Published for SISSA by Springer

Received: November 11, 2025
Revised: January 28, 2026
Accepted: February 28, 2026
Published: April 23, 2026

Neural fake factor estimation using data-based inference

Jan Gavranovič[a,b] **Lara Čalić**[c] **Jernej Debevc**[a,b] **Else Lytken**[c] **and Borut Paul Kerševan**[a,b]

[a] *Jožef Stefan Institute, Jamova cesta 39, Ljubljana, Slovenia*
[b] *Faculty of Mathematics and Physics, University of Ljubljana, Jadranska ulica 19, Ljubljana, Slovenia*
[c] *Division of Particle and Nuclear Physics, Lund University, Professorsgatan 1b, Lund, Sweden*

E-mail: jan.gavranovic@ijs.si, lara.calic@fysik.lu.se, jernej.debevc@ijs.si, else.lytken@fysik.lu.se, borut.kersevan@ijs.si

Abstract: In a high-energy physics data analysis, the term "*fake*" backgrounds refers to events that would formally not satisfy the (signal) process selection criteria, but are accepted nonetheless due to mis-reconstructed particles. This can occur, e.g., when leptons from secondary decays are incorrectly identified as originating from the hard-scatter interaction point (known as *non-prompt leptons*), or when other physics objects, such as hadronic jets, are mistakenly reconstructed as leptons (resulting in *mis-identified leptons*). These *fake leptons* are usually estimated using data-driven techniques, one of the most common being the Fake Factor method. This method relies on predicting the fake lepton contribution by reweighting data events, using a scale factor (i.e. fake factor) function. Traditionally, fake factors have been estimated by histogramming and computing the ratio of two data distributions, typically as functions of a few relevant physics variables such as the transverse momentum p_{T} and pseudorapidity η. In this work, we introduce a novel approach of fake factor calculation, based on density ratio estimation using neural networks trained directly on data in a higher-dimensional feature space. We show that our method enables the computation of a continuous, unbinned fake factor on a per-event basis, offering a more flexible, precise, and higher-dimensional alternative to the conventional method, making it applicable to a wide range of analyses. A simple LHC open data analysis we implemented confirms the feasibility of the method and demonstrates that the ML-based fake factor provides smoother, more stable estimates across the phase space than traditional methods, reducing binning artifacts and improving extrapolation to signal regions.

Keywords: Electroweak Precision Physics, Jets and Jet Substructure, Left-Right Models

ArXiv ePrint: 2511.06972

https://doi.org/10.1007/JHEP04(2026)188

Contents

JHEP04(2026)188

1 Introduction

Monte Carlo (MC) simulations are widely used in high-energy physics (HEP) to model physics processes and the behavior of particles and their interactions within a detector. However, while this approach is very successful in general, it specifically fails to accurately describe occurrences where the detector and reconstruction algorithms *failed* to perform as expected, e.g. to correctly reconstruct and identify leptons. The underlying cause is that the detector response is complex and consequently not understood well enough to be accurately modeled in MC simulation. Furthermore, while in the recent years there has been significant progress in the accuracy of MC simulations describing these mis-reconstructed and mis-identified objects, the other intrinsic MC limitation remains, namely the limited statistics of the simulated samples. This is especially true for processes with large cross-sections, where the mis-reconstructed objects appear only in small fractions of events, leading to very limited MC statistics in the kinematic regions of interest. A good example also comes from flavor physics, where an additional limitation comes from the large number of hadron decay modes that must be accounted for, often involving partially reconstructed final states, which makes a complete simulation-based description impractical [1]. To sum up, the *data-driven* approaches are used to estimate the contributions of such types of processes directly from data.

One of the approaches commonly used for this purpose is the Fake Factor method [2, 3]. It is used to estimate the contribution of events with mis-reconstructed and/or mis-identified leptons (collectively called *fake leptons*) in the kinematic region where one tries to measure

(discover) a physics process of interest, often referred to as the *signal region* (SR). The estimation is done by extrapolating the fake lepton contribution from data in a kinematically adjacent, *looser* kinematic region ($\mathrm{SR^L}$),[1] where the term *loose* is commonly used and refers to the implemented lepton selection requirements that are less stringent than the usual (tight) selection used in the SR. The extrapolation is done using a scale factor (i.e. fake factor) function.

The fake factor is thus defined as the ratio of two probability density functions, given approximately by the number of events with a fake lepton passing the nominal selection criteria to the number of events with a fake lepton passing looser selection criteria. The fake factor function itself is typically evaluated in a dedicated, fake lepton enriched, kinematic *control region* (CR) by using either nominal or looser selection criteria ($\mathrm{CR^L}$). The fake factor function, derived in a standard way, has binned (piece-wise constant) values, since it is estimated by binning (histogramming) the data and computing the ratio of the two data distributions in CR and $\mathrm{CR^L}$, typically as functions of a few relevant physics variables such as p_{T} and $|\eta|$. Depending on the analysis, additional variables correlated with the overall event topology, such as $E_{\mathrm{T}}^{\mathrm{miss}}$, may also be included. Such variables are not intrinsically related to lepton identification, but are instead sensitive to variations in event topology, background composition, and global event kinematics. In practice, this indirect (non-intrinsic) dependence is often reduced by parameterizing the fake rate in terms of variables more directly connected to the fake-lepton production mechanism, accounting for the mother parton type and kinematics, for example reconstructing the mother particle p_{T} [4]. The CR is designed to be enriched in fake leptons, while still being kinematically similar to the SR. The fake factor function is then applied to the events in $\mathrm{SR^L}$ to extrapolate the background contribution coming from fakes to the SR. This binned Fake Factor method however suffers from several challenges, such as binning selection, extrapolation uncertainties to the SR, and the limited ability to be parameterized along more than a couple of variables due to lack of statistics, to name a few. To overcome these limitations, we propose a novel approach, based on machine learning (ML) using (real) data, which can be applied in higher-dimensional feature spaces and provides a continuous, unbinned fake factor estimate on a per-event basis that extrapolates better to the SR.

In this paper we aim to give an introduction to the ML-based Fake Factor method and demonstrate its advantages compared to the traditional binned Fake Factor method, especially in terms of interpolation power lost due to binning artifacts, as well as its potential to extend to higher-dimensional feature spaces. The paper aims to demonstrate broad applicability of this method to diverse analyses in high energy physics, especially those involving leptons in final states. The paper is organized as follows. Section 2 provides an overview of the binned Fake Factor method and its connection to the related Matrix and ABCD methods, and highlights the challenges of these methods. Section 3 introduces the ML-based Fake Factor method. In section 4, we validate the ML-based approach using the ATLAS Open

[1] In this paper, the label $\mathrm{SR^L}$ corresponds to the region defined by the loose lepton selection and excluding events passing the nominal (tight) selection. This region corresponds to what is sometimes also referred to as the *application region* (AR) in the Fake Factor method. Analogously, $\mathrm{CR^L}$ refers to the control region with the loose lepton selection applied.

JHEP04(2026)188

Data sample and present the training results. Section 5 compares the final results of the ML-based and binned Fake Factor methods. Finally, section 6 summarizes our conclusions.

2 Overview of data-driven background estimation methods

The most popular data-driven methods for estimating fake lepton backgrounds in HEP analyses are the Matrix method, the Fake Factor method, and the ABCD method [2, 3]. All three methods rely on defining two different lepton selection criteria: a *tight* selection (the nominal selection used in a physics analysis, i.e. the signal region) and a *loose* selection (a looser selection that defines an adjacent kinematic region, excluding the tight one). The union of both categories is called the *baseline* selection. By definition, the set of leptons passing the tight selection must be a subset of those passing the baseline selection and does not include loose leptons.

We define the real efficiency r as the probability that a correctly reconstructed (real) lepton passing the baseline selection also passes the tight selection, and, analogously, the fake efficiency f as the probability that a fake lepton passing the baseline selection also passes the tight selection. In the simple case of a single lepton, the relationship between the numbers of tight and loose leptons in data and their composition in terms of real and fake leptons can be written as

$$\begin{pmatrix} N^{\mathrm{T}} \\ N^{\mathrm{L}} \end{pmatrix} = \begin{pmatrix} r & f \\ 1-r & 1-f \end{pmatrix} \begin{pmatrix} N_r \\ N_f \end{pmatrix} . \tag{2.1}$$

Both r and f are to be estimated in separate dedicated kinematic regions (control regions). Using (state-of-the-art) MC simulation to estimate the real efficiency r can be considered as reliable due to a good understanding of real lepton reconstruction and identification efficiencies in the detector. Additional data-driven corrections (scale factors) can be applied to MC to further improve the modeling of real leptons. On the other hand, the fake efficiency f is generally not modeled to the same precision in MC simulation, so it is typically estimated directly from data in a fake-lepton enriched control region.

2.1 The Matrix method

The Matrix method estimates the number of fake leptons in the tight region by explicitly inverting eq. (2.1), relating the numbers of tight and loose leptons in data to their composition in terms of real and fake leptons. The method can in principle be made fully data-driven, as it does not need to rely on MC simulation to estimate any contributions, but it can suffer from possible measurement bias, as it uses data information from the signal region itself.[2]

The number of events with a fake lepton using the tight (i.e. nominal) selection, found by inverting the above equation, is given by

$$N_f^{\mathrm{T}} = f N_f = \frac{f}{f-r}\left((1-r)N^{\mathrm{T}} - rN^{\mathrm{L}}\right) . \tag{2.2}$$

[2]In other words, the analysis is not completely *blinded* in the SR before the final statistical analysis when using this method, and it can thus impact the evaluation of possible signal presence.

JHEP04(2026)188

This equation relates the number of events in the tight and loose regions to the expected number of fake leptons in the tight region. As already stated, since N^{T} (i.e. the measured data in the SR) appears in the equation, information from the signal region enters the estimate, so the method is not fully blinded.

2.2 The Fake Factor method

The Fake Factor method estimates the contribution of events that only contain real leptons, i.e. $N_r^{\mathrm{T}} = rN_r$ and $N_r^{\mathrm{L}} = (1-r)N_r$ in eq. (2.1), from MC simulation predictions. The method is thus no longer fully data-driven, as it relies on MC to estimate the contribution of events with real leptons. Nonetheless, as already argued, real leptons are generally well modeled in MC, so this is not a significant degradation. On the other hand, the method does not utilize any data information from the SR, which gives it a formal advantage in terms of possible measurement bias with respect to the Matrix method, as this method can be used with a fully blinded signal region.

The number of fake leptons in the loose region can be obtained from eq. (2.1) as

$$N_f^{\mathrm{L}} = N^{\mathrm{L}} - N_r^{\mathrm{L}} = (1-f)N_f \,. \tag{2.3}$$

Using eq. (2.1) with eq. (2.3), we can express the number of fake leptons in the tight region as

$$N_f^{\mathrm{T}} = fN_f = FN_f^{\mathrm{L}} \,, \tag{2.4}$$

where F is the fake factor defined as

$$F = \frac{f}{1-f} \,. \tag{2.5}$$

Alternatively, eq. (2.4) can be derived directly from eq. (2.1) by multiplying the left-hand side by a row vector with the entries $(1, -F)$, as is shown in [2].

The fake factor in eq. (2.5) needs to be evaluated in a control region where the fake lepton contribution can be identified without looking at the data in the ('blinded') signal region. We can write the fake factor as

$$F = \frac{N_f^{\mathrm{T}}}{N_f^{\mathrm{L}}} = \frac{N^{\mathrm{T}} - N_r^{\mathrm{T}}}{N^{\mathrm{L}} - N_r^{\mathrm{L}}} = \frac{N_{\mathrm{data}}^{\mathrm{T}} - N_{\mathrm{MC}}^{\mathrm{T}}}{N_{\mathrm{data}}^{\mathrm{L}} - N_{\mathrm{MC}}^{\mathrm{L}}} \,, \tag{2.6}$$

which can be estimated in a control region by subtracting the contribution of real leptons (N_r) from data using MC simulation predictions to estimate the real lepton contribution.

