Measurement of $\mathcal{R}(D)$ and $\mathcal{R}(D^*)$ Using Semileptonic Tags and Hadronic τ Decays

Thesis by
Jae Hong Kim

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2019
Defended January 15, 2019

Jae Hong Kim
ORCID: 0000-0003-1968-2753

Acknowledgements

Thank you —

To my wife Diana — for your love and making me feel complete.

To my advisor Prof. David Hitlin and Prof. Frank Porter — for providing me with a nurturing and learning environment for me to grow as a physicist and as a person, and for giving me your trust and confidence on countless occasions.

To my fellow (ex-)group members Daniel Chao, Bernard Echenard, and Yunxuan Li — for the support and discussions that made this thesis possible.

To Viktor Shcherbatyuk — for the tea and maintaining the computers essential for this thesis.

To my friends — for helping me feel right at home on the opposite side of the country.

And to my mom, dad, and sisters — for your unconditional love and support.

Abstract

We present a measurement of $\mathcal{R}(D^{(*)}) = \mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)/\mathcal{B}(B \to \overline{D}^{(*)}\ell\nu_\ell)$ using semileptonic tagging and hadronic τ decays on the 433 fb^{-1} of data collected at the $\Upsilon(4S)$ resonance using the *BABAR* detector at the PEP-II collider. We obtain a high statistics data sample using loose selection criteria. The signal is extracted by performing a 2-dimensional fit of the component densities to the kernel density estimate of the data, which is made computationally tractable by algorithmic improvements and speedups provided by graphics processing units. We obtain two distinct central values based on the model used to represent the $B\overline{B}$ background densities: $\mathcal{R}(D) = 0.231 \pm 0.028 \pm 0.028$ and $\mathcal{R}(D^*) = 0.127 \pm 0.019 \pm 0.031$ with a correlation of 0.06 and $\mathcal{R}(D) = 1.454 \pm 0.028 \pm 0.028$ and $\mathcal{R}(D^*) = 1.507 \pm 0.019 \pm 0.031$ with a correlation of 0.06. The region encompassed by the two results are consistent with both the Standard Model prediction and the world average.

Contents

Chapter 1

Introduction

The layout of this thesis can be broken down into two parts: the first is a description of the background in which we clearly define the objective of this analysis. The second consists of exposition into how the measurement, including uncertainty estimation, was carried out, and a discussion of the results.

The first part starts in Chapter 1, where we discuss the framework in which the measurement takes place. It is meant to be a high-level overview of the topic for a general audience. In Chapter 2, we delve deeper into the theory behind our current understanding of the quantities of interest, $\mathcal{R}(D^{(*)})$, which results in their theoretical prediction values. In Chapter 3, we review the *BABAR* detector and the hardware that collected the data used in this analysis.

1.1 Background

The ultimate goal of the field of particle physics is to completely specify the laws of physics that determine the characteristics and interactions of the *elementary* particles, commonly referred to as the Theory of Everything (TOE). A particle is elementary, or fundamental, if it is not composed of any other particles[1]. In a sense, they are the building blocks of larger particles and atoms, and in turn of molecules and the universe.

One might be tempted to think that if we completely understand the laws that govern these elementary particles, that we have in a sense, 'solved' physics. This is a way of thinking called *reductionism*, and as described by Anderson [1], it does not account for the *emergent* phenomena that occur when we widen our field of view. As a particle physicist, one is entitled to feel a certain pride about investigations of the most fundamental aspect of nature, but would be foolish to let that pride blind oneself from the advancements in other fields of physics.

1.1.1 The Standard Model

The Standard Model (SM) can be thought of as our best attempt at unifying every interaction besides gravity. The keyword being 'best', as we know certain limitations of the current version of SM, one being the subject of this thesis, the prediction of $\mathcal{R}(D^{(*)})$, but others include the matter-antimatter asymmetry and gravity. The SM, if it were the true description of nature, should be able to give us the correct predictions of all experimentally verifiable quantities involving elementary particles. On the other hand, the fact that the SM disagrees with some experimental evidence does not mean that all hope is lost. In fact, the SM is able to explain most observations remarkably well, and gives us great confidence that

[1] This statement, along with many others throughout the thesis, should be followed by the words "as far as we know".

we are heading in the right direction. The SM, like all other models, can be iterated upon to improve its correctness, which involves great communication between the experimentalists and the theorists.

The current version of the Standard Model can actually be written down as a single equation, but that would not be very informative. In addition, the SM is a *field theory*, meaning it treats particles as quantum fields. However, for the sake of clarity, we can view the particles of SM as classical particles and the SM as a collection of such particles, and defer the relevant formalisms to later chapters.

The elementary particles can be divided into two groups: *fermions*, which consists of quarks u, d, s, c, b, t and leptons $e, \mu, \tau, \nu_e, \nu_\mu, \nu_\tau$, and *bosons*, $\gamma, g, W^\pm, Z^0$, H^0. The bosons are often referred to as force-carrier particles, as all interactions of fermions can be described as exchanges of such particles. Some particles, such as the leptons, are stable particles happy to be by themselves while others such as the quarks are always found in a bound-state with other quarks. This is crucial because it determines what we can observe. More specifically, we always observe quarks in pairs (mesons) or triplets (baryons). Some examples of composite particles that are relevant to this analysis are the $\pi, \rho, K, \Upsilon, B, D$ mesons.

As one can see, all particles are represented by a symbol, which is sometimes stylized with a bar ($\bar{}$), superscripts (*,**,$^+$,$^-$,^{0}), or subscripts (e.g. $_{\mu,S}$). In most cases, Greek letters with different annotations in fact represent different particles, but since they are minor variations of each other, they can be represented by the same letter. For example, π^+ and π^- make up a particle-antiparticle pair, representing particles with different charges, whereas π^0 and $\overline{\pi^0}$ are actually the same particle (it is its own antiparticle). Still, all three particles are considered to be π-mesons since they all consist of a quark-antiquark pair of u and d quarks.

1.1.2 Decay rate and branching fraction

In order to verify the SM, we need to first establish which predictions are measurable and which are not. One of the most commonly measured quantity is the decay rate Γ of a particle, which can be thought of as the inverse of its lifetime τ. For instance, the electron, being a stable particle, has in theory an infinite lifetime, and consequently a decay rate of zero[2].

Let us be a bit more specific and ask given a particle X, what is its decay rate if we restrict the possible output to be only particles A and B? This is equivalent to asking what is the *branching fraction* of the *decay mode* $X \to AB$? In other words, what is the probability of X decaying into AB given that it decays? The main task of this thesis is to measure the branching fractions $\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)$, where $D^{(*)}$ denotes D and D^* in the sense that we are measuring two quantities, $B \to \overline{D}\tau\nu_\tau$ and $B \to \overline{D}^*\tau\nu_\tau$. These branching fractions in turn will be used to calculate $\mathcal{R}(D^{(*)})$.

The predictions of the branching fractions of interest are calculated using the theories of electroweak (EW) and quantum chromodynamics (QCD). EW theory describes the interactions mediated by the γ, Z^0, and $W^\pm$ bosons, while QCD describes the interactions between quarks and gluons (g). There are many ways within QCD to calculate the same quantity, the differences being the assumptions made in each method. This allows for differing approximations of the same quantity. The approach taken to calculate the most precise predictions $\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)$will be the topic of Chapter 2.

The fact that we are estimating the probability of something should give hints on *how* the measurement can be performed. The idea is simple: suppose we observe n number of B mesons, and m of them decay into our mode of interest. It follows that our estimate of the branching fraction is m/n, with a statistical uncertainty that depends on m and n. To give you a sense of n, the *BABAR* experiment collected 471 million $B\overline{B}$ pairs.

[2]Measuring the lifetimes of stable particles such as e and p is an active area of research.

1.1.3 Particle accelerators and detectors

Just like many ideas in science, the concept of the measurement is simple but the execution is difficult. In the context of our measurement, where do we find these B mesons, and how do we identify them and the decay products $D^{(*)}, \tau$, and ν_τ? The short answer is that we produce B's by colliding particles at a very high energy and identify particles by studying their behavior when propagating through our detector. While the engineering innovation that must occur to build such accelerators and detectors takes great effort, as an analyst of the dataset, the bottleneck lies in the fact that it is impossible to identify the particles correctly every single time. Indeed, the neutrinos ν cannot even be directly detected in our experiment, we only hypothesize their existence for each event based on the missing energy. The accelerator and the detector for the *BABAR* experiment will be discussed further in Chapter 3.

1.2 Analysis objective

As stated above, the main objective of this analysis is to measure $\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)$, which will be then used to calculate the quantity

$$\mathcal{R}(D^{(*)}) = \frac{\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)}{\mathcal{B}(B \to \overline{D}^{(*)}\ell\nu_\ell)}, \tag{1.1}$$

where l denotes e or μ. Rather than measuring both the numerator and the denominator, we will only measure the numerator, and use the world-average value for the denominator as its precision is already quite good. This allows us to tune our analysis to maximize the sensitivity to the numerator.

1.2.1 Discrepancy between theory and experiment

The most precise predictions of $\mathcal{R}(D^{(*)})$ as of writing are [2]:

$$\begin{aligned} \mathcal{R}(D)_{\text{SM}} &= 0.299 \pm 0.003, \\ \mathcal{R}(D^*)_{\text{SM}} &= 0.258 \pm 0.005. \end{aligned} \tag{1.2}$$

While the world average of previous measurements are:

$$\begin{aligned} \mathcal{R}(D)_{\text{exp}} &= 0.407 \pm 0.039 \pm 0.024, \\ \mathcal{R}(D^*)_{\text{exp}} &= 0.306 \pm 0.013 \pm 0.007. \end{aligned} \tag{1.3}$$

These imply $2.3(3.0)\sigma$ discrepancies[3] between the prediction and the measurement of $\mathcal{R}(D)(\mathcal{R}(D^*))$, and a combined discrepancy of 3.78σ.

In the field of experimental high energy physics, we assign arbitrary thresholds to indicate the statistical power of the discoveries. For example, a 3σ result is referred to as an *evidence* of the discrepancy, whereas a 5σ result is called a *discovery*, and is the threshold where physicists can comfortably state that the discrepancy is indeed real.

We can summarize the goal of this analysis as simply providing another data point contributing to the world average. That is inherently different from saying our goal is to push the discrepancy into the discovery territory, since as analysts we should not bias ourselves by searching for the discrepancy as if it is real. In fact, the results of this analysis could turn out to be consistent with the Standard Model predictions and decrease the discrepancy.

[3] σ denotes the deviation of the observations when compared to the prediction assuming the prediction is correct.

1.2.2 Previous measurements

There have been six previous measurements, two of which measured both $\mathcal{R}(D^{(*)})$ and the other four only measured $\mathcal{R}(D^*)$. They are shown in Figure 1.1 and Table 1.1.

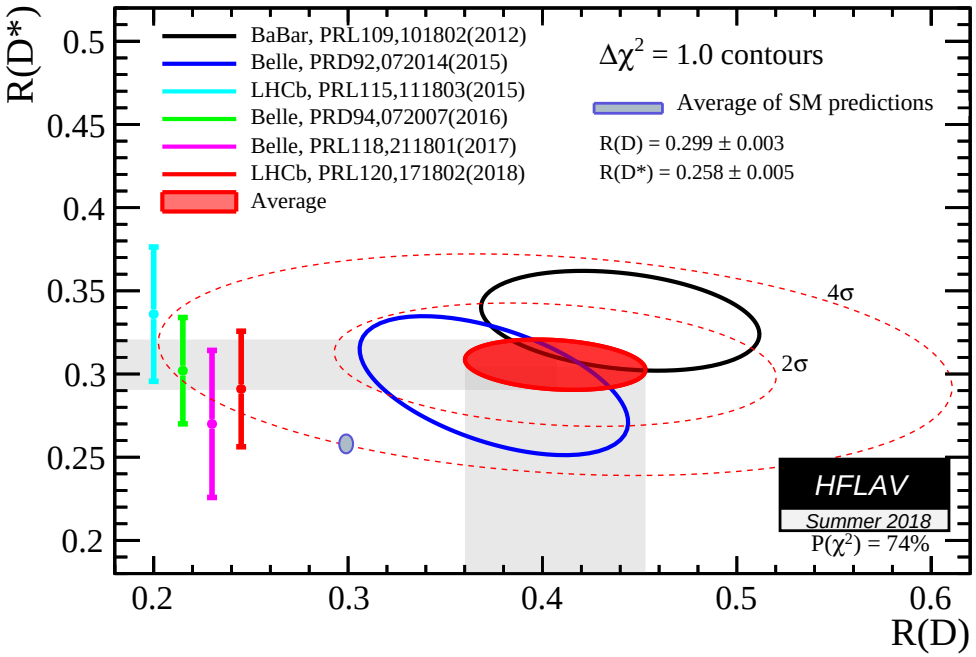


Figure 1.1: Visual summary of recent results [2].

	$\mathcal{R}(D)$	$\mathcal{R}(D^*)$
BABAR 2013 [3]	$0.440 \pm 0.058 \pm 0.042$	$0.332 \pm 0.024 \pm 0.018$
Belle 2015 [4]	$0.375 \pm 0.064 \pm 0.026$	$0.293 \pm 0.038 \pm 0.015$
LHCb 2015 [5]	-	$0.336 \pm 0.027 \pm 0.03$
Belle 2016 [6]	-	$0.302 \pm 0.030 \pm 0.011$
Belle 2017 [7]	-	$0.270 \pm 0.035^{+0.028}_{-0.025}$
LHCb 2018 [8]	-	$0.291 \pm 0.019 \pm 0.029$
Average (HFLAV Summer 2018)	$0.407 \pm 0.039 \pm 0.024$	$0.306 \pm 0.013 \pm 0.007$
Standard Model [2]	0.299 ± 0.003	0.258 ± 0.005

Table 1.1: Experimental results.

Much of the work of this thesis overlaps with that of Daniel Chao's thesis [9] since this measurement was a collaboration between the author and him. The major difference between [9] and the results shown in this thesis is the attempt to reduce the observed discrepancies between the simulation and the detector data.

Chapter 2

Phenomenology

In this chapter, we discuss the theory behind the predictions shown in (1.2). More specifically, we will assume that the reader has a basic understanding of quantum field theory and is familiar with concepts such as Feynman diagrams, Fermi's golden rule, and gamma matrices.

The topic of interest greatly depends on the theory behind the semileptonic quark transition $b \to c\ell\overline{\nu}$ with $\ell = e, \mu, \tau$, which will be used in the prediction of $\mathcal{B}(\overline{B} \to D^{(*)}\ell\overline{\nu}_\ell)$. This becomes clear when we examine the Feynman diagram shown in Fig. 2.1: the spectator quark $\overline{d}$, which was initially bound with the b quark to make up the B meson, simply binds with the decay product c quark which we detect as a D meson. Thus, the most interesting part of the decay indeed is the $b \to c\ell\overline{\nu}$.

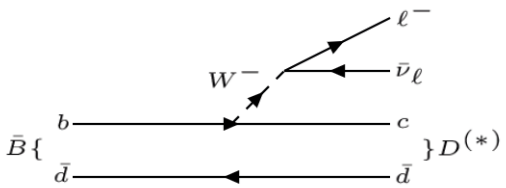


Figure 2.1: Feynman diagram of the $\overline{B} \to D^{(*)}\ell\overline{\nu}_\ell$.

The most precise predictions above are actually naive averages (i.e. they do not take correlations into account) of four separate results. Looking at the individual results, we can see that there are two fundamentally different approaches: heavy quark effective theory (HQET) calculations and lattice QCD calculations. We will focus on the HQET predictions, and in turn on how the branching fractions are parametrized and what contributes to the theoretical uncertainties.

2.1 The physical picture

The initial state of the system is a single B meson containing a heavy quark Q and a light antiquark $\overline{q}$. Let p be the momentum of the meson in the laboratory frame. At time $t = t_0$, Q decays into a W and another heavy quark Q', where the W decays into a pair of leptons $\ell\overline{\nu}$ and Q' hadronizes with $\overline{q}$ to produce a new meson with momentum p', in this case a $D^{(*)}$.

This simple description of the system does not take into account the complex interactions between Q and $\overline{q}$ through gluons and the self-interactions of the gluons themselves, whose strength of interaction is of order $\Lambda_{\text{QCD}} \sim 220$ MeV. This makes the calculation of observables such as the decay rate from first principles intractable. This secondary sub-system is

aptly referred to as the *brown muck*, and our goal becomes that of providing quantitative description of the interaction between the brown muck and Q. Nevertheless, we are still able to derive the expressions of the decay rates in terms of *form factors* which are functions of p and p'. But it is precisely the parameterization of these form factors that become intractable and require additional assumptions about the system.

2.2 QCD, Λ_{QCD}, and the heavy quark symmetry

The recipe for calculations of $\mathcal{R}(D^{(*)})$ is as follows: using the QCD part of the SM Lagrangian $\mathcal{L}_{\mathrm{SM}}$, we define the amplitude of the decay. To allow for predictions of physical observables, we define an effective theory based on the heavy quark symmetry [10], which is the symmetry of the effective Lagrangian $\mathcal{L}_{\mathrm{eff}}$ in the limit $m_Q \to \infty$, where Q indicates the heavy quark, c or b. In other words, when the masses of the heavy quarks are much larger than the energy scale of the interaction, the heavy quarks can be treated as a static source of color.

Note that the heavy quark symmetry is *not* a symmetry of $\mathcal{L}_{\mathrm{SM}}$, which means that the validity of the predictions from the heavy quark effective theory (HQET) depends on the kinematic region and only appropriate for $m_Q \gg \Lambda_{\mathrm{QCD}}$. The reader is directed towards the excellent overview of this topic by Neubert [11].

Once we have parameterized the amplitude, the expressions of the branching fractions are straight-forward application of Fermi's golden rule. It is then we input the measured values of the form factors along with QCD constants for the SM prediction of $\mathcal{R}(D^{(*)})$.

2.3 Standard Model amplitudes

There are two vertices in the Feynman diagram above: the brown muck and a vertex between the W and the lepton pairs. They correspond to the following quark and leptonic currents [12]:

$$J^{\mu}_{\nu\ell} \equiv \overline{\psi}_{\nu}\gamma^{\mu}\left(1-\gamma^{5}\right)\psi_{\ell}, \qquad J^{\mu}_{cb} \equiv \overline{\psi}_{c}\gamma^{\mu}\left(1-\gamma^{5}\right)\psi_{b}, \tag{2.1}$$

and the Lagrangian of such a system is

$$\mathcal{L} = -\frac{G_{\mathrm{F}}}{\sqrt{2}}|V_{cb}|J^{\dagger\mu}_{\nu\ell}J_{cb\mu} + \mathrm{h.c.}, \tag{2.2}$$

where G_{F} is the Fermi coupling constant the and V_{cb} is the CKM matrix element between c and b quarks.

The idea is to start in the rest frame of the $\overline{B}$ and define the appropriate kinematic variables and polarization vectors for the hadronic current, then boost into the rest frame of the virtual W to account for the leptonic current. See Fig. 2.2 for the definitions of kinematic variables in both frames of reference.

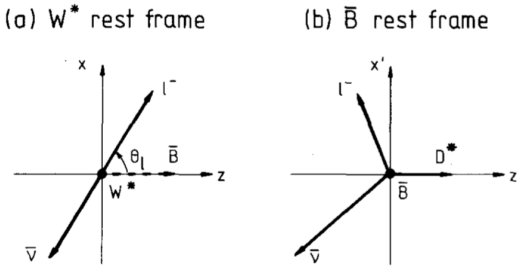


Figure 2.2: $\overline{B} \to D^{(*)}\ell\overline{\nu}_{\ell}$ seen from a) W^* rest frame and b) $\overline{B}$ rest frame. [12]

Let $M = D$ or D^* and define the following kinematic variables in the rest frame of $\overline{B}$:

$$p_B^\mu = (m_B, 0, 0, 0), \qquad p_M^\mu = (E_M, 0, 0, p_M), \qquad q^\mu \equiv p_B - p_M = (q^0, 0, 0, -p_M). \tag{2.3}$$

Defining $Q_\pm \equiv (m_B \pm m_M)^2 - q^2$ gives us the following relationships:

$$2m_B E_M = m_B^2 + m_M^2 - q^2, \qquad 2m_B p_M = \sqrt{Q_+ Q_-}, \qquad 2m_B q^0 = m_B^2 - m_M^2 + q^2,$$

and the following variable:

$$x \equiv p_{\ell \cdot B} / m_B^2 = m_B E_\ell. \tag{2.4}$$

Note that q^2 and x are Lorentz invariant.

Using the variables defined above, the amplitudes of the scattering processes are:

$$\mathcal{M}^{\lambda_\ell}_{\lambda_M}(q^2, x) = \frac{G_\mathrm{F}}{\sqrt{2}} V_{cb} \sum_{\lambda_W} \eta_{\lambda_W} L^{\lambda_\ell}_{\lambda_W} H^{\lambda_M}_{\lambda_W}. \tag{2.5}$$

To specify the exact form of the hadronic and leptonic amplitudes, let λ be the particle helicities, $\epsilon(q, \lambda)$ be the polarization vectors, and η_λ be the metric. For example, for $M = D$ we have $\lambda_M = 0$, whereas for $M = D^*$ we have $\lambda_M \in \{-1, 0, 1\}$. Helicities of the virtual W $\lambda_W \in \{-1, 0, 1, s\}$, where s is the zero helicity state in the case of pseudoscalar $M = D$.

This gives the following relationship of the metric tensor given $\eta_\pm = \eta_0 = -\eta_s = 1$:

$$-g^{\mu\nu} = \sum_{\lambda_W} \eta_{\lambda_W} \epsilon^\mu_W \epsilon^{*\nu}_W. \tag{2.6}$$

The hadronic ($\overline{B} \to MW^*$)[1] and leptonic ($W^* \to \ell\overline{\nu}$) amplitudes can be expressed as

$$\begin{aligned} H^{\lambda_M}_{\lambda_W}(q^2) &\equiv \epsilon^*_{W_\mu} \langle M(p_M, \lambda_M) | J^\mu_{cb} | \overline{B}(p_{\bar{B}}) \rangle \\ L^{\lambda_M}_{\lambda_W}(q^2, x) &\equiv \epsilon_{W_\mu} \langle \ell^-(p_\ell, \lambda_\ell)\, \overline{\nu}(p_{\overline{\nu}}) | J^{\dagger\mu}_{\ell\overline{\nu}} | 0 \rangle, \end{aligned} \tag{2.7}$$

where the polarzation vectors are

$$\epsilon(q, \pm)^\mu = \mp\frac{1}{2}(0, 1, \mp i, 0), \qquad \epsilon(q, 0)^\mu = \frac{1}{q^2}(p_M, 0, 0, -q^0), \qquad \epsilon(q, s)^\mu = \frac{1}{\sqrt{q^2}} q^\mu. \tag{2.8}$$

2.3.1 Leptonic amplitudes

In the rest frame of W^*, (2.8) becomes:

$$\epsilon(q, \pm)^\mu = \mp\frac{1}{2}(0, 1, \pm i, 0), \quad \epsilon(q, 0)^\mu = \frac{1}{q^2}(0, 0, 0, -1), \quad \epsilon(q, s)^\mu = \frac{1}{\sqrt{q^2}} q^\mu = (1, 0, 0, 0). \tag{2.9}$$

Then the leptonic amplitudes for the pseudoscalar case $M = D$ are

$$L^-_\pm(q^2, x) = 2\sqrt{q^2} v d_\pm \qquad L^-_0(q^2, x) = -2\sqrt{q^2} v d_0, \qquad L^-_s(q^2, x) = 0, \tag{2.10}$$

while for $M = D^*$ we have

$$L^+_\pm(q^2, x) = \pm\sqrt{2} m_\ell v d_0, \qquad L^+_0(q^2, x) = \sqrt{2} m_\ell v (d_+ - d_-), \qquad L^+_s(q^2, x) = -2 m_\ell v, \tag{2.11}$$

where

$$v = \sqrt{1 - \frac{m_\ell^2}{q^2}}, \qquad d_\pm = \frac{1 + \cos\theta_\ell}{\sqrt{2}}, \qquad d_0 = \sin\theta_\ell. \tag{2.12}$$

[1] The * references the virtuality of W.

2.3.2 Hadronic amplitudes

The matrix elements of interest for the $\overline{B} \to M$ transitions are

$$\langle D|V_{cb}^{\mu}|\overline{B}\rangle, \qquad \langle D^*|V_{cb}^{\mu}|\overline{B}\rangle, \qquad \langle D^*|A_{cb}^{\mu}|\overline{B}\rangle. \tag{2.13}$$

It turns out that these three matrix elements can be fully described by six *form factors*, all functions of q^2: $f_{\pm}(q^2)$ and $f_i(q^2), i = 1, 2, 3, 4$.

This results in the following amplitudes for the case of $M = D$:

$$\begin{aligned} H_0^s(q^2) &= f_+(q^2)\frac{\sqrt{Q_+Q_-}}{\sqrt{q^2}}, \\ H_s^s(q^2) &= f_+(q^2)\frac{m_B^2 - m_M^2}{\sqrt{q^2}} + f_-(q^2)\sqrt{q^2}, \end{aligned} \tag{2.14}$$

and for the case of $M = D^*$:

$$\begin{aligned} H_{\pm}^{\pm}(q^2) &= f_2(q^2) \mp f_1(q^2)\sqrt{Q_+Q_-}, \\ H_0^0(q^2) &= -\frac{1}{2m_M\sqrt{q^2}}\left\{(m_B^2 - m_M^2 - q^2)f_2(q^2) + Q_+Q_-f_3(q^2)\right\}, \\ H_s^0(q^2) &= -\frac{\sqrt{Q_+Q_-}}{2m_M\sqrt{q^2}}\left\{f_2(q^2) + (m_B^2 - m_M^2)f_3(q^2) + q^2 f_4(q^2)\right\}. \end{aligned} \tag{2.15}$$

2.3.3 Form factor parametrization

Notice that up to this point, we have not made any approximation based on heavy quark symmetry. It turns out that in the heavy quark limit, the amplitudes can be reduced to depend on a single function $\xi(q^2)$, commonly referred to as the Isgur-Wise function [10].

We will now proceed with the most commonly used form factor parameterization model called CLN, short for Caprini-Lellouch-Neubert [13]. The idea is to use the dispersion constraints introduced by the heavy quark symmetry to expand the form factors about the zero-recoil point ($v = v'$).

To do this, we first introduce a new variable $w = v \cdot v'$, which is related to the Lorentz invariant quantity q^2 by

$$q^2 = m_B^2 + m_M^2 - 2m_B m_M w. \tag{2.16}$$

This allows us to replace any function $f(q^2)$ with a function $g(w)$ with the zero-recoil point being $w = 1$.

2.3.4 Form factor parametrization for $M = D$

Rewriting (2.14) as functions of w gives us:

$$\begin{aligned} H_0^s(w) &= \sqrt{m_B m_D}\frac{m_B + m_D}{\sqrt{q^2(w)}}\sqrt{w^2 - 1}V_1(w), \\ H_s^s(w) &= \sqrt{m_B m_D}\frac{m_B + m_D}{\sqrt{q^2(w)}}(w + 1)S_1(w), \end{aligned} \tag{2.17}$$

where $S_1(w)$ is also commonly called $\mathcal{G}(w)$ in literature.

The CLN model uses dispersion relations to expand the form factors above in terms of $z(w) = \frac{\sqrt{w+1}-\sqrt{2}}{\sqrt{w+1}+\sqrt{2}}$:

$$\begin{aligned} \frac{V_1(w)}{V_1(1)} &= 1 - 8\rho_1^2 z + (51\rho_1^2 - 10)z^2 - (262\rho_1^2 - 84)z^3 + \mathcal{O}(z^4) \\ \frac{S_1(w)}{V_1(w)} &= (1 + \Delta(w)), \end{aligned} \tag{2.18}$$

where $\Delta(w) = -0.019 + 0.041(w-1) - 0.015(w-1)^2 + \mathcal{O}(w^3)$.

We see that the amplitudes only depend on two parameters, whose experimental values are [14]:

$$V_1(1) = 1.054 \pm 0.004 \pm 0.008, \qquad \rho_1 = 1.128 \pm 0.024 \pm 0.023. \tag{2.19}$$

It is worth noting that the numerical value of $V_1(1)$ does not matter in the measurement of $\mathcal{R}(D)$, since it cancels out in the fraction, leaving ρ_1 as the only relevant parameter when estimating the effects of form factor uncertainties when using the CLN parametrization scheme.

2.3.5 Form factor parametrization for $M = D^*$

Rewriting (2.15) in terms of w gives us:

$$\begin{aligned}
H_{\pm}^{\pm}(w) &= (m_B + m_{D^*})A_1(w) \mp \frac{2m_B|p|}{m_B + m_{D^*}}V(w), \\
H_0^0(w) &= \frac{1}{2m_{D^*}\sqrt{q^2}}\left[(m_B^2 - m_{D^*}^2 - q^2)(m_B + m_{D^*})A_1(w) - \frac{4m_B^2|p|^2}{m_B + m_{D^*}}A_2(w)\right], \\
H_s^0(w) &= \frac{2m_B|p|}{\sqrt{q^2}}A_0(w),
\end{aligned} \tag{2.20}$$

where p is the momentum of D^* in the rest frame of B (p_M in (2.3)).

It is customary to represent the four form factors V, A_0, A_1, A_2 in terms of the following form factor ratios:

$$\begin{aligned}
R_0(w) &= r\frac{A_0(w)}{h_{A_1}(w)}, \\
R_1(w) &= r\frac{V(w)}{h_{A_1}(w)}, \\
R_2(w) &= r\frac{A_2(w)}{h_{A_1}(w)},
\end{aligned} \tag{2.21}$$

where $r = 2\sqrt{m_B m_{D^*}}/(m_B + m_{D^*})$ and

$$h_{A_1}(w) = A_1(w)\frac{1}{r}\frac{2}{w+1}. \tag{2.22}$$

The expansions of these form factors are:

$$\begin{aligned}
\frac{h_{A_1}(w)}{h_{A_1}(1)} &= 1 - 8\rho^2 z + (53\rho^2 - 15)z^2 - (231\rho^2 - 91)z^3 + \mathcal{O}(z^4), \\
R_0(w) &= R_0(1) - 0.11(w-1) + 0.01(w-1)^2 + \mathcal{O}(w^3), \\
R_1(w) &= R_1(1) - 0.12(w-1) + 0.05(w-1)^2 + \mathcal{O}(w^3), \\
R_2(w) &= R_2(1) + 0.11(w-1) - 0.06(w-1)^2 + \mathcal{O}(w^3),
\end{aligned} \tag{2.23}$$

leading us to conclude that there are five free parameters, whose experimental values are [2]:

$$\begin{aligned}
\mathcal{F}(1) \equiv h_{A_1}(1) &= 0.906 \pm 0.013, \\
\rho^2 &= 1.205 \pm 0.015 \pm 0.021, \\
R_0(1) &= 1.14 \pm 0.07, \\
R_1(1) &= 1.404 \pm 0.032, \\
R_2(1) &= 0.854 \pm 0.020.
\end{aligned} \tag{2.24}$$

2.4 Standard Model prediction of $\mathcal{R}(D^{(*)})$

2.4.1 Decay rates

Fermi's golden rule gives us the recipe to calculate the differential decay rate given the scattering amplitude from (2.5):

$$d\Gamma = \frac{1}{2m_B} \sum_{\lambda_\ell \lambda_M} |\mathcal{M}^{\lambda_\ell}_{\lambda_M}|^2 d\Phi_3, \tag{2.25}$$

with

$$d\Phi_3 = \frac{(q^2 - m_\ell^2)\sqrt{Q_+ Q_-}}{256\pi^3 m_B^2 q^2} dq^2 d\cos\theta_\ell, \tag{2.26}$$

where θ_ℓ is the angle between ℓ and B in the rest frame of B.

Integrating over θ_ℓ gives us the q^2 spectra of the decay rate [15]:

$$\frac{d\Gamma}{dq^2} = \left(1 - \frac{m_\ell^2}{q^2}\right)^2 \left((|H_+^+|^2 + |H_-^-|^2 + |H_0^0|^2) \left(1 + \frac{m_\ell^2}{2q^2}\right) + \frac{3}{2}\frac{m_\ell^2}{q^2} |H_s^0|^2 \right). \tag{2.27}$$

Finally, we calculate the branching fractions by integrating (2.27) over the valid q^2 region: $m_\ell^2 \leq q^2 \leq (m_B - m_M)^2$, and input the best known values for the various constants and form factors.

2.4.2 Predicted values of $\mathcal{R}(D^{(*)})$

While the above recipe of plugging in the best values of parameters into the expressions gives us *an* answer, the most precise predictions are calculated a bit differently [16].

The authors of [16] use the information regarding the form factors from lattice QCD calculations, QCD sum rules, and experimental data (from the Belle experiment) to perform a global fit. This results in a highly constrained fit with low statistical uncertainty.

