Slight ϕ non–uniformity in measured data due to detector segmentation and gaps. Only invariant under discrete rotations matching module periodicity.
Boost	Physics unchanged under change of inertial frame; kinematic distributions described in Lorentz-invariant terms.	Detector at fixed orientation not boost-invariant. Boost pushes more particles into forward regions, worsens some resolutions. Observed distributions depend on the detector rest frame.
Parity	If interaction is P–symmetric, processes occur equally in mirrored coordinates.	Detector not mirror–symmetric. Differences between $+z$ and $-z$ hemispheres introduce forward—backward asymmetries in measured yields; calibrations are required.
Charge Conjugation	If physics is C–symmetric, particles and antiparticles produced at equal rates with identical kinematics.	Charge asymmetric response is common. Magnetic bending plus finite acceptance can cause differential detection. Likewise e^+/e^- or μ^+/μ^- efficiencies can differ, biasing measured particle/antiparticle counts.
Permutation/Relabelling	Complete symmetry under exchange; the joint distribution $P(z_1, z_2)$ is invariant if two identical particles' momenta or labels are swapped.	Measured joint distributions can change upon swapping when the two objects land in detector regions with different responses. A jet in the central region and one forward suffer different energy smearing than the opposite arrangement. Imposing an arbitrary ordering (leading/sub=-leading) can hide the underlying symmetry.

How Symmetries Manifest in Measured Cross Sections

Given the above considerations, one can now examine how symmetries and symmetry violations are reflected in the measured distributions that experiments record and report. A measured cross section differential in some observable is effectively a statistical aggregate of many collision events, after selection cuts and corrections. If the underlying physics possesses a symmetry, one might expect the differential cross section to reflect that, provided the measurement process does not hide or distort it. In practice, one observes in histograms a mixture of genuine physical symmetry patterns and effects of detector acceptance or sample selection.

Exact Symmetries and Flat Distributions

A hallmark of a symmetry in a distribution is a repeated pattern indicating invariance. For example, consider azimuthal invariance in a proton–proton collision. Since the colliding protons provide a cylindrically symmetric initial state of two identical beams head-on, no physics process at the parton level prefers a particular ϕ direction. Consequently, the true differential cross section $\frac{d\sigma}{d\phi}$ for an inclusive process is invariant under the transformation $\phi \mapsto \phi + \delta\phi$ (aside from small QED effects or residual detector magnetization influences). If the detector has uniform ϕ coverage and the analysis has no $\phi-$dependent cuts, the measured distribution of events as a function of ϕ should be approximately flat. Any significant deviation from flatness might indicate an instrumental problem or a selection bias.

Once the symmetry has been established, one might combine data from all ϕ slices (since they are equivalent) to improve statistical precision, effectively using the symmetry to gather more data. However, as noted, small modulations can appear if certain detector modules deviate from the rest; these are corrected or quoted as systematic uncertainties. Another example is rapidities in symmetric collisions: in a pp collider at equal beam energies, the center-of-mass frame coincides with the lab frame, and the process is symmetric under exchanging the two beam directions. This implies that the distribution of particles in rapidity y is symmetric about $y = 0$ for processes that do not involve a bias (for instance, pure QCD dijet production should yield a symmetric $d\sigma/dy$ for jets, with equal activity in the forward ($+y$) and backward ($-y$) hemispheres). This is why measurements of inclusive jet or hadron yields often present results as a function of $|y|$ or $|\eta|$ (absolute rapidity or pseudorapidity), invoking the symmetry $y \leftrightarrow -y$ to double the statistics and simplify presentation. The physical symmetry (identical proton beams) justifies this, and one checks that, within uncertainties, the $+y$ and $-y$ distributions are consistent before merging. Thus, a symmetry in initial conditions and dynamics (here, invariance under $y \to -y$) leads to a clear symmetry in the measured cross section (equal yields for $\pm y$). If an unexpected asymmetry were observed, it would either signal new physics (e.g. a CP-violating effect or a

bias in parton distribution functions) or, more likely, an issue like a mismodeled detector efficiency gradient.

Symmetries in Kinematic Shapes

Symmetries often impose recognizable shapes or constraints on distributions. For example, energy and momentum conservation require that for each event, the vector sum of momenta of final state particles equals that of initial state. As a result, distributions of total transverse momentum in events would be expected to peak at zero, and any significant imbalance indicates e.g. neutrinos or detector holes. This is not a symmetry in the sense of a group acting on one event's space, but rather a deterministic constraint on the ensemble. The distribution of missing momentum should be centered at zero and isotropic in azimuth. Experiments can thus verify that the missing transverse momentum vector has no preferred direction to validate rotational symmetry and momentum conservation in aggregate.