The Fake Factor method (and the Matrix method) can be extended to multiple leptons. For two leptons, one distinguishes four categories: *tight-tight (TT)*, *tight-loose (TL)*, *loose-tight (LT)*, and *loose-loose (LL)*. Efficiency matrices analogous to eq. (2.1) can be constructed that relate the fake and real composition of the different categories. Using the Fake Factor method, the number of events with two fake leptons in the tight region can be expressed as

$$N_{\mathrm{data}}^{\mathrm{TT}} = F_1\Big(N_{\mathrm{data}}^{\mathrm{LT}} - N_{\mathrm{MC}}^{\mathrm{LT}}\Big) + F_2\Big(N_{\mathrm{data}}^{\mathrm{TL}} - N_{\mathrm{MC}}^{\mathrm{TL}}\Big) - F_1F_2\Big(N_{\mathrm{data}}^{\mathrm{LL}} - N_{\mathrm{MC}}^{\mathrm{LL}}\Big) \,, \tag{2.7}$$

where F_1 and F_2 are fake factors calculated for the first and second lepton candidate respectively. This expression assumes that the probabilities for the two lepton candidates

JHEP04(2026)188

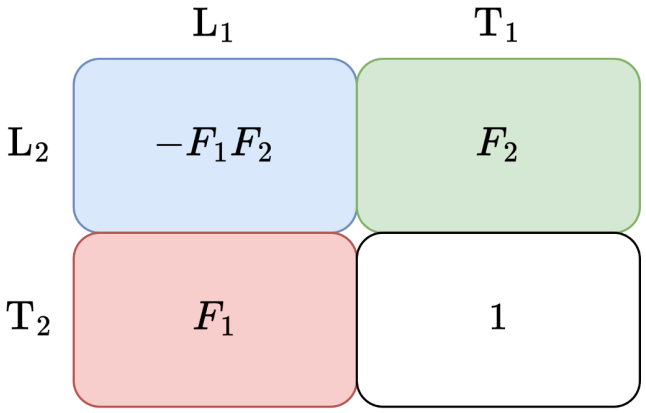

Figure 1. Fake factor method diagram in case of two leptons. In events with two leptons, the Fake Factor method is applied individually to each lepton. For LT and TL combinations, only one fake factor is assigned. In the signal region, where both leptons are tight (TT), the event weight is 1. When both leptons are loose (LL), two fake factors are applied for both leptons.

to satisfy the tight selection are factorizable (i.e. the fake rates of the two leptons are independent). This assumption may not be valid in the case where fake leptons are produced in a correlated way, for example when 2 leptons have an origin from the same heavy-flavor jet, and in which case we need to use dedicated methods (as discussed e.g. in the long-lived HNL search in ref. [5]).

This equation is visualized in figure 1 and is used in analyses with two leptons in the final state, for example in refs. [4, 6–8]. Related fake-rate implementations in multi-lepton final states are used in [5].

2.3 The binned Fake Factor method

Using simple event-counts, as the methods were in fact presented above, the fake factor is a single number. In such a case, the Matrix and Fake Factor methods are equivalent to the ABCD method [9, 10]. The ABCD relation follows directly from the same tight/loose factorization used above. In the single-lepton case, the binned fake factor estimate (eq. (2.4)) is equivalent to the ABCD relation:

$$N_{\mathrm{A}} = N_{\mathrm{C}} \frac{N_{\mathrm{B}}}{N_{\mathrm{D}}} = (N^{\mathrm{L}}_{\mathrm{data}} - N^{\mathrm{L}}_{\mathrm{MC}}) \frac{N'^{\mathrm{T}}_{\mathrm{data}} - N'^{\mathrm{T}}_{\mathrm{MC}}}{N'^{\mathrm{L}}_{\mathrm{data}} - N'^{\mathrm{L}}_{\mathrm{MC}}}\,, \tag{2.8}$$

with $N_f^{\mathrm{T}} = N_{\mathrm{A}}$, $N_f^{\mathrm{L}} = N_{\mathrm{C}}$ and $F = N_{\mathrm{B}}/N_{\mathrm{D}}$. Here, regions A and C correspond to the tight and loose signal regions, while regions B and D correspond to the tight and loose control regions, respectively. The primed quantities refer to the control region. Real-lepton contamination is subtracted in both regions before building the ABCD relation. The visualization of the method is shown in figure 2.

In the actual implementations of LHC analyses, however, the fake factor cannot be considered a constant, but rather a function of some relevant kinematic variables, such as transverse momentum p_{T}, pseudorapidity $|\eta|$, with additional topology-correlated variables (e.g. $E_{\mathrm{T}}^{\mathrm{miss}}$) included in some analyses. Consequently, the fake factor is evaluated as a function

JHEP04(2026)188

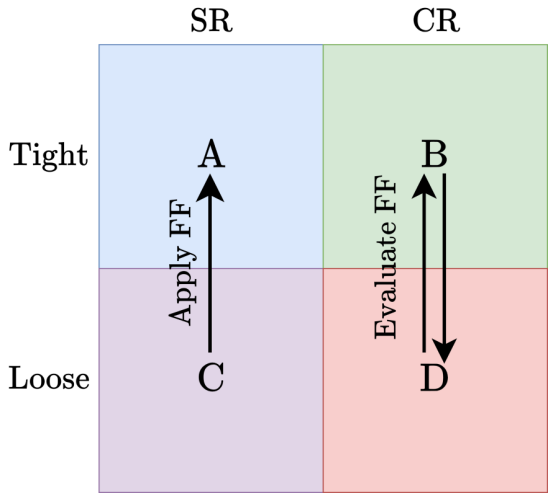

Figure 2. Visualization of the ABCD method, which is equivalent to the Fake Factor method when using only event counts. To estimate the number of fake leptons in region A, the ratio of the number of fake leptons in kinematically orthogonal tight and loose regions B and D is evaluated first, giving the value of F. Since this ratio is assumed to be equal in both SR and CR, the number of fake leptons in region A can be obtained by applying F as a transfer factor to the number of fake leptons in region C.

of these variables by binning (histogramming) the data in the control region and computing the ratio of the two data distributions in the nominal (tight) CR and loose CR^{L}. The resulting binned fake factor function is then applied to data in the loose SR^{L} region on a per-event basis, by assigning each event the fake factor corresponding to the bin in which the event falls.

The number of fake leptons in the SR can thus be estimated by applying the fake factor function to data in the loose SR^{L} region, subtracting the contribution of real leptons using MC simulation. Equation (2.4) then evolves into predicting the number of fake leptons in the tight region for each kinematic bin b, defined by the chosen binning scheme and kinematic observables:

$$N_{f,b}^{\mathrm{T}} = \sum_{i \in b} F_{\mathrm{data},i} - \sum_{j \in b} F_{\mathrm{MC},j} \, . \tag{2.9}$$

In the above equation, the fake factor introduces an event weight given as a function of p_{T}, $|\eta|$ and, where applicable, $E_{\mathrm{T}}^{\mathrm{miss}}$, applied as a weight to events in the loose region. This is referred to as the *binned Fake Factor method*.

The binned Fake Factor method suffers from several challenges. Most importantly, the choice of binning can significantly impact the results, as too coarse binning can lead to a loss of important kinematic features, while too fine binning can result in statistical fluctuations and empty (or even negative) bins, due to poor data and MC statistics. Furthermore, the method is typically limited to a few dimensions due to statistical limitations, which can restrict its ability to capture complex dependencies in the data and degrade the extrapolation to the SR. Finally, the method can suffer from artifacts due to binning, producing discontinuities in the fake factor function and affecting the stability of the background estimation.

JHEP04(2026)188

3 The ML-based Fake Factor method

The machine-learning based Fake Factor method, presented in this paper, is a novel approach to the traditional binned Fake Factor method. The main idea is to use machine learning to estimate an *unbinned (continuous)* fake factor function. This is achieved by estimating the probability densities, specifically the density ratio, of the real-subtracted tight and loose samples using neural networks. This is in fact related to the very active topic of simulation-based inference (SBI) approaches for likelihood ratio estimation [11], albeit in this case based on data itself, which we can refer to as the data-based inference (DBI) approach.

3.1 Neural fake factor estimation

The goal is to estimate the fake factor defined in eq. (2.6). By substituting the numbers of events with the corresponding probability densities, i.e. normalizing the numerator and denominator by the total number of baseline selection events $N_f = N_f^{\mathrm{T}} + N_f^{\mathrm{L}}$, we can rewrite the fake factor as a probability ratio:

$$F = \frac{N_f^{\mathrm{T}}}{N_f^{\mathrm{L}}} = \frac{p^{\mathrm{T}}}{p^{\mathrm{L}}} = \frac{f}{1-f}\,, \quad p^{\mathrm{T,L}} = \frac{N_f^{\mathrm{T,L}}}{N_f}\,, \tag{3.1}$$

reinterpreting F of eq. (2.5) as the ratio of the probability densities of fake leptons in the tight and loose regions. Given event observables $\mathbf{x}$, this translates to determining the functional dependence of F on the observables $\mathbf{x}$ as estimating the *density ratio*:

$$F(\mathbf{x}) = r_F(\mathbf{x}) = \frac{p^{\mathrm{T}}(\mathbf{x})}{p^{\mathrm{L}}(\mathbf{x})} \tag{3.2}$$

So far, this has been done using binning/slicing in some low-dimensional space, using a few observables, e.g. p_{T} and $|\eta|$, where the binning was necessarily coarse due to limited statistics. Here we propose to estimate the fake factor as the density ratio in a higher-dimensional feature space using neural networks, thus avoiding binning artifacts and enabling a more precise and flexible estimation of the fake factor.

The density ratio $r(\mathbf{x})$ of two probability densities can be estimated by training a binary classifier to distinguish two hypotheses (loose and tight) using the *likelihood ratio trick* [12–14]. There are many choices for the loss function for training the classifier [15]. We use the binary cross-entropy and a squared regularization term to prevent 'exploding' densities. The loss function is given by:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} w_i \left[-y_i \log \sigma(q(\mathbf{x}_i;\boldsymbol{\theta})) - (1-y_i)\log(1-\sigma(q(\mathbf{x}_i;\boldsymbol{\theta})))\right] - \sum_{i=1}^{N} \lambda q(\mathbf{x}_i;\boldsymbol{\theta})^2\,, \tag{3.3}$$

where σ is the sigmoid function, $y_i \in \{0,1\}$ are the binary labels corresponding to the two hypotheses, w_i are the event weights, $q(\mathbf{x};\boldsymbol{\theta})$ is the classifier output (in this case used as the *logit* value), $\boldsymbol{\theta}$ are the model parameters, λ is a regularization parameter, and N is the batch size. After the classifier has been trained, the density ratio can be estimated as $r(\mathbf{x}) = \exp(q(\mathbf{x};\boldsymbol{\theta}))$. The training dataset this contains events from both the tight and loose regions, with labels y_i assigned accordingly. We can then use the ratio as the fake

JHEP04(2026)188

factor function on a per-event basis, reweighting loose events by assigning each event the fake factor corresponding to its features $\mathbf{x}$.

A special consideration is needed to account for the real lepton contamination in both the tight and loose regions, which needs to be subtracted before estimating the density ratio, i.e. $N_f^{\mathrm{T,L}} = N_{\mathrm{data}}^{\mathrm{T,L}} - N_{\mathrm{MC}}^{\mathrm{T,L}}$ as given by eq. (2.6). Simulated MC events (and sometimes also data events) are in practice weighted events, to account for scaling to the data luminosity and process cross-section, simulation corrections (scale factors) and similar. The probabilities are thus given as (cf. eq. (3.1)):

$$p^{\mathrm{T,L}}(\mathbf{x}) \simeq \frac{1}{N_f}\left[\sum_{i\in\Omega(\mathbf{x})} w_{\mathrm{data,i}}^{\mathrm{T,L}} - \sum_{j\in\Omega(\mathbf{x})} w_{\mathrm{MC,j}}^{\mathrm{T,L}}\right], \tag{3.4}$$

where $\Omega(\mathbf{x})$ denotes the set of events with features close to the value of $\mathbf{x}$. The number of data and MC events in the tight and loose regions needs to be high enough to avoid too sparsely populated kinematic regions, resulting in locally negative probability estimation, but the formula is valid and only requires careful checks of possible negative probability values in low-statistics regions.