2.4.3 Theoretical explanations and implications

As with any possible source new physics, the discrepancy of $\mathcal{R}(D^{(*)})$ has been a very active area of theoretical research. Some of the reasons for a strong theoretical and experiment interests are the fact that the process $\overline{B} \to D^{(*)}\tau\overline{\nu}_\ell$ is a tree-level process and that the amount of discrepancy seems to be large ($\sim 30\%$).

If the discrepancy between the prediction and the measurement of $\mathcal{R}(D^{(*)})$ is confirmed to be real (i.e. a 5σ deviation between the two), it would be an example of the violation of lepton universality. In the current version of the SM, the three generations of leptons are assumed to have identical behaviors besides the effects due to the differing masses. This characteristic of the leptons is denoted as lepton universality and its violation would have profound new physics implications, which can differ based on the amount of violation observed.

One of such implications is discussed in the initial *BABAR* measurement of $\mathcal{R}(D^{(*)})$ from 2013 [3], where the analysts focused on the evidence towards a charged Higgs boson of a two-Higgs doublet model [17, 18]. Other new physics possibilities include leptoquarks and composite fermions [19].

Chapter 3

PEP-II and *BABAR*

In this chapter, we review the hardware that comprises the *BABAR* detector and the surrounding environment. At a high level, the electron and positron beams are designed such that they collide at the center of the detector. The energies of the beams are tuned to the mass of the $\Upsilon(4S)$, which decays into a $B\overline{B}$ pair more than 95% of the time [14]. The B mesons subsequently decay into charged or neutral particles, which are detected by specialized subsystems. The responses of the subsystems are simply hits in the detector which only records the magnitude of the hit (i.e. energy of the particle that passed through the system). It is the job of the offline system to perform the track- and cluster-finding using the hits for the purpose of *particle identification*, or PID. The track-level information is the abstraction of this data to be used by analysts who perform event reconstruction and extract relevant kinematic features.

The content of this chapter is as follows: first we review the accelerator facility that produces the electron-positron beams at the specified energy. Next, we review the components of the *BABAR* detector. Lastly, we briefly discuss the trigger system. Much of the material presented is taken from [20] and [21].

3.1 The PEP-II accelerator

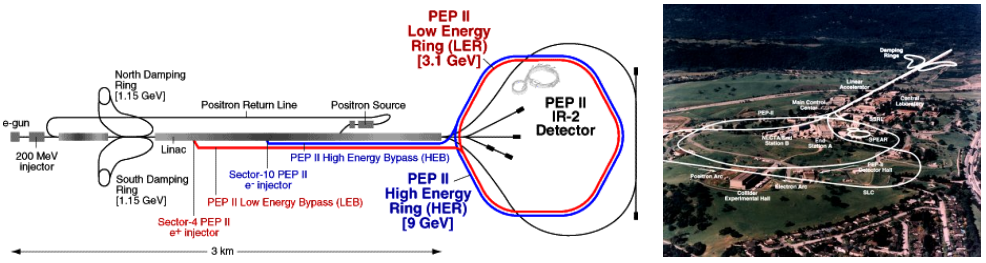


Figure 3.1: PEP-II

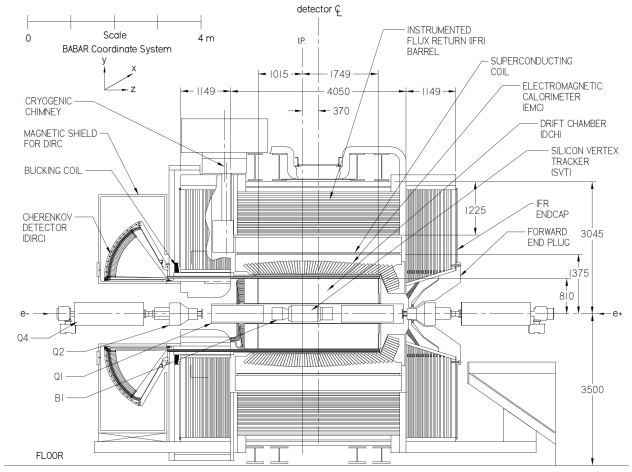
PEP-II was an e^+e^- storage ring system located at the Stanford Linear Accelerator Center (SLAC) in Menlo Park, CA. The origin of the electrons is a Ti-sapphire laser that feeds the linear accelerator while the positrons are produced by colliding a part of the electron beam with a titanium target. When the beams are accelerated to the specified energies, they are each injected into the two storage rings of PEP-II (one for e^+ and another for e^-). An interesting historical note is that its predecessor PEP was designed to search for the top quark at 29 GeV (the top quark is measured to have a mass of 172 GeV).

The key design feature of the PEP-II is the asymmetry between the energies of the electron beam (9 GeV) and the positron beam (3.1 GeV). This turned out to be a great

decision in improving the identifiability of B mesons due to their significant momenta.

In addition to running the accelerator at the peak of the $\Upsilon(4S)$ resonance (on-peak), the *BABAR* experiment also collected data just below the peak (off-peak) by reducing the electron beam energy down to 8.9 GeV. This dataset provided invaluable information on handling the background in many analyses.

3.2 The *BABAR* detector



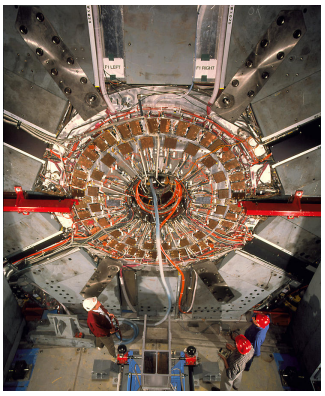
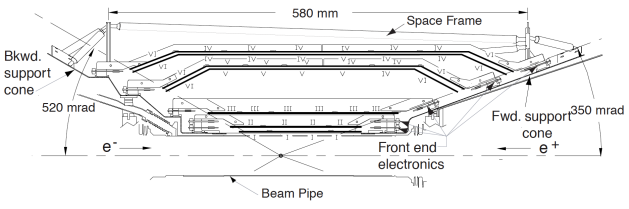


Figure 3.2: The *BABAR* detector.

The *BABAR* detector consists of six major subsystems: Silicon Vertex Tracker (SVT), Drift Chamber (DCH), Detector of Internally Reflected Cherenkov light (DIRC), CsI Electromagnetic Calorimeter (EMC), the superconducting coil, and Instrumented Flux Return (IFR).

In order to discuss the design choices for the detector, it is useful to understand how the design requirements were determined. The *BABAR* experiment was motivated by a singular physics goal: to measure the CP asymmetry in neutral B decays. Most, if not all, design choices of the detector are results of the balancing between meeting the specified physics sensitivity while staying under the budget acquired prior to the construction.

3.2.1 Charged particle tracking: SVT and DCH



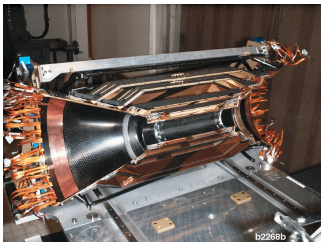


Figure 3.3: Silicon vertex tracker.

The SVT is the innermost component of the detector; its primary purpose is to reconstruct the decay vertices of the $B\overline{B}$ pair by tracking the positions and angles of the charged

particles close to the interaction point. It also has a secondary purpose of tracking low-energy particles that are otherwise missed by the outer components. Physically, it consists of 340 silicon microstrip detectors with a total of 150000 readout channels.

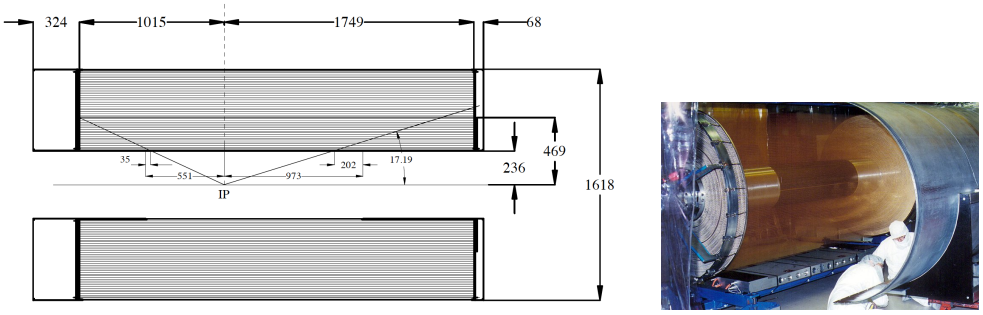


Figure 3.4: Drift chamber.

The SVT is surrounded by a cylindrical chamber filled with helium-isobutane, the DCH. Its primary purpose is to measure the momenta of charged particles passing through the detector, along with the dE/dx information. In addition, it also provides triggering information to the Level 1 trigger system.

3.2.2 DIRC

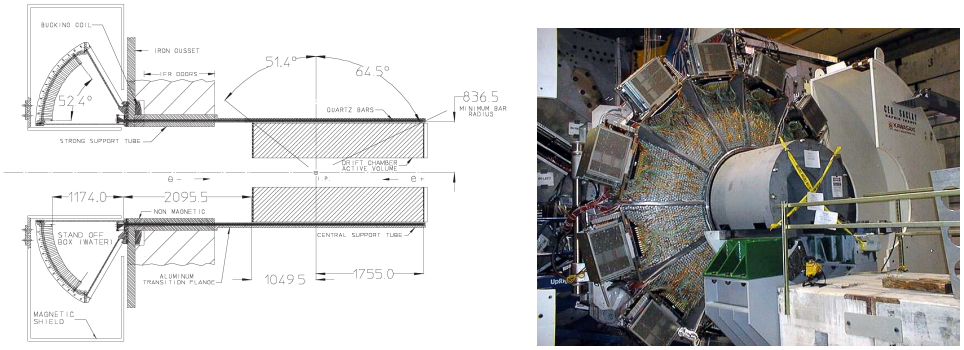


Figure 3.5: The DIRC.

The DIRC is designed to provide excellent identification of kaons, and secondarily that of muons. The ability to distinguish kaons from pions greatly increases the precision of many analyses performed using the *BABAR* data. Physically, the DIRC uses fused synthetic silica as the scintillator with photomultiplier tubes sensing the Cherenkov light.

3.2.3 Electromagnetic calorimeter

The primary purpose of the electromagnetic calorimeter is to detect, with excellent energy and angular resolution, the various charged and neutral particles that result from the decays of B mesons. The particles interact with the thallium-doped cesium iodide (CsI(Tl)) crystals, producing photoelectrons that are detected by photodiodes.

The readout system requires careful calibration at regular interval throughout its lifetime due to the degradation of the crystals and the hardware from high exposure to radiation. This was carried out using a neutron generator producing 6.1 MeV photons.

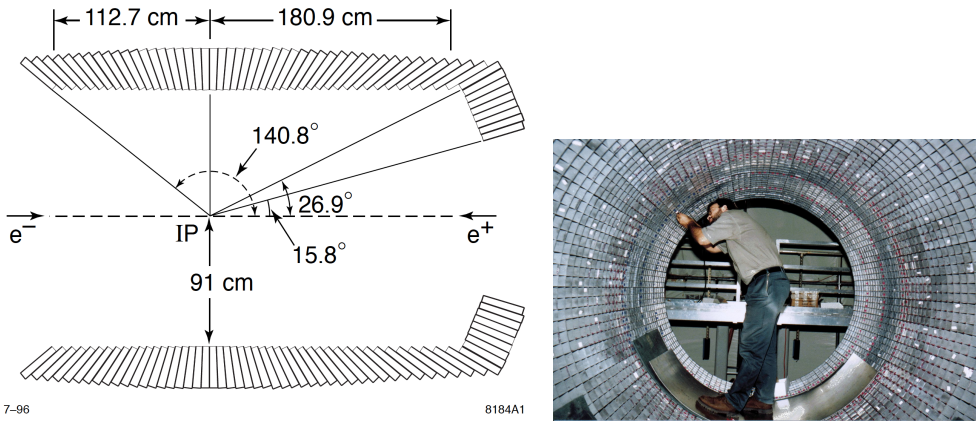


Figure 3.6: The EMC.

3.2.4 Superconducting coil

Figure 3.7: The superconducting coil.

The above subsystems, SVT, DCH, DIRC, and EMC, together define the inner detector which is in a 1.5T magnetic field produced by a superconducting solenoid. The magnetic field enables identification and quantification of energies and momenta for charged particles.

3.2.5 Instrumented flux return

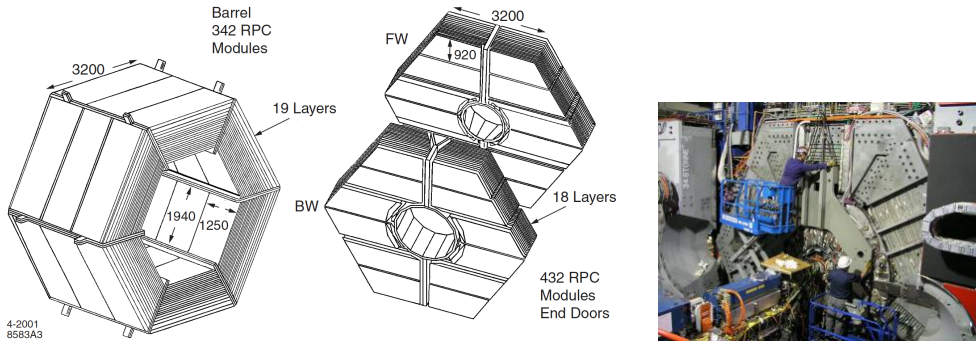


Figure 3.8: The IFR.

The outer detector, IFR, is designed to identify muons and aid the EMC in neutral hadron identification. It is a clever re-use of the iron yoke used for the superconducting

coil by attaching resistive plate counters (RPC) to the iron plates. The RPCs in the barrel section of the IFR were replaced by limited streamer tubes in 2004 due to the degradation of the RPC efficiency.

3.3 Trigger system

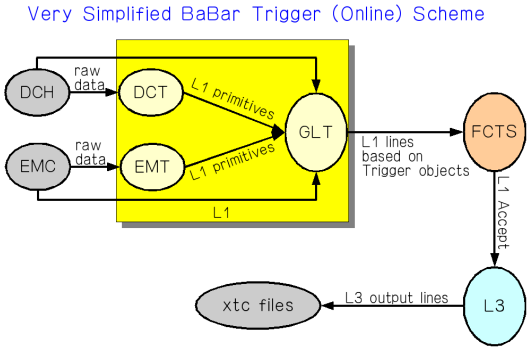


Figure 3.9: The superconducting coil.

The trigger system at *BABAR* consists of two levels: Level 1 (L1) and Level 3 (L3). The L1 trigger is a hardware trigger built into the front-end electronics of the detector.

The design goal of the L1 trigger is to remove as much beam background as possible by enforcing various energy and track/cluster requirements at a rate of 2 kHz with only an 11 μs delay. In comparison, the L3 trigger is a software trigger designed for maximum sensitivity towards physics-relevant events.

Chapter 4

Analysis Strategy

4.1 Overview

In this chapter, we present the analysis strategy for measuring $\mathcal{R}(D^{(*)})$, and how the picture of viewing this as a counting experiment becomes complex. Much of the of the statistical derivations and the descriptions of design choices is taken from [9], which was the Ph.D. thesis of an analyst of this project.

Let us restate the basic idea of the analysis: since we are interested in the probability of a B decaying into $D^{(*)}\tau_h\overline{\nu}$, it would seem sufficient to simply count the fraction of B's in our sample that decays in our mode of interest *conditioned* on the probability that these B's make into our data sample. In essence, this is the analysis strategy: First we need the *proportion* of the decays of interest and second we need the *efficiency* of these decays making into our dataset in the first place. Let us proceed by examining the key tasks that we must perform. To simplify the picture, let us consider the case when our data is just a single event corresponding to a single $\Upsilon(4S)$ and subsequently a single $B\overline{B}$ pair.

The first task is to identify whether these two B mesons decay into our modes of interest. At this point, the data simply consists of tracks and clusters for which we have only educated guesses of their true identities. Furthermore, heavy particles such as the B only live for a very short time and thus leave behind very short tracks that could be smaller than the resolution of our detector. It turns out that we essentially only have information about the *final state particles*, which are:

$$e^{\pm}, \mu^{\pm}, \pi^{\pm}, K^{\pm}, K_L, p, n, \gamma, \text{ and } \nu.$$

What makes this even more complicated is the fact that particle identification algorithms can misidentify particle species, and tracks can be missed by the pattern recognition algorithm. Nonetheless, given what we have, we can attempt to *reconstruct* the decay that produced the tracks we see. This again is a probabilistic process that can, and often does, have multiple solutions; there are many decays that could explain a single event. This is a key challenge in the analysis: how do we pick the best reconstruction candidate that is most likely to be the truth?

Suppose we are able to pick the best candidate. If the dataset was indeed just a single event, then we would be done. However, our actual goal is given a large dataset of many B decays, extract the *best estimates* of $\mathcal{R}(D^{(*)})$. The way we will accomplish this is by projecting every event into a space that maximizes the difference between the *signal* and *background* events. In this space, we can learn the *probability densities* of the events which can be used to extract the proportion of signal events from the overall distribution of data.

The secret weapon that enables this strategy is the *simulated* data. The simulated data is generated using the Monte Carlo method and is commonly referred to as simply the MC. The inputs to the simulation include the detector specifications and the physics parameters that govern the decay probabilities. The form factors described in Chapter 2 are examples of

physics parameters. The advantage of simulated data is that we actually know the truth that generated the data we see in our fictional detector. This means that we can, for instance, train a supervised classifier that picks the best candidate for us. Due to the fact that the simulated data has known deficiencies, we also attempt to use new learning algorithms from the field of computer vision called unsupervised domain adaptation. While they are not used in the extraction of the final result, the description of our exploration can be found in Appendix A.

Simulated data also gives us a way to avoid experimentalist's bias, or data snooping. If we work directly with the detector data when constructing the analysis strategy, we inevitably tweak some parts of the analysis based on what we see in the data.

However, the MC has its own disadvantages. They stem from the fact that no simulation can ever be perfect. We quantify the apparent differences between the simulation and detector data in the form of *systematic uncertainties*.

4.2 Event types and estimation of $\mathcal{R}(D^{(*)})$

Measuring $\mathcal{R}(D^{(*)})$ requires the measurement of four different branching fractions: $\mathcal{B}(B \to \overline{D}^{(*)}\ell\nu_\ell)$ and $\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)$. Since $\mathcal{B}(B \to \overline{D}^{(*)}\ell\nu_\ell)$ are well known, this analysis focuses on measuring $\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)$. This is a decision made by the analysts, and other measurements have measured all four with the advantage of possible cancellation of some systematic uncertainties.

It is clear that we actually have two types of signal events based on whether there is a D or a D^*. Furthermore, it turns out to be a good idea to partition the background events into three separate categories, which allows us to better probe the deficiencies of their simulation.

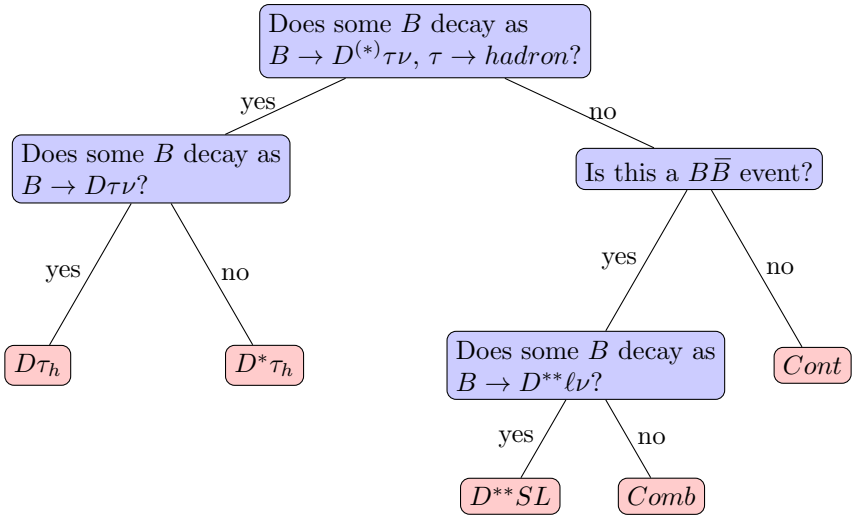


Figure 4.1: The leaves of this decision tree define a partitioning of *BABAR* events.

The logic of event type categorization is that of a decision tree as shown in Figure 4.1.

Consider the probability space of all possible outcomes of a collision event in the *BABAR* detector. Let E_i for $i \in C = \{D\tau_h, D^*\tau_h, D^{**}SL, Comb, Cont\}$ denote partition of the probability space, meaning that any observed event must fall under one of these categories. $D^{**}SL$ denotes events with semileptonic decays involving a D^{**}, $Comb$ denotes the combinatoric $B\overline{B}$ events, and $Cont$ denotes the events from the continuum (i.e. $e^+e^- \to q\overline{q}$ rather than $e^+e^- \to \Upsilon(4S)$).

Let

$$\begin{aligned} P &= \mathbb{P}\left[E_{D\tau_h}|E_{B\overline{B}}\right], \\ Q &= \mathbb{P}\left[E_{D^*\tau_h}|E_{B\overline{B}}\right], \\ p &= \mathcal{B}(B \to \overline{D}\tau\nu) \times \mathcal{B}(\tau \to \text{hadrons}), \\ q &= \mathcal{B}(B \to \overline{D}^*\tau\nu) \times \mathcal{B}(\tau \to \text{hadrons}), \end{aligned} \tag{4.1}$$

where $E_{B\overline{B}}$ refers to union of all event types except *cont*. Now, we use the above to derive the following relationship:

$$\begin{aligned} P &= p^2 + 2p(1-p) \\ \implies p &= 1 - \sqrt{1-P}, \end{aligned} \tag{4.2}$$

hence

$$\boxed{\mathcal{B}(B \to \overline{D}\tau\nu) = \frac{1-\sqrt{1-P}}{\mathcal{B}(\tau \to \text{hadrons})}}. \tag{4.3}$$

Similarly,

$$\begin{aligned} Q &= q^2 + 2q(1-p-q) \\ \implies q &= (1-p)\left(1 - \sqrt{1 - \frac{Q}{(1-p)^2}}\right), \end{aligned} \tag{4.4}$$

gives us

$$\boxed{\mathcal{B}(B \to \overline{D}^*\tau\nu) = \frac{1-p}{\mathcal{B}(\tau \to \text{hadrons})}\left(1 - \sqrt{1 - \frac{Q}{(1-p)^2}}\right)}. \tag{4.5}$$

(4.3) and (4.5) tell us that the problem of estimating $\mathcal{R}(D^{(*)})$ reduces to estimating $\mathbb{P}\left[E_{D^{(*)}\tau_h}|E_{B\overline{B}}\right]$ since all other parameters can be taken from the literature [14, 2]:

$$\begin{aligned} \mathcal{B}(B \to D\ell\nu_\ell) &= \mathbb{P}\left[B \to D\ell\nu_\ell | B = B^\pm\right]\mathbb{P}\left[B = B^\pm\right] \\ &\quad + \mathbb{P}\left[B \to D\ell\nu_\ell | B = B^0\right]\mathbb{P}\left[B = B^0\right] \\ &= 0.487 \times \mathcal{B}(B^0 \to \overline{D}^0\ell\nu_\ell) + 0.513 \times \mathcal{B}(B^+ \to D^+\ell\nu_\ell) \\ &= (2.22 \pm 0.10)\%, \end{aligned} \tag{4.6}$$

$$\begin{aligned} \mathcal{B}(B \to D^*\ell\nu_\ell) &= \mathbb{P}\left[B \to D^*\ell\nu_\ell | B = B^\pm\right]\mathbb{P}\left[B = B^\pm\right] \\ &\quad + \mathbb{P}\left[B \to D^*\ell\nu_\ell | B = B^0\right]\mathbb{P}\left[B = B^0\right] \\ &= 0.487 \times \mathcal{B}(B^0 \to \overline{D}^{*0}\ell\nu_\ell) + 0.513 \times \mathcal{B}(B^+ \to D^{*+}\ell\nu_\ell) \\ &= (5.13 \pm 0.11)\%, \end{aligned} \tag{4.7}$$

$$\begin{aligned} \mathcal{B}(\tau \to \text{hadrons}) &= 1 - \mathcal{B}(\tau \to e\bar{\nu}_e\nu_\tau) - \mathcal{B}(\tau \to \mu\bar{\nu}_\mu\nu_\tau) \\ &\quad - \mathcal{B}(\tau \to e\bar{\nu}_e\nu_\tau\gamma) - \mathcal{B}(\tau \to \mu\bar{\nu}_\mu\nu_\tau\gamma) \\ &= (63 \pm 0.18)\%, \end{aligned} \tag{4.8}$$

where we have used the following values:

$$\begin{aligned}
\mathcal{B}(B^0 \to \overline{D}^0 \ell \nu_\ell) &= (2.13 \pm 0.095)\% \\
\mathcal{B}(B^+ \to D^+ \ell \nu_\ell) &= (2.30 \pm 0.10)\% \\
\mathcal{B}(B^0 \to \overline{D}^{*0} \ell \nu_\ell) &= (4.93 \pm 0.11)\% \\
\mathcal{B}(B^+ \to D^{*+} \ell \nu_\ell) &= (5.31 \pm 0.12)\% \\
\mathcal{B}(\tau^- \to e \bar{\nu}_e \nu_\tau) &= (17.83 \pm 0.04)\% \\
\mathcal{B}(\tau^- \to e \bar{\nu}_e \nu_\tau \gamma) &= (1.75 \pm 0.18)\% \\
\mathcal{B}(\tau^- \to \mu \bar{\nu}_\mu \nu_\tau) &= (17.41 \pm 0.04)\% \\
\mathcal{B}(\tau^- \to \mu \bar{\nu}_\mu \nu_\tau) &= (0.0036 \pm 0.0004)\%.
\end{aligned} \tag{4.9}$$

4.3 Estimating $\mathbb{P}\left[E_{D^{(*)}\tau_h} | E_{B\overline{B}}\right]$

In order to discuss the probability of observing a certain event type in the dataset, we need to account for the process that reduces all events collected at *BABAR* to our dataset. This filtering is necessary due to the large amount of irrelevant events that exist in the raw data along with the need for computational tractability. Let $\mathcal{F}$ be such filter and U be the set of outcomes that pass the filter.

We can then write the efficiency and sample proportions as

$$\epsilon_i \equiv \mathbb{P}\left[U_i | E_i\right] = \frac{|U_i|}{|E_i|}, \tag{4.10}$$

$$p_i \equiv \mathbb{P}\left[U_i | U\right] = \frac{|U_i|}{|U|}, \tag{4.11}$$

which gives

$$\mathbb{P}\left[E_{sig} | E_{B\overline{B}}\right] = \frac{|E_{sig}|}{|E_{B\overline{B}}|} = \frac{p_{sig}|U|}{\epsilon_{sig}|E_{B\overline{B}}|}, \tag{4.12}$$

where *sig* refers to $D\tau_h, D^*\tau_h$.

Each variable in (4.12) can be estimated as follows: $\hat{p}_{sig}$ is the proportion of the signal event types in our dataset, $\hat{\epsilon}_{sig}$ is the proportion of signal events that passes $\mathcal{F}$, $|U|$ is simply the number of events in our dataset N, and $|E_{B\overline{B}}|$ is the total number of events *before* the filter $N_{B\overline{B}}$.

This gives the following estimate of $\mathbb{P}\left[E_{D^*\tau_h} | E_{B\overline{B}}\right]$:

$$\boxed{\hat{\mathbb{P}}\left[E_{D^*\tau_h} | E_{B\overline{B}}\right] = \frac{\hat{p}_{D^{(*)}\tau_h} N}{\hat{\epsilon}_{D^{(*)}\tau_h} N_{B\overline{B}}}}. \tag{4.13}$$

The only remaining obstacles are how to construct the optimal filter $\mathcal{F}$ and how to best estimate $\hat{p}_{D^{(*)}\tau_h}$ and $\hat{\epsilon}_{D^{(*)}\tau_h}$, which will be the focus of the rest of the chapter.

4.4 Data filtering

In high energy physics experiments, it is not unusual to see the same experiment reporting multiple measurements of the same quantity. These measurements are considered to be independent for the purpose of calculating the world average. This independence relies entirely on the construction of $\mathcal{F}$ for each measurement that ensures the datasets do not overlap.

The distinction of this analysis from the previous *BABAR* measurement [3] is the method of *tagging* used (semileptonic vs. hadronic) and the reconstruction method of the τ (τ_h vs. τ_ℓ).

The tagging, or B tagging, method is a clever tool used to study B physics. The idea is that since the $\Upsilon(4S)$ produces $B\overline{B}$ pairs, we first attempt to reconstruct a single B_{tag}, called the tagging B, and assign all other tracks in the event for the purpose of reconstructing the signal B_{sig}, in this case $B \to \overline{D}^{(*)}\tau\nu_\tau$. Thus semileptonic tagging refers to reconstructing B_{tag} by looking for semileptonic decays of B.

The decay modes that we use to reconstruct the τ on the signal side are hadronic decays of the τ, as opposed to the leptonic decays of the τ.

The exact modes used in the reconstruction will be discussed in Chapter 5.

4.5 Estimating $\hat{p}_{sig}$

Consider the probability density of our dataset in some variable space z, i.e. $f(z)$. We can think of f as being made up of its five component densities. More formally,

$$\begin{aligned} f(z) &= \sum_{j \in C} \mathbb{P}[U_j] f(z|U_j) \\ &= \sum_{j \in C} p_j f_j(z), \end{aligned} \tag{4.14}$$

where $f_j(z) = f(z|U_j)$.

Suppose we know the conditional, or component, densities $f_j(z)$. Then, we can proceed to extract the maximum likelihood estimator of $\hat{p}_{sig}$ by solving the following optimization problem:

$$\begin{aligned} \underset{p \in \mathbb{R}^{|C|}}{\text{minimize}} \quad & -\sum_{i=1}^{N} \log\left(\sum_{j \in C} p_j f_j(z_i)\right) \\ \text{subject to} \quad & \sum_{j \in C} p_j = 1. \end{aligned} \tag{4.15}$$

Note that there is no non-negativity constraints on the p_j, meaning the proportions are allowed to be negative.

4.5.1 Feature engineering and density estimation

The dataset can be thought of as a $n \times m$ matrix X, corresponding to n data points with m features describing each point. A probability density of such dataset would be an m-dimensional density. Beyond the concerns of curse of dimensionality[1] and computational efficiency, there is a statistical reason for trying to keep m as small as we can [9], pushing us to learn a mapping $\mathbb{R}^m \to \mathbb{R}^{|z|}$. It turns out that by seeking z that best satisfies the following heuristics, we *minimize* $\text{Var}[\hat{p}_{sig}]$:

1. Low dimensionality.
2. Continuous.
3. Large difference between signal and background densities.

This analysis will proceed by using a 2-dimensional $z = (z_1, z_2)$:

- $z_1 = s_1(X)$, where s_1 is a regression function trained to separate signal events from background events.
- $z_2 = s_2(X)$, where s_2 is a regression function trained to separate $D\tau_h$ and $D^*\tau_h$.

[1] Curse of dimensionality refers to the effect where the number of data points required to train a model with sufficient power increases as the dimensionality of the feature set increases. Thus given a fixed size data sample, more features are not necessarily better.

We can then proceed to learn the 2-dimensional densities $f(z)$ and $f_j(z)$ using kernel density estimation.

It should be noted that training s_1 and s_2 by construction requires the use of simulated data as the training set. In fact, the conditional densities $f_j(z)$ are also learned using the MC, and the only place we use the detector data is to learn $f(z)$ when performing the maximum likelihood estimation.

4.6 Estimating $\hat{\epsilon}_{sig}$

Estimating the signal efficiencies is much more straightforward.

We have

$$\begin{aligned}\epsilon_i &= \mathbb{P}\left[U_i|E_i\right] = \frac{\mathbb{P}\left[U_i \cap E_i\right]}{\mathbb{P}\left[E_i\right]} = \frac{\mathbb{P}\left[U_i\right]}{\mathbb{P}\left[E_i\right]} \\ &= \frac{\mathbb{P}\left[U_i|E_{B\bar{B}}\right]\mathbb{P}\left[E_{B\bar{B}}\right] + \mathbb{P}\left[U_i|E^c_{B\bar{B}}\right]\mathbb{P}\left[E^c_{B\bar{B}}\right]}{\mathbb{P}\left[E_i\right]} = \frac{\mathbb{P}\left[U_i|E_{B\bar{B}}\right]\mathbb{P}\left[E_{B\bar{B}}\right]}{\mathbb{P}\left[E_i\right]} \end{aligned} \tag{4.16}$$

and

$$\mathbb{P}\left[E_i|E_{B\bar{B}}\right] = \frac{\mathbb{P}\left[E_i \cap E_{B\bar{B}}\right]}{\mathbb{P}\left[E_{B\bar{B}}\right]} = \frac{\mathbb{P}\left[E_i\right]}{\mathbb{P}\left[E_{B\bar{B}}\right]} \tag{4.17}$$

$$\Rightarrow \mathbb{P}\left[E_i\right] = \mathbb{P}\left[E_i|E_{B\bar{B}}\right]\mathbb{P}\left[E_{B\bar{B}}\right]. \tag{4.18}$$

Combining the two results, we get

$$\epsilon_i = \frac{\mathbb{P}\left[U_i|E_{B\bar{B}}\right]}{\mathbb{P}\left[E_i|E_{B\bar{B}}\right]} \qquad \text{for } i = D\tau_h,\, D^*\tau_h. \tag{4.19}$$

As with the density estimation, (4.19) cannot be estimated using detector data as we do not know their true event types. Thus we estimate the signal efficiencies using simulation and quantify their degree of disagreement with the detector data in the form of systematic uncertainties.