If one measures the transverse momentum spectrum of the first jet against that of the second jet in dijet events (with jets ordered by p_T), there is no fundamental reason for these spectra to differ except for the ordering bias. The leading jet p_T distribution will be harder by construction, and the subleading softer, any jet is equally likely to be at a given p_T as its partner, aside from that ordering. This symmetry can therefore by verified through the similarity between the distribution of subleading jet p_T and the leading jet p_T distribution of a lower–energy subset, or by symmetrizing the dataset by swapping jets event by event and seeing no change in overall two jet correlation distributions. In summary, wherever a symmetry exists, one finds redundancies or equalities in the measured spectra: sections of phase space that should mirror other sections. Experimental analyses often exploit this to measure detector backgrounds, by assuming that an uninstrumented region should have similar counts as a well-instrumented region after normalization.

Interplay of Physical and Detector Symmetries

It is important to disentangle which symmetries in a measured cross section come from physics and which from measurement procedure An analysis might impose a cut that itself introduces a symmetry or asymmetry. When presenting a measured cross section, unfolding detector effects to reconstruct particle level distributions to the extent possible, to report a cross section as it would appear with an ideal detector, can restore the symmetries that belong to the physics by removing the distortions of measurement**[cite –KD]**. For example, if the raw data show a $\phi-$dependence due to detector inefficiency, the unfolded cross section vs ϕ should be flat (with larger uncertainties reflecting the correction). In this sense, symmetries provide a consistency check. If after unfolding one still sees a symmetry violation in a quantity that should be symmetric by physics, the unfolding procedure might

be flawed. Conversely, if a symmetry is expected to be broken by physics, one must be careful to ensure the detector is not distorting that asymmetry. For instance, measuring a forward-backward asymmetry in top quark production, a sign of potential new physics or weak-interaction interference, requires excellent control of any detector differences between the forward and backward directions so that the observed asymmetry can be trusted as physical.

A physical symmetry is a property of the underlying probability law. It requires equal probabilities for events and their transformed versions. A statistical symmetry of a dataset entails that the finite sample of observed data appears invariant under some transformation, within the limits of noise, so that with large data, one expects the symmetry to become apparent as the fluctuations average out. If deviations persist significantly beyond expected fluctuations, that flags either a real symmetry violation or unaccounted systematics.

Challenges in Identifying Symmetries from Noisy Data

Identifying symmetries in experimental data is not always straightforward. Noisy data, stemming from finite statistics, background processes, and detector imperfections, can obscure or mimic symmetry signals. This section outlines the main challenges one faces in discerning true invariances or symmetry violations within collider datasets, and the need for methods like the SymmetryGAN approach developed later in this work to address these challenges.

A fundamental challenge is that any empirical distribution has random fluctuations. If an underlying distribution is perfectly symmetric (say truly uniform in ϕ), a finite sample will still exhibit some variation across ϕ bins. Hence any symmetry discovery method must have a mechanism to distinguish a real asymmetry from a mere fluctuation.

Conversely, an underlying asymmetry can be washed out by limited statistics. This is especially pertinent in searches for new symmetries or violations. The signals are often at the level of small deviations and can be difficult to detect over statistical fluctuation. Moreover, multiple comparisons increase the probability that one finds an apparent “symmetric pattern" in some projection of the data purely by chance.

As discussed, detector effects can induce or conceal asymmetries. Often the largest uncertainties in measuring symmetry come from how well we understand the detector. For example, in measuring a forward–backward asymmetry, uncertainties in the relative efficiency of the forward and backward region directly translate to uncertainty in the asymmetry observable. If those efficiencies are poorly known, one might not be able to distinguish a symmetry violation from detector bias.

Similarly, backgrounds, other processes that mimic the signal, might not share the symmetry of the signal. Suppose one is looking for a symmetry in a certain particle decay distribution; if there is a significant background from a different process that does not

respect that symmetry, the combined data will appear to break the symmetry even if the signal alone is symmetric. Careful background subtraction or isolation is required. In practice, identifying a symmetry often involves comparing two distributions (e.g. $P(x)$ vs $P(Tx)$ for some transformation T) and seeing if they differ. If they do, one must estimate if the difference is due to known systematic effects. This typically demands high-precision calibration. For instance, to confirm CP symmetry in production of particle vs antiparticle to the 10^{-3} level, one needs detector efficiencies known to better than 0.1% between positively and negatively charged particle detection.**[cite –KD]**

Dimensionality challenges

Collider events are high dimensional, consisting of many particles with various kinematic attributes. A symmetry might not be evident in any single one dimensional projection, but rather in a complicated combination of variables. For example, Lorentz invariance is best seen when considering all four–momenta together or invariants like masses; a naive look at just one momentum component would not show it. Permutation symmetry in a multijet event is a property of the joint distribution of all jet momenta, not necessarily obvious if one only looks at single–jet spectra.