Subsequently, to incorporate this necessary subtraction, the ML training procedure needs to account for weighted events, including events contributing negative weights due to subtraction. Using negative weights in the ML loss function definition has been an open topic of discussion in the HEP ML community (advanced MC generators in fact produce a fraction of negative-weighted events, see e.g. ref. [16]), but we note that in this case it is a well-defined problem, because what we are doing in ML training is estimating the model density using cross-entropy, as described in [17]:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= -\mathbb{E}_{\mathbf{x}\sim p^{\mathrm{T,L}}}\left[\log p_{\mathrm{model}}^{\mathrm{T,L}}(\mathbf{x};\boldsymbol{\theta})\right] \\ &\simeq \frac{1}{N_f}\left[\sum_{i\in\Omega(\mathbf{x})} w_{\mathrm{data,i}}^{\mathrm{T,L}} \log p_{\mathrm{model}}^{\mathrm{T,L}}(\mathbf{x}_i;\boldsymbol{\theta}) - \sum_{j\in\Omega(\mathbf{x})} w_{\mathrm{MC,j}}^{\mathrm{T,L}} \log p_{\mathrm{model}}^{\mathrm{T,L}}(\mathbf{x}_j;\boldsymbol{\theta})\right].\end{aligned} \tag{3.5}$$

As long as the data and simulation statistics are sufficiently high, the model density $p_{\mathrm{model}}(x)$ can be learned.

We can improve the method further by again introducing the density ratio trick to estimate the (real-)subtracted densities in both the tight and loose regions separately, determining a ratio that would reweight either data or MC events to the subtracted density. This is similar to the approach used in ref. [18], where the authors use classifiers to reweight simulated samples to match data distributions. Here, we apply this idea to reweight either data or MC samples to obtain the subtracted densities $p^{\mathrm{T,L}}(\mathbf{x})$ in tight and loose regions. We introduce a ratio of data to MC (it works symmetrically for the opposite MC to data ratio as well), using eq. (2.6):

$$r^{\mathrm{T,L}} = \frac{N_{\mathrm{data}}^{\mathrm{T,L}}}{N_{\mathrm{MC}}^{\mathrm{T,L}}}, \qquad N_f^{\mathrm{T,L}} = N_{\mathrm{data}}^{\mathrm{T,L}} - N_{\mathrm{MC}}^{\mathrm{T,L}} = N_{\mathrm{data}}^{\mathrm{T,L}}\left(1 - \frac{1}{r^{\mathrm{T,L}}}\right) = N_{\mathrm{MC}}^{\mathrm{T,L}}\left(r^{\mathrm{T,L}} - 1\right). \tag{3.6}$$

By learning this ratio, we can effectively reweight either data or MC to obtain the subtracted densities in both tight and loose regions. Furthermore, this approach has to ensure that the derived correction factors are positive by construction, which means we need to ensure

JHEP04(2026)188

that the ratio $r^{\mathrm{T,L}}$ is always greater than one. The ratio approach also greatly improves the stability of the learning process, which we find to be an important achievement in itself and crucial in practice. The loss function remains the same as in eq. (3.3), with the labels assigned accordingly to distinguish the data and MC events in this case.

The complete ML fake factor estimation method consists of two steps: modeling the *subtraction* and *ratio*. The subtraction step is applied separately in the tight and loose regions to obtain real-subtracted densities. The ratio step then estimates the fake factor as the ratio of the real-subtracted tight and loose samples. The overall procedure is shown in figure 3 and detailed below.

Subtraction step. We construct the numerator and denominator terms of the fake factor density ratio r_F of eq. (3.2) by applying the subtraction procedure separately in the tight and loose regions, as defined in eq. (3.6). We train two independent (*subtraction*) classifiers and use them to derive correction factors $r^{\mathrm{T,L}}$. These correction factors are then used to reweight either data or MC samples on an event-by-event basis. Since the data and MC events are weighted, we include the event weights and write the reweighting procedure on an event-by-event basis as:

$$w_{f,i}^{\mathrm{T,L}} = w_{\mathrm{data,i}}^{\mathrm{T,L}} \left(1 - \frac{1}{r^{\mathrm{T,L}}(\mathbf{x}_i)}\right) \quad \text{or} \quad w_{f,j}^{\mathrm{T,L}} = w_{\mathrm{MC,j}}^{\mathrm{T,L}} \left(r^{\mathrm{T,L}}(\mathbf{x}_j) - 1\right) . \tag{3.7}$$

To obtain the functions $r^{\mathrm{T,L}}(\mathbf{x}_i)$, we train two classifiers to estimate these two ratios separately on the tight (T) and loose (L) datasets, constructed as a union of data and MC samples, setting MC labels to 0 and data labels to 1:

$$\mathcal{D}^{\mathrm{T,L}} = \{(\mathbf{x}_{\mathrm{data}}^{\mathrm{T,L}}, \mathbf{w}_{\mathrm{data}}^{\mathrm{T,L}}, \mathbf{1})\} \cup \{(\mathbf{x}_{\mathrm{MC}}^{\mathrm{T,L}}, \mathbf{w}_{\mathrm{MC}}^{\mathrm{T,L}}, \mathbf{0})\} . \tag{3.8}$$

In our implementation, we chose to reweight the data rather than the MC to stay more data-driven (left formula of eq. (3.7)), although either approach is valid. We need to ensure that the ratio condition $r^{\mathrm{T,L}}(\mathbf{x}) > 1$ is met, to keep the weights positive. This is straightforward to achieve by requiring the classifier logit outputs $q^{\mathrm{T,L}}(\mathbf{x};\boldsymbol{\theta})$ to be positive, since the ratio function is then $r^{\mathrm{T,L}}(\mathbf{x}) = \exp(q^{\mathrm{T,L}}(\mathbf{x};\boldsymbol{\theta}))$, as defined by the likelihood ratio trick and the loss function of eq. (3.3). The actual implementation of this constraint is given in section 3.2.

Ratio step. After subtraction, we train a third (*ratio*) classifier to distinguish the numerator (tight subtraction) from the denominator (loose subtraction), using the new weights derived in the subtraction steps described above in the new loss function, constructed again as given by eq. (3.3). This will give us the final ratio $r_F(\mathbf{x})$ to be used as the fake factor function $F(\mathbf{x})$. According to the reweighting choice made in the subtraction step, we train an ML model on data

$$\mathcal{D}_{\mathrm{data}} = \{(\mathbf{x}^{\mathrm{T}}, \mathbf{w}_f^{\mathrm{T}}, \mathbf{1})\} \cup \{(\mathbf{x}^{\mathrm{L}}, \mathbf{w}_f^{\mathrm{L}}, \mathbf{0})\} . \tag{3.9}$$

The end result is a parametric model $r_F : \mathbf{x} \in \mathbb{R}^d \to \mathbb{R}^+$ that estimates the fake factor given an event that has features $\mathbf{x}$ of dimension d.

JHEP04(2026)188

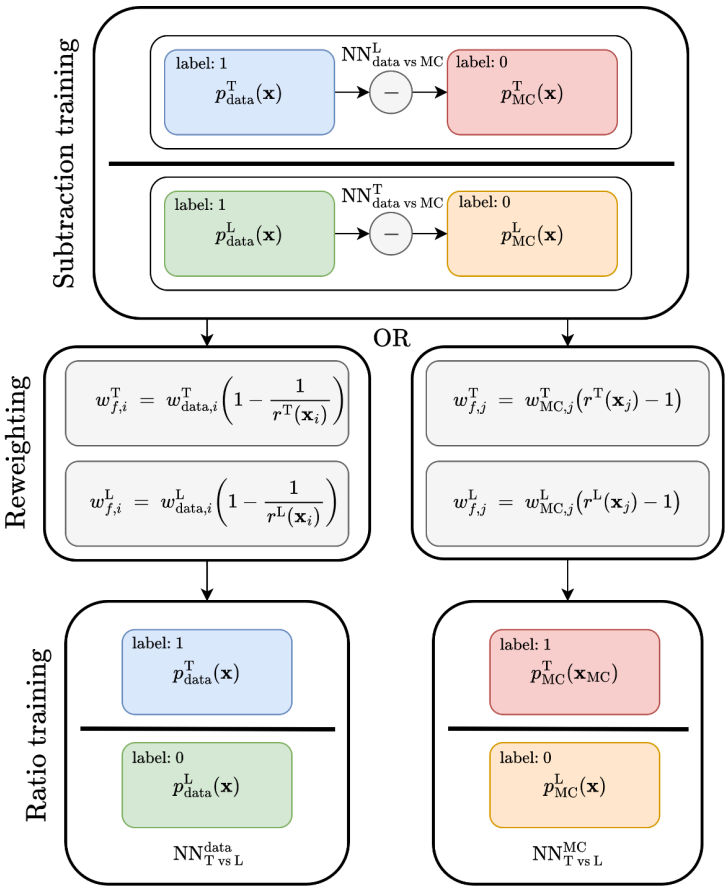

Figure 3. Flow diagram of the ML-based method to obtain the fake factor F as a density ratio $r_F(\mathbf{x})$. Firstly, two independent classifiers are trained in the tight and loose regions to model the ratios $r^{\mathrm{T,L}}$ between data and MC. These can then be used as correction factors to obtain prompt-subtracted densities by reweighting either data or MC events, giving the two branches of the diagram. Lastly, a third classifier is trained on reweighted events to separate loose and tight prompt-subtracted distributions, which gives the final density ratio $r_F(\mathbf{x})$.

3.2 Model architecture and training setup

We begin by preprocessing the data. Numerical features are standardized to have zero mean and unit variance, while categorical features are label-encoded. The dataset is then split into training (50 %), validation (25 %), and test (25 %) sets. During training, we monitor the validation loss to prevent over-fitting and select the best neural network classifier model based on validation performance. If the validation loss does not decrease for five consecutive epochs, the learning rate is reduced by a factor of 0.5.

Before feeding the data into the model, we apply an embedding layer to both categorical and numerical features [19, 20]. This maps each feature into a higher-dimensional space, enabling the model to learn richer representations of the input data. All embeddings are concatenated into a single feature vector, and mean pooling is applied across the feature dimension to obtain the final input representation for the model.

JHEP04(2026)188

We train three binary classifiers with the same architecture for density ratio estimation using the loss function defined in eq. (3.3). Training is performed with the AdamW optimizer [21], using a learning rate of 3×10^{-4}, a weight decay of 10^{-5}, a batch size of 1024 epochs, and training taking up to 100 epochs. Early stopping is applied based on validation loss. The regularization parameter λ in the loss function is set to 0.01 for the ratio classifier and 0 for the subtraction classifiers. We notice that the higher regularization translates into better extrapolation to the signal region, but worse performance in the control region, so this is a hyperparameter that can be tuned based on the specific analysis needs.

For the classifier model architecture, schematically shown in figure 4, we adopt a *pre-activation ResNet* [22] with four residual blocks, each containing two layers of 128 neurons. We found that batch normalization and dropout did not improve performance, so they are omitted in the final model, although they can be easily added if needed. ReLU activations are used after each layer except for the output layer. The network input consists of the concatenated feature embeddings, with an optional projection layer applied if the embedding dimension does not match the network input dimension. The output layer is a single projection layer with one neuron that produces the result q (interpreted as the logit value) for the binary classification task.

We use the ResNet architecture because residual connections mitigate the problem of vanishing gradients, enabling deeper networks to be trained effectively. This design allows the classifier to learn more expressive transformations without suffering from performance degradation as depth increases. In practice, we found that using residual connections leads to more stable training dynamics and consistently better density ratio estimate compared to equivalent plain feed-forward networks, where we found that a classifier can over-fit easily.

The output activation function depends on the specific task. For the subtraction classifiers, we use a *soft absolute* activation function to ensure non-negative outputs q (interpreted as the logit value) for both subtraction classifiers, which is required to keep the data correction weights $(1 - 1/r)$ of equation (3.7) positive, since $r = \exp(q)$. In the ratio classifier, we use a linear activation function to allow for unbounded outputs.