Chapter 5

Data Collection

In this chapter, we delve into the details of the criteria of the data filter $\mathcal{F}$, which includes the reconstruction criteria. We also tabulate the resulting simulated and detector datasets.

In the following sections, the MC and the detector data are treated on equal footing with regards to reconstruction and the term *data* will be used for both with no ambiguity.

5.1 Data filtering

5.1.1 Event pre-screen

The first part of the filter is a set of broad selection criteria; broad in the sense that only the events that we can very confidently classify as background will not pass. These criteria are collectively referred to as event pre-screening, and are:

- Size of `ChargedTracks` ≤ 14: Number of charged particle candidates must be no greater than 14.
- Size of `GoodPhotonsLoose` ≤ 10: Number of photon candidates must be no greater than 10.
- $-2 \leq Q_{total} \leq 2$: Total charge of the event must be between -2 and 2.
- Apply tag filter `BGFMultiHadron`: Require at least 3 charged tracks in the event and require the second Fox-Wolfram moment of the event $R_2 < 0.98$.
- Apply tag filter `TagL3`: Require L3 trigger in the DCH and the EMC.

The terms in the teletype font are jargon specific to the *BABAR* analysis framework.

5.1.2 Event reconstruction

The events that pass the pre-screen are used to reconstruct the B_{tag} and B_{sig}. Recall that initially each event is simply a set of tracks of final state particles. The idea is to recursively group these tracks until we have a possible candidate for the $\Upsilon(4S)$.

The most suitable data structure to represent the reconstruction is a directed acyclic graph (DAG) with multiple roots. The roots are the $\Upsilon(4S)$ candidates, the leaves are the final state particles, and the intermediate nodes represent the particle hypotheses. Indeed, the terms candidates and hypotheses are the most appropriate, as these are abstractions of the *possible* explanations of the events. Note that we do not gauge which candidate is the most likely within an event, only its validity. The edges of the graph indicate mother-daughter relations such that for nodes u, v, a directed edge (u, v) exists if v is a decay product of u.

In order to construct particle hypotheses, we choose a set of valid decay modes far smaller than the set of all possible decay modes. The primary condition for choosing the reconstruction modes is the branching fraction; it is a balance between covering as many modes as possible while keeping the size of the reconstruction graph manageable. The secondary condition is the cleanliness of the modes. For example, the mode $D^0 \to K^-\pi^+$ is a must-include in our set of decay modes: the small number of daughter particles and their ease of identification satisfy the cleanliness condition, on top of being the dominant D decay mode. On the other hand, decays such as $D^0 \to K^-\pi^+\pi^-\pi^+\pi^0$ are not included due to its high multiplicity of pions.

Within the *BABAR* framework, the reconstruction can be defined by explicitly writing down the modes or using pre-built *lists*.

Starting from the final state particles, the decay modes used in reconstruction are:

1. Final state particle lists.
   - Electrons: `eCombinedSuperLoose` — Electron candidates that pass at least one of the three electron PID selector algorithms that make up `eCombinedSuperLoose`.
   - Muons: `muCombinedVeryLooseFakeRate` — Muon candidates that pass at least one of the three muon PID selector algorithms that make up the list. "Fake rate" refers to the optimization of the algorithms to lower misidentification of pions as muons at the cost of lower efficiency.
   - Pions: `ChargedTracks` and `GoodTracksVeryLoose` — Charged pions candidates that satisfy the (loose) kinematic criteria to be a "good" track such as being close to the beam center.
   - Kaons: `KCombinedSuperLoose` — Kaon candidates that pass at least one of the four Kaon PID selector algorithms that make up the list.
   - Photons: `GoodPhotonLoose` with $0.01 \leq Lat \leq 0.6$ — Photon candidates from the EMC hits with minimum energy of 0.03 GeV and lateral moment between 0.01 and 0.6. Lateral moment is a ratio quantifying the spread of the hits in the EMC.
2. Light meson lists.
   - π^0: `pi0AllDefault` and `pi0SoftDefaultMass` — Using the photon list above, reconstruct $\pi^0 \to \gamma\gamma$ with the mass of the pion candidate constrained to be between $[0.115, 0.15]$ GeV.
   - $\rho^+ \to \pi^+\pi^0$: The π^+ is refined such that it cannot be a member `eKMTight` or `muBDTVeryTight`. It is also required to have mass between $[0.45, 1.09]$ GeV and a χ^2 of at least 0.001.
   - K_S: `KsTight` — Reconstruct $K_S \to \pi^+\pi^-$ with χ^2 of at least 0.001.
3. D meson lists. All reconstructed D mesons without π^0 daughters are required to have masses within 0.06 GeV of the PDG value. For those with π^0 daughters, the masses must be within 0.1 GeV of the PDG value.
   - $D^+ \to K^-\pi^+\pi^+$.
   - $D^0 \to K^-\pi^+$.
   - $D^0 \to K^-\pi^+\pi^0$.
   - $D^0 \to K^-\pi^+\pi^-\pi^+$.
4. D^* meson lists. The soft pions used are required to have a center of mass 3-momentum magnitude of at most 400 MeV. Charged pions are refined from `GoodTracksVeryLoose` while neutral pions are refined from `pi0SoftDefaultMass`. Soft photons are refined from `GoodPhotonLoose`, but are required to have $Lat \leq 0.8$ and center of mass energy of at least 100 MeV and 3-momentum magnitude of at most 400 MeV.

- $D^{*0} \to D^0\pi^0$. Require Δm to be within $[0.135, 0.175]$ GeV.
- $D^{*0} \to D^0\gamma$. Require Δm to be within $[0.13, 0.155]$ GeV.
- $D^{*+} \to D^0\pi^+$. Require Δm to be within $[0.135, 0.165]$ GeV.
- $D^{*+} \to D^+\pi^0$. Require Δm to be within $[0.13, 0.15]$ GeV.

5. B_{tag} meson list. Masses are required to be at most 5.2791 GeV, and the χ^2 must be at least 0.001. Furthermore, $\cos\theta_{BY}$ must be between $[-5, 1.0]$.
   - $B^+ \to \overline{D}^0 e^+$.
   - $B^+ \to \overline{D}^0 \mu^+$.
   - $B^0 \to D^- e^+$.
   - $B^0 \to D^- \mu^+$.
   - $B^+ \to \overline{D}^{*0} e^+$.
   - $B^+ \to \overline{D}^{*0} \mu^+$.
   - $B^0 \to D^{*-} e^+$.
   - $B^0 \to D^{*-} \mu^+$.
6. B_{sig} meson list. Masses are required to be at most 5.2791 GeV, and the χ^2 must be at least 0.001.
   - $B^+ \to \overline{D}^0 \pi^+$.
   - $B^+ \to \overline{D}^0 \rho^+$.
   - $B^0 \to D^- \pi^+$.
   - $B^0 \to D^- \rho^+$.
   - $B^+ \to \overline{D}^{*0} \pi^+$.
   - $B^+ \to \overline{D}^{*0} \rho^+$.
   - $B^0 \to D^{*-} \pi^+$.
   - $B^0 \to D^{*-} \rho^+$.
7. $\Upsilon(4S) \to B_{tag} B_{sig}$. The B daughters must conserve charge, and B's are allowed to mix. The daughters cannot have overlapping final states.

Figure 5.1 shows an example of the reconstruction graph. In the case that the graph is empty, the event is simply discarded.

5.2 Data collection

The *BABAR* experiment collected 433 fb^{-1}, or 471 million $B\overline{B}$ pairs, at the $\Upsilon(4S)$ resonance over 6 runs from 1999 to 2008. The integrated luminosity over time can be seen in Figure 5.2.

For simulated events, the total number of generated events exceeds that of the detector data by a factor of 2 or 3 depending on the run.

5.2.1 Detector data

Tables 5.1 and 5.2 show the names of the processed data samples that are used to perform this analysis. The off-peak data is collected just below the $\Upsilon(4S)$ resonance by reducing the energy of the electron beam from 9 GeV to 8.9 GeV as discussed in Chapter 3.

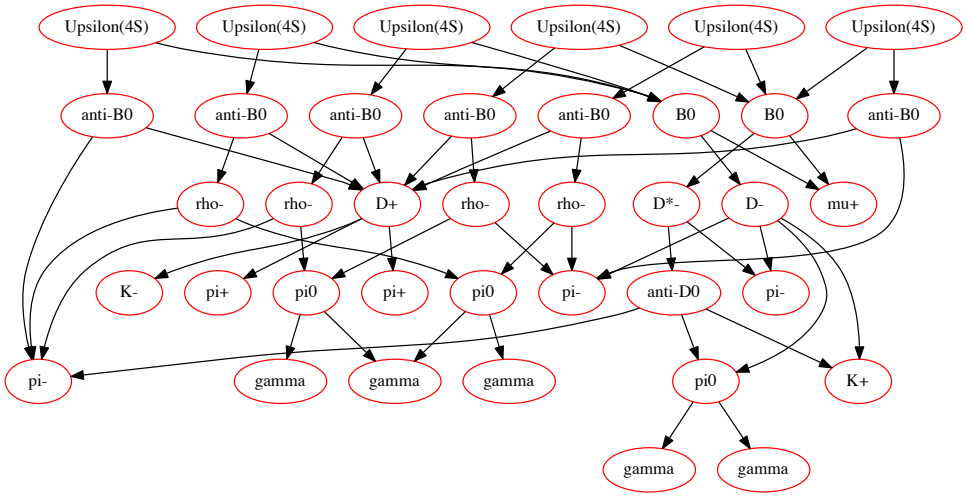


Figure 5.1: Example of event reconstruction.

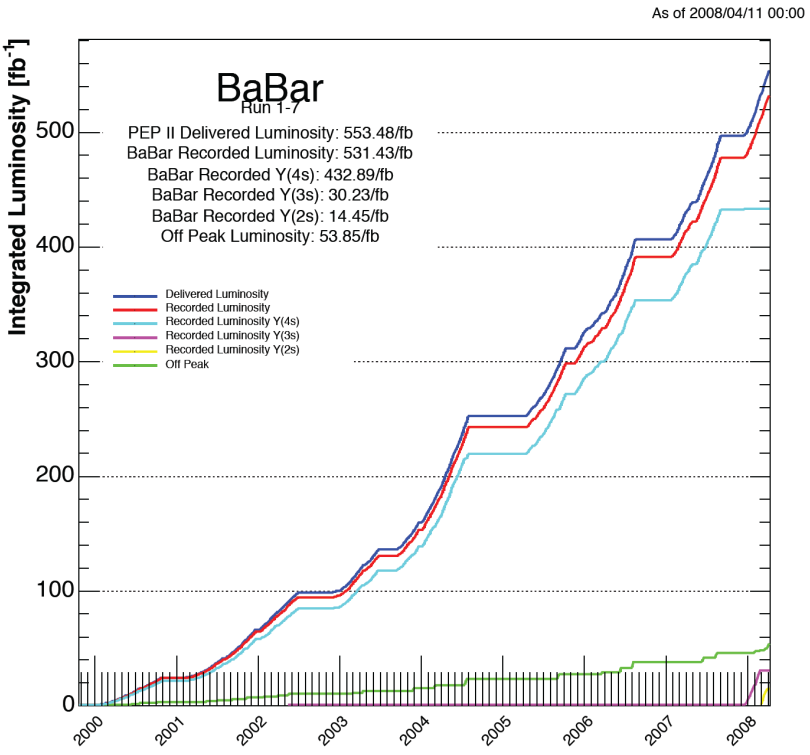


Figure 5.2: Integrated luminosity recorded at the *BABAR* experiment through its lifetime.

5.2.2 Simulated data

The simulated data consists of two types: generic MC and signal MC.

The generic MC seeks to represent the data collected in the detector by faithfully generating the possible results of the e^+e^- collisions. The generic MC itself consists of two

Collider Dataset Name	Luminosity (pb^{-1})	$N_{B\overline{B}}(10^4)$
AllEventsSkim-Run1-OnPeak-R24a1	20372 ± 91	2256 ± 14
AllEventsSkim-Run2-OnPeak-R24a1	61300 ± 257	6844 ± 41
AllEventsSkim-Run3-OnPeak-R24a1	32300 ± 132	3576 ± 22
AllEventsSkim-Run4-OnPeak-R24a1	99600 ± 418	11143 ± 67
AllEventsSkim-Run5-OnPeak-R24a1	132400 ± 582	14762 ± 89
AllEventsSkim-Run6-OnPeak-R24a1	78300 ± 352	8519 ± 51
Total	424300 ± 854	47100 ± 132

Table 5.1: Detector on-peak datasets. $N_{B\overline{B}}$ is the estimated number of $B\overline{B}$ pairs contained in the specified dataset; it includes both neutral and charged B pairs.

Collider Dataset Name	Luminosity (pb^{-1})	Weight
AllEventsSkim-Run1-OffPeak-R24a1	2564.0 ± 12.1	7.95
AllEventsSkim-Run2-OffPeak-R24a1	6869.1 ± 30.2	8.93
AllEventsSkim-Run3-OffPeak-R24a1	2443.6 ± 10.5	13.21
AllEventsSkim-Run4-OffPeak-R24a1	10016.0 ± 43.1	9.95
AllEventsSkim-Run5-OffPeak-R24a1	14276.8 ± 67.1	9.27
AllEventsSkim-Run6-OffPeak-R24a1	7752.6 ± 36.4	10.10
Total	43922.0 ± 94.1	9.66

Table 5.2: Detector off-peak datasets. Weight is the factor by which the size of the corresponding on-peak dataset exceeds that of the given off-peak dataset.

parts: $B\overline{B}$ and continuum. The generic $B\overline{B}$ MC, which includes both charged and neutral B pairs, emulates the case when the $\Upsilon(4S)$ is produced. On the other hand, the continuum MC represents $e^+e^- \to q\overline{q}$ events where $q \in \{u, d, s, c\}$.

In order to mix the various types of generic MC in the right proportions, the number of events must be weighted by its probability to occur in the detector. This is accomplished by taking the cross sections σ of each component into account through the following formula:

$$w_i = \mathcal{L}\frac{\sigma_i}{N_i}, \tag{5.1}$$

where the i represents the component, e.g. $B^0\overline{B}^0$, $\mathcal{L}$ is the integrated detector data luminosity, and N_i is the number of generated events for component i. The cross sections used in this analysis are shown in Table 5.3 and the components of the generic MC dataset are listed in Table 5.4.

The signal MC data is generated by forcing every $\Upsilon(4S)$ to decay into $B\overline{B}$ that subsequently decay into our signal modes. More specifically, it is the case in which one B, the tag B, decays semileptonically and the other into $D^{(*)}\tau_h\nu_\tau$. This dataset is crucial for training the supervised classifiers, as the number of signal events in the generic MC is small. The

SP Mode	Mode type	Cross section (pb)
1235	B^+B^-	525.0
1237	$B^0\overline{B}^0$	525.0
1005	$c\overline{c}$	1300.0
998	uds	2090.0

Table 5.3: Cross sections used to convert the sizes generic simulated data to the equivalent on-peak dataset.

Simulated Dataset Name	Mode Type	Collisions Generated	Weight
SP-1235-AllEventsSkim-Run1-R24a1	B^+B^-	34878000	0.306
SP-1235-AllEventsSkim-Run2-R24a1	B^+B^-	105561000	0.305
SP-1235-AllEventsSkim-Run3-R24a1	B^+B^-	56035000	0.303
SP-1235-AllEventsSkim-Run4-R24a1	B^+B^-	166784000	0.314
SP-1235-AllEventsSkim-Run5-R24a1	B^+B^-	215168000	0.323
SP-1235-AllEventsSkim-Run6-R24a1	B^+B^-	130336000	0.316
SP-1237-AllEventsSkim-Run1-R24a1	$B^0\overline{B}^0$	34941000	0.306
SP-1237-AllEventsSkim-Run2-R24a1	$B^0\overline{B}^0$	104188000	0.308
SP-1237-AllEventsSkim-Run3-R24a1	$B^0\overline{B}^0$	57888000	0.292
SP-1237-AllEventsSkim-Run4-R24a1	$B^0\overline{B}^0$	169801000	0.307
SP-1237-AllEventsSkim-Run5-R24a1	$B^0\overline{B}^0$	215953000	0.321
SP-1237-AllEventsSkim-Run6-R24a1	$B^0\overline{B}^0$	135224000	0.304
SP-1005-AllEventsSkim-Run1-R24a1	$c\overline{c}$	55254000	0.479
SP-1005-AllEventsSkim-Run2-R24a1	$c\overline{c}$	164722000	0.483
SP-1005-AllEventsSkim-Run3-R24a1	$c\overline{c}$	88321000	0.475
SP-1005-AllEventsSkim-Run4-R24a1	$c\overline{c}$	267308000	0.484
SP-1005-AllEventsSkim-Run5-R24a1	$c\overline{c}$	344275000	0.499
SP-1005-AllEventsSkim-Run6-R24a1	$c\overline{c}$	208664000	0.488
SP-998-AllEventsSkim-Run1-R24a1	uds	176404000	0.241
SP-998-AllEventsSkim-Run2-R24a1	uds	525504000	0.243
SP-998-AllEventsSkim-Run3-R24a1	uds	276381000	0.244
SP-998-AllEventsSkim-Run4-R24a1	uds	845899000	0.246
SP-998-AllEventsSkim-Run5-R24a1	uds	1110944000	0.249
SP-998-AllEventsSkim-Run6-R24a1	uds	655152000	0.250

Table 5.4: Generic simulated data. Weight is the factor by which the size of the corresponding on-peak dataset exceeds that of the given simulated dataset calculating using (5.1).

Simulated Dataset Name	Mode Type	Collisions Generated
SP-11444-Run1-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	694000
SP-11444-Run2-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	2044000
SP-11444-Run3-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	1152000
SP-11444-Run4-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	3347000
SP-11444-Run5-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	4546000
SP-11444-Run6-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	2732000
SP-11445-Run1-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	644000
SP-11445-Run2-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	1937000
SP-11445-Run3-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	955000
SP-11445-Run4-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	3207000
SP-11445-Run5-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	4627000
SP-11445-Run6-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	2349000
SP-11446-Run1-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	651000
SP-11446-Run2-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	1919000
SP-11446-Run3-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	1025000
SP-11446-Run4-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	3402000
SP-11446-Run5-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	4276000
SP-11446-Run6-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	2685000
SP-11447-Run1-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	750000
SP-11447-Run2-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	1702000
SP-11447-Run3-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	901000
SP-11447-Run4-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	3120000
SP-11447-Run5-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	4637000
SP-11447-Run6-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	2505000

Table 5.5: Simulated signal data.

components of the signal MC dataset are listed in Table 5.5. Since this dataset does not attempt to emulate the detector data, we do not calculate any weights.

Chapter 6

Candidate Selection

In Chapter 4, we briefly mentioned the idea of viewing our dataset as a matrix X where the columns represent the features. The features can be of two types: event level and candidate level. Event level features are those that apply to the event as a whole and are shared by all candidates of the event, whereas candidate level features are unique to each candidate.

The features of each event will be used to train the two regressors s_1 and s_2. It is entirely feasible to feed them the event level features and the candidate level features of all candidates for each event. While this method would take advantage of the possible information provided by many candidates, it has significant disadvantages. The key concept is the curse of dimensionality previously discussed; the number of training data required for a sufficiently powerful regressor increases along with the dimensionality of the dataset. Furthermore, it is likely that the features among the candidates within an event show large correlation, which means that we would not be introducing any new information. For these reasons, we perform a best candidate selection where we discard all candidate level features except that of the best candidate.[1]

6.1 Truth matching

The criteria for a candidate being the best one is clear: if the reconstruction graph of a candidate matches exactly that of the truth, that is our best candidate. Thus the algorithm should seek to choose the candidate that is the most likely to *match* the truth, which is available for us in MC. In short, the binary label of whether a candidate is truth matched is the training data input to the best candidate selector.

The truth matching process is described in detail in [9]. The setting of the problem is as follows: we have two rooted DAG's, the reconstructed graph and the truth graph. The two graphs match if one is *subgraph isomorphic* to the other. In other words, if there exists a bijective mapping between the vertices and edges of the two graphs.

6.1.1 Software tools

The graphs are stored as adjacency lists in a PostgreSQL instance. The workflow is as follows:

1. Download data from database as a CSV file.
2. For each candidate in every event, perform truth matching. Label as 0 if not truth matched, 1 if truth matched.
3. Upload truth matching result into a new table in the database.

[1] Best candidate selection can be thought of as a dimensionality reduction process.

The truth matching software is written in C++ using the Boost Graph Library [22] that runs in linear $\mathcal{O}(V + E)$ time, where V is the number of vertices of the graph and E is the number of edges of the graph.

6.1.2 Results

Figures 6.1a and 6.1b show example input graphs for truth matching. The MC truth graphs are pruned from the original graphs by removal of disconnected components, decay products of final state particles, and particle irrelevant to the truth matching process (i.e. neutrinos). Figure 6.1c shows the output of the truth matching process, which was able to find a candidate that was truth matched.

6.2 Best candidate selection

The best candidate selector is a random forest classifier [23] that outputs the probability of a given candidate being truth matched. We use the scikit-learn implementation [24].

6.2.1 Training sample

Since we are the most interested in selecting the correct candidate for the signal events, we use the signal MC candidates labeled by the truth matched outcomes as the training set. In other words, we are optimizing for the recall of our selection. This is a choice made by the analysts, and one can argue for using a classifier trained to choose the best candidate regardless of its event type.

The training sample consists of 600k signal MC events, each represented by the following set of features:

`mmiss2prime`, `eextra`, `tag_lp3`, `tag_cosby`, `tag_costhetadl`, `tag_dmass`, `tag_deltam`, `tag_costhetadsoft`, `tag_softp3magcm`, `sig_hp3`, `sig_cosby`, `sig_costhetadtau`, `sig_vtxb`, `sig_dmass`, `sig_deltam`, `sig_costhetadsoft`, `sig_softp3magcm`, `sig_hmass`, `sig_vtxh`, `tag_isbdstar`, `sig_isbdstar`, `tag_dmode`, `tag_dstarmode`, `sig_dmode`, `sig_dstarmode`.

The descriptions of each feature can be found in Chapter 7.

Figure 6.2 shows the distribution of each feature for the two classes.

6.2.2 Model selection

We first standardize the data by centering and scaling the numerical features. The categorical features are one-hot encoded. The parameters of the random forest classifier are grid searched to minimize the five-fold cross validation error.

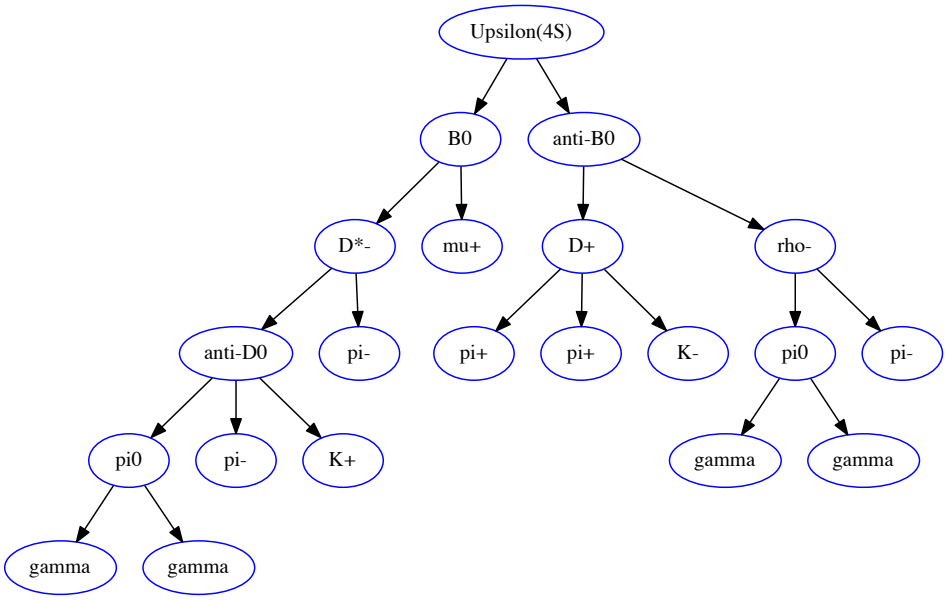
The best performing hyperparameters are the default settings with the number of trees, `n_estimators`, set to 1000.

6.2.3 Results

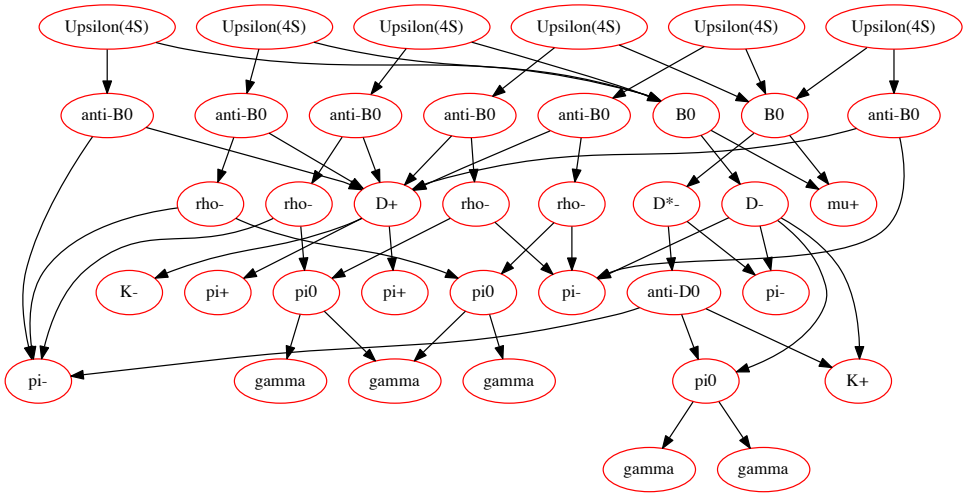
To quantify the performance of the best candidate selector, we define the following recall metric: out of the total possible truth matchable candidates, how many does our classifier predict as truth matched?

To establish a baseline, we use the conventional method of best candidate selection used in *BABAR* analyses: choose the candidate with the minimum value of E_{extra}. Lower E_{extra} translates to how well the reconstruction was performed and is a good indicator of the quality of a candidate.

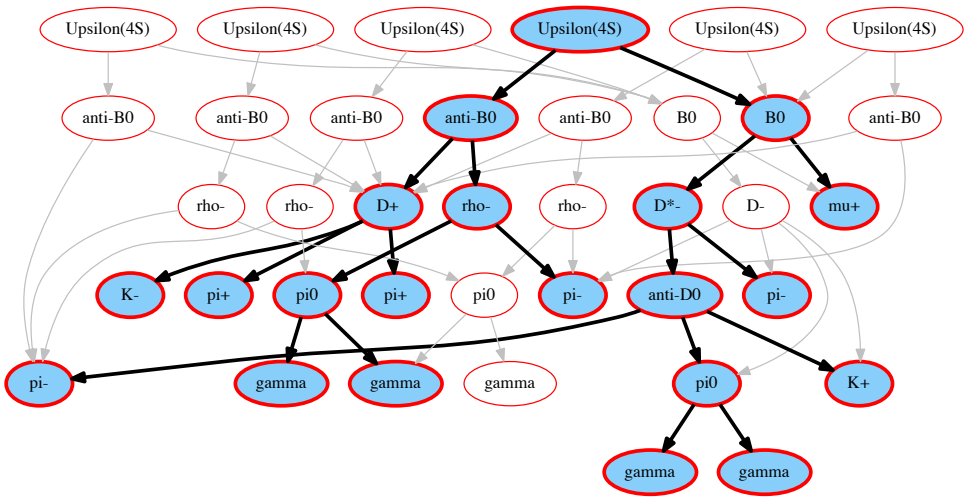
Table 6.1 shows the results for the baseline selector and the random forest selector and demonstrates the superiority of the supervised classifier.



(a) Generated decay graph with irrelevant particles removed.



(b) The corresponding reconstruction graph.

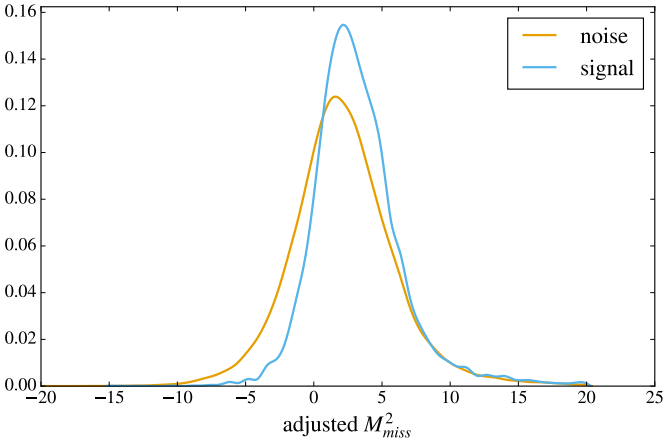


(c) Result of truth matching process.

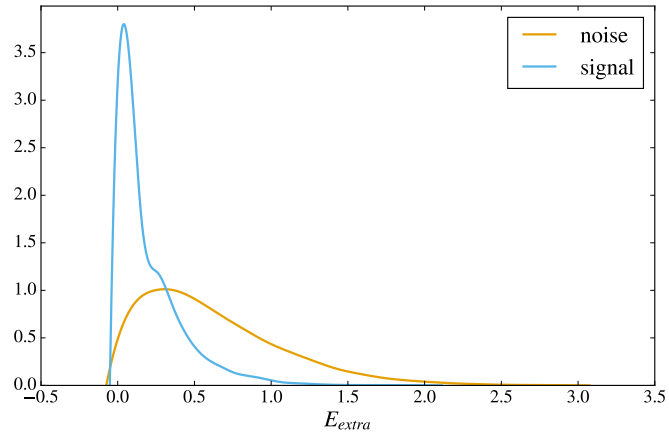
Figure 6.1: Truth matching example (cont.).

	minimum E_{extra}	maximum candidate score
total possible	6643	6643
number chosen	4597	5916
recall	0.69	0.89

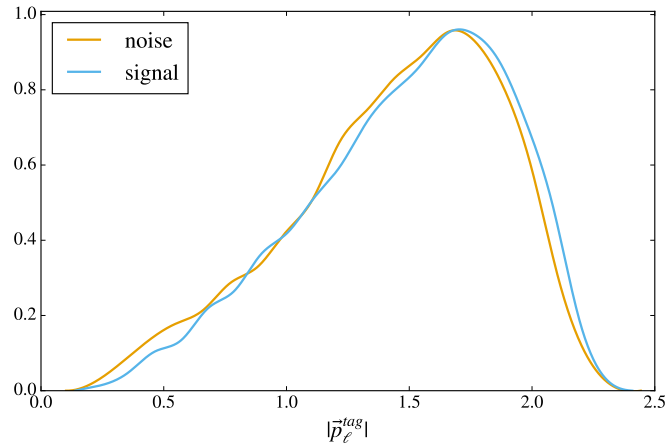
Table 6.1: "Total possible" is the number of collision events that have a matched candidate that can be chosen. "Number chosen" is the number of those collision events that the specified criteria was able to correctly identify as the matched candidate. "Recall" is simply the ratio between the rows above.