This is where machine learning methods become attractive, because they can, in principle, detect subtle patterns in high dimensional data. However, even ML models need guidance. The space of possible transformations is huge, and hence searching it naively for invariances is intractable. Hence traditional methods often restrict attention to physically motivated symmetry transformations (rotations, reflections, boosts, particle exchanges, etc.).

Since scanning for symmetries by comparing all possible pairs of transformed distributions is computationally prohibitive, as data volumes grow and analysis spaces become more complex, we need more automated symmetry discovery mechanisms. The SymmetryGAN approach discussed in this thesis is one attempt to automate the discovery of symmetries by leveraging generative adversarial networks. Conceptually, SymmetryGAN will trains a generator (applying candidate transformations) against a discriminator to test if the transformed data looks statistically identical to the original data. If the generator generates a transformation under which the discriminator is maximally confounded, that transformation corresponds to a symmetry of the data distribution. Implementing this is challenging: the model must search a continuous space of transformations, handle approximate symmetries, and avoid trivial solutions. A careful choice of network architecture using known equivariances are needed to make such learning robust.**[cite –KD]**

In summary, identifying symmetries from noisy collider data requires

1. Sufficient statistics and rigorous statistical tests to differentiate real invariances from fluctuations,
2. Precise control of detector systematics to avoid mistaking detector effects for or against symmetry,
3. Methods to probe high-dimensional and subtle symmetry patterns that might elude simple binned analyses, and
4. Methodological consideration to handling approximate symmetries in a principled way.

These challenges motivate the development of tools like SymmetryGAN, which I will introduce in the next sections. Such tools aim to combine physical insight with machine learning's ability to detect patterns, thereby providing a statistical discovery framework for symmetries. SymmetryGAN and similar approaches offer a promising path to unveil symmetries that are latent in complex data. The rigorous understanding of symmetry and symmetry–breaking provided in this section will form the foundation on which those computational methods build, ensuring that any discovered "symmetry" is physically meaningful and relevant to the challenges of unfolding and analyzing collider data.

9.3 Statistical Definition of Dataset Symmetries

The concept of symmetry in physics typically evokes images of rotational invariance in crystals, parity conservation in weak interactions, or gauge transformations in field theory. Yet when we turn our attention to experimental data, especially the high dimensional datasets emerging from modern collider experiments, the notion of symmetry becomes surprisingly subtle. What does it mean for a collection of measured events to possess a symmetry? This question, deceptively simple in appearance, reveals profound connections between statistical inference, group theory, and the fundamental challenge of unfolding detector effects from observed data.

Distinction between point and dataset symmetries

Statistical symmetries in datasets represent a fundamental departure from traditional geometric symmetries, requiring careful consideration of probability measures, transformation Jacobians, and reference densities. This section explores the mathematical foundations, practical applications, and machine learning approaches to understanding and leveraging dataset symmetries. The distinction between symmetries of individual data elements and entire datasets lies at the heart of statistical theory of symmetries. For individual data points,

symmetry is characterized by simple invariance: a transformation g preserves element x if $g(x) = x$. This represents a straightforward geometric notion where specific points remain fixed under transformation. Distribution level symmetries, however, operate on probability measures rather than individual points. A measure space X with probability measure μ exhibits symmetry under group G when the measure remains invariant.**[–KD]**

$$\forall A \subseteq X \, \forall g \in G \, \mu(A) = \mu(g(A)) \tag{9.2}$$

This measure theoretic definition captures the statistical properties of entire distributions rather than individual elements.

The critical insight, as formalized in the SymmetryGAN framework, is that dataset symmetries are ambiguous due to Jacobian factors introduced during coordinate transformations.**[–KD]** Unlike point symmetries, where transformations either preserve or don't preserve specific locations, dataset symmetries must account for how probability densities transform under coordinate changes. This fundamental difference necessitates the introduction of inertial reference densities to properly define statistical symmetries.