The soft absolute activation function is defined as

$$f(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } |x| < 1 \,, \\ |x| - \frac{1}{2}, & \text{otherwise} \,, \end{cases} \tag{3.10}$$

which is inspired by the Huber loss [23]. The function is smooth and differentiable everywhere, and behaves like the absolute value function for large inputs, while being quadratic near zero. This ensures that the network outputs are always non-negative as depicted in figure 5, which is important for the subtraction step where we need to avoid negative weights.

4 Method validation using a representative LHC analysis

To validate the ML-based Fake Factor method, we define a straightforward data analysis to measure the W boson transverse mass m_{T} from the W boson decaying into an electron and a corresponding neutrino ($W \to e\nu$). We use the ATLAS Open Data sample from Run 2 collected in the years 2015 and 2016 [24]. Events are required to pass single-electron

JHEP04(2026)188

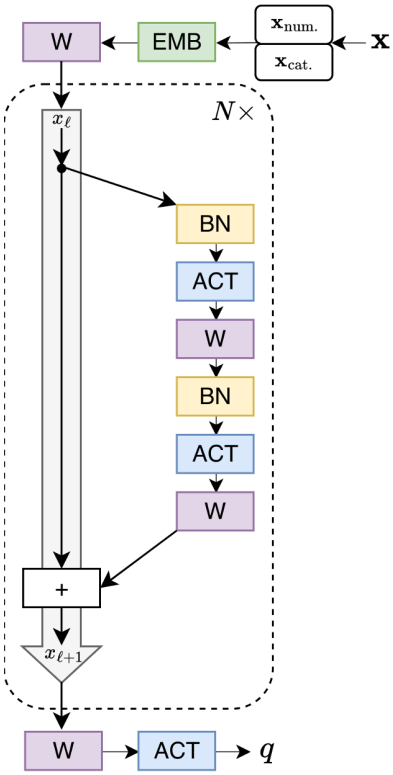

Figure 4. Schematic illustration of the classifier model architecture used in this work. Our classifiers use a pre-activation residual network (ResNet) architecture. The numerical features $\mathbf{x}_{\text{num.}}$, concatenated with the categorical features $\mathbf{x}_{\text{cat.}}$, are embedded through an embedding layer (EMB) and then passed through a projection layer (if needed) before being fed into the ResNet. The ResNet is schematically shown as a stack of batch normalization (BN), weight multiplication (W), and activation function (ACT) layers, with residual connections between them. The output layer uses either a soft absolute or linear activation function to produce the final output (logit) value q, as described in the text.

triggers, with the reconstructed electron required to match the triggering object. Additionally, the requirements are: p_{T} of the electron above 25 GeV, missing transverse energy in the event $E_{\text{T}}^{\text{miss}} > 30\,\text{GeV}$, at least one jet present and the b-jet veto implemented. The main background contributions in this channel after these selection cuts are W+jets and $t\bar{t}$ events. To define our control and signal kinematic regions, we use transverse mass m_{T} defined as

$$m_{\text{T}} = \sqrt{2\,p_{\text{T}}\,E_{\text{T}}^{\text{miss}}\,\left(1-\cos\left(\Delta\phi\right)\right)}\,, \tag{4.1}$$

where $\Delta\phi$ is the azimuthal angle between the electron and the missing transverse energy directions. The control region (CR) is defined with the requirement $m_{\text{T}} < 60\,\text{GeV}$, while the signal region (SR) is defined with $m_{\text{T}} > 60\,\text{GeV}$, to ensure orthogonality. This selection is loosely inspired by the measurement of the W-boson mass in ref. [25]. We deliberately set up our benchmark analysis to be quite simple in selection, with a minimal set of cuts to demonstrate that it works well for describing the fake contribution across a wide range of kinematics and for substantial real lepton contamination (subtraction) as well.

JHEP04(2026)188

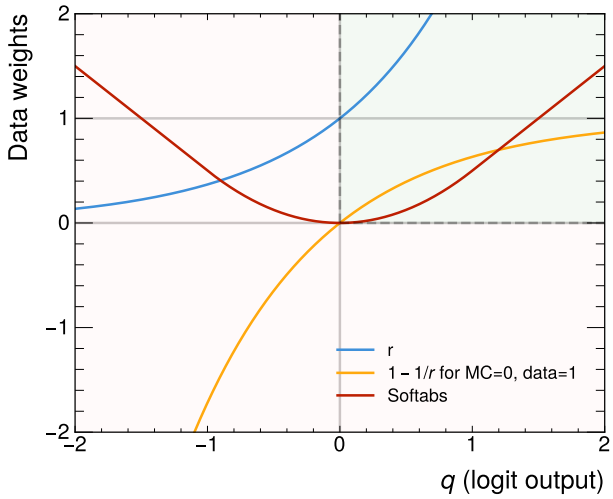

Figure 5. The soft absolute activation function (red) constrains the (logit) outputs q of both subtraction classifier networks to be non-negative, which is required to keep the data correction weights positive. The exponential of the output $r = \exp(q)$, is used to obtain the density ratio estimate (blue), which will be $r > 1$ when using the proposed activation function. Data reweighting function (orange) is given in eq. (3.7). In our implementation, we reweight data with labels 0 for MC and labels 1 for data. Using the soft absolute activation ensures that the reweighting function remains non-negative and bounded within $[0, 1]$, as required.

The events are further divided into two categories, *tight* and *loose*, referring to the electron reconstruction requirements, that can be used in the Fake Factor method. In these two categories, tight leptons pass the tight identification and loose isolation criteria, while loose leptons fail both.

We have chosen five different kinematic observables for the ML training: p_{T}, $E_{\mathrm{T}}^{\mathrm{miss}}$, η, number of jets (N_{jets}), and m_{T}. The control region distributions of a subset of these observables for both tight and loose selections are shown in figure 6. The machine learning model is trained using these observables (four continuous and one discrete) as input features in the control region to estimate the fake factor.

4.1 ML subtraction step in the analysis control region

To validate the ML-based Fake Factor method, we first examine the performance of the subtraction step in the CR. In order to validate the subtraction, we apply the weights derived from the subtraction step in eq. (3.7) to reweight the data distribution in the control region. We then bin the reweighted data and compare it to the binned subtraction of data and MC in the control region. This allows us to assess the effectiveness of the subtraction step. The results are shown in figure 6. We observe a good agreement between ML and binned subtraction, indicating that the ML subtraction step is working as intended.

In more detail, the difference in the total number of events between the two methods is within 1 % for both numerator and denominator, which is acceptable given the statistical uncertainties in the data and MC samples. This demonstrates that the ML-based subtraction step is effective and can be used as part of the overall fake factor estimation process.

JHEP04(2026)188

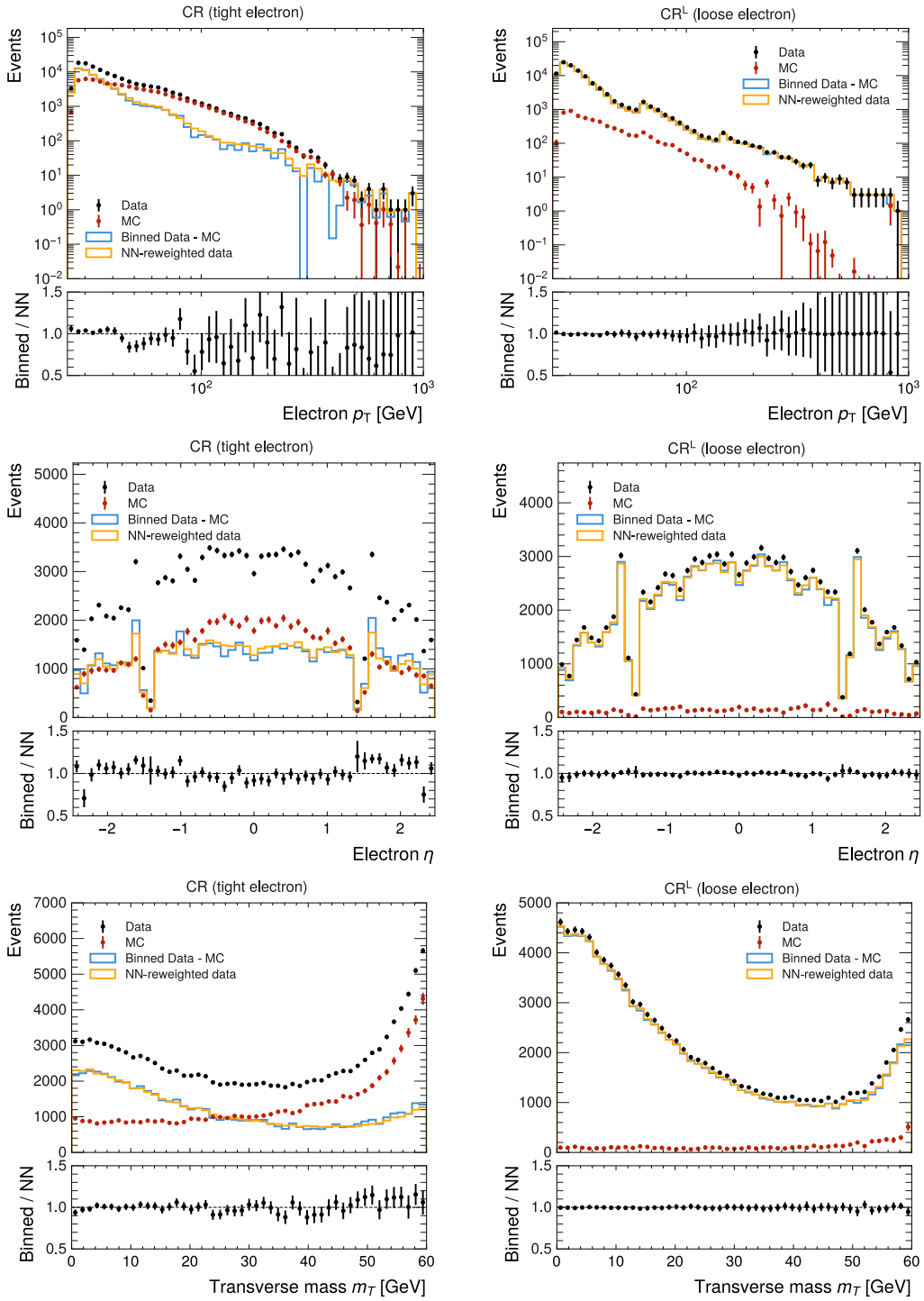

Figure 6. Distributions of p_{T}, η and m_{T} in the control region of the implemented analysis for both the tight (left column) and loose (right column) selections. Data and MC distributions are shown by the black and red error bars, respectively. The error bars show the statistical uncertainties of data or MC only. For reference, the blue histogram shows the values of the subtraction of MC from data for each bin individually. Distributions derived by using the subtraction step of the ML-based approach, i.e. obtained by reweighting data events by the correction factor $r^{\mathrm{T,L}}$, are shown in orange. One can observe that the main objective of the ML-based subtraction, which is to obtain per-event weights that reweight the data, resulting in adequate agreement with the reference, per-bin subtracted distributions, is achieved to a good precision.

JHEP04(2026)188

In the loose region (CR^L), the real lepton contamination is relatively small, so the subtraction step has a limited impact on the final fake factor estimate. However, it is still important to perform the subtraction to ensure that the derived fake factor is not biased by any residual real contributions in the loose region.

4.2 ML ratio step in the analysis control region

After validating the subtraction step, we proceed to validate the ratio step in the ML-based Fake Factor method, as denoted in eq. (3.2). Once the final ratio classifier has been trained, we can visualize the resulting predicted value of the fake factor as a continuous function of the input features, which allows us to see how the fake factor varies across different regions of the feature space. We create 2D projections of the fake factor function as slices in the p_T-η plane (figure 7) and p_T-E_T^{miss} plane (figure 8). We observe that the fake factor varies smoothly across the feature space, indicating that the classifier model has learned a meaningful representation of the data, and that the fake factor values increase with the jet multiplicity N_{jets}, which is consistently incorporated in the ML-based fake factor parameterization.[3]

With increasing N_{jets} and E_T^{miss}, we also encounter lower data and MC statistics, which can lead to larger fluctuations in the derived fake factor values. This is particularly evident in the high E_T^{miss} regions and with more than one jet, where the fake factor values do fluctuate significantly. Overall, the 2D projections provide a useful way to visualize the learned fake factor and assess its behavior across different regions of the feature space.