(a) Adjusted M^2_{miss}

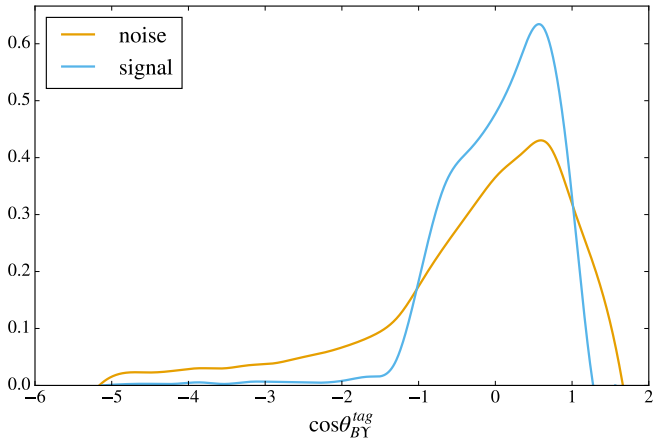


(b) E_{extra}

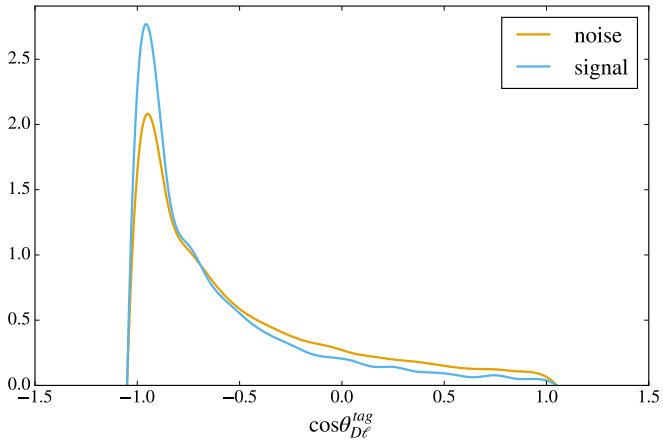


(c) $|\vec{p}^{\,tag}_{\ell}|$

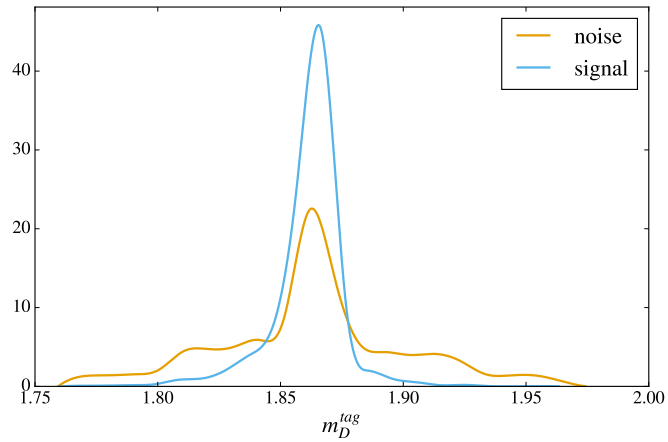
Figure 6.2: Density functions of features used in best candidate selection.



(d) $\cos\theta^{tag}_{B\Upsilon}$

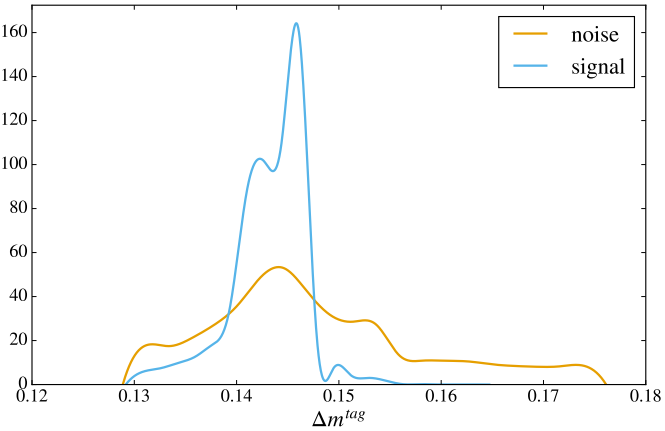


(e) $\cos\theta^{tag}_{D\ell}$

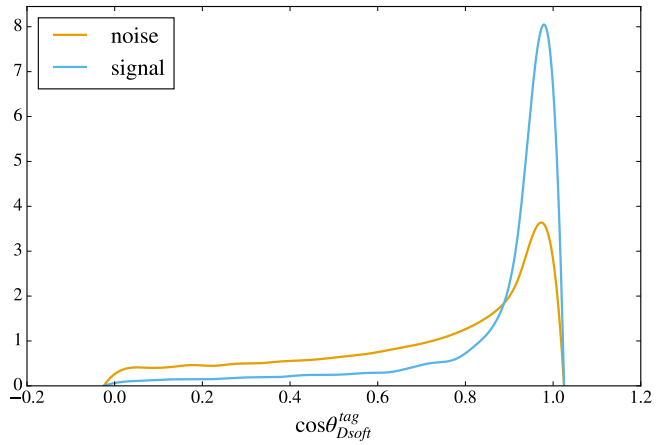


(f) m^{tag}_{D}

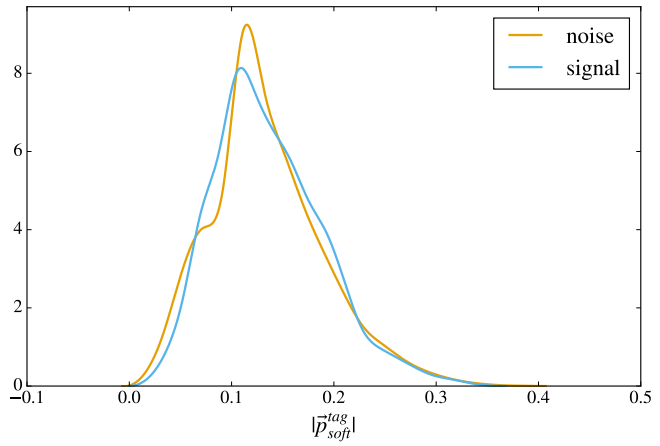
Figure 6.2: Density functions of features used in best candidate selection (cont.).



(g) Δm^{tag}

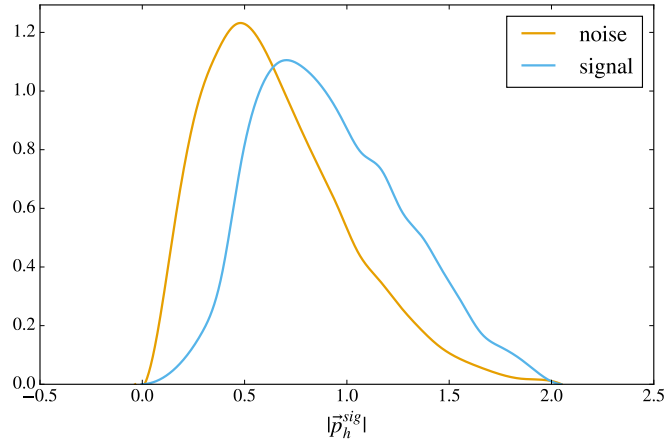


(h) $\cos\theta^{tag}_{Dsoft}$

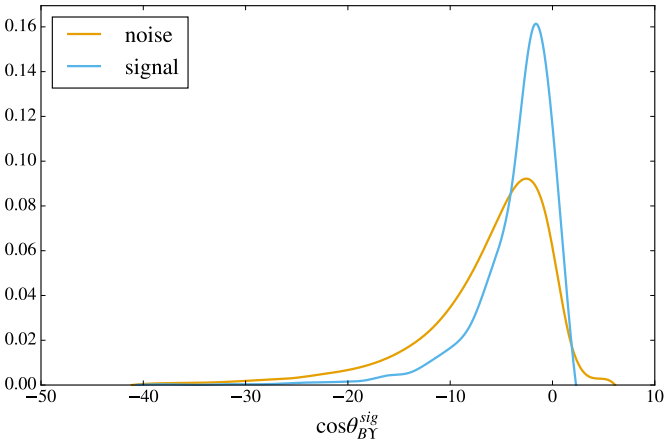


(i) $|\vec{p}^{tag}_{soft}|$

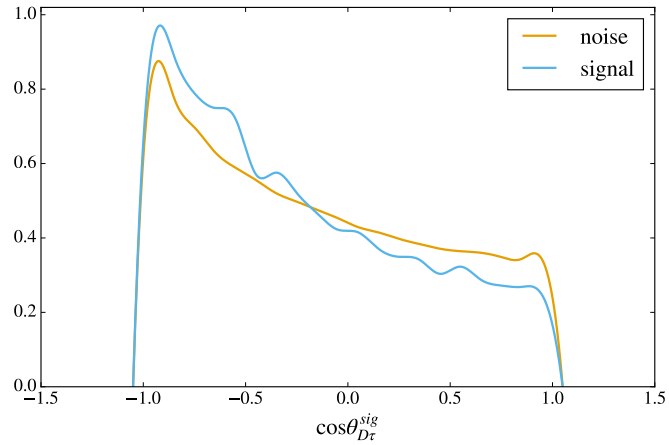
Figure 6.2: Density functions of features used in best candidate selection (cont.).



(j) $|\vec{p}_h^{sig}|$

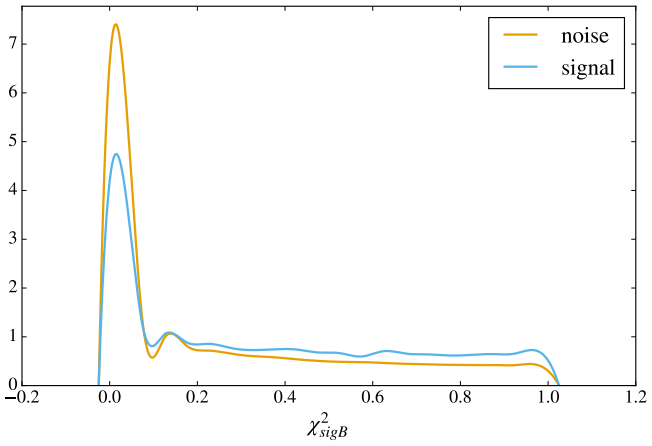


(k) $\cos\theta_{BY}^{sig}$

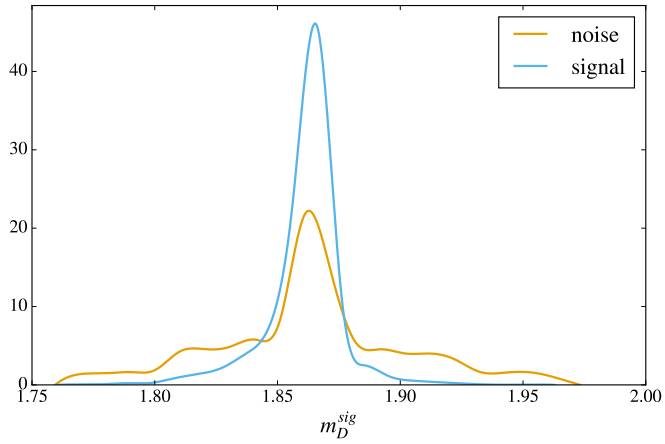


(l) $\cos\theta_{D\tau}^{sig}$

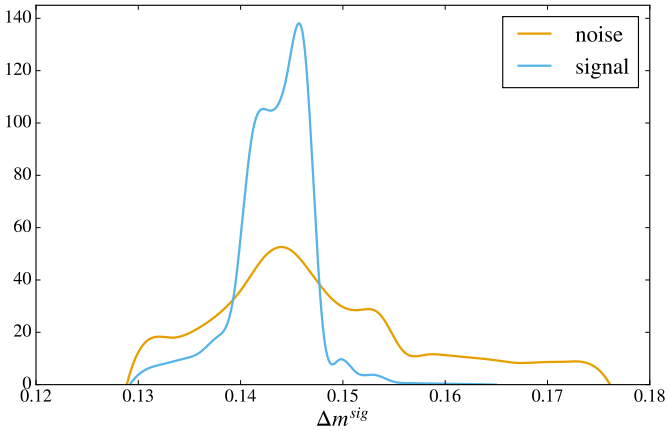
Figure 6.2: Density functions of features used in best candidate selection (cont.).



(m) χ^2_{sigB}

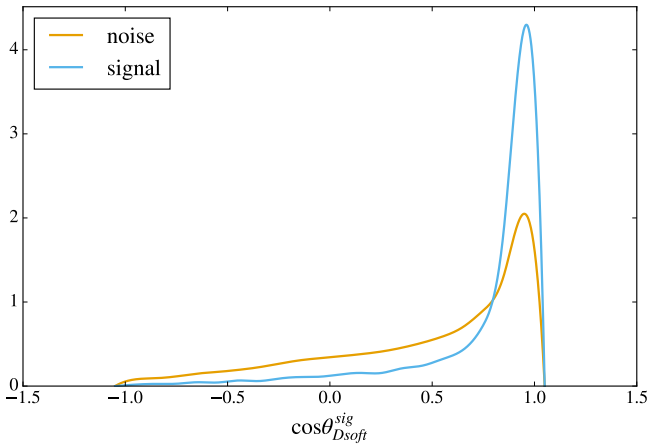


(n) m_D^{sig}

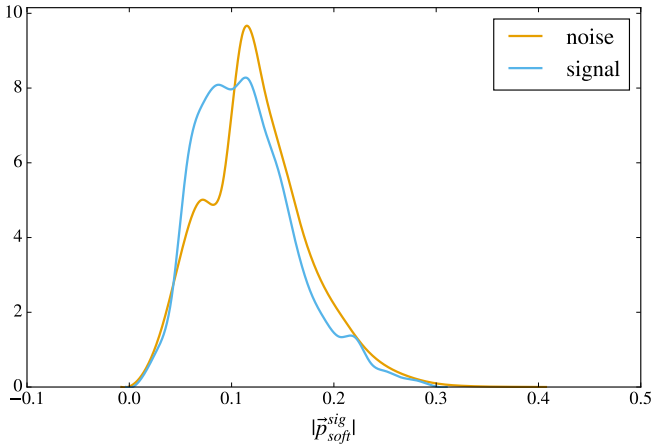


(o) Δm^{sig}

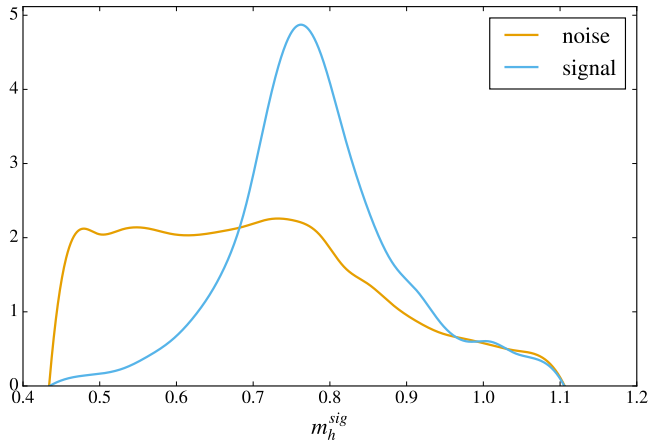
Figure 6.2: Density functions of features used in best candidate selection (cont.).



(p) $\cos\theta^{sig}_{Dsoft}$

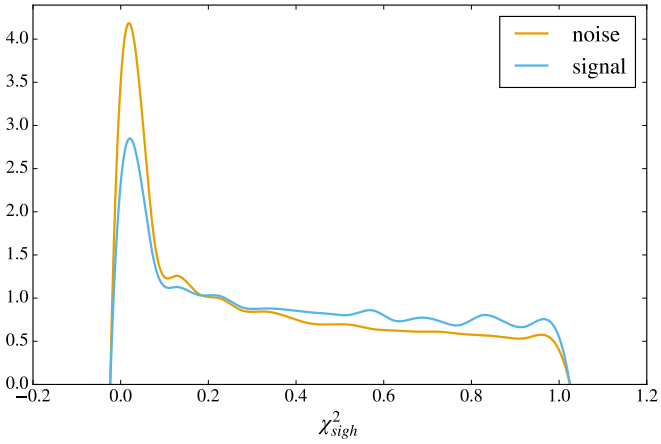


(q) $|\vec{p}^{sig}_{soft}|$

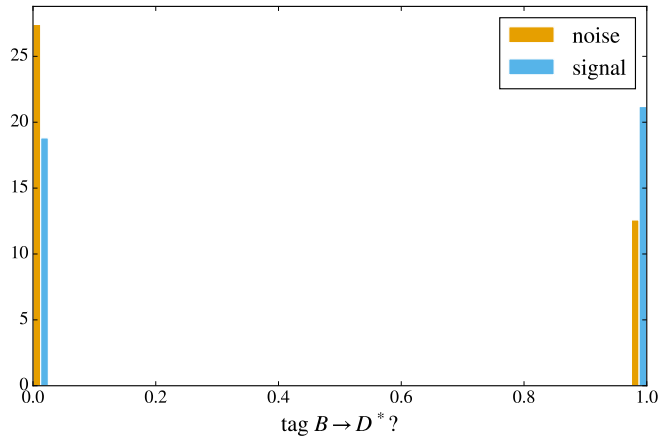


(r) m^{sig}_{h}

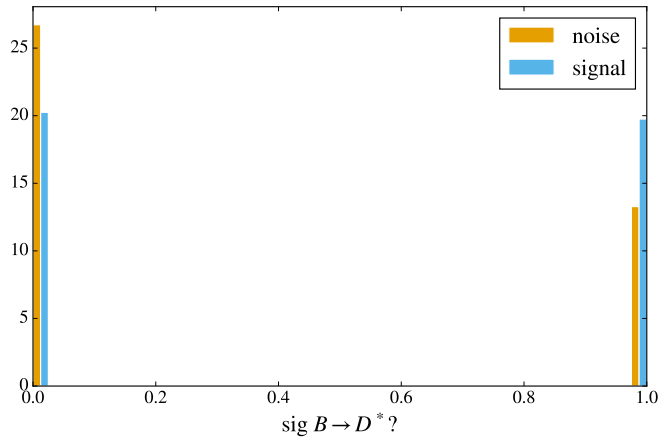
Figure 6.2: Density functions of features used in best candidate selection (cont.).



(s) χ^2_{sigh}

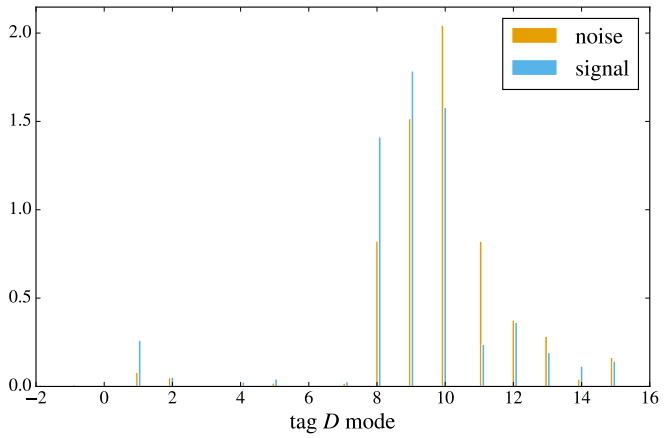


(t) tag $B \to D^*$?

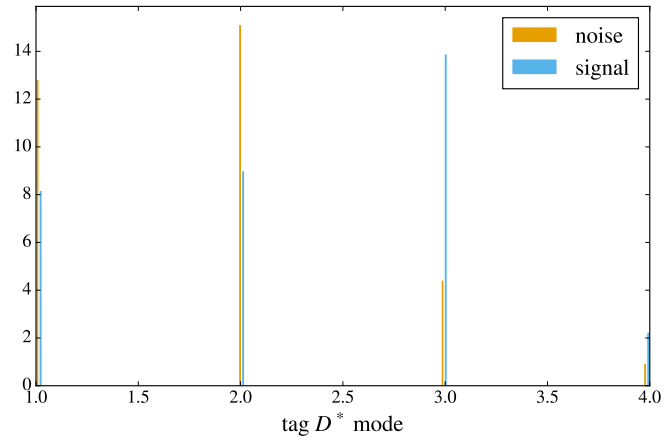


(u) sig $B \to D^*$?

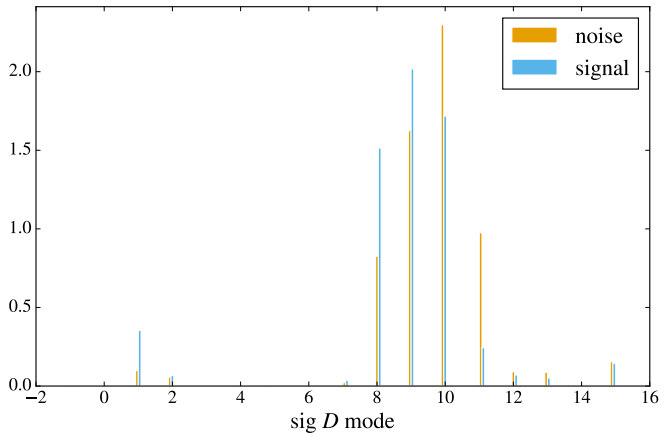
Figure 6.2: Density functions of features used in best candidate selection (cont.).



(v) tag D mode

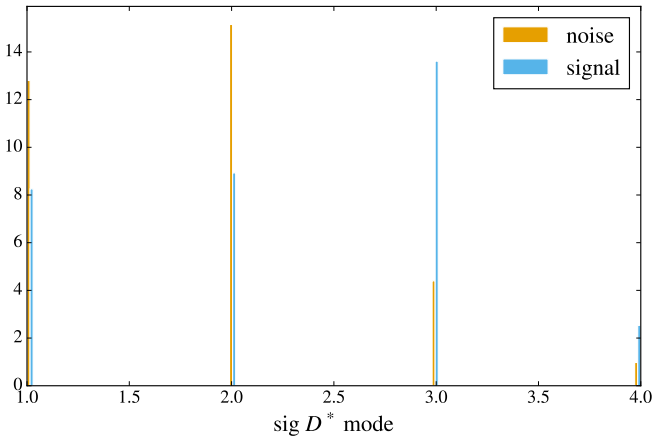


(w) tag D^* mode



(x) sig D mode

Figure 6.2: Density functions of features used in best candidate selection (cont.).



(y) sig D^* mode

Figure 6.2: Density functions of features used in best candidate selection (cont.).

Chapter 7

Feature Exploration

In this chapter we describe the features that make up the design matrix X. The candidate level features will now consist of those of the best candidate, which is determined by the candidate with the highest score predicted by the best candidate selector.

By visualizing our dataset in projections to each feature dimension, we can get a sense of how well the simulation agrees with the physical intuition for each event type. For example, we expect the M^2_{miss} distributions of the signal events to be centered at higher values than those of the background events due to the neutrinos.

In addition, we also explore 1% of the detector data that has been reserved for this purpose. While the statistics of this sample is low, we can qualitatively compare MC and detector data to detect any *significant* mismodeling with the understanding that some deviations are expected.

7.1 Feature description

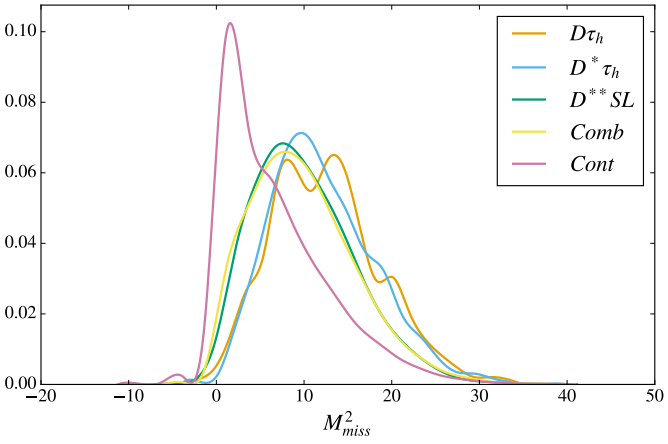
The feature names and their descriptions are as follows:

- $n\Upsilon$ (`ny`): Number of $\Upsilon(4S)$ candidates.
- N_{track} (`ntracks`): Number of `GoodTracksVeryLoose`.
- R_2 All (`r2all`): Second Fox-Wolfram moment.
- Candidate Score (`cand_score`): Candidate score from chapter 6.
- M^2_{miss} (`mmiss2`): Missing mass squared.
- Adjusted M^2_{miss} (`mmiss2prime`): Adjusted missing mass squared.
- E_{extra} (`eextra`): Leftover photon energy.
- $\cos\theta_T$ (`costhetat`): Cosine of the thrust angle between the B_{tag} and the rest of the event.
- $|\vec{p}^{\,tag}_{\ell}|$ (`tag_lp3`): 3-momentum magnitude of the B_{tag}'s lepton.
- $\cos\theta^{tag}_{BY}$ (`tag_cosby`): Cosine of the angle between the 3-momentum of the B_{tag} and the 3-momentum sum of its D and lepton daughters.
- $\cos\theta^{tag}_{D\ell}$ (`tag_costhetadl`): Cosine of the angle between the 3-momenta of the D and the lepton daughter of the B_{tag}.
- m^{tag}_D (`tag_dmass`): Mass of the B_{tag}'s D meson daughter.

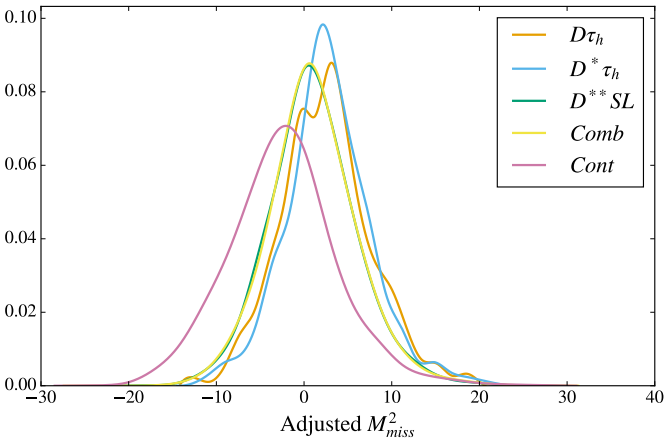
- Δm^{tag} (`tag_deltam`): Δm of the B_{tag}'s D^* meson daughter if it exists.
- $\cos\theta^{tag}_{Dsoft}$ (`tag_costhetadsoft`): Cosine of the angle between the D^* mesons' daughters.
- $|\vec{p}^{tag}_{soft}|$ (`tag_softp3magcm`): 3-momentum magnitude of the D^*'s soft daughter.
- $|\vec{p}^{sig}_{h}|$ (`sig_hp3`): 3-momentum magnitude of the B_{sig}'s hadron daughter.
- $\cos\theta^{sig}_{BY}$ (`sig_cosby`): Cosine of the angle between the 3-momentum of the B_{sig} and the 3-momentum sum of its D and lepton daughters.
- $\cos\theta^{sig}_{D\tau}$ (`sig_costhetadtau`): Cosine of the angle between the 3-momentum of the B_{sig} and the 3-momentum sum of its D and hadron daughters.
- χ^2_{sigB} (`sig_vtxb`): χ^2 of the B_{sig}'s vertex fit.
- m^{sig}_{D} (`sig_dmass`): Mass of the B_{sig}'s D meson daughter.
- Δm^{sig} (`sig_deltam`): Δm of the B_{sig}'s D^* meson daughter if it exists.
- $\cos\theta^{sig}_{Dsoft}$ (`sig_costhetadsoft`): Cosine of the angle between the D^* mesons' daughters.
- $|\vec{p}^{sig}_{soft}|$ (`sig_softp3magcm`): 3-momentum magnitude of the D^*'s soft daughter.
- m^{sig}_{h} (`sig_hmass`): Mass of the B_{sig}'s hadron daughter, if it exists.
- χ^2_{sigh} (`sig_vtxh`): χ^2 of the B_{sig}'s composite hadron daughter, if it exists.
- tag D mode (`tag_dmode`): The mode in which the B_{tag}'s D daughter is reconstructed.
- tag D^* mode (`tag_dstarmode`): The mode in which the B_{tag}'s D^* daughter is reconstructed.
- sig D mode (`sig_dmode`): The mode in which the B_{sig}'s D daughter is reconstructed.
- sig D^* mode (`sig_dstarmode`): The mode in which the B_{sig}'s D^* daughter is reconstructed.
- tag ℓ electron PID (`tag_l_epid`): B_{tag}'s lepton daughter's electron PID level.
- tag ℓ muon PID (`tag_l_mupid`): B_{tag}'s lepton daughter's muon PID level.
- sig h electron PID (`sig_h_epid`): B_{sig}'s hadron daughter's electron PID level.
- sig h muon PID, (`sig_h_mupid`): B_{sig}'s hadron daughter's muon PID level.
- Is $B_{tag} \to D^*$? (`tag_isbdstar`): Flag to indicate whether B_{tag} is reconstructed as a semileptonic D^* decay.
- Is $B_{sig} \to D^*$? (`sig_isbdstar`): Flag to indicate whether B_{sig} is reconstructed as a semileptonic D^* decay.

7.2 Event type densities of features

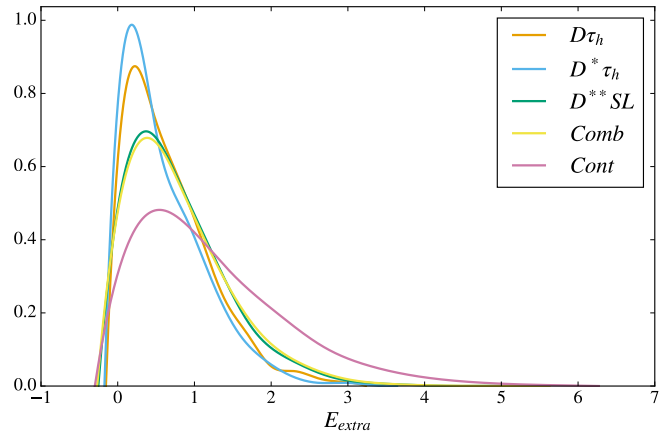
Figure 7.1 shows the densities of each event type for all features used in this analysis. We do not observe any features that exhibit characteristics that contrast with our physical intuitions.



(a) M^2_{miss}.

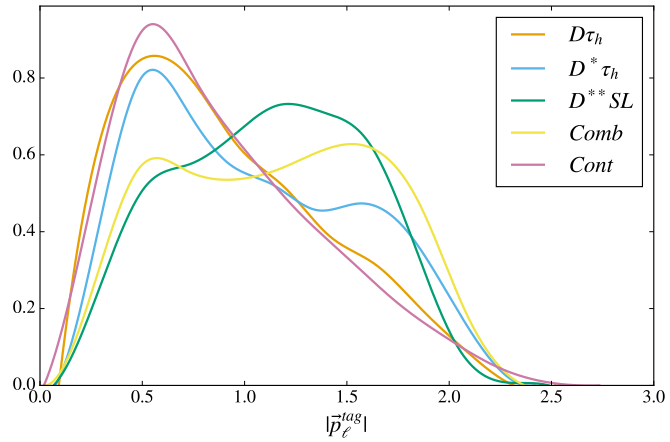


(b) Adjusted M^2_{miss}.

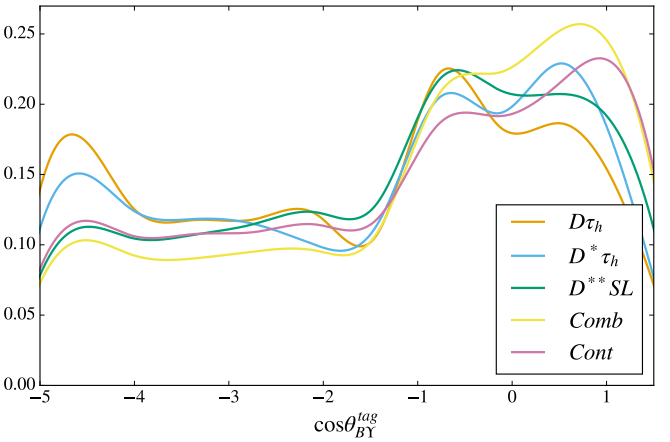


(c) E_{extra}.

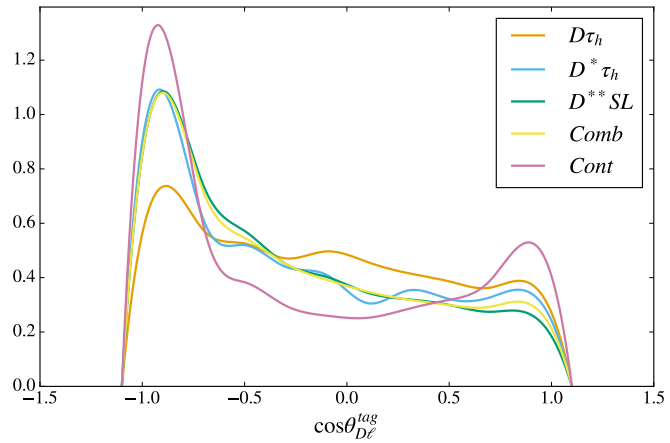
Figure 7.1: Feature density functions for each event type.



(d) $|\vec{p}_\ell^{tag}|$.

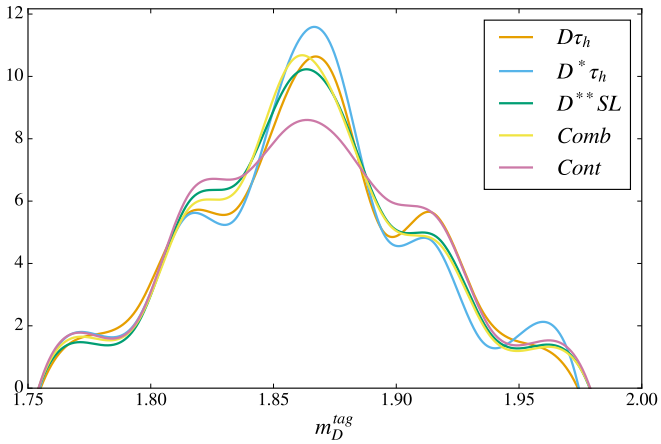


(e) $\cos\theta_{BY}^{tag}$.