Inertial reference densities and their theoretical role

The concept of inertial reference densities emerges as a necessary theoretical construct, analogous to inertial frames in classical mechanics.**[–KD]** These reference densities provide a baseline against which statistical symmetries can be meaningfully defined, resolving the ambiguity inherent in coordinate transformations of probability measures.

In the formal framework, a reference density $\rho(x)$ establishes a canonical measure for comparing probability distributions, enabling the definition of relative entropy

$$H_\rho[X] = -\mathbb{E}_\mu[\log \frac{\mathrm{d}\mu}{\mathrm{d}\rho}]. \tag{9.3}$$

Additionally, it provides a coordinate–independent way to specify symmetry transformations, ensuring that symmetry definitions remain consistent across different parameterizations of the same statistical manifold. This can be analogized to phase space shifting operations that leave the Gibbs integration measure invariant can be understood as gauge transformations, with the reference density playing the role of a gauge fixing condition.**[Phys. Rev. Lett. 133, 217101 (2024) –KD]**

The mathematical machinery for statistical symmetries centers on how probability densities transform under coordinate changes. Under a transformation $X = g(Y)$, probability densities transform as

$$p_y(y) = p_x(g(y))|\det(g'(y))| \tag{9.4}$$

This transformation law, involving the Jacobian determinant $|\det(g'(y))|$, ensures probability conservation. The Jacobian factor measures how volumes scale under transformation. $|\det(J)| = 1$ characterises volume preserving transformations like rotations and reflections.

For a probability distribution to exhibit symmetry under group action G, it must satisfy the condition

$$\forall g \in G \, \forall x \in X \, p(x) = p(g(x)) \, |\det g'(x)| \tag{9.5}$$

This condition is far more restrictive than point symmetry, as it demands global consistency across the entire probability measure.

The group-theoretic formulation provides additional structure. A group G acts on a probability space (, F,) through measurable maps that preserve the -algebra structure. ScienceDirect The orbit-stabilizer theorem applies, decomposing the action into orbits and stabilizers, while representation theory enables systematic construction of invariant functions and decomposition of function spaces into irreducible components. Jacobian factors and volume preservation Jacobian determinants play a pivotal role in statistical symmetries, serving as the bridge between geometric transformations and probability conservation. The determinant encodes how infinitesimal volumes transform, with several key properties making it essential for statistical applications. Wikipedia The multiplicative property of Jacobians, det(D(g g)(x)) = det(Dg(g(x))) · det(Dg(x)), ensures that group composition translates correctly to probability transformations. For inverse transformations, det(Dg¹(g(x))) = 1/det(Dg(x)), maintaining consistency of the transformation group structure. Volume preservation, characterized by |det(Dg(x))| = 1, defines a special class of transformations called unimodular transformations. Wikipedia These transformations, including rotations, reflections, and translations, preserve statistical properties like moments and correlations. Wikipedia For such transformations, E[f(X)] = E[f(g(X))] for any measurable function f, making them particularly important for statistical inference. Recent theoretical work quantifies how group symmetries improve sample complexity. For finite groups of size n, the sample complexity reduction is proportional to n for Wasserstein-1 distances and Lipschitz-regularized divergences. For infinite compact groups, the improvement depends on the intrinsic dimension of the fundamental domain, characterized by covering numbers. arxiv +2 Applications to particle physics data analysis Particle physics provides a rich domain for applying statistical symmetries, particularly in detector calibration, data unfolding, and inference tasks at Large Hadron Collider experiments. arXiv The field distinguishes three types of symmetries: geometrical (space-time invariance), dynamical (observer relationships), and statistical (field theory properties). WikipediaNational Academies Press In detector response modeling, statistical symmetries constrain how identical particles must be treated. Permutation symmetry requires that detector analysis respect the indistinguishability of identical particles, affecting event reconstruction algorithms, particle identification methods, and background estimation techniques. PNAS