In order to validate the newly derived ML procedure, we compare the ML-based fake factor to the results the (customary) binned fake factor approach would give. For the binned fake factor calculation, we use the same control region as for the ML-based method, and apply the same event selection criteria. The binned fake factor is calculated by taking the ratio of the number of tight events to the number of loose events in bins of p_T and $|\eta|$ after the data and real-lepton MC subtraction, i.e. subtracting and dividing the required histograms. This provides the comparison baseline to assess the performance of the ML-based method. The calculated binned fake factors are shown in figure 9.

To compare the ML-based fake factor with the binned fake factor, we produce one-dimensional projections of the ML fake factor across different p_T and $|\eta|$ ranges (bins). For all other variables, we perform an averaging (integration) over the ranges where the binned fake factor is defined. This procedure ensures a consistent and direct comparison between the two methods. Additionally, uncertainty bands are included for the ML-based fake factor, reflecting the variation introduced by the averaging process. The results of this comparison are presented in figures 10 and 11.

We observe good agreement in both shape and magnitude between the two methods across all bins given the uncertainties, indicating that the ML-based approach successfully captures the same underlying physics as the traditional binned method. In poor-statistics regions the ML-based method is expected to produce a more robust result, since it uses more

[3]It is worth emphasizing that using a more optimal parameterization (e.g. mother parton p_T [4]) such dependence would be absent or at least greatly reduced, which is, however, beyond the scope of this paper. In this study, the successful reproduction of the kinematics in a multidimensional representation of a fake factor parameterization was the main emphasis.

JHEP04(2026)188

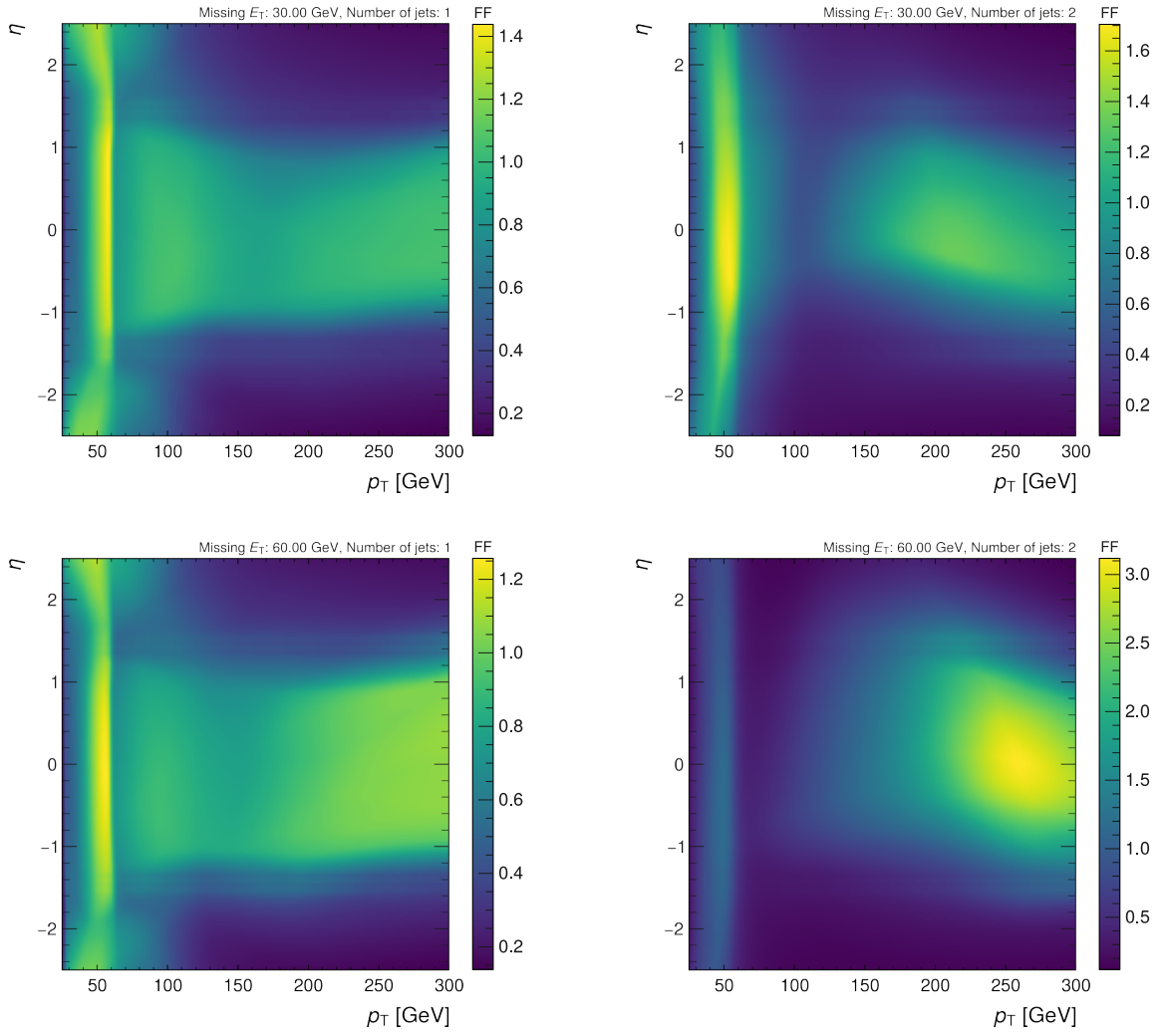

Figure 7. A 2D projection of the fake factor obtained with the ML-based method in the p_T-η plane for different fixed values of the other features. The values are approximately symmetrical in η, as would be expected, but have an otherwise non-trivial dependence across all input features. Note that all values are strictly positive by construction.

event information in a properly correlated way, which explains the occasional deviations of the two, e.g. in high-p_T regions. This demonstrates that the ML-based method is a valid and robust alternative for estimating fake factors, offering greater flexibility and the potential for improved accuracy.

5 Predicting the fake lepton background in the analysis

In this section, we present the results of applying both the binned and ML-based Fake Factor methods to estimate the fake electron background in the implemented analysis setup using the ATLAS Open Data sample. We validate the performance of both methods in the control region (CR) and signal region (SR) defined by transverse mass (m_T) cuts, as described in section 4. Performance is evaluated by comparing the estimated fake contributions to

JHEP04(2026)188

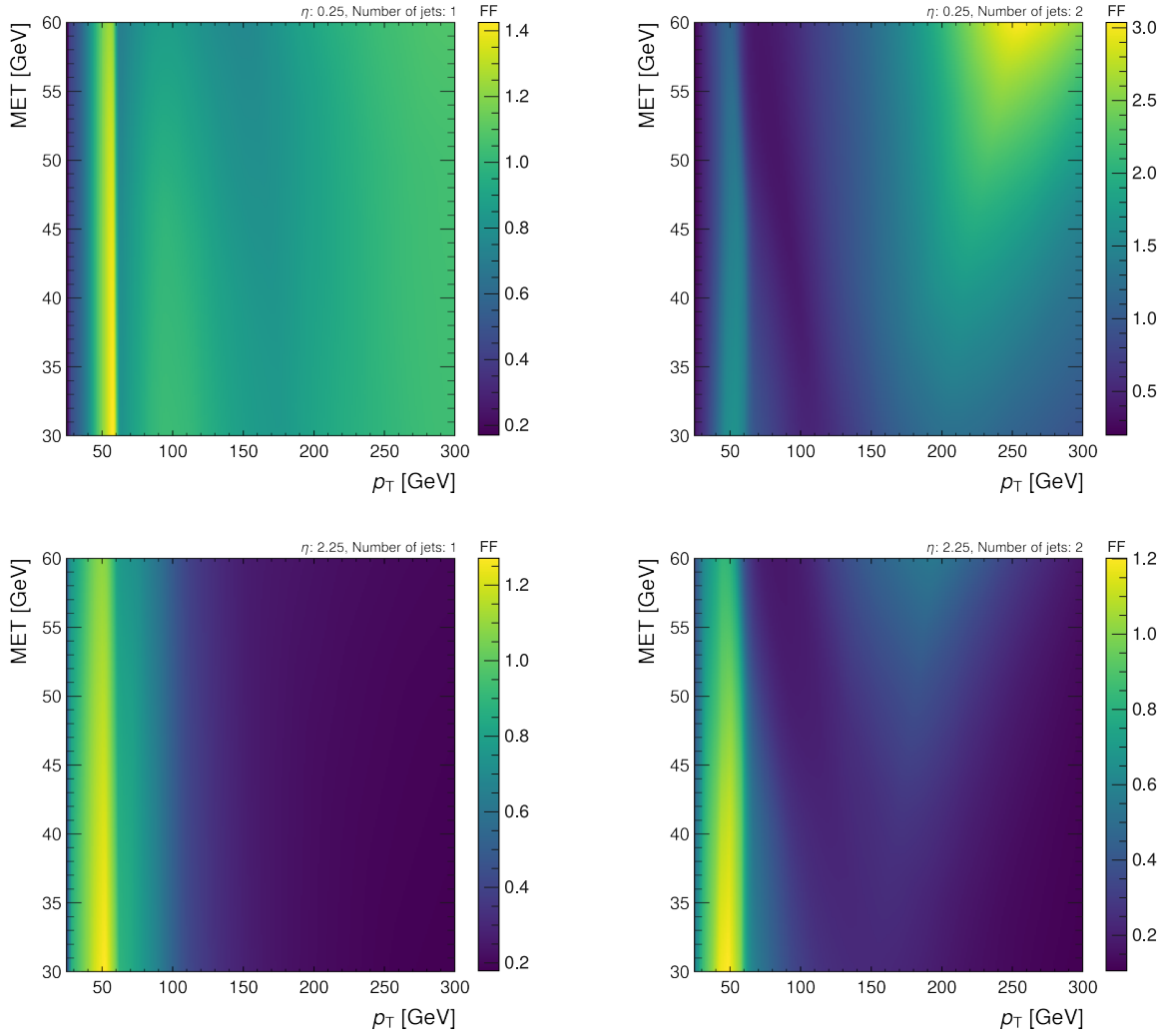

Figure 8. A 2D projection of the fake factor obtained with the ML-based method in the p_T-E_T^{miss} plane for different fixed values of the other features.

the observed data and MC predictions across several key kinematic variables, including lepton transverse momentum (p_T), pseudorapidity (η), missing transverse energy (E_T^{miss}), jet multiplicity (N_{jets}), and transverse mass (m_T). In order to ensure a fair comparison, we evaluated both methods using the same events and applied identical selection criteria.

We assess the agreement between data and fake process predictions, after applying the fake factor corrections, focusing on both the shape and normalization of the distributions.

5.1 Control region performance

Firstly, we validate both the binned and ML-based Fake Factor methods in the control region defined by $m_T < 60\,GeV$, corresponding to the fake-enriched region described in more detail in section 4. In figure 12 and figure 13, we show the closure tests for both methods in p_T, η, E_T^{miss}, N_{jets}, and m_T. Both methods show good agreement between data and MC across all variables, which is expected since the fake factors are derived in this region. This demonstrates

JHEP04(2026)188

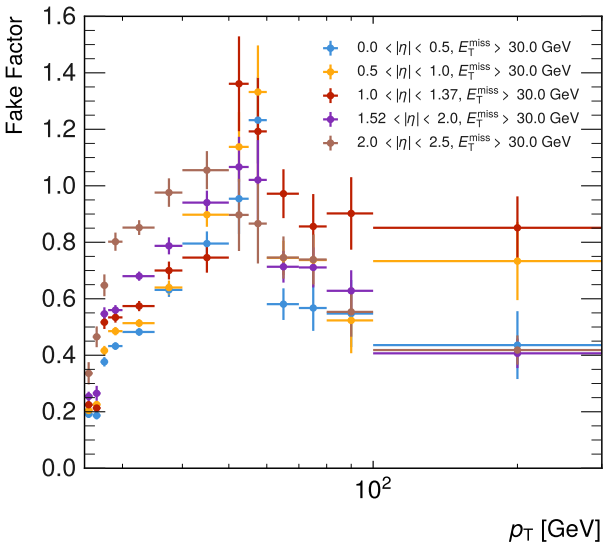

Figure 9. Values of the fake factor obtained using the binned Fake Factor method as a function of p_T for different bins in $|\eta|$. Uncertainties are driven by the finite statistics of data and MC samples, limiting the possibility to further segment the binning or add an additional dimension.

that the estimated fake contributions accurately account for the shape and normalization of the fake-lepton background within the fake-enriched control region for both methods.