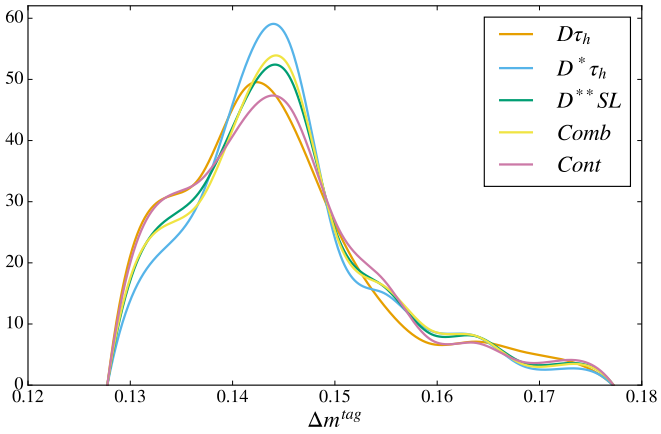


(f) $\cos\theta_{D\ell}^{tag}$.

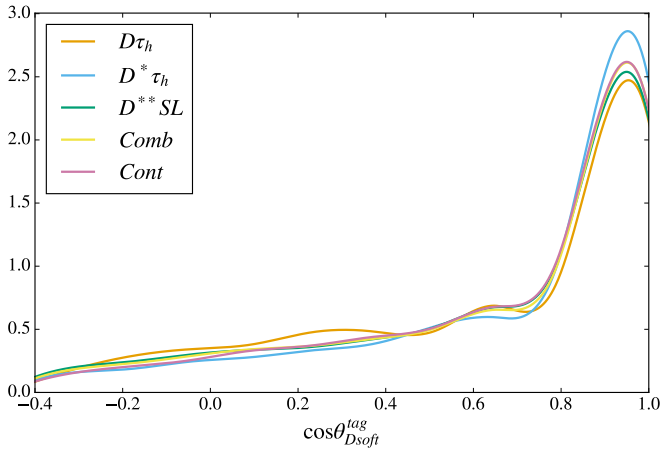
Figure 7.1: Feature density functions for each event type (cont.).



(g) m_D^{tag}

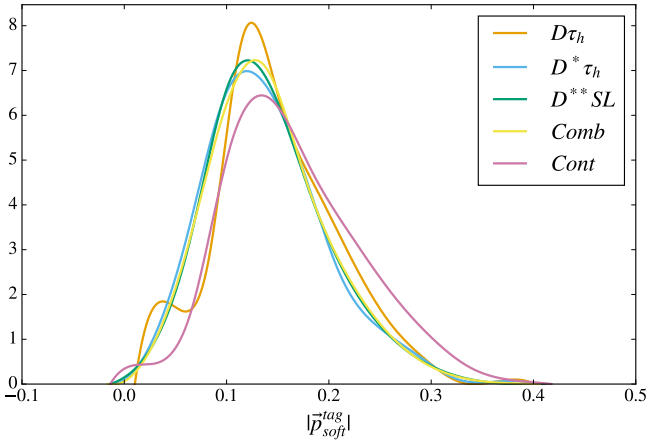


(h) Δm^{tag}.

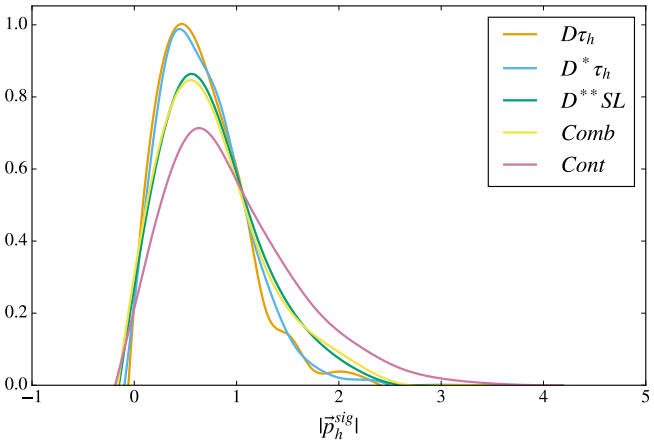


(i) $\cos\theta_{Dsoft}^{tag}$.

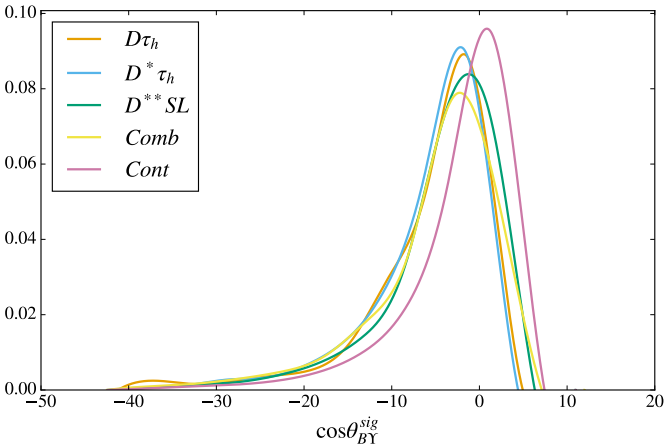
Figure 7.1: Feature density functions for each event type (cont.).



(j) $|\vec{p}^{tag}_{soft}|$.

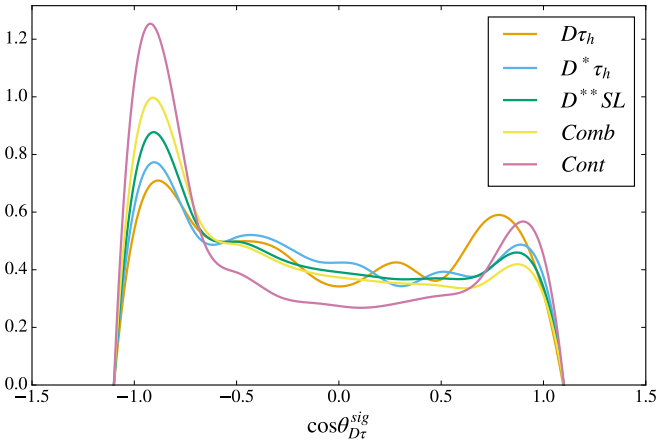


(k) $|\vec{p}^{sig}_{h}$.

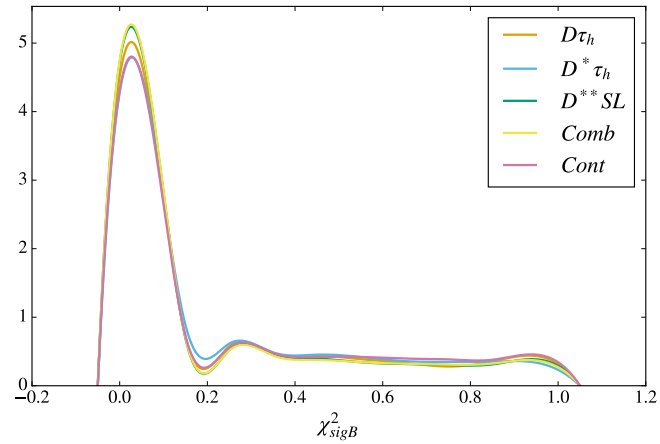


(l) $\cos\theta^{sig}_{BY}$.

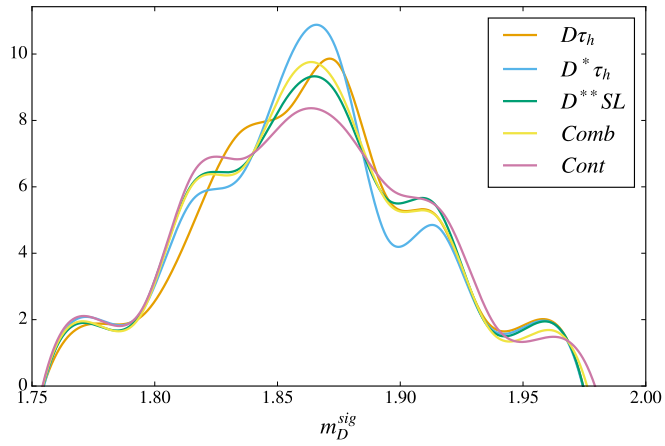
Figure 7.1: Feature density functions for each event type (cont.).



(m) $\cos\theta_{D\tau}^{sig}$.

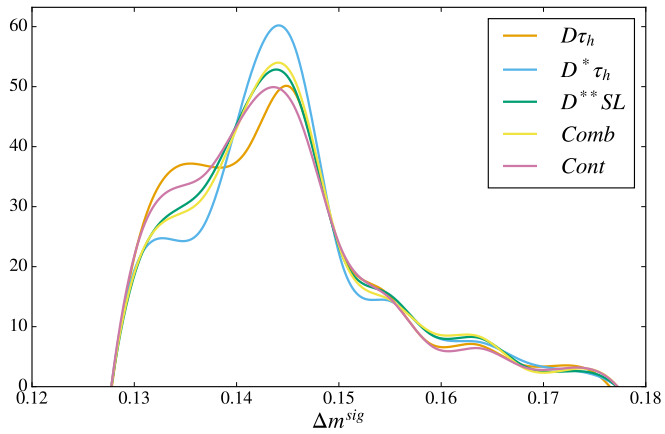


(n) χ^2_{sigB}.

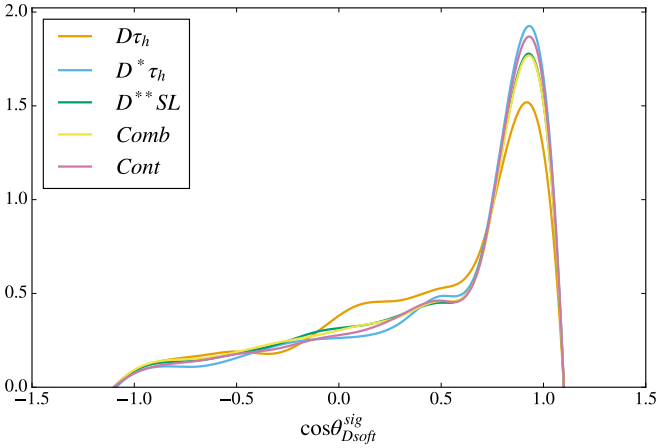


(o) m_D^{sig}

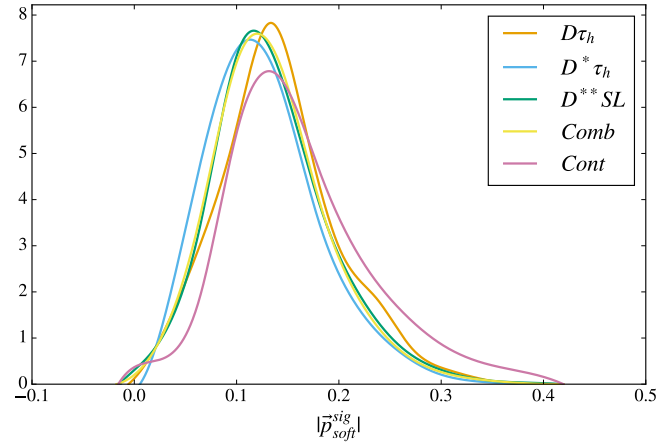
Figure 7.1: Feature density functions for each event type (cont.).



(p) Δm^{sig}.

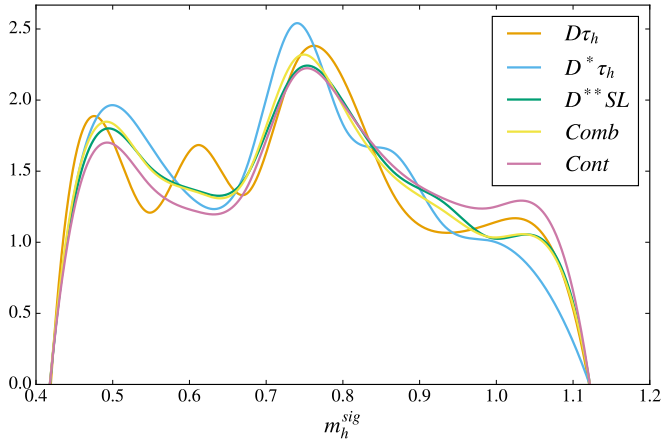


(q) $\cos\theta^{sig}_{Dsoft}$.

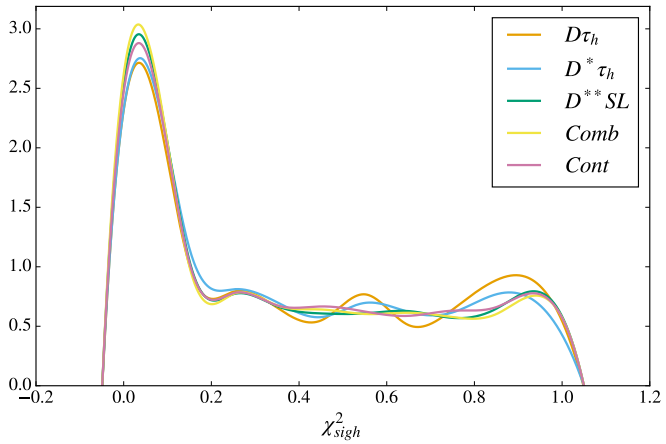


(r) $|\vec{p}^{sig}_{soft}|$.

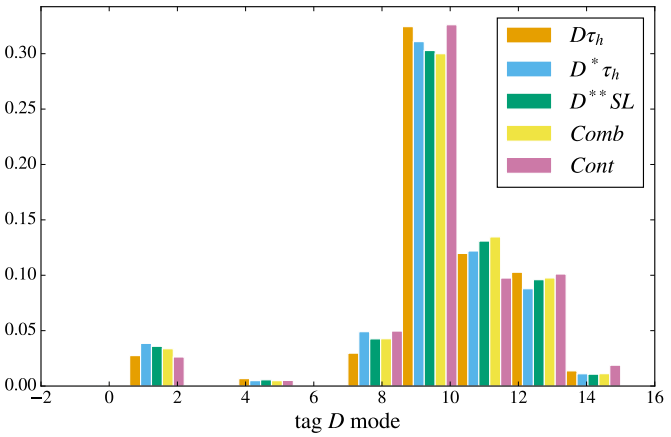
Figure 7.1: Feature density functions for each event type (cont.).



(s) m_h.

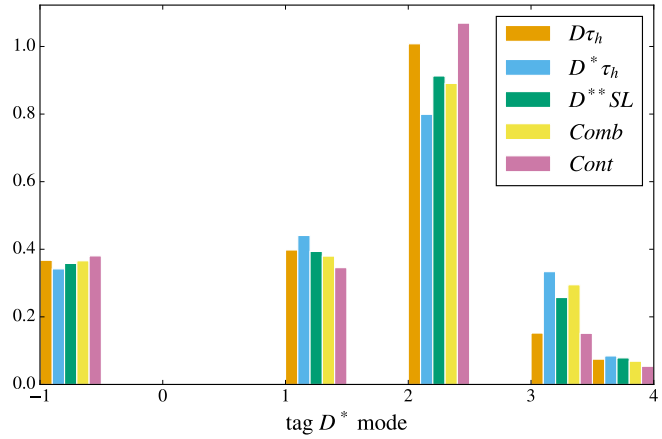


(t) χ^2_{sigh}.

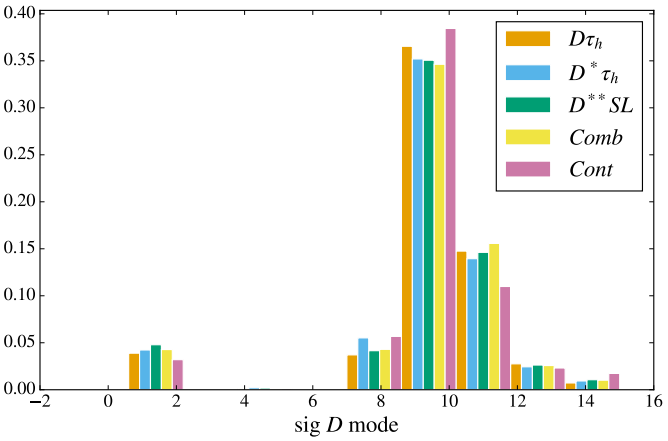


(u) B_{tag} D mode.

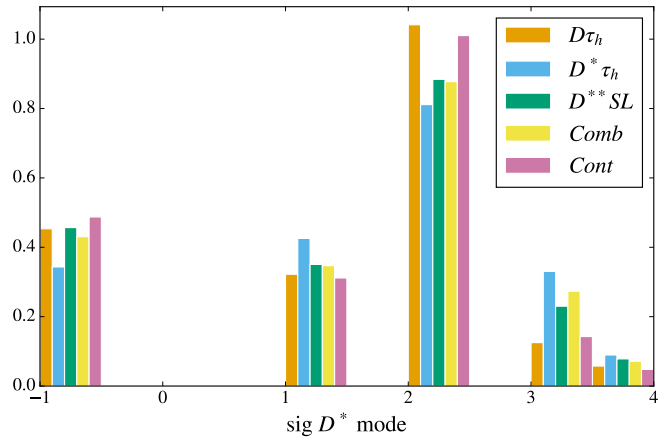
Figure 7.1: Feature density functions for each event type (cont.).



(v) B_{tag} D^* mode.

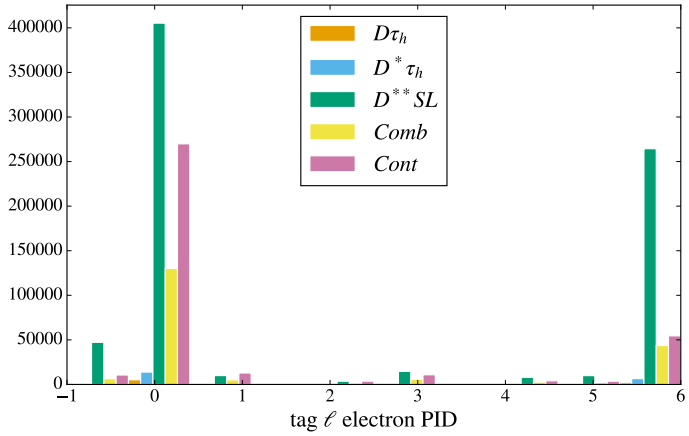


(w) B_{sig} D mode.

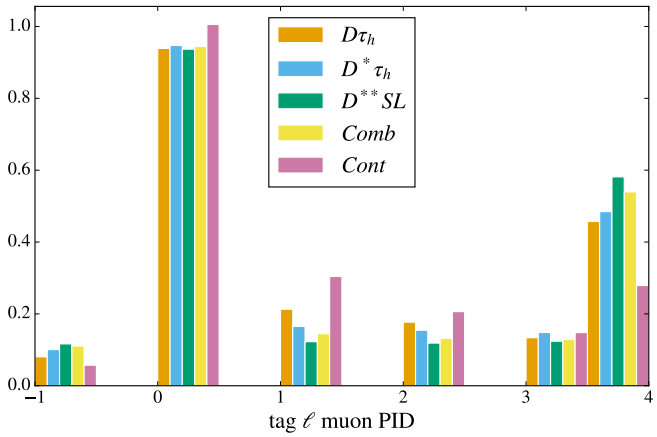


(x) B_{sig} D^* mode.

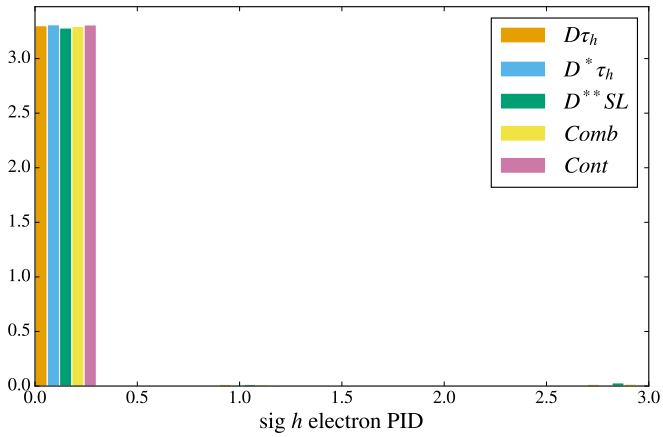
Figure 7.1: Feature density functions for each event type (cont.).



(y) B_{tag}'s ℓ daughter electron PID level.

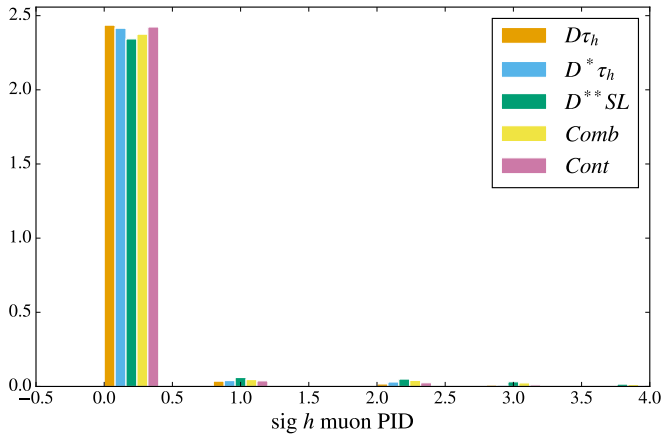


(z) B_{tag}'s ℓ daughter muon PID level.

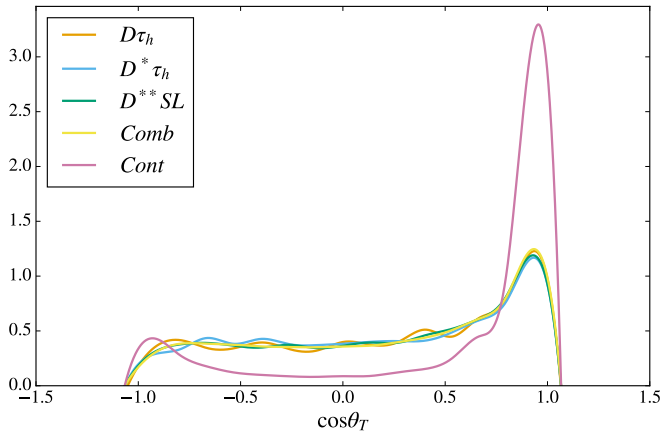


(aa) B_{sig}'s hadron daughter electron PID level.

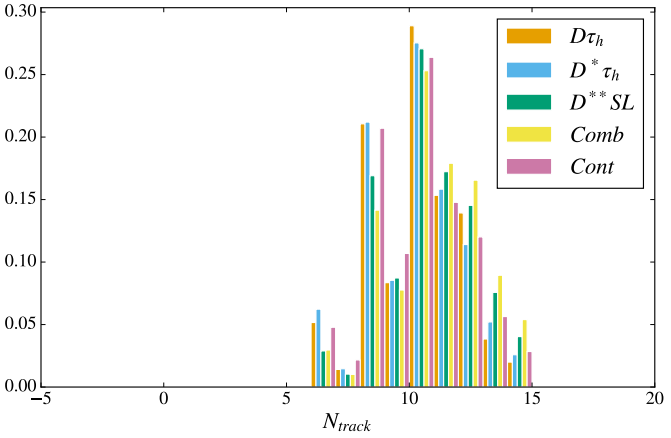
Figure 7.1: Feature density functions for each event type (cont.).



(ab) B_{sig}'s hadron daughter muon PID level.

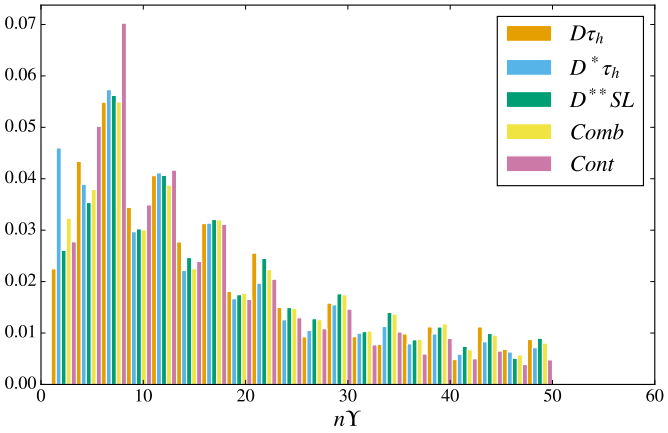


(ac) $\cos\theta_T$.

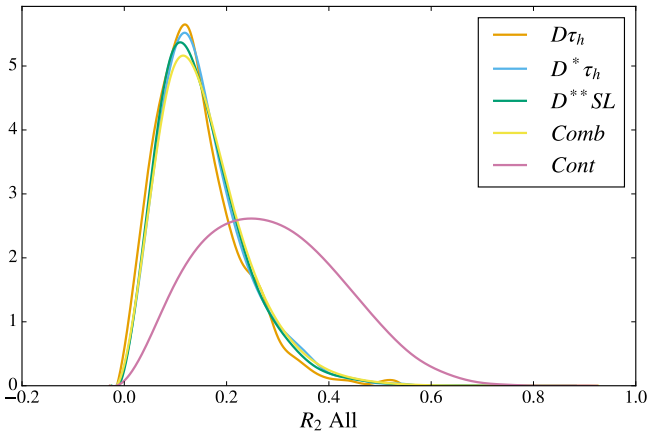


(ad) N_{tracks}.

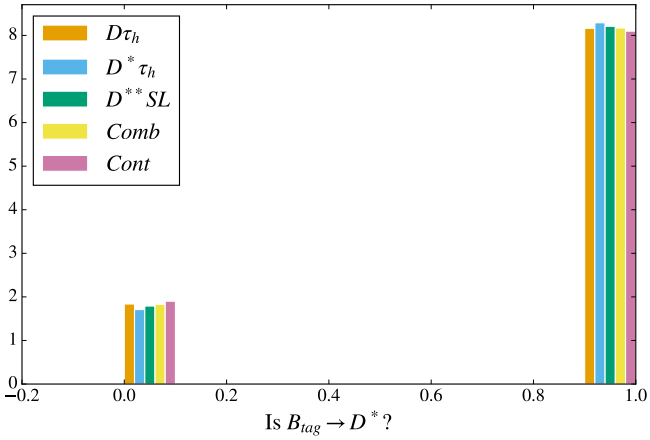
Figure 7.1: Feature density functions for each event type (cont.).



(ae) n_Υ.

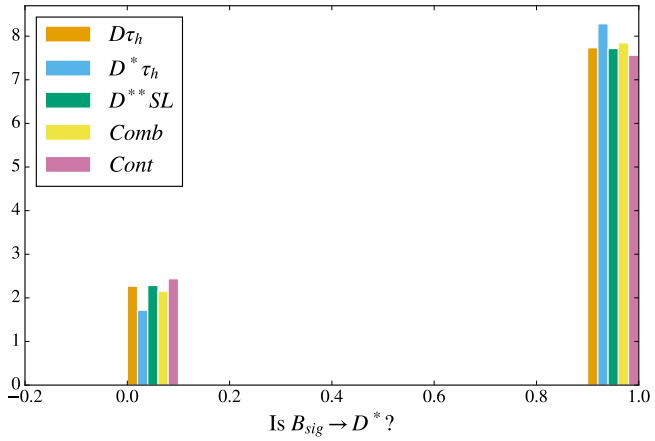


(af) R_2 All.

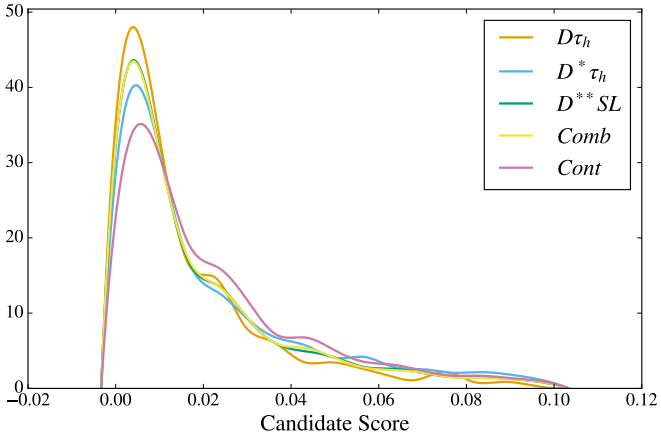


(ag) Does $B_{tag} \to D^*$?

Figure 7.1: Feature density functions for each event type (cont.).



(ah) Does $B_{sig} \to D^*$?



(ai) Score of the best candidate.

Figure 7.1: Feature density functions for each event type (cont.).

7.3 Simulation fidelity

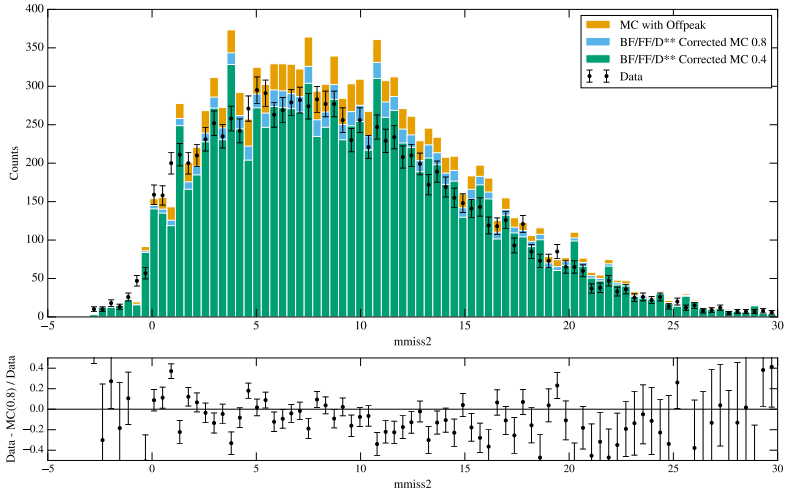
The reserved detector data can be used directly without any re-weighting. However, we present the 4% of the generic MC used for exploration with the following adjustments:

1. Replace the continuum component with the off-peak data.
2. Re-weight the branching fractions of the most frequently occurring B decay modes to the most recent PDG and HFLAV values. This will be discussed in more detail in section 11.3.

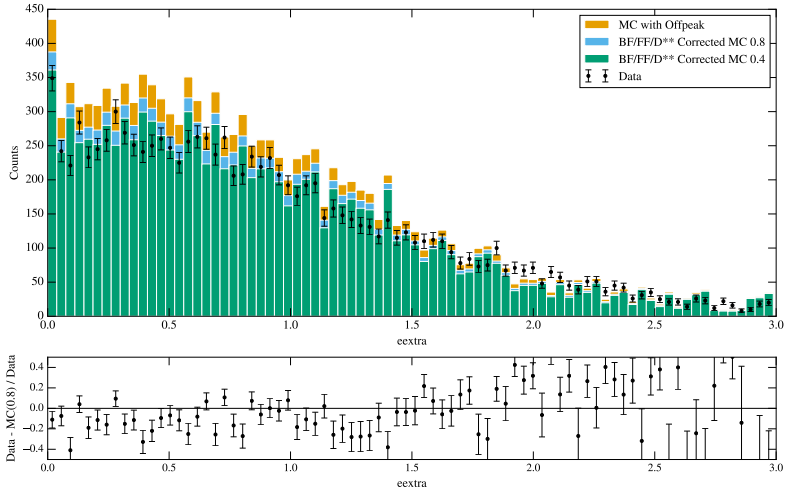
In addition, we also re-weight the branching fraction of the B semileptonic D^{**} modes under two scenarios to show the impact of its branching fraction uncertainty on the overall normalization. More specifically, we apply correction factors of 0.8 and 0.4 to the branching fractions, which is quite reasonable given the large uncertainties of these branching fractions. The intent is to compare the overall shapes of the densities rather than the normalizations.

Figure 7.2 shows the comparisons between data and MC for each feature.

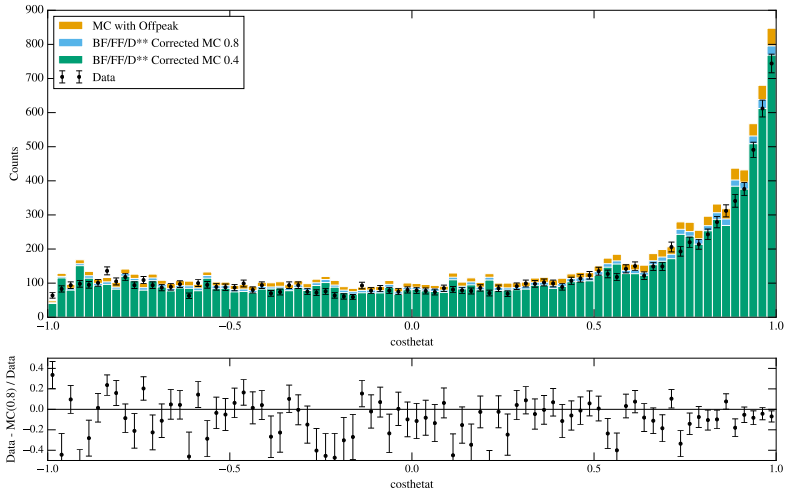
Aside from the uncertainty in the MC normalization, we note that the overall shapes of each feature agree reasonably well.



(a) M^2_{miss}.