Response matrices used in unfolding procedures must respect particle exchange symmetries, with regularization procedures preserving these symmetry properties. Unfolding represents a critical application where statistical symmetries guide the correction of detector effects. The process must preserve the statistical symmetries of the underlying physics, maintaining correlations between identical particles and ensuring conservation laws implied by symmetries remain valid. Modern approaches like "Full Phase Space Unfolding" demonstrate how statistical symmetries constrain unfolding procedures in high-dimensional phase spaces. LHC experiments extensively utilize these principles. The combination of ATLAS and CMS results for Higgs properties required consistent treatment of gauge symmetries across experiments, proper handling of statistical correlations, and symmetry-preserving combination procedures. BNL Newsroom Background estimation methods exploit symmetry properties through carefully designed control regions, while systematic uncertainties account for potential symmetry violations. The SymmetryGAN approach SymmetryGAN represents a breakthrough in automatically discovering dataset symmetries through deep learning. The framework addresses the fundamental question of identifying symmetries without prior knowledge, using a modified generative adversarial network architecture combined with the theoretical foundation of inertial reference densities. arXivarXiv The architecture consists of a generator network that learns symmetry transformations by parameterizing group elements, a discriminator that distinguishes between original and transformed data, and crucially, an inertial reference dataset that ensures statistical validity of discovered symmetries. The training process encourages the discovery of true symmetries while maintaining the data distribution through a carefully designed loss function. Empirical results demonstrate SymmetryGAN's effectiveness across diverse applications. On Gaussian distributions, it accurately discovers rotational and translational symmetries. Applied to simulated Large Hadron Collider dijet events, it successfully identifies the complex symmetries present in high-energy physics data. arXivarXiv The approach handles both discrete and continuous symmetries, providing a general framework for symmetry discovery. Beyond SymmetryGAN, the field has developed numerous approaches. Latent LieGAN (LaLiGAN) extends symmetry discovery to nonlinear group actions by mapping data to latent spaces where symmetries become linear. IBM Neural symmetry discovery methods use deep networks to model both transformations and generators, constructing loss functions that ensure closed algebra structure. These approaches connect to broader themes in representation learning, where symmetry-aware learning naturally leads to disentangled representations. Connections to measurement and inference Statistical symmetries fundamentally shape measurement and inference tasks through multiple pathways. The gauge invariance framework in statistical mechanics reveals that phase-space shifting operations preserve all physically meaningful quantities while providing new computational approaches for molecular simulations. APS Physics These symmetries lead to exact correlation relations between forces and observable properties, offering systematic ways to derive new statistical relations. In machine learning,

encoding known symmetries dramatically improves performance. MIT research proves that symmetry encoding reduces sample complexity exponentially, with multidimensional symmetries providing disproportionately large returns. MIT NewsTilos Equivariant neural networks, designed to respect known symmetries through architectural constraints, achieve superior data efficiency and generalization. arXiv +2 The theoretical framework connects reference measures, symmetry principles, and statistical inference in profound ways. Reference measure selection can be guided by symmetry considerations, leading to principled approaches for prior selection in Bayesian inference. Encyclopedia of Mathematics Symmetry principles suggest natural classes of invariant estimators, while statistical analogs of physical conservation laws emerge from symmetry considerations, constraining the behavior of statistical systems. PNAS This unified framework reveals deep connections between statistical mechanics, information theory, and quantum theory. Gauge invariance provides a unifying perspective on various statistical phenomena, offering concrete advantages for computational statistics and machine learning while opening new directions for both theoretical development and practical applications. Conclusion Statistical dataset symmetries represent a fundamental advance in our understanding of how symmetries apply to probabilistic systems. The distinction from point symmetries, necessitating inertial reference densities and careful treatment of Jacobian factors, reflects the inherently different nature of statistical versus geometric symmetries. From the mathematical foundations rooted in measure theory and group actions to practical applications in particle physics and machine learning, this framework provides powerful tools for understanding and exploiting the structure in statistical data. The SymmetryGAN approach and related machine learning methods demonstrate that these theoretical insights can be translated into practical algorithms for symmetry discovery and utilization. As the field continues to develop, the connections between symmetry, information, and inference promise to yield further insights into the fundamental nature of statistical systems and enable more powerful approaches to data analysis across scientific domains.

Chapter 10

Synthesis and Comparative Analysis

- Unified framework for understanding unfolding approaches
- Progressive evolution from binned to fully unbinned methods
- Comparative analysis: accuracy, precision, and computational demands
- Decision framework for method selection based on needs
- Relevance to Current and Future Experiments
- Common themes and lessons learned
- Interplay between physics knowledge and data-driven approaches

Chapter 11

Conclusions and Future Directions

- Summary of key contributions
- Impact on cross section measurements
- Open challenges in unfolding methodology
- Potential extensions
- Future research directions

Appendix A

Mathematical Derivations

- Various analytical proofs
- Statistical properties of the methods

Appendix B

Implementation Details

- Datasets
- Specific network architectures
- Hyperparamter sets
- Optimization startegies
- Code repositories

Appendix C

Supplementary Results

- Other studies performed not included in main text
- Robustness and sensitivity checks when varying hyperparameters