Focusing on the p_T distribution, the binned method performs better at lower p_T ($p_T < 100\,\text{GeV}$), where we have sufficient statistics in each bin and where the binning was optimized to capture the finer features of the distribution. The ML-based method, however, provides better modeling in E_T^{miss} and m_T since these variables were also used in the ML training, while they cannot be used in the binned method due to statistical constraints, as already described. In the N_{jets} distribution, where we expect the jet multiplicity to be correlated with the fake contribution (since there is a higher probability of non-real leptons in multi-jet events), both methods capture this trend.

5.2 Signal region performance

In the signal region (SR), defined by $m_T > 60\,\text{GeV}$, we validated both Fake Factor methods by extrapolating from the results derived in the control region to a kinematic regime in the SR, where the fake contribution is smaller and real backgrounds dominate. In figure 14 and figure 15, we show the distributions across the main observables.

In all the distributions, we observe that the ML-based method provides a better prediction in both shape and normalization with reduced fluctuations. This can be seen particularly in the E_T^{miss} and m_T distributions where the ML-based method captures the shape more accurately. The binned method shows larger discrepancies due to its statistical constraints on coarse binning, which can lead to inaccuracies when extrapolating to regions with limited statistics or complex correlations between variables.

With these validations, we demonstrate that the ML-based Fake Factor method can effectively extrapolate from the control region to the signal region, providing a reliable estimate

JHEP04(2026)188

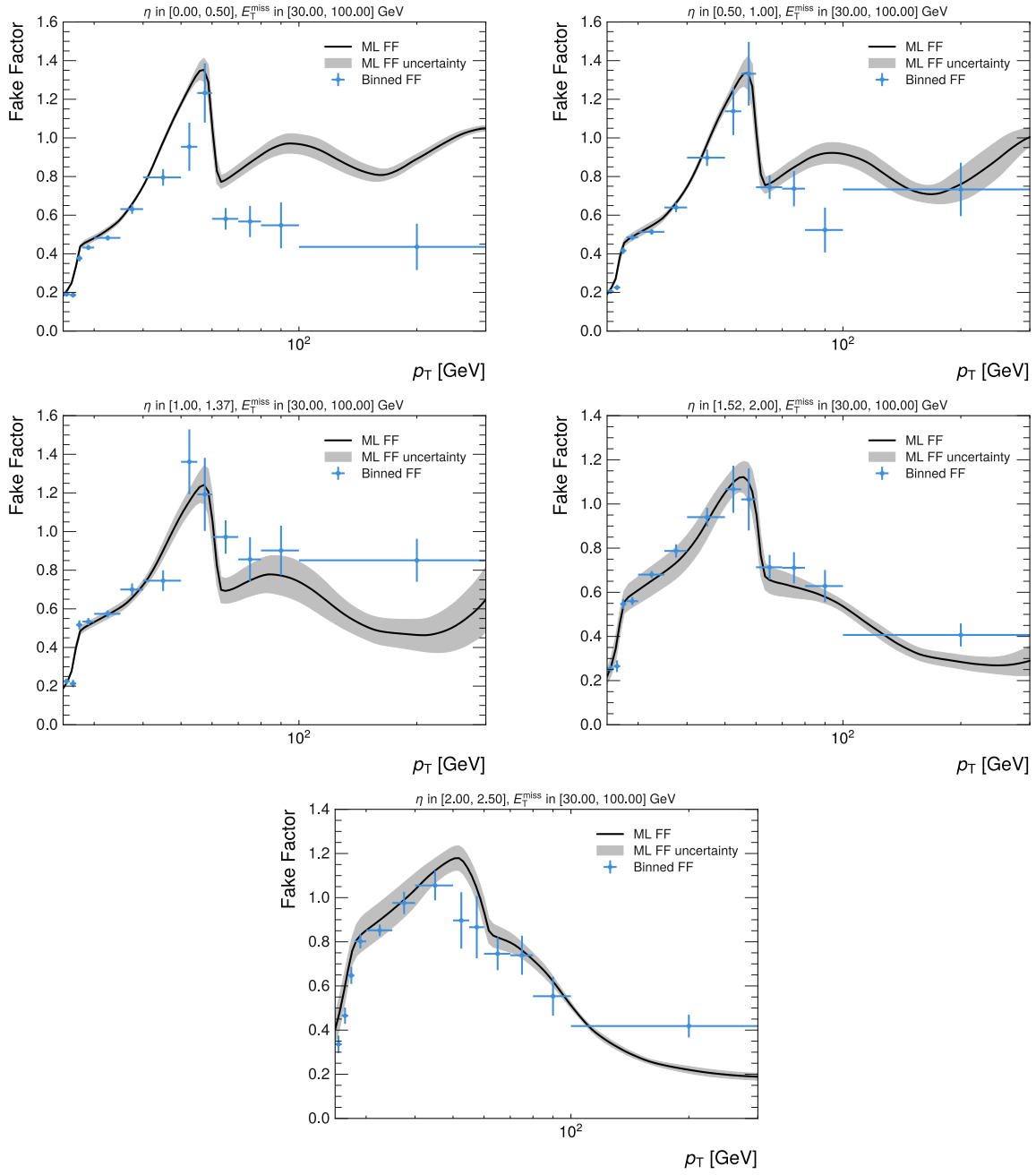

Figure 10. Comparison of binned fake factors and 1D projections of the fake factors obtained with the ML-based method as a function of p_T. Individual plots are showing different bins (ranges) in $|\eta|$. 1D projections of the ML-based method are obtained by integrating out (averaging over) all additional non-relevant dimensions (E_T^{miss}, N_{jets} and m_T) as well as the applicable range in $|\eta|$. The uncertainty band represents the standard deviation of the fake factor values in the integration range.

JHEP04(2026)188

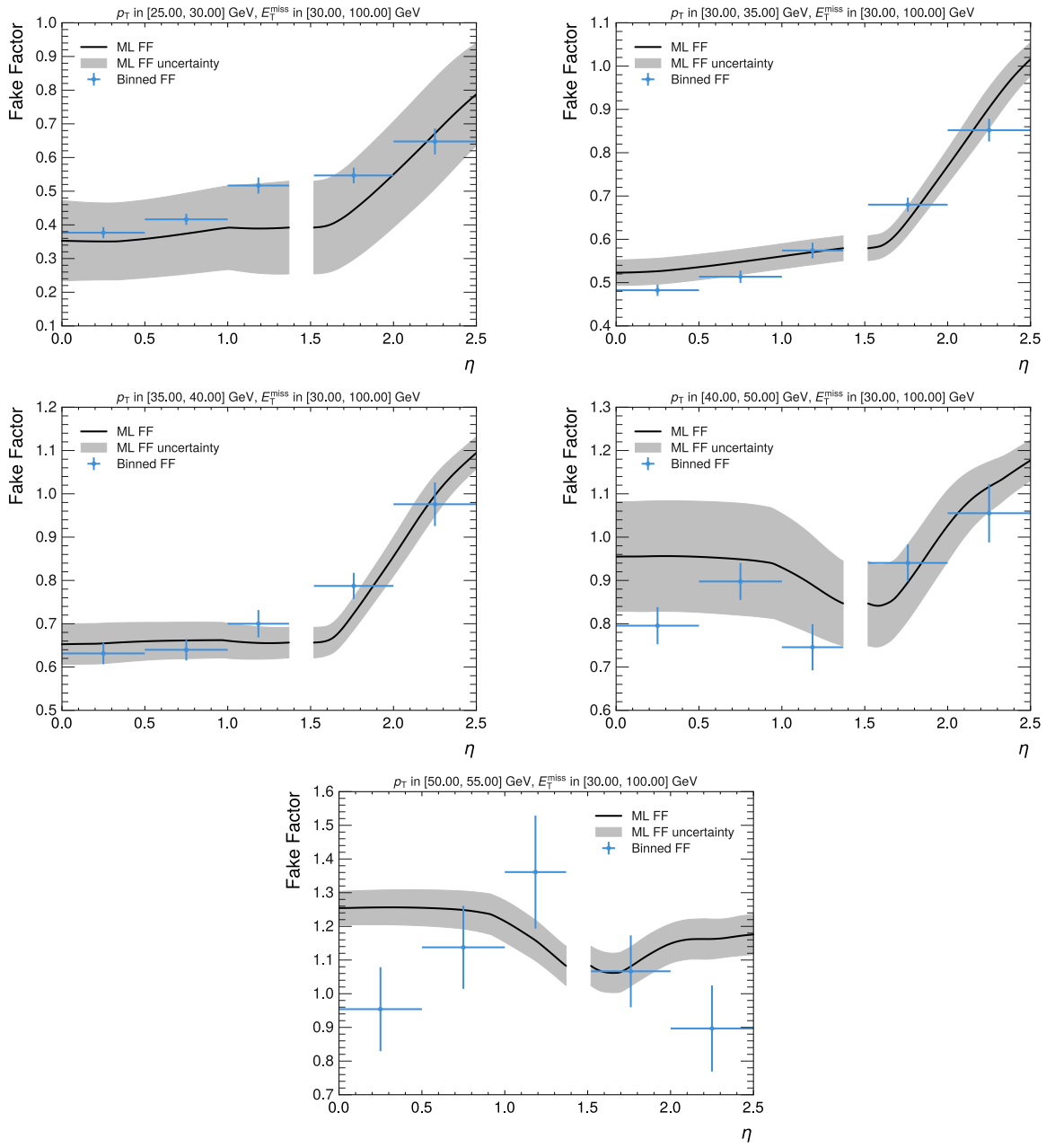

Figure 11. Comparison of binned fake factors and 1D projections of the fake factors obtained with the ML-based method as a function of $|\eta|$. Individual plots are showing different bins (ranges) in p_{T}. 1D projections of the ML-based method are obtained by integrating out (averaging over) all additional non-relevant dimensions ($E_{\mathrm{T}}^{\mathrm{miss}}$, N_{jets} and m_{T}) as well as the applicable range in p_{T}. The uncertainty band represents the standard deviation of the fake factor values in the integration range.