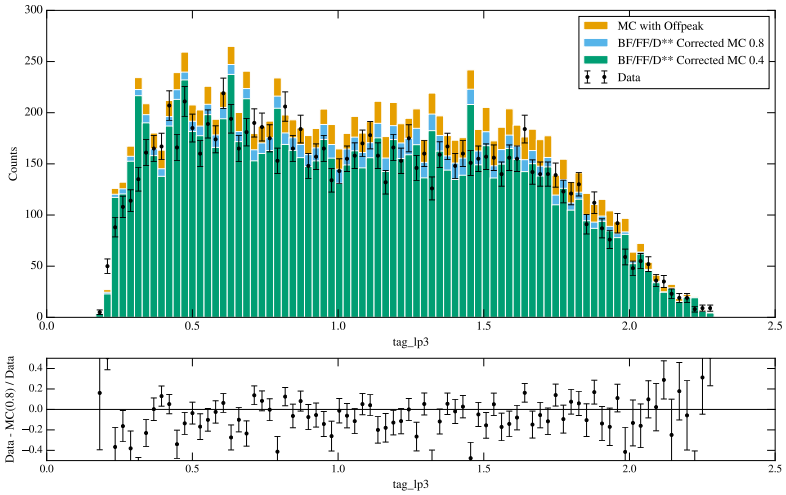


(b) E_{extra}.

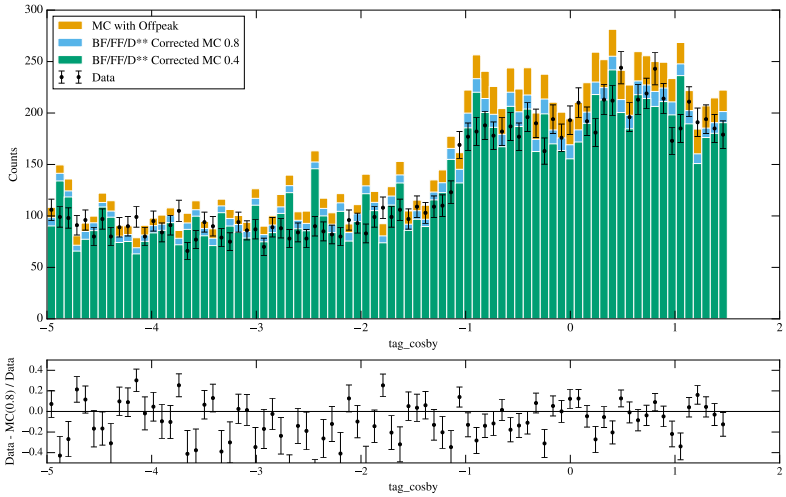


(c) $\cos\theta_T$.

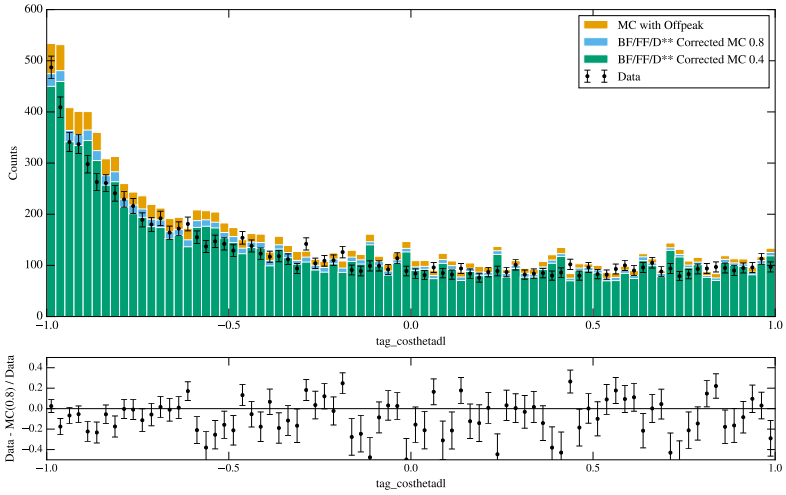
Figure 7.2: Comparisons between data and MC for each event type.



(d) $|\vec{p}_\ell^{\,tag}|$.

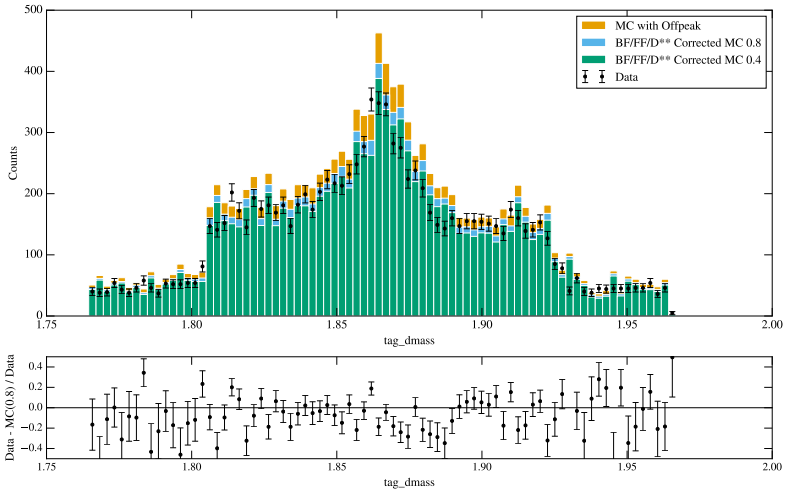


(e) $\cos\theta_{BY}^{tag}$.

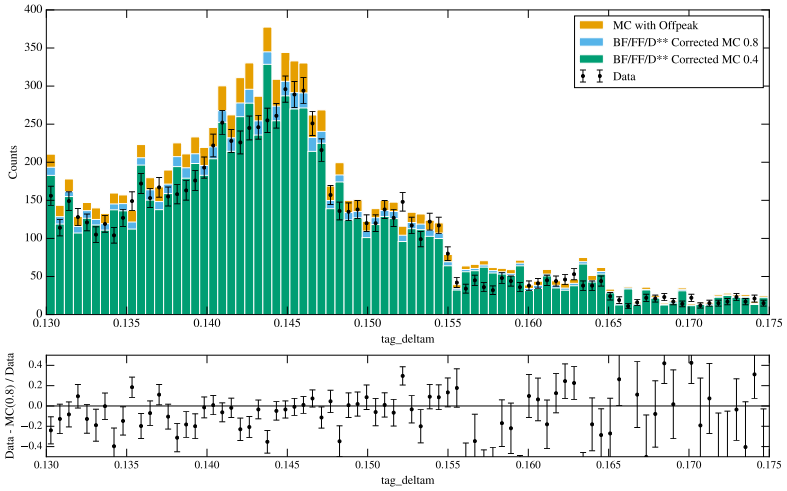


(f) $\cos\theta_{D\ell}^{tag}$.

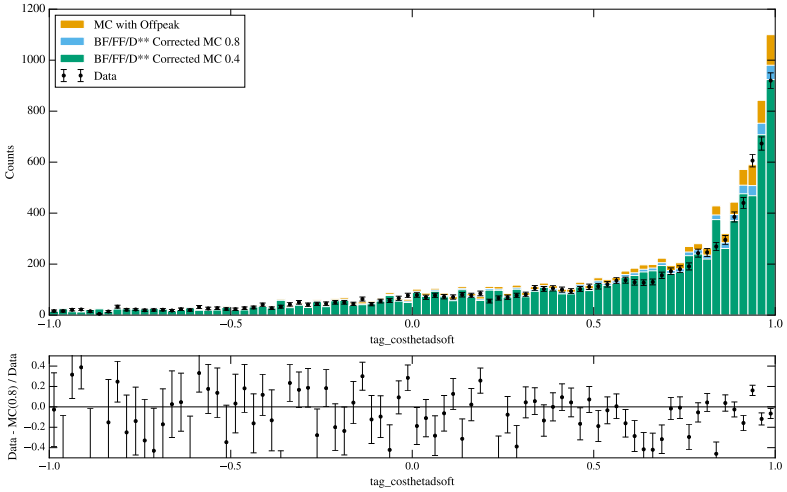
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(g) m_D^{tag}

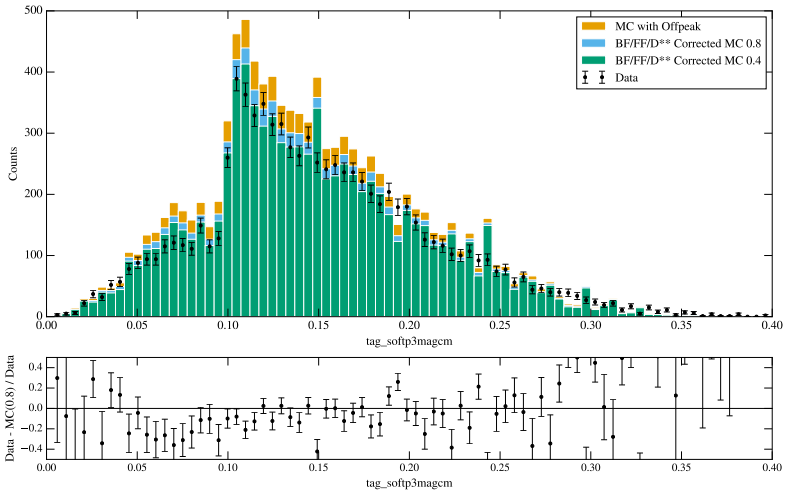


(h) Δm^{tag}.

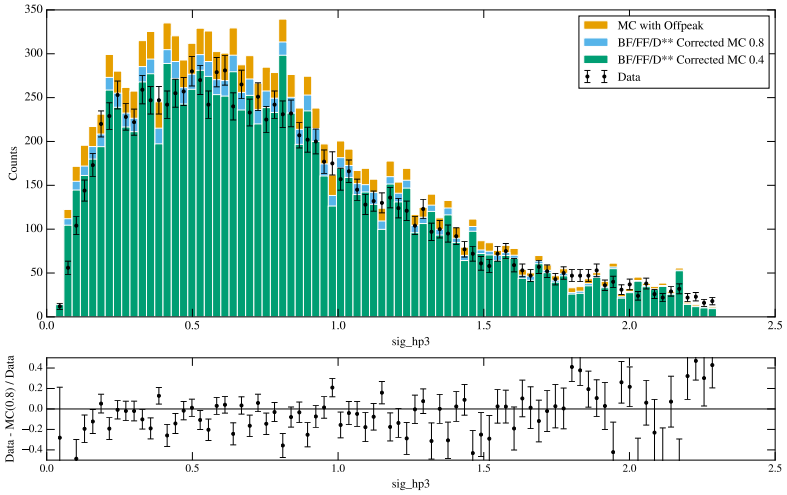


(i) $\cos\theta_{Dsoft}^{tag}$.

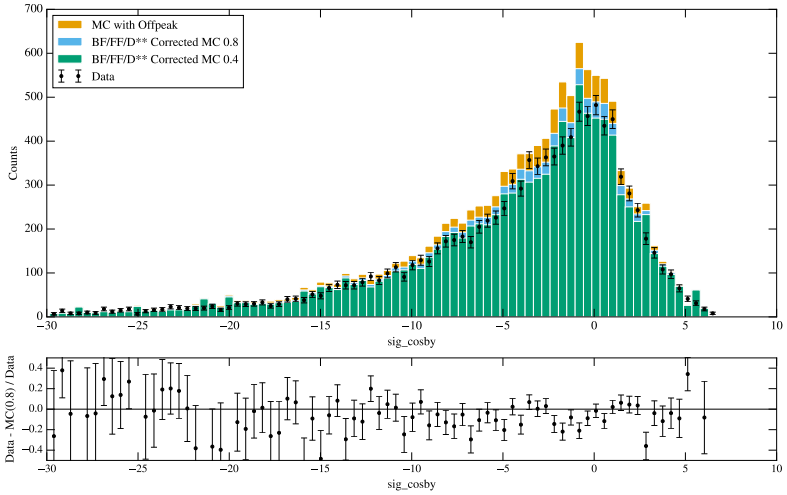
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(j) $|\vec{p}^{tag}_{soft}|$.

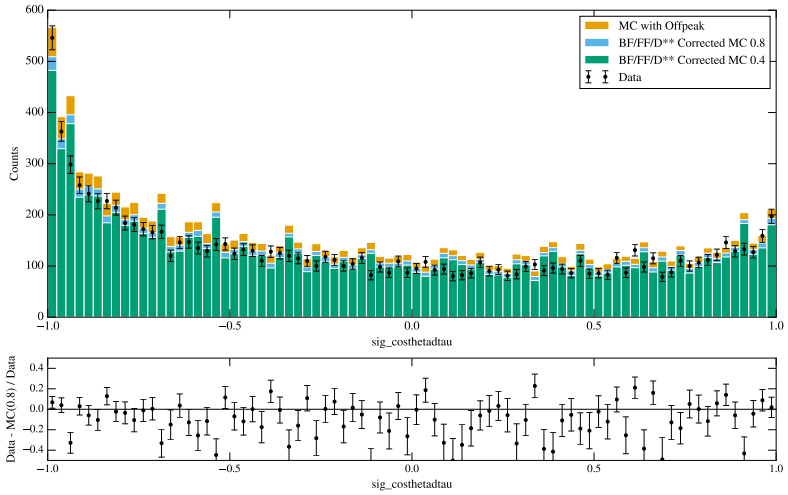


(k) $|\vec{p}^{sig}_{h}$.

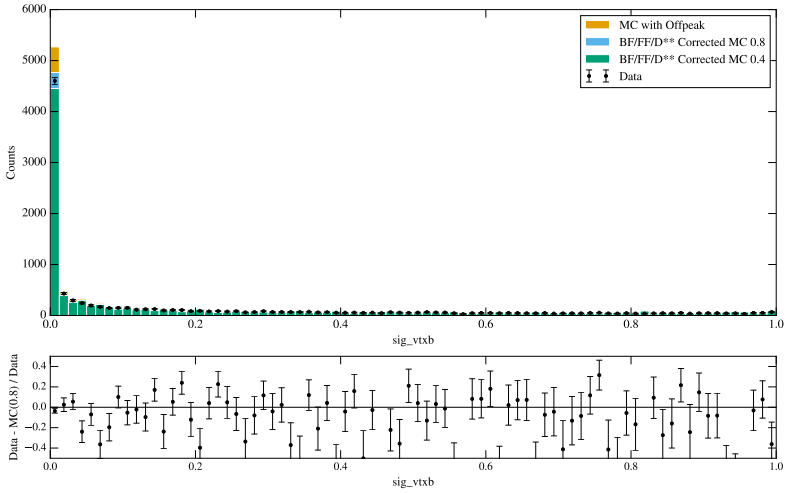


(l) $\cos\theta^{sig}_{BY}$.

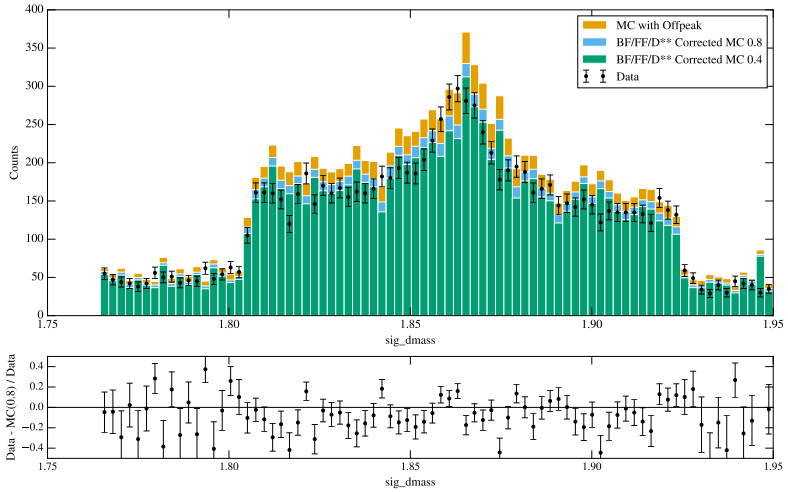
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(m) $\cos\theta_{D\tau}^{sig}$.

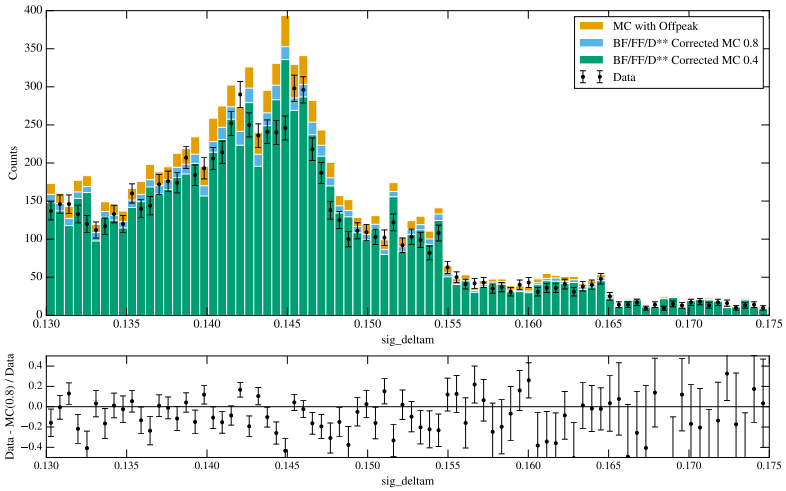


(n) χ^2_{sigB}.

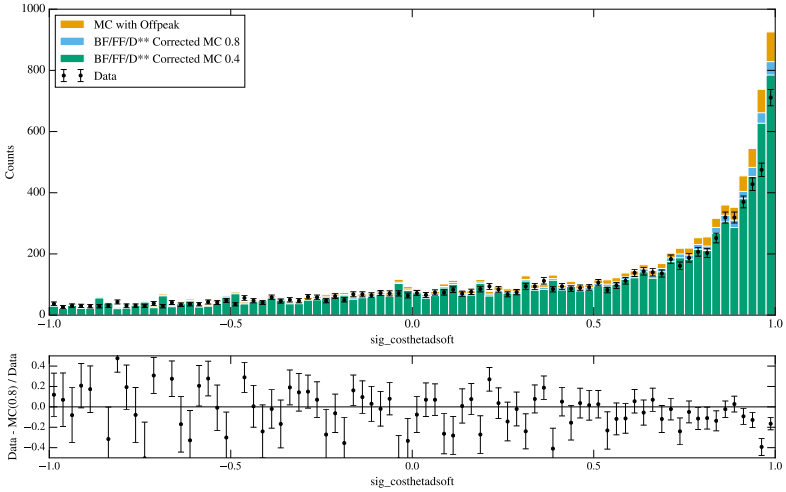


(o) m_D^{sig}

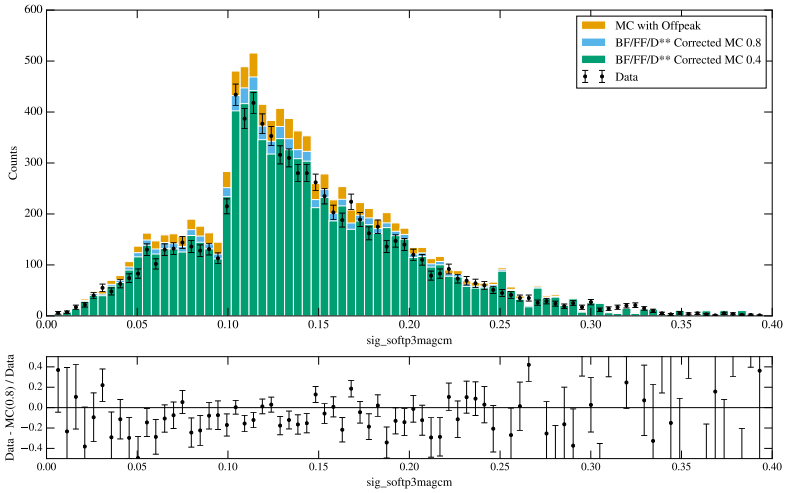
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(p) Δm^{sig}.

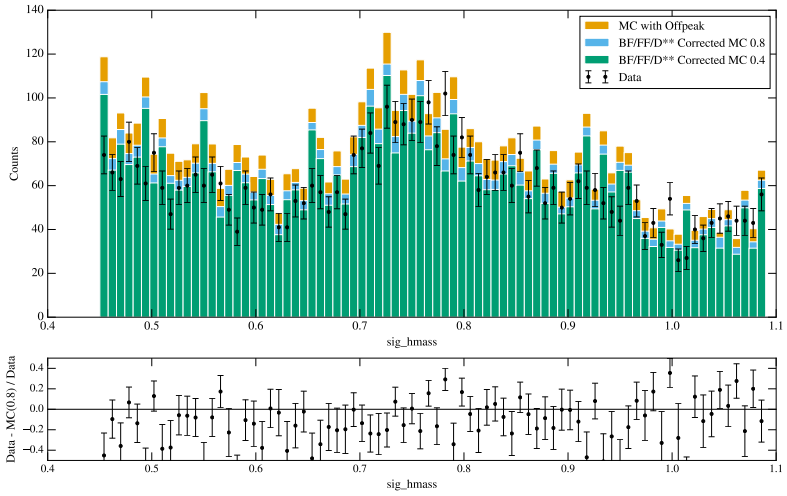


(q) $\cos\theta^{sig}_{Dsoft}$.

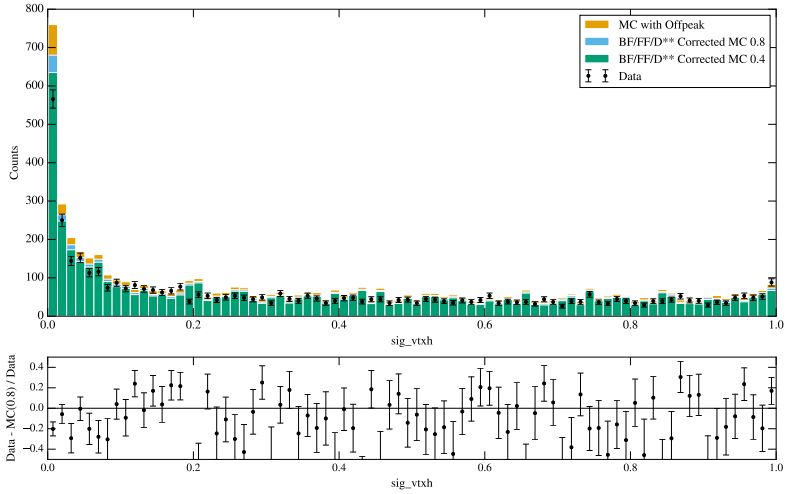


(r) $|\vec{p}^{sig}_{soft}|$.

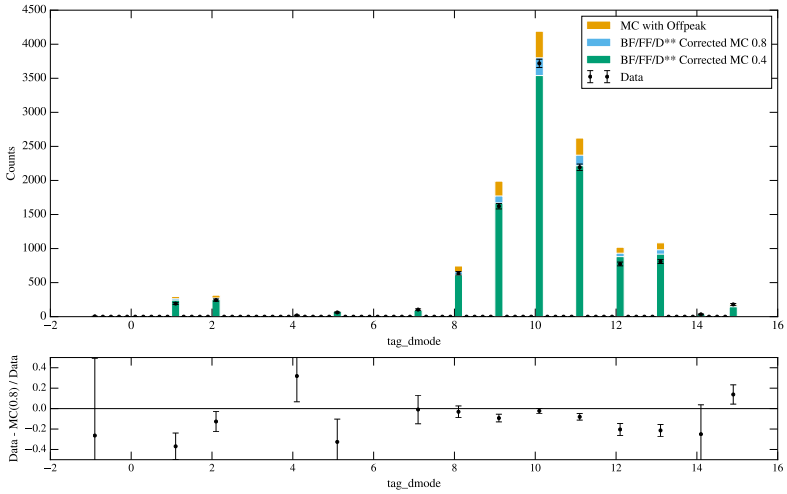
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(s) m_h.

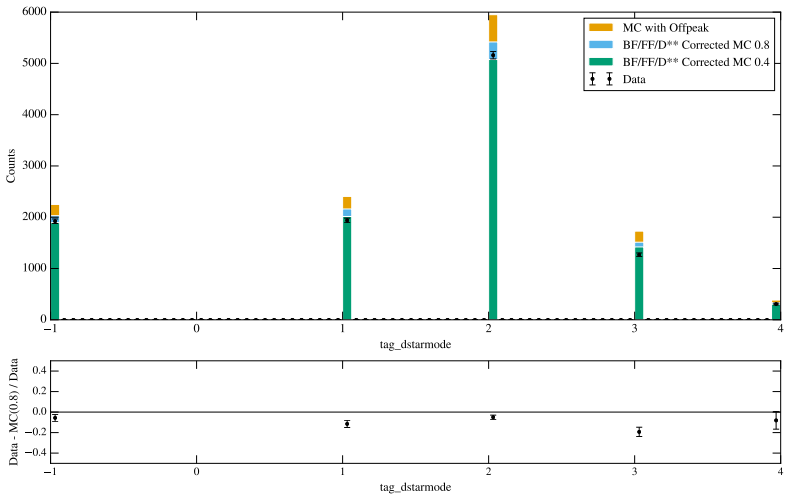


(t) χ^2_{sigh}.

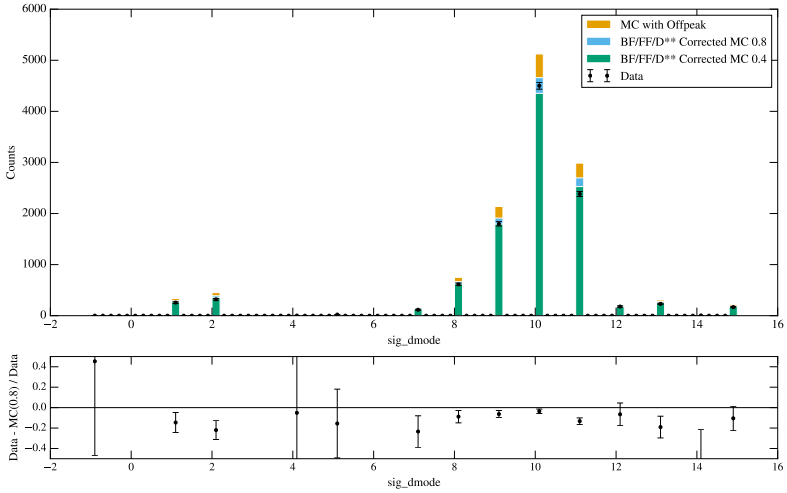


(u) B_{tag} D mode.

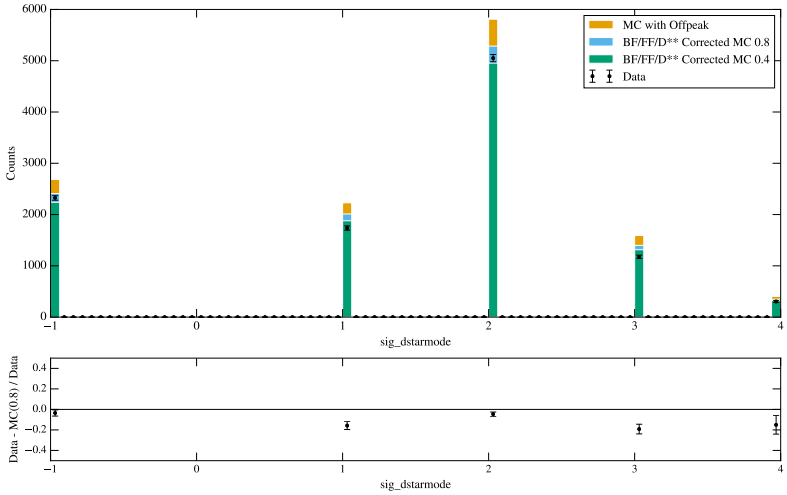
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(v) B_{tag} D^* mode.

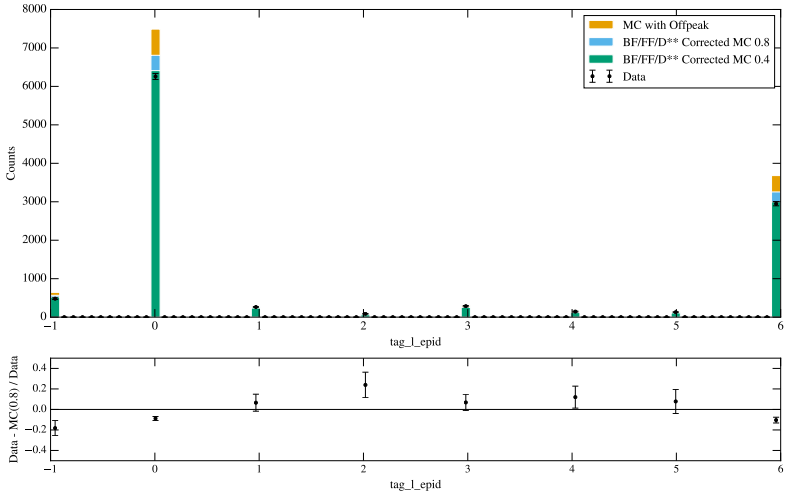


(w) B_{sig} D mode.

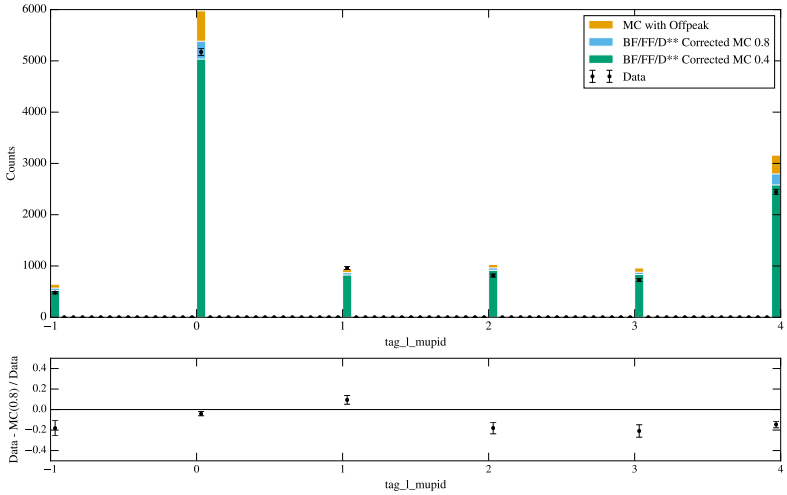


(x) B_{sig} D^* mode.

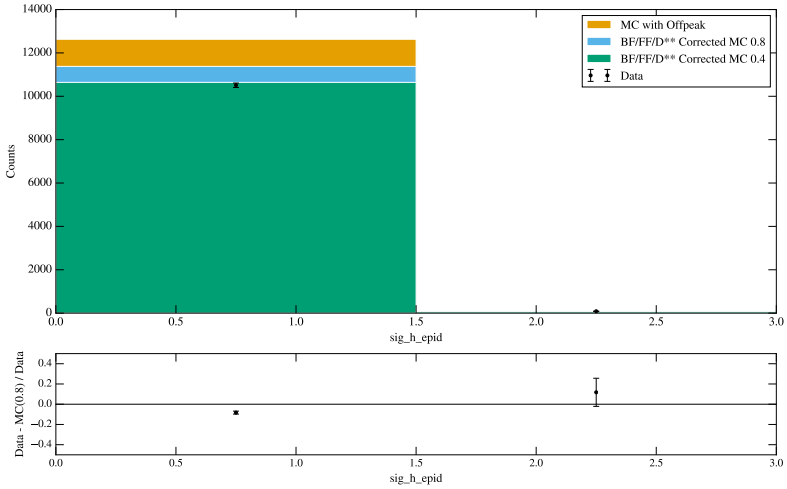
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(y) B_{tag}'s ℓ daughter electron PID level.

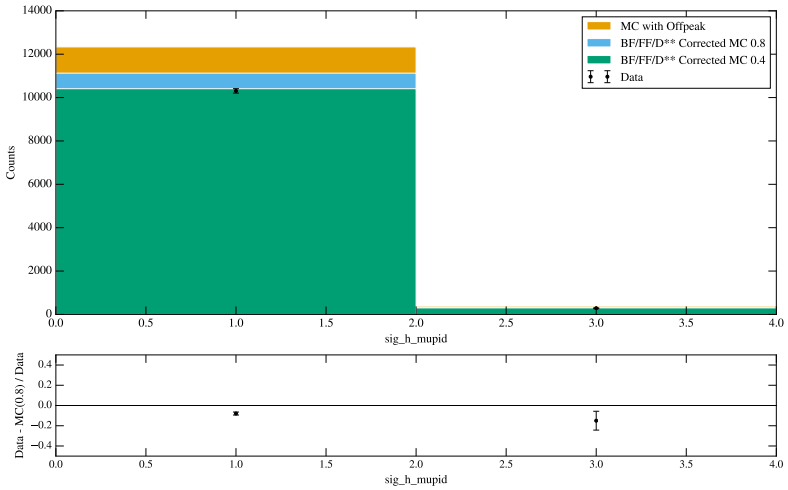


(z) B_{tag}'s ℓ daughter muon PID level.