JHEP04(2026)188

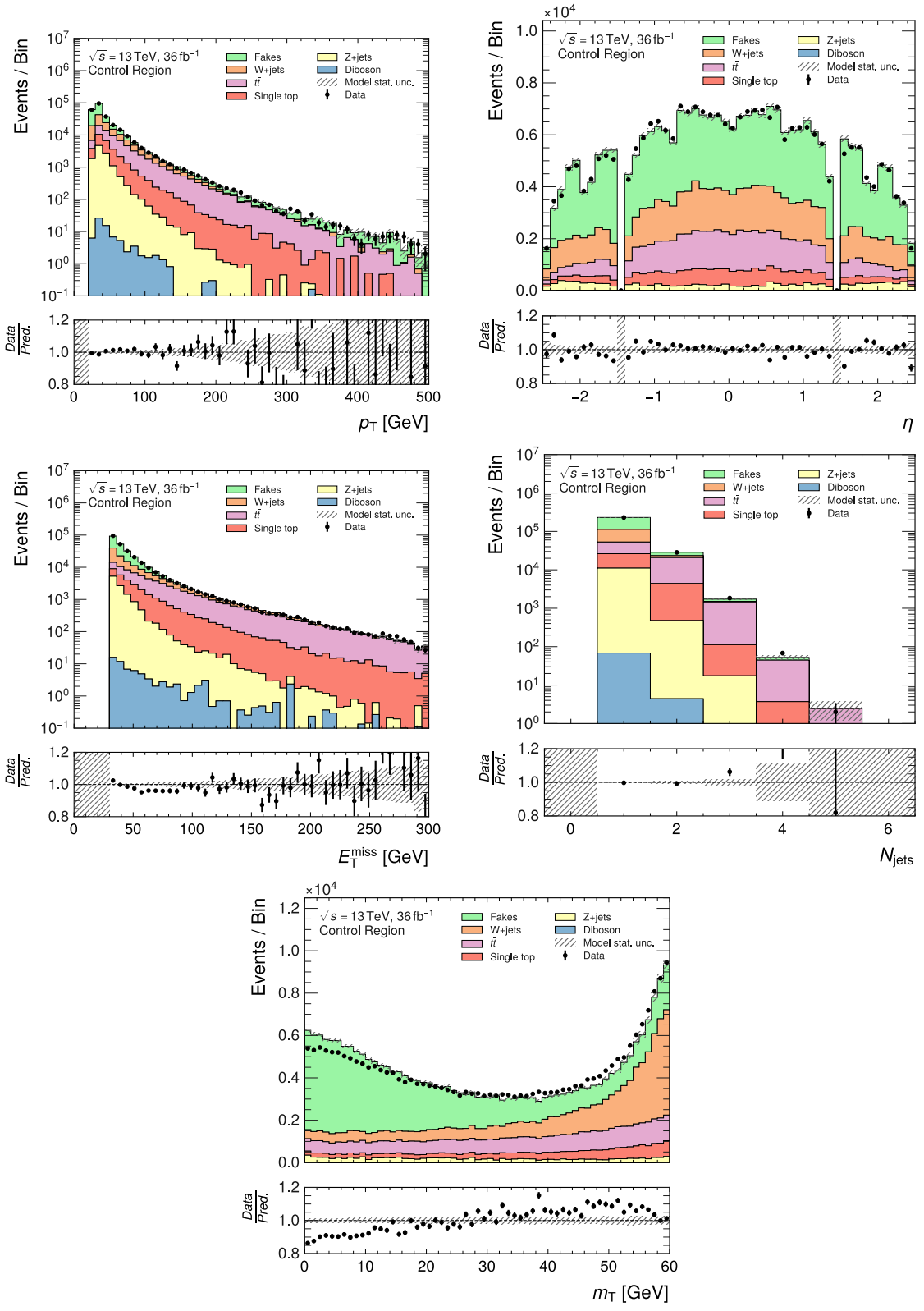

Figure 12. Closure test in the control region (CR) of the implemented analysis using fake factors obtained from the binned method. As expected, distributions in p_{T} and η show good agreement since the binned method was parameterized in these variables and optimized to match the input data (i.e. 'close' well). Distributions with respect to other variables that were not included in the parameterization all show a greater systematic mis-modeling, which is especially noticeable in the m_{T} distribution.

JHEP04(2026)188

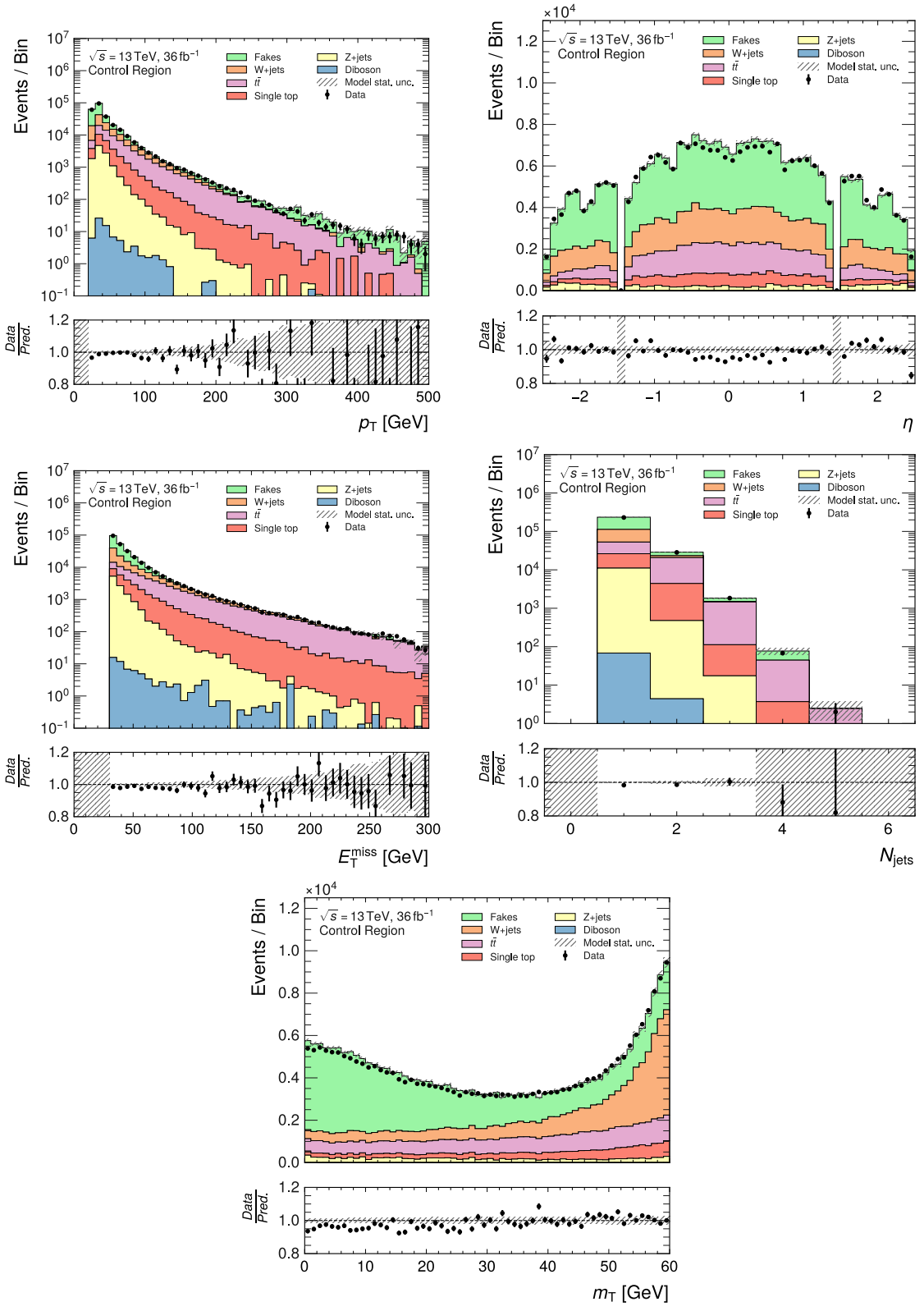

Figure 13. Closure test in the control region (CR) of the implemented analysis using fake factors obtained from the ML-based method. Comparing to the results of the binned method in figure 12, with p_{T} and η distributions giving a slightly degraded agreement, an overall improvement in the closure is seen over all considered kinematic variables. This especially noticeable in the $E_{\mathrm{T}}^{\mathrm{miss}}$ and m_{T} distributions, where the ML-based method captures the non-trivial dependence of the fake factor.

JHEP04(2026)188

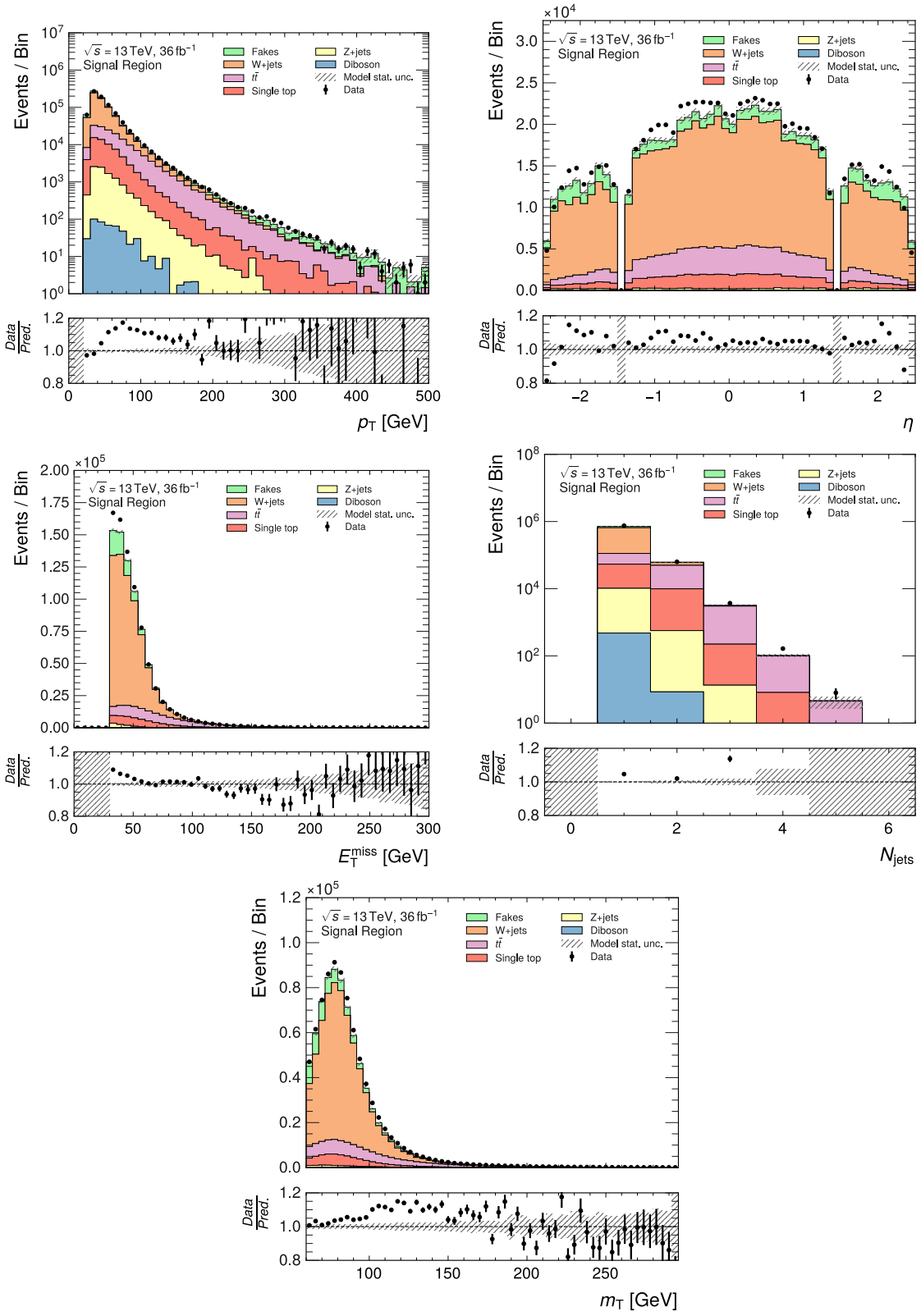

Figure 14. Signal region (SR) of the implemented analysis with fake events modeled using binned fake factors. Compared to the closure study in the CR, the p_{T} and η distributions show a significant systematic mis-modeling when the binned fake factors are extrapolated to the signal region. The same is reflected in the distributions of other variables as well, showing that the contributions from fake events are consistently underestimated.