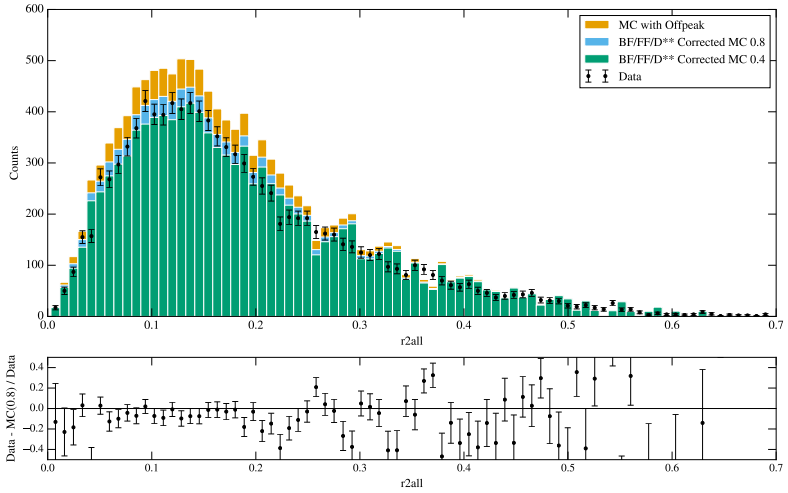


(aa) B_{sig}'s hadron daughter electron PID level.

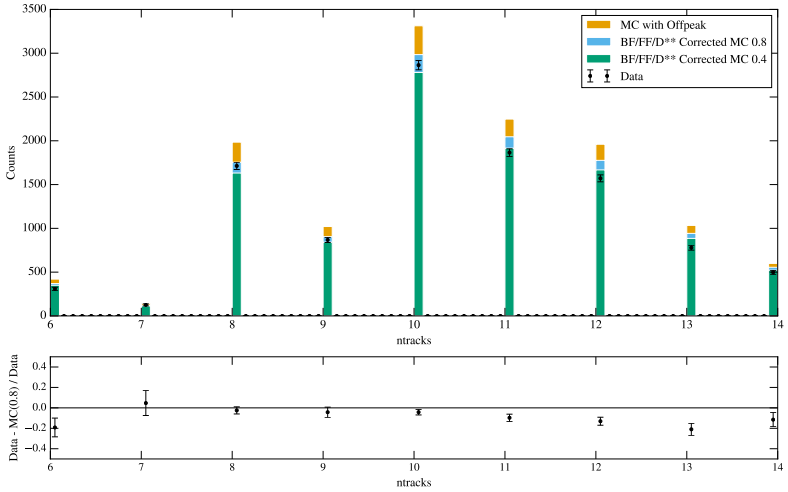
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(ab) B_{sig}'s hadron daughter muon PID level.

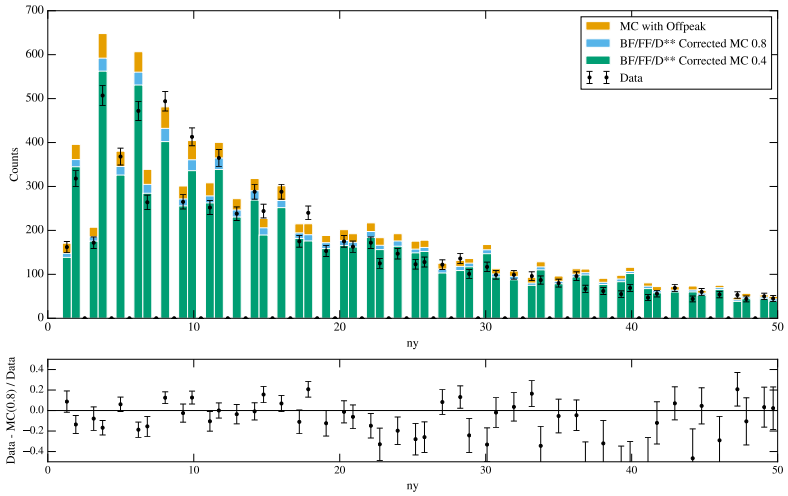


(ac) R_2 All.

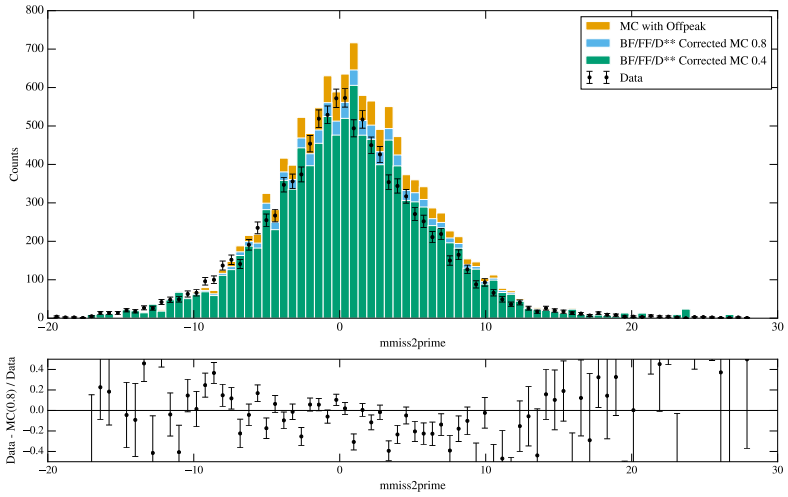


(ad) N_{tracks}.

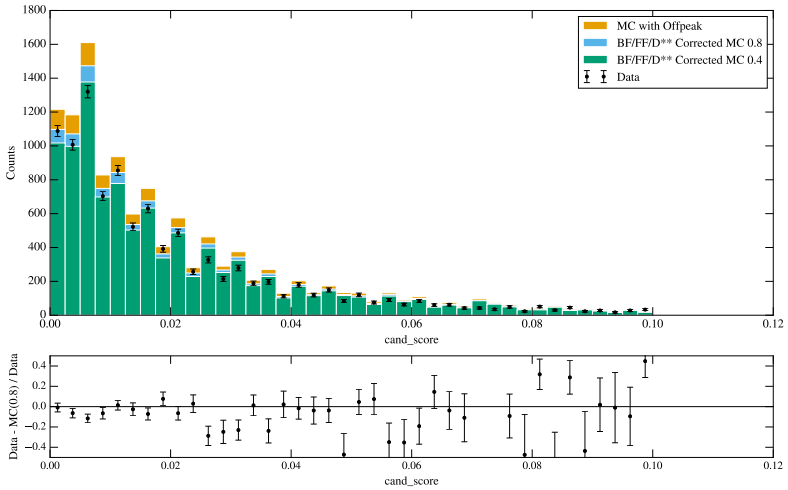
Figure 7.2: Comparisons between data and MC for each event type (cont.).



(ae) n_{Υ}.

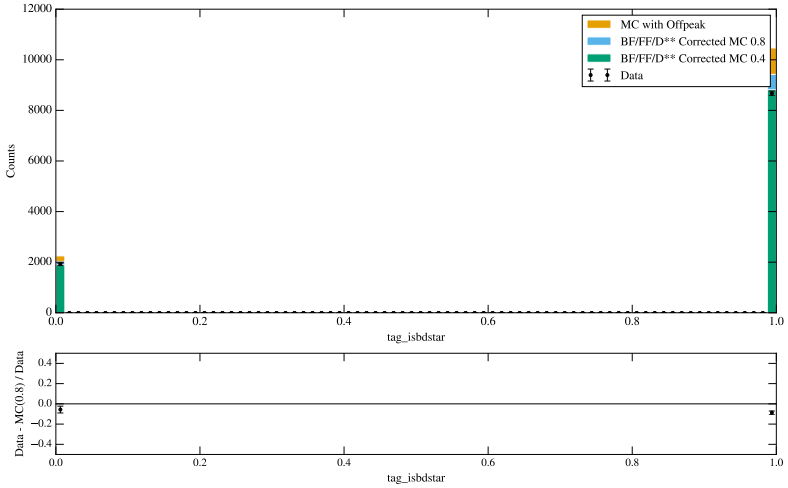


(af) Adjusted M^2_{miss}.

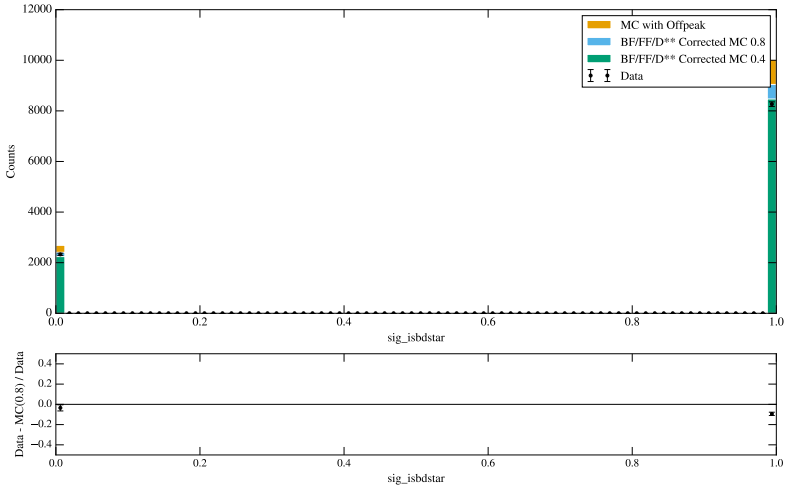


(ag) Score of the best candidate.

Figure 7.2: Comparisons between data and MC for each event type (cont.).



(ah) Does $B_{tag} \to D^*$?



(ai) Does $B_{sig} \to D^*$?

Figure 7.2: Comparisons between data and MC for each event type (cont.).

Chapter 8

Signal Detection

In this chapter we construct the regressors s_1 and s_2 that reduce the feature set into two representative statistics Z_1 and Z_2.

8.1 Signal detector

s_1 is the *signal detector* that is trained to separate the signal event types, $D\tau_h$ and $D^*\tau_h$, from the background event types, $D^{**}SL$, $Comb$, and $Cont$.

8.1.1 Training sample

The training sample consists of 85k events from the generic MC. The data is preprocessed by the standard centering (mean subtraction), scaling (set variance to 1), and one-hot encoding[1] of categorical features. Each event is given a label of 0 if it is a background event type and 1 if it is a signal event type based on the MC truth. The difficulty of learning on this dataset stems mostly from its inherent noise and the class imbalance. While the noise is a characteristic of the data that has no great solution, the class imbalance (ratio of signal to background events is 3%) is rectified somewhat by up-sampling the smaller class.

8.1.2 Model selection

Implementations of common machine learning algorithms are widely available and convenient to use. These algorithms range from logistic regression and tree-based models to more complex deep learning algorithms. Model selection also entails choosing the best set of hyperparameters given an algorithm. Thus, the same algorithm with different hyperparameters must be considered as different models in this context.

The criteria for choosing the best model are two-fold: first, the model must attain a low cross validation error for the given metric, area-under-curve or AUC. Second, we seek to choose the simplest model among those that perform well. The second condition is inspired by Occam's razor, which states that typically the simplest solution is the best. More quantitatively, this is telling us to choose the model with smaller variance at the possible cost of larger bias.

The algorithms that we explore are: logistic regression, random forest, support vector machine (SVM), gradient boosted decision tree (GBDT), and multilayer perceptron (MLP).

[1] One-hot encoding transforms categorical features into numerical bit states, making them suitable for learning.

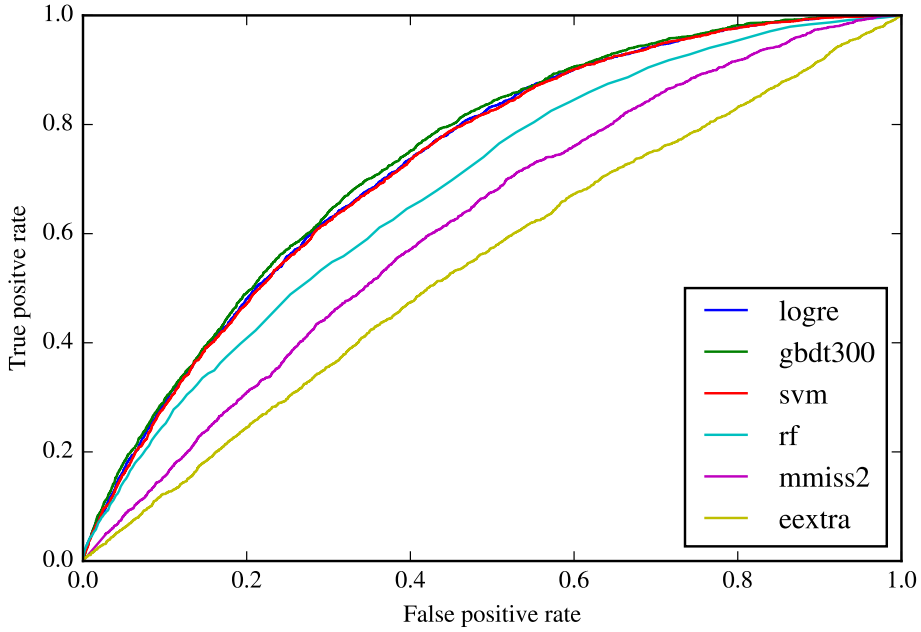


Figure 8.1: ROC curve for the Z_1 learners.

8.1.3 Results

The final model chosen is logistic regression with the ℓ_2 regularization strength $C = 100$. The performance of select models is shown in Figure 8.1, where *eextra* and *mmiss2* refer to the case where we use a single observable to classify the events. The three top performing models, logistic regression, random forest, and gradient boosted decision trees (GBDT), all result in similar performances, but the logistic regression model is chosen due to its simplicity. We also note that the univariate model using E_{extra} perform no better than random guessing, displaying the power of multivariate methods.

Figure 8.2 shows the learning curve of the logistic regression model, which tells us that we indeed have a sufficient number of training events.

We also explore the efficacy of various features, which might be useful for dimensionality reduction had the learning curve indicated a lack of training sample statistics. We calculate two metrics commonly used in machine learning to quantify feature importance:

1. Pearson correlation coefficient is the correlation between each feature and the output label. Powerful features tend to have larger correlation with the output label.

2. Mutual information is measurement of the (in)dependence of two random variables, in this case of a feature and the output label. Empirically, it is calculated for a feature column X and label column Y with N observations as

$$MI(X,Y) = \sum_{i\in|X|}\sum_{j\in|Y|}\frac{|X_i\cap Y_j|}{N}\log\frac{N|X_i\cap Y_j|}{|X_i||Y_j|}. \tag{8.1}$$

Figure 8.3 shows the two above metrics for the training dataset.

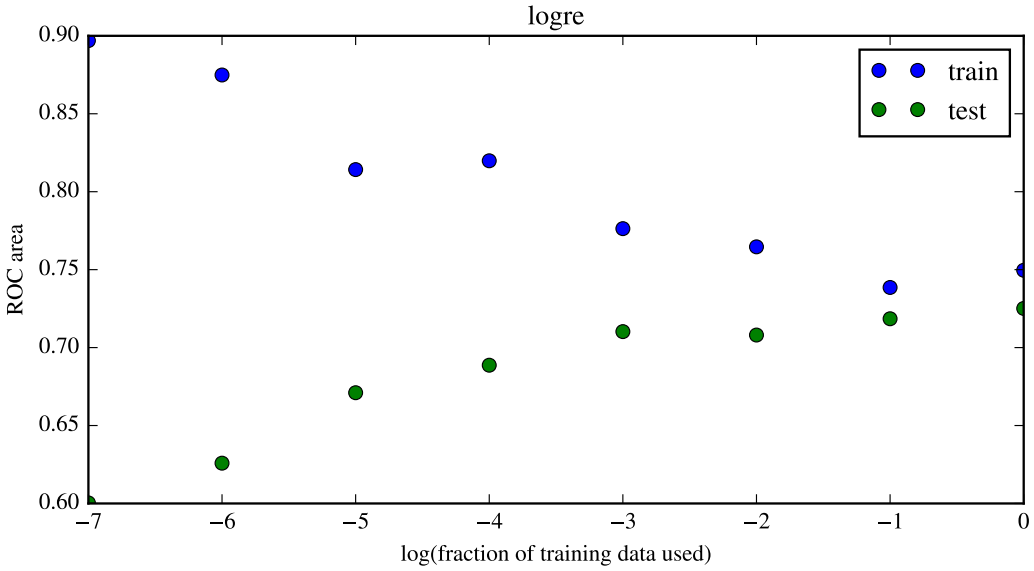


Figure 8.2: Learning curve for the logistic regression learner.

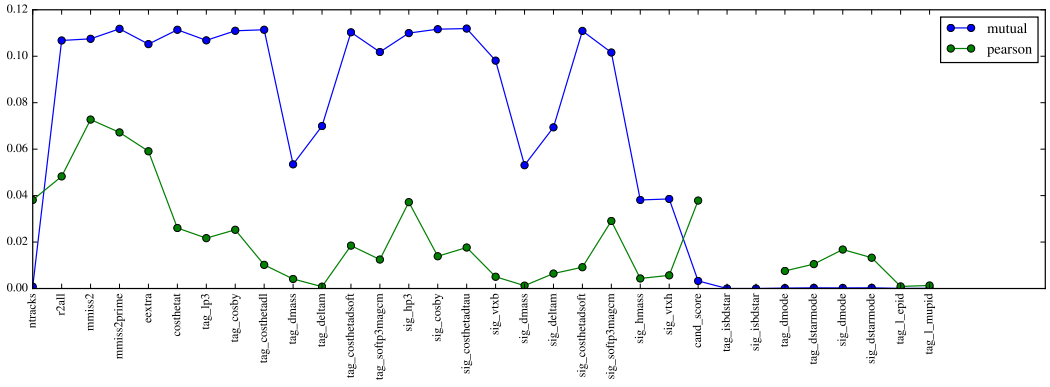


Figure 8.3: Relative importance of each feature for learning Z_1.

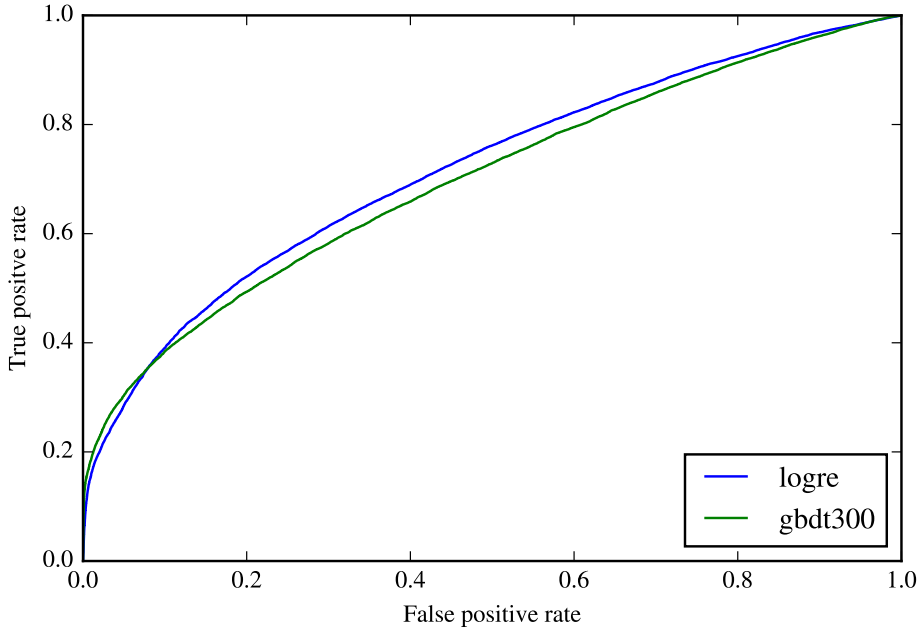


Figure 8.4: ROC curve for the Z_2 learners.

8.2 $D^*\tau$ detector

s_2 is the $D^*\tau$ *detector* that is trained to separate the two signal event types. This means that the performance of s_2 when given a background event is not important.

8.2.1 Training sample and model selection

The training sample consists of 85k events from the signal MC. The features are the same as those used in the signal detector with the addition of the signal detector score. The data is preprocessed in the same manner as the data for the signal detector. Each event is labeled 0 if it is of type $D\tau$ and 1 if it is of type $D^*\tau$. The model selection methodology of the signal detector is reused for the $D^*\tau$ detector.

8.2.2 Results

Again, the logistic regression algorithm wins out due to its performance and simplicity, with the ℓ_2 regularization parameter $C = 1000$.

As with the signal detector, we show the performance of select models, the learning curve of the final model, and the feature importance in Figures 8.4, 8.5, and 8.6, respectively.

8.3 Choosing Z_1 and Z_2

The scores output by s_1 and s_2 could be used directly as the representative statistics. However, we choose to use a transformed version of the statistics:

- $Z_1 = \text{logit}(s_1(X))$,
- $Z_2 = \text{logit}(s_2(X))$,

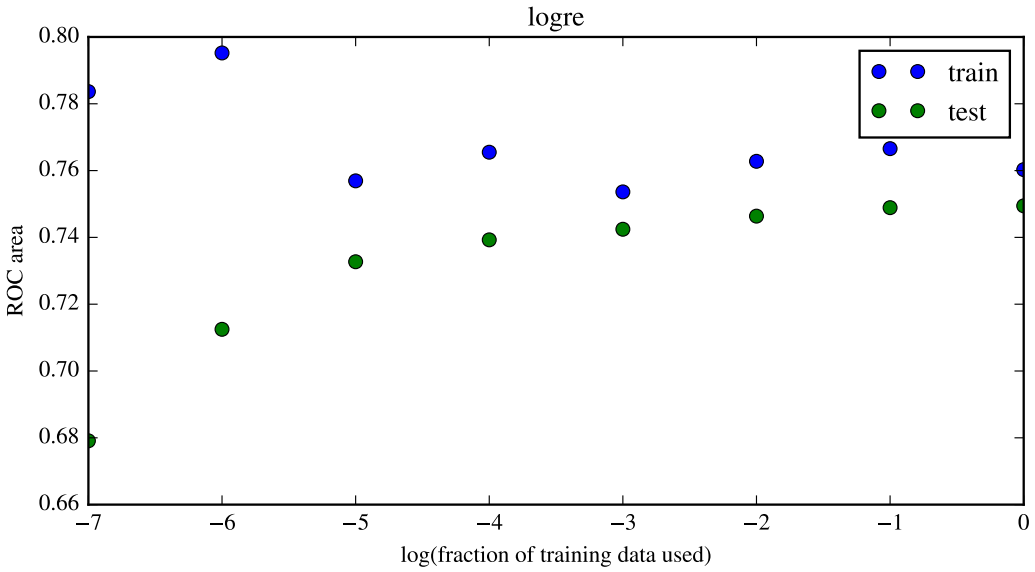


Figure 8.5: Learning curve for the logistic regression learner.

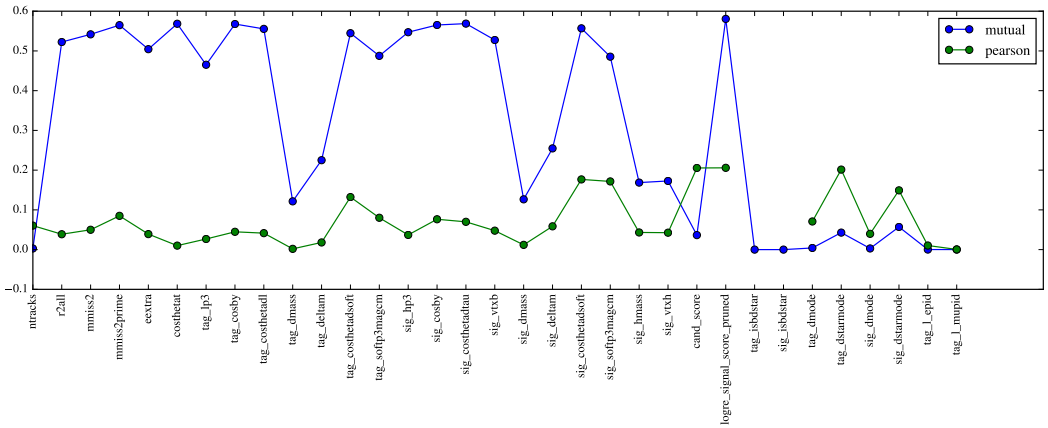


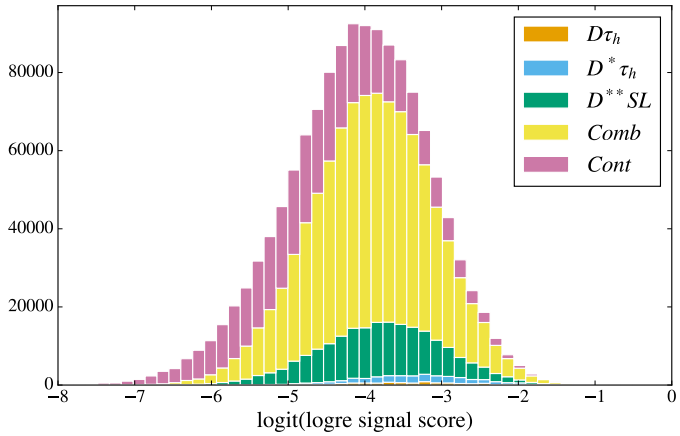
Figure 8.6: Relative importance of each feature for learning Z_2.

where the logit function, the inverse of logistic function, is defined as:

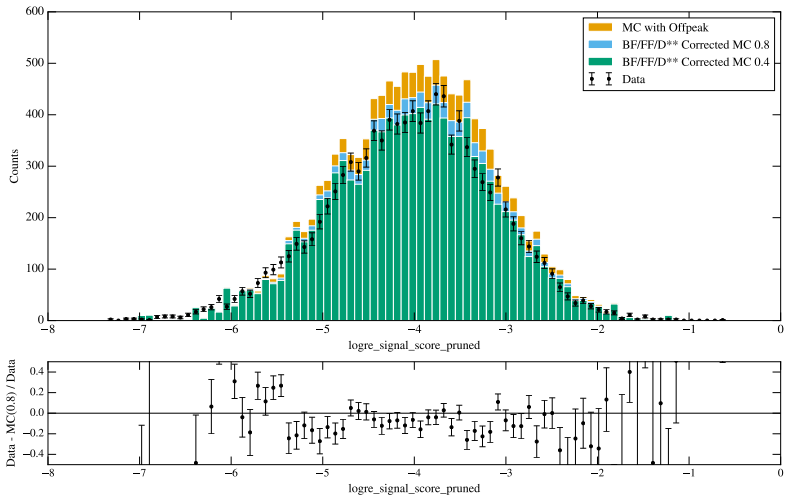
$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right). \tag{8.2}$$

The logit transformation is essentially undoing the activation function of the output layer in our model, meaning Z_1 and Z_2 are raw outputs of the hidden layer. This allows the output scores to not be restricted between 0 and 1, and has a smoothing effect on the distributions. This is actually quite crucial for the next steps in the analysis which involve estimating these densities using kernel density estimation, or KDE. The density estimation will be the topic of the next chapter, but in short, KDE does not perform well for distributions with sharp peaks, and the logit transformation helps to combat this shortcoming.

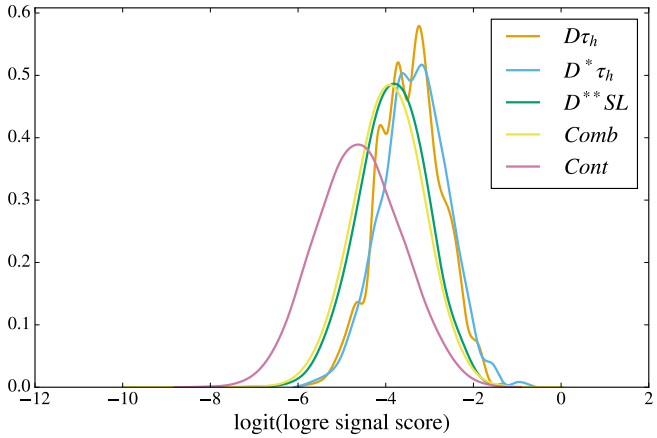
Figures 8.7 and 8.8 show the densities of the exploratory samples projected to Z_1 and Z_2, respectively, in the same manner as presented in Chapter 7.



(a) Event type stacked histograms.

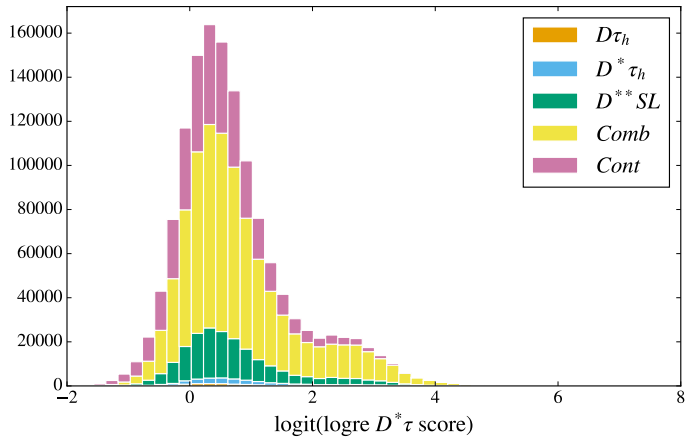


(b) Data-Simulation comparison.

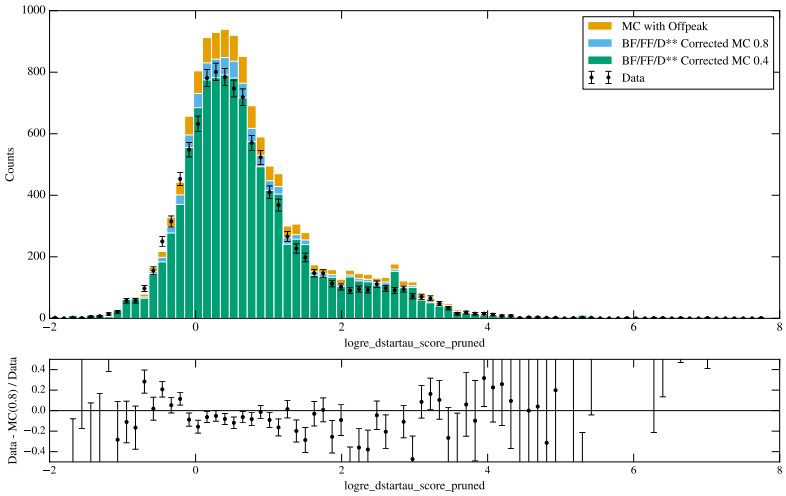


(c) Event type densities. The fluctuations of the signal event type densities are due to limited statistics.

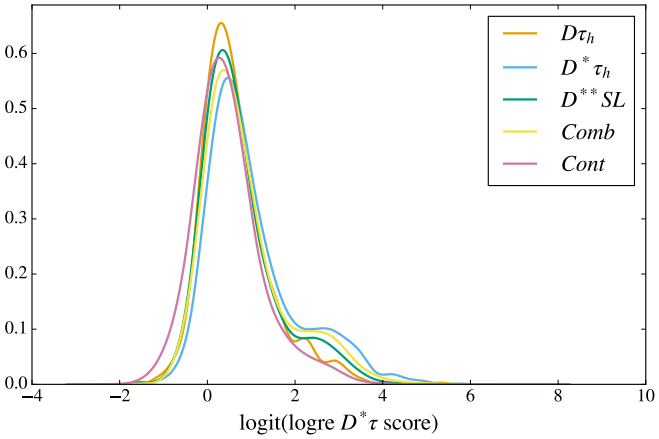
Figure 8.7: Z_1 score.



(a) Event type stacked histograms.



(b) Data-Simulation comparison.



(c) Event type densities. The fluctuations of the signal event type densities are due to limited statistics.

Figure 8.8: Z_2 score.

Chapter 9

Signal Extraction

In this chapter we discuss how the proportions of the signal event types, $\hat{p}_{sig}$ are estimated. More specifically, we have to solve the optimization problem posed in (4.15) which we write down again for convenience:

$$\begin{aligned} \underset{p \in \mathbb{R}^{|C|}}{\text{minimize}} \quad & -\sum_{i=1}^{N} \log \left(\sum_{j \in C} p_j f_j(z_i) \right) \\ \text{subject to} \quad & \sum_{j \in C} p_j = 1, \end{aligned} \tag{9.1}$$

where

- C: The set of event type categories. They are $\{D\tau_h,\, D^*\tau_h,\, D^{**}SL,\, Comb,\, Cont\}$.
- N: Total number of events in the test set.
- z_i: Observed value of the signal detector score $Z = (Z_1, Z_2)$ for event i.
- p_j: The optimization variable. It represents the proportion of events that belong to event type j.
- f_j: The (conditional) density function of Z of event type j.

It is clear that the test set of interest is the detector data. However, we can also use a test set reserved from the generic MC. Since the event type proportions of such test set are known, we use it to validate our methods.

We must know how to perform the following two things in order to proceed:

- Density estimation.
- Maximum likelihood estimation.

The purpose of this chapter is to describe our methods of performing the two above tasks.

9.1 Maximum likelihood estimation

The estimation of the event type proportions is implemented using the CVXOPT [25] python package. The variance of the estimate is estimated using the bootstrap method, which is computed in parallel in the department cluster.

9.2 Density estimation

The problem of density estimation is as follows: suppose we observe N instances of a random variable X that follows the probability distribution f. Further suppose that f is unknown. The task is to estimate f based on the N observations X_i, $i = 1, \ldots, N$.

There are two general approaches to density estimation: parametric and non-parametric. By introducing assumptions about the distribution, the parametric methods achieve great computational efficiencies at the cost of the bias from the assumptions. On other hand, non-parametric methods let the data speak for itself at the cost of being more computationally intensive.