JHEP04(2026)188

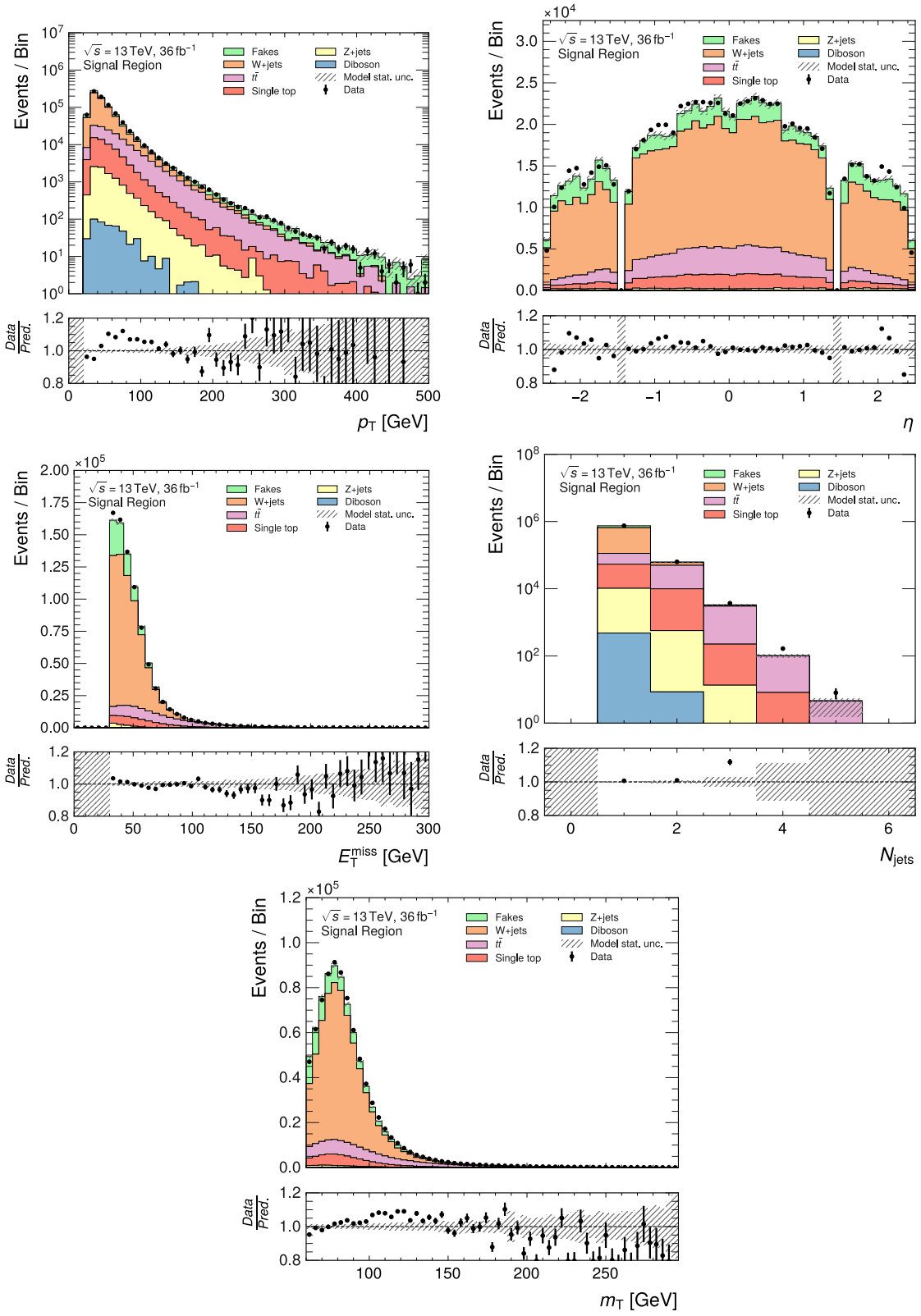

Figure 15. Signal region (SR) of the implemented analysis with fake events modeled using fake factors from the ML-based method. In comparison with the binned method on figure 14, a significant improvement in the modeling is seen in the distributions of all variables — both in the normalization and non-trivial shape effects. This indicates an improved ability of the ML-based method to accurately extrapolate the fake factor to the signal region, capturing a more complex and accurate kinematic dependence.

JHEP04(2026)188

of the fake lepton background in a kinematic regime where real backgrounds dominate. This is particularly important for analyses searching for rare signals, where accurate background estimation is crucial for maximizing sensitivity and avoiding potential appearance of a spurious (accidental) signal due to mis-modeling.

6 Conclusion

In this study, we have introduced a novel data-based inference method using machine learning for estimating fake lepton backgrounds in high-energy physics analyses, demonstrating its advantages over traditional binned fake factor approaches. By leveraging neural density ratio estimation, our approach enables the calculation of continuous, unbinned fake factors on a per-event basis, allowing for precise modeling in high-dimensional feature spaces. This flexibility mitigates common limitations of conventional methods, such as coarse binning, extrapolation uncertainties, and the inability to capture complex correlations between variables.

Applying the method to a straightforward $W \to e\nu$ transverse mass analysis using ATLAS Open Data, we validated the method's performance through a two-step procedure consisting of subtraction and ratio calculations. The subtraction step effectively removes contamination from real leptons, ensuring that the derived fake factors reflect the true background contribution. The ratio step then generates smooth and physically consistent estimates across the feature space. Comparison with traditional binned fake factor results shows good agreement in both normalization and distribution shapes, while also highlighting the ML approach's strength in sparsely populated or high-dimensional regions where conventional methods often struggle.

Beyond demonstrating methodological improvements, this work highlights the broader potential of machine learning for data-driven background estimation in high-energy physics. The proposed method is inherently adaptable, capable of being extended to multi-lepton final states or other types of mis-identified objects. Future work will focus on incorporating an advanced approach to systematic uncertainty estimation, exploring probabilistic models to quantify the uncertainty in ML-based fake factor predictions beyond simple statistical variations available through data resampling techniques (bootstrapping, Jackknife, etc.), which are already available.

Additionally, integrating this method into more complex analyses of the LHC experiments, including those searching for new physics phenomena, will further showcase its utility and robustness.

In conclusion, the machine learning based Fake Factor method represents a significant step toward more precise, flexible, and robust data-driven background estimation in particle physics. By combining data-driven techniques with advanced computational tools, it enables better exploitation of the rich datasets produced by modern experiments, ultimately improving the sensitivity of searches for rare processes and new phenomena. As high-energy physics continues to explore more intricate final states and higher-dimensional datasets, methods like this will play a crucial role in ensuring that background modeling keeps pace with experimental and simulation capabilities, enabling utilization of the full physics potential of the data collected.

JHEP04(2026)188

Acknowledgments

We would like to thank the ATLAS Collaboration for providing the Open Data used in this paper, as well as the developers of the various software tools and libraries that facilitated this work. The work of B.P. Kerševan, J. Debevc and J. Gavranovič is supported by the research grant J1-60028 and research program P1-0135, funded by the Slovenian Research and Innovation Agency (ARIS). L. Čalić and E. Lytken wish to acknowledge financial support from the Swedish Research Council (VR).

Data Availability Statement. This article has associated data in a data repository. The data is available at https://doi.org/10.7483/OPENDATA.ATLAS.B5M9.44TN.

Code Availability Statement. This article has associated code in a code repository. The code used in this work is available at https://github.com/j-gavran/NeuralFakeFactor.

References

[1] LHCb collaboration, *Measurement of lepton universality parameters in $B^+ \to K^+\ell^+\ell^-$ and $B^0 \to K^{*0}\ell^+\ell^-$ decays*, *Phys. Rev. D* **108** (2023) 032002 [arXiv:2212.09153] [INSPIRE].

[2] K. Lehmann and B. Stelzer, *The Fake Factor Method and its relation to the Matrix Method*, *Nucl. Instrum. Meth. A* **1054** (2023) 168376 [INSPIRE].

[3] ATLAS collaboration, *Tools for estimating fake/non-prompt lepton backgrounds with the ATLAS detector at the LHC*, 2023 *JINST* **18** T11004 [arXiv:2211.16178] [INSPIRE].

[4] CMS collaboration, *Search for new physics in same-sign dilepton events in proton-proton collisions at $\sqrt{s} = 13$ TeV*, *Eur. Phys. J. C* **76** (2016) 439 [arXiv:1605.03171] [INSPIRE].

[5] CMS collaboration, *Search for heavy neutral leptons in final states with electrons, muons, and hadronically decaying tau leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV*, *JHEP* **06** (2024) 123 [arXiv:2403.00100] [INSPIRE].

[6] ATLAS collaboration, *Search for doubly charged Higgs boson production in multi-lepton final states using 139 fb^{-1} of proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, *Eur. Phys. J. C* **83** (2023) 605 [arXiv:2211.07505] [INSPIRE].

[7] ATLAS collaboration, *Search for type-III seesaw heavy leptons in leptonic final states in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, *Eur. Phys. J. C* **82** (2022) 988 [arXiv:2202.02039] [INSPIRE].

[8] CMS collaboration, *Search for long-lived heavy neutral leptons with displaced vertices in proton-proton collisions at $\sqrt{s} = 13$ TeV*, *JHEP* **07** (2022) 081 [arXiv:2201.05578] [INSPIRE].

[9] W. Buttinger, *Background estimation with the ABCD method. Featuring the TRooFit Toolkit*, https://twiki.cern.ch/twiki/pub/Main/ABCDMethod/ABCDGuide_draft18Oct18.pdf, (2018).

[10] G. Kasieczka, B. Nachman, M.D. Schwartz and D. Shih, *Automating the ABCD method with machine learning*, *Phys. Rev. D* **103** (2021) 035021 [arXiv:2007.14400] [INSPIRE].

JHEP04(2026)188

[11] K. Cranmer, J. Brehmer and G. Louppe, *The frontier of simulation-based inference*, *Proc. Nat. Acad. Sci.* **117** (2020) 30055 [arXiv:1911.01429] [INSPIRE].

[12] S. Rizvi, M. Pettee and B. Nachman, *Learning likelihood ratios with neural network classifiers*, *JHEP* **02** (2024) 136 [arXiv:2305.10500] [INSPIRE].

[13] K. Cranmer, J. Pavez and G. Louppe, *Approximating Likelihood Ratios with Calibrated Discriminative Classifiers*, arXiv:1506.02169 [INSPIRE].

[14] M. Sugiyama, T. Suzuki and T. Kanamori, *Density Ratio Estimation in Machine Learning*, Cambridge University Press (2012) [DOI:10.1017/cbo9781139035613].

[15] A. Menon and C. S. Ong, *Linking losses for density ratio and class-probability estimation*, in *Proceedings of the 33rd International Conference on Machine Learning*, *Proc. Mach. Learn. Res.* **48** (2016) 304, https://proceedings.mlr.press/v48/menon16.html.

[16] B. Nachman and J. Thaler, *Neural resampler for Monte Carlo reweighting with preserved uncertainties*, *Phys. Rev. D* **102** (2020) 076004 [arXiv:2007.11586] [INSPIRE].

[17] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*, MIT Press (2016) http://www.deeplearningbook.org.

[18] A. Andreassen and B. Nachman, *Neural Networks for Full Phase-space Reweighting and Parameter Tuning*, *Phys. Rev. D* **101** (2020) 091901 [arXiv:1907.08209] [INSPIRE].

[19] Y. Gorishniy, I. Rubachev, V. Khrulkov and A. Babenko, *Revisiting Deep Learning Models for Tabular Data*, arXiv:2106.11959.

[20] Y. Gorishniy, I. Rubachev and A. Babenko, *On Embeddings for Numerical Features in Tabular Deep Learning*, *Adv. Neural Inf. Process. Syst.* **35** (2022) 24991, [arXiv:2203.05556].

[21] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, arXiv:1711.05101 [INSPIRE].

[22] K. He, X. Zhang, S. Ren and J. Sun, *Identity Mappings in Deep Residual Networks*, in *Computer Vision — ECCV 2016*, Springer International Publishing (2016), p. 630–645 [DOI:10.1007/978-3-319-46493-0_38].

[23] P.J. Huber, *Robust Estimation of a Location Parameter*, *Annals Math. Statist.* **35** (1964) 73.

[24] ATLAS collaboration, *ROOT ntuple format 2015-2016 proton-proton open data for education and outreach beta release from the ATLAS experiment*, CERN Open Data Portal (2025), https://doi.org/10.7483/OPENDATA.ATLAS.B5M9.44TN.

[25] ATLAS collaboration, *Measurement of the W-boson mass in pp collisions at* $\sqrt{s} = 7$ *TeV with the ATLAS detector*, *Eur. Phys. J. C* **78** (2018) 110 [*Erratum ibid.* **78** (2018) 898] [arXiv:1701.07240] [INSPIRE].

[26] J. Gavranovič, *NeuralFakeFactor*, https://github.com/j-gavran/NeuralFakeFactor.

JHEP04(2026)188