We will proceed with using non-parametric density estimation methods. For a more in-depth discussion, see [26].

The most basic non-parametric density estimation method is the histogram method. Given the range of X_i, we partition the range into m equal partitions or bins. Then we assign each observed data point into a specific bin. The counts of each bin normalized to the number of points gives us $\hat{f}_m$.

For a histogram, the only parameter is m, the number of bins. We can then ask how do we choose m? The answer is to choose m that minimizes a metric that quantifies how well the histogram describes the true unknown distribution f.

There are many such metrics and heuristics of choosing the optimal m. The metric we will minimize is the mean integrated squared error, or MISE, which is defined as:

$$MISE \equiv \int (\hat{f}_m - f)^2 = \int \hat{f}_m^2 - 2 \int \hat{f}_m f + \int f^2. \tag{9.2}$$

In practice, one can evaluate the integrals by summing over each observed data point.

The histogram method in effect discretizes the probability density, which is typically known to produce a less accurate estimate of the true density. The method we will use for density estimation is kernel density estimation, or KDE, which is known to perform better than the simple histograms. The KDE is also known as the Parzen window method.

The KDE of f is:

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right), \tag{9.3}$$

where h is the bandwidth parameter for the kernel function K. Common choices of the kernel K include the Gaussian, radial basis, and Epanechnikov functions.

As with the histogram method, we have a parameter controlling the smoothness of the estimate, h. We can choose the h that minimizes the MISE. But in addition, we have the choice of the kernel.

It turns out [27] that rather than minimizing the MISE directly, we can minimize the cross validation score, which achieves the same asymptotic accuracy:

$$CV(h) = \int \hat{f}^2 - 2N^{-1} \sum_i \hat{f}_{-i}(x_i), \tag{9.4}$$

where $\hat{f}_{-i}(x_i) = \frac{1}{Nh} \sum_{i \neq j} K\left(\frac{x_i - x_j}{h}\right)$; that is, simply evaluate (9.3) but remove the contribution due to point i.

We use the Epanechnikov function as the kernel K, which has a slight advantage in theoretical guarantees such as having bounded support.

9.2.1 Implementation of kernel density estimation

We implement the KDE as a custom software library `bbrcit_kde` that takes advantage of the speedups provided by GPU computation and implements the algorithm that performs

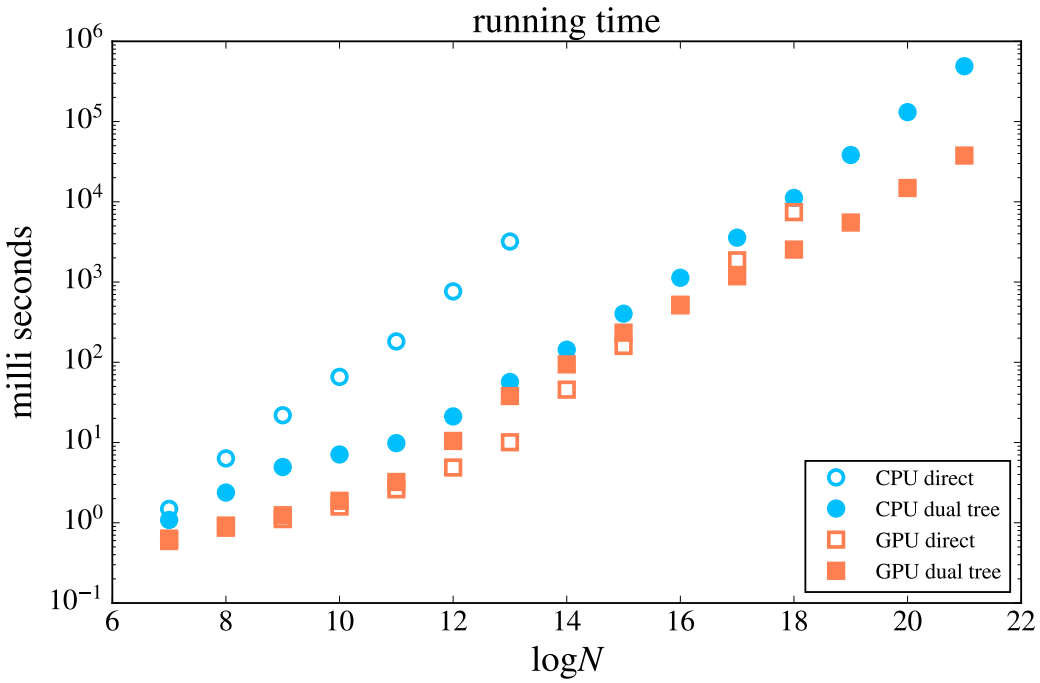


Figure 9.1: Runtime benchmark. N is the number of training and query points and *log* is in base 2. $1M$ is approximately when $\log N = 20$

KDE in $\mathcal{O}(n)$ [28]. We will only provide an overview of the software and the algorithm; the full details of the algorithm implementation can be found in [9].

The naive algorithm of KDE has a $\mathcal{O}(n^2)$ time complexity. This can be easily be shown to be true since we have to evaluate the kernel for all possible pairs of points. The so-called dual tree algorithm is inspired by the two following observations:

1. $K(x) \to 0$ as $x \to \infty$, meaning the contributions of points far apart is negligible.

2. The above observation leads to the idea of partitioning the space into cells, in which we calculate the pair-wise contributions for points only in the same cell.

By using a space partitioning scheme such as the kd-tree [29], we can evaluate any query point x_i in $\mathcal{O}(\log n)$. The dual-tree algorithm takes this a step further and also partitions the query points, resulting in a linear time algorithm for KDE.

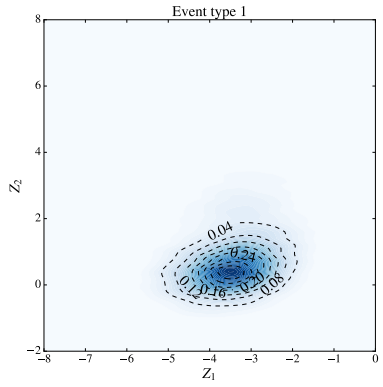
In the implementation of the dual-tree algorithm, we rely on the high-performance parallel computation capabilities of graphical processing units (GPU).

The performances of the dual-tree algorithm and the naive algorithm on the CPU and the GPU are shown in Figure 9.1, which exhibits a factor of 10,000 improvement in speedup between the naive-CPU case and the dual-tree-GPU case.

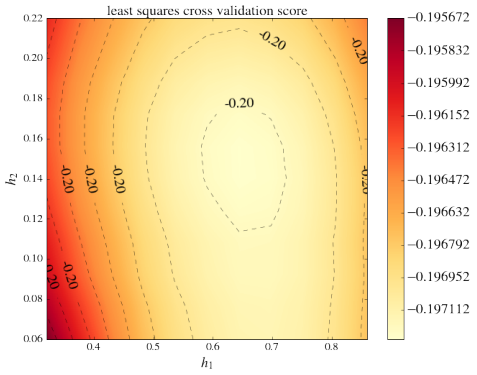
9.2.2 Density estimation training sample

The conditional densities f_j are learned from the generic MC. This is not the optimal choice of the training data, as detector data *control samples* for each event type would result in much more accurate descriptions of the components.

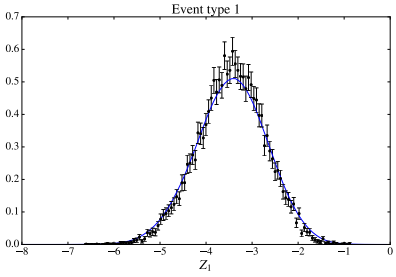
For each event type sample, we perform the cross validation to choose the best bandwidths.



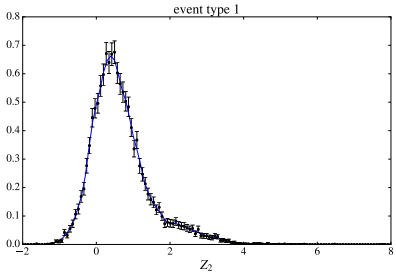
(a) Estimated 2D density.



(b) Least square cross validation scores.



(c) Marginal density in Z_1.



(d) Marginal density in Z_2.

Figure 9.2: Kernel density estimates for the $D\tau$ event type.

9.2.3 Results

Figures 9.2, 9.3, 9.4, 9.5, and 9.6 show the estimated densities for each event type. Each figure contains 4 subfigures ordered from left to right, and then top to bottom. They show the following:

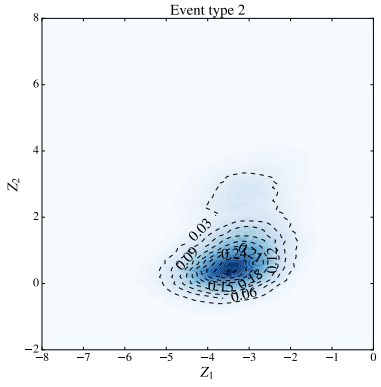
1. The 2D density estimate f_j.
2. The least square cross validation score evaluated over a grid for the purpose of finding the optimal bandwidth.
3. The 1D marginal density in Z_1 along with a histogram of the training points.
4. The 1D marginal density in Z_2 along with a histogram of the training points.

The estimated densities all appear to behave as expected.

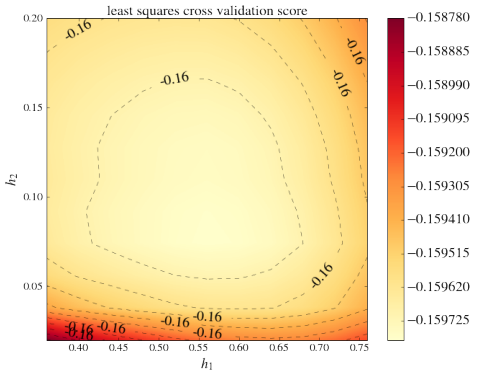
9.3 Bias correction of extracted signal proportions

A typical analysis in particle physics is performed *blind*, as is the case here. This means that the analysis strategy is first performed on the MC in its entirety, as described in this chapter. It is only when the analysis procedures are locked in that we *unblind* the data and perform the actual analysis that will give us the final result.

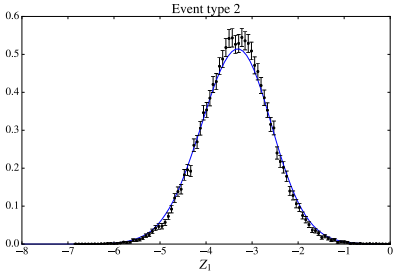
This implies that we use a test set of simulated data to validate our procedure in extraction of the $\hat{p_i}$'s. This MC test set is composed of the five event types in proportions that we believe best describe the truth, in this case based on the Standard Model prediction and the world average.



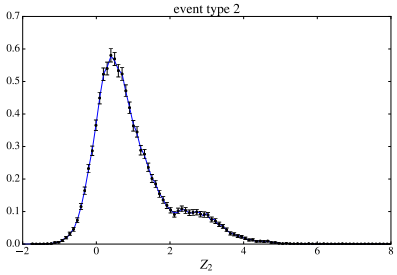
(a) Estimated 2D density.



(b) Least square cross validation scores.

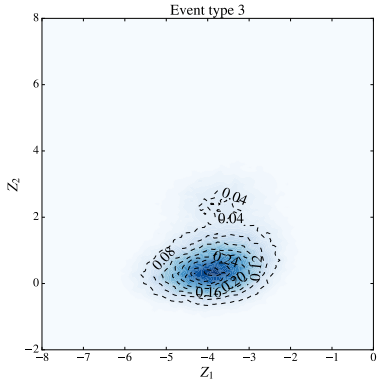


(c) Marginal density in Z_1.

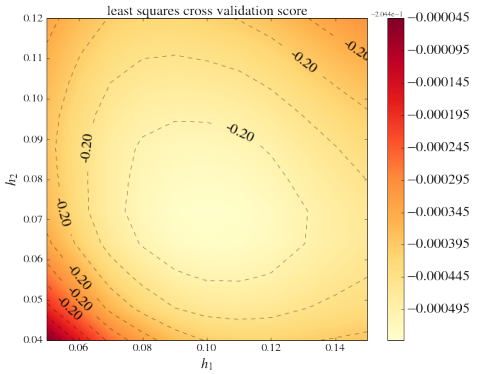


(d) Marginal density in Z_2.

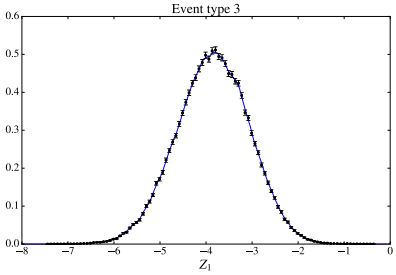
Figure 9.3: Kernel density estimates for the $D^*\tau$ event type.



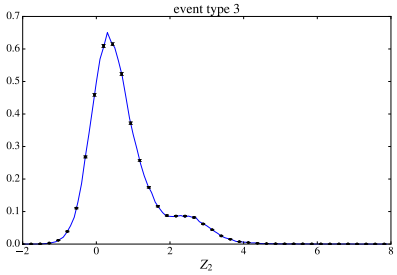
(a) Estimated 2D density.



(b) Least square cross validation scores.

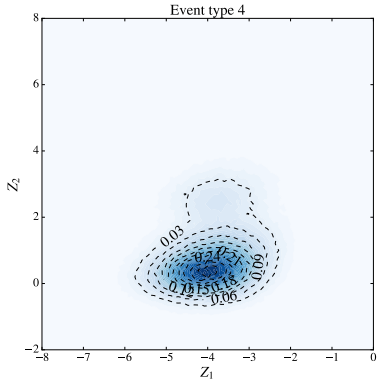


(c) Marginal density in Z_1.

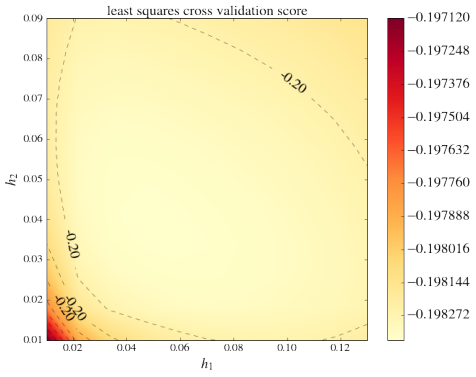


(d) Marginal density in Z_2.

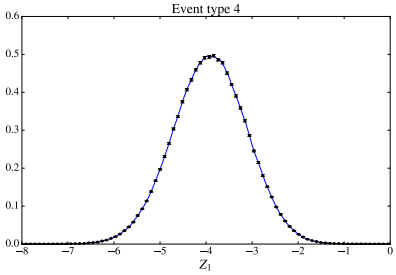
Figure 9.4: Kernel density estimates for the $D^{**}SL$ event type.



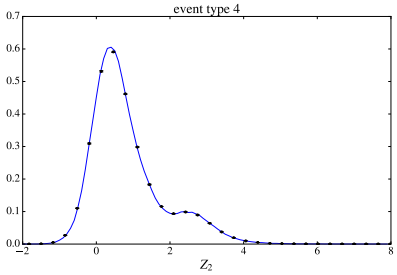
(a) Estimated 2D density.



(b) Least square cross validation scores.

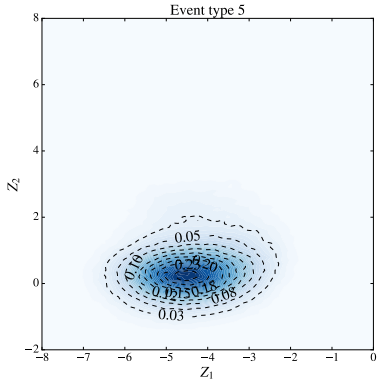


(c) Marginal density in Z_1.

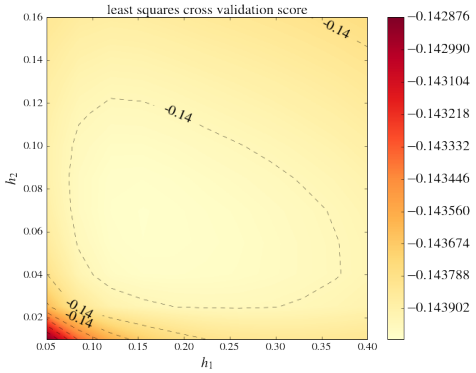


(d) Marginal density in Z_2.

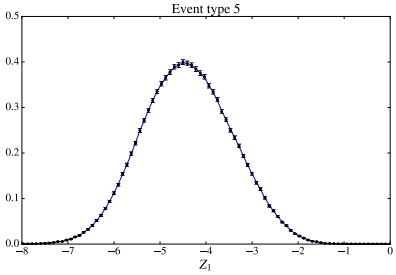
Figure 9.5: Kernel density estimates for the *Comb* event type.



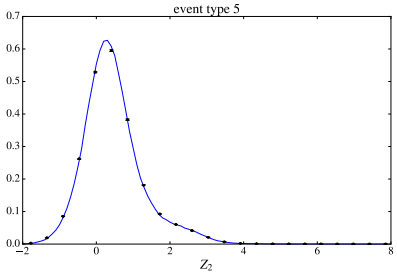
(a) Estimated 2D density.



(b) Least square cross validation scores.



(c) Marginal density in Z_1.



(d) Marginal density in Z_2.

Figure 9.6: Kernel density estimates for the *Cont* event type.

While the result from the MC test set will be discussed in Chapter 10, in this section we describe a method to correct for the biases in the extracted proportions.

It is helpful to distinguish the three ways simulated data is used in this analysis:

1. **Train**: A set of events used to build the conditional KDE for each event type.
2. **Tuning**: A set of events used to estimate the bias due to the choice of bandwidths of each event type density in construction of KDE's.
3. **Test**: A set of events that serves as a data emulation set. The KDE's will be used as input f_j's to extract the event type proportions of this data set.

The *bias* mentioned above can be explicitly stated as the difference between the extracted proportions of the test set and the true proportions in which the test set was generated.

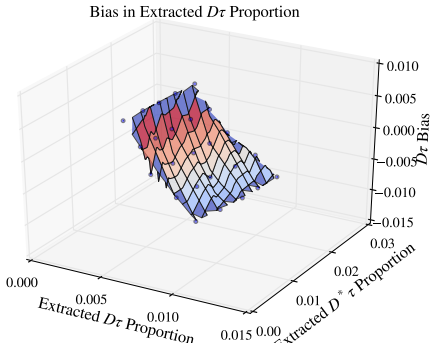
Mechanically, the bias correction is implemented in the form of a look-up table. There have been other methods considered, but we found them to introduce complexity to the analysis that were deemed to outweigh the benefits.

The bias (lookup-)table is mapped out by 25 test points. The value of each test point, or the bias at a given test proportion, is determined as follows:

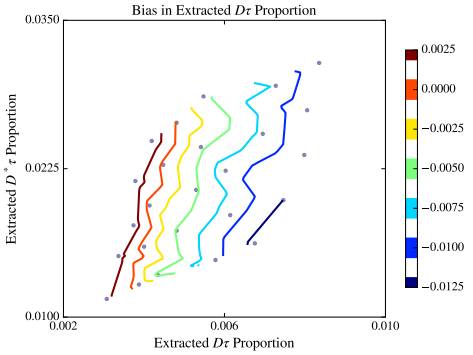
1. Generate 40 samples based on a given proportion from the MC. This is done by throwing a weighted die that chooses a random event from the MC with replacement.
2. Extract the signal proportions from the 40 test sets using the (fixed) component densities.
3. $bias = \text{avg}(extracted - true)$; the statistical uncertainty on the bias is estimated as the standard deviation among the 40 results.

The resulting bias tables in 3D and contour plots are shown in Figure 9.7.

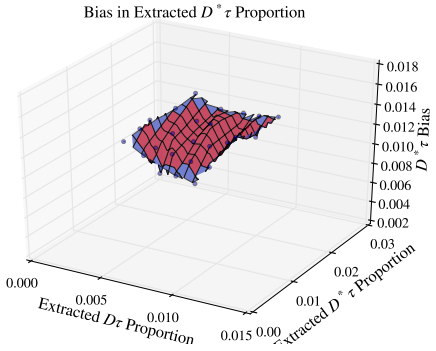
It is worth noting that the only sources of this bias seem to be the choice of the test proportions and the choice of bandwidths, as both training and test data are sampled from a single simulated data set. The look-up table as described above assumes that the biases only depend on the differences of signal proportions; the effects of the differences between the background proportions are explored as a systematic uncertainty in Chapter 11.



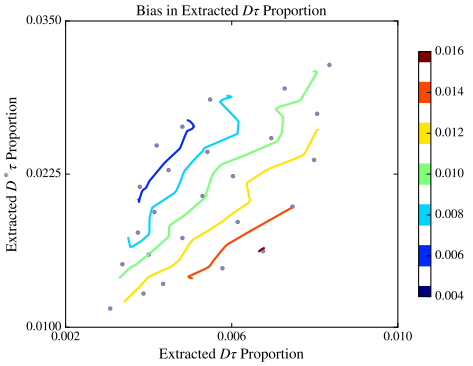
(a) Bias in extracted $D\tau$ proportion.



(b) Contour plot of bias in extracted $D\tau$ proportion.



(c) Bias in extracted $D^*\tau$ proportion.



(d) Contour plot of bias in extracted $D^*\tau$ proportion.

Figure 9.7: Biases of the signal proportions as a function of the extracted signal proportions. The points are linearly interpolated to better showcase the overall shape.

Chapter 10

Solving for $\hat{p}_j$ and $\hat{\epsilon}_j$ on Simulated Data

In this chapter we validate our analysis strategy on a test set reserved from the generic MC. This allows us to fine tune and debug any part of the analysis without biasing ourselves from the answers from the detector data. If the results on the MC appear to be consistent with our expectations, then we can simply swap the test sets from MC to detector data for our final answer.

10.1 Solving for $\hat{p}_j$

To obtain the central value of the sample proportion, we simply solve the optimization problem once. We calculate the standard error of the estimate from solving the same optimization problem on 1000 bootstrap samples.

The distributions of the bootstrap results are shown in figure 10.2, from which we extract the statistical uncertainties of the $\hat{p}_{D\tau}$ and $\hat{p}_{D^*\tau}$ as $\boxed{0.0011}$ and $\boxed{0.0013}$, respectively.

Given that the test set is MC, we know exactly the proportions of each event type. If we compare our extracted proportions with the true proportions, we see that they do not agree. This is expected, since maximum likelihood estimates are consistent but can be biased. Indeed, we can attribute the biases of the extracted signal proportions to the biases of the KDE's learned from MC. More specifically, the minimum MISE criteria used for the bandwidth selection minimizes the sum of both the bias and the variance. As we will see in Chapter 11, the differences between the extracted and the true proportions are exactly the biases we need to correct when extracting signal proportions from the detector data. Figure 10.1 shows the result of this trivial bias correction using the look-up table described in Section 9.3.

Figure 10.3 shows the marginal distributions when the extracted values of the $\hat{p}_j$'s are used to stack the five component densities. Visual inspection shows that the fitting procedure has been completed successfully. We also do not observe any significant differences between the densities when stacked with the extracted proportions versus the true proportions.

10.2 Solving for $\hat{\epsilon}_j$

For each signal category j, we estimate the efficiency as follows:

$$\hat{\epsilon}_j = \frac{\sum_{i=1}^{N^{(j)}} w_i}{\sum_{i=1}^{M^{(j)}} w_i}, \tag{10.1}$$

where

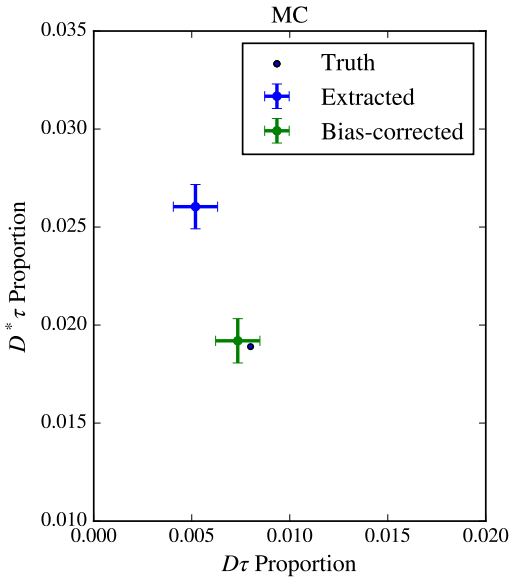
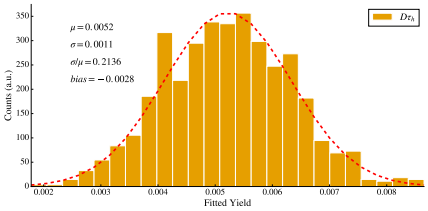


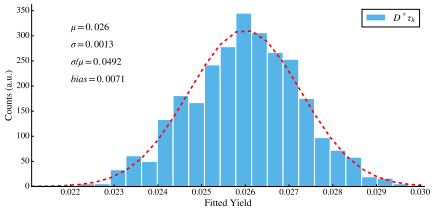
Figure 10.1: Result of applying bias correction to extracted signal proportions of the MC test set.

- $N^{(j)}$: number of events of event type j that is in the sample that has passed the data filter $\mathcal{F}$.
- $M^{(j)}$: number of events of event type j that was generated in the simulation.
- w_i: the weight of the event based on the luminosity and the cross section.

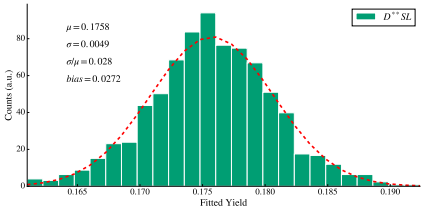
Applying the formula above, we obtain efficiencies of $\boxed{0.18\%}$ and $\boxed{0.23\%}$, for the $D\tau$ and $D^*\tau$ categories, respectively. The statistical errors are negligible.



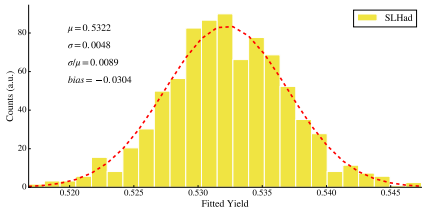
(a) $D\tau$.



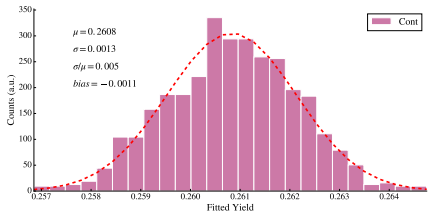
(b) $D^*\tau$.



(c) $D^{**}SL$.

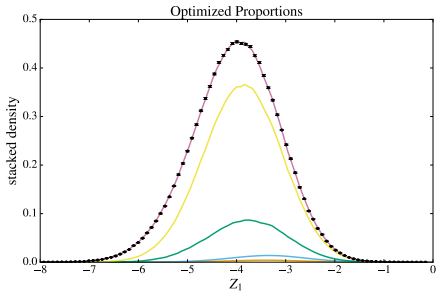


(d) $Comb$.

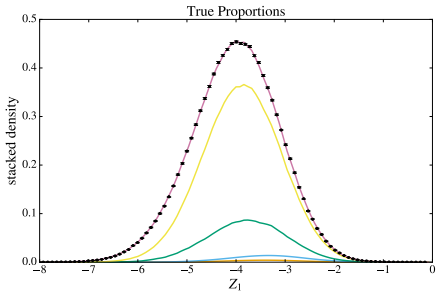


(e) $Cont$.

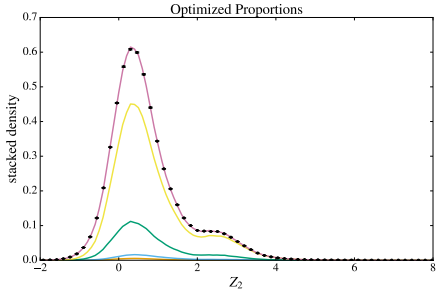
Figure 10.2: Bootstrapped results for all components.



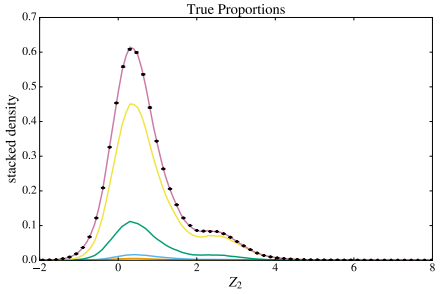
(a) Z_1 stacked with extracted proportions.



(b) Z_1 stacked with true proportions.



(c) Z_2 stacked with extracted proportions.



(d) Z_2 stacked with true proportions.

Figure 10.3: Stacked event type densities based on the extracted and true proportions. Black dots are histograms of the MC test set.

Chapter 11

Systematic Uncertainties

In Chapter 7, we observed slight discrepancies between the simulation and the detector data. This is very much expected, and in fact it would have been a bit strange to see the simulation agree exactly with the data.

The purpose of this chapter is to quantify the effects of these differences on the extracted signal proportions and signal efficiencies, and in turn on the measured $\mathcal{R}(D^{(*)})$.

We proceed by enumerating all possible ways that we know the simulation can be different from nature. This certainly is not an exhaustive list, but the goal is to have enough coverage over all major aspects of the deficiencies from the knowledge that have been accumulated over many years of experimental particle physics.

The concept of quantifying such uncertainties is simple: given a baseline model assumption, we change a single parameter of the model while keeping all others fixed. The proportions and efficiencies estimated using the new model can then be used to estimate the systematic uncertainty associated with that particular parameter.

A summary of the systematic uncertainties is shown at the end of this chapter in Table 11.13.

11.1 Bias correction

One common theme throughout this chapter is the comparison between extracted proportions of datasets generated from different model assumptions. We need to be careful on how the comparison is performed as there are different sources of bias that exist in the extracted proportions.

Our analysis so far has distinguished two types of datasets: training and test. The *training* data is simulated (i.e. Monte Carlo), and is used to construct the KDE's. The *test* data, which could be simulated or real, is the dataset whose signal proportions we would like to estimate.

The extracted proportions can be written as:

$$\mathbb{E}[\hat{p}_{i,j,h_0}] = p_j + b_{i,j,h_0}, \tag{11.1}$$

where p_j is the true signal proportion of the test set j, h_0 is the set of bandwidths used in construction of the KDE's, and b_{i,j,h_0} is the overall bias term.

Now we make the assumption that b_{i,j,h_0} can be decomposed into two parts:

$$b_{i,j,h_0} = b_{\text{KDE}}(i, h_0) + b_{\text{sys}}(i, j) + b_{\text{other}}, \tag{11.2}$$

where the $b_{\text{KDE}}(i, h_0)$ is the bias of the KDE due to our choice of the bandwidth, and $b_{\text{sys}}(i, j)$ is the contribution to the bias from the fact that model i and j are different. b_{other} describes the possibilities of other biases that might exist that cannot be attributed to the other bias terms. It is clear that the purpose of assessing systematic uncertainties is

estimating $b_{\text{sys}}(i,j)$ for various models i and $j \in \{\texttt{mc.central}, \text{detector data}\}$, then quoting the systematic uncertainty as

$$\Delta b_{\text{sys},i} = b_{\text{sys}}(i,j) - b_{\text{sys}}(\texttt{mc.central}, j)., \tag{11.3}$$

where `mc.central` refers to the default simulation model.

The expressions above imply that if we were to naively compare the extracted signal proportions from two choices of training data, i and i', then we are not accurately estimating Δb_{sys}. More specifically, we need to estimate the contribution from the b_{KDE} to the overall bias.

Consider estimating the signal proportions of dataset of model i using KDE's constructed using data points from the same model, i.e. the tuning set. We can say

$$\mathbb{E}[\hat{p}_{i,i,h_0}] = p_i + b_{\text{KDE}}(i, h_0) \tag{11.4}$$

since $b_{\text{sys}}(i,i) = 0$.

Since p_i is known, we can estimate the KDE bias as

$$\hat{b}_{\text{KDE}}(i, h_0) = \mathbb{E}[\hat{p}_{i,i,h_0}] - p_i, \tag{11.5}$$

and thus we can estimate the systematic uncertainty as defined in (11.3) as

$$\begin{aligned}
\Delta\hat{b}_{\text{sys}} &= (\mathbb{E}[\hat{p}_{i,j,h_0}] - b_{\text{KDE}}(i,h_0)) \\
&\quad - (\mathbb{E}[\hat{p}_{\texttt{mc.central},j,h_0}] - b_{\text{KDE}}(\texttt{mc.central}, h_0)) \\
&= \mathbb{E}[\hat{p}_{i,j,h_0}] - \mathbb{E}[\hat{p}_{\texttt{mc.central},j,h_0}] \\
&\quad - (\mathbb{E}[\hat{p}_{i,i,h_0}] - \mathbb{E}[\hat{p}_{\texttt{mc.central},\texttt{mc.central},h_0}]) \\
&\quad + (p_i - p_{\texttt{mc.central}}),
\end{aligned} \tag{11.6}$$

which reduces to 0 when $i = \texttt{mc.central}$.

In the following sections, when we consider extracted signal proportions, the reader can assume them to be *bias-corrected* proportions, as the procedure to perform the bias correction is well-determined.

11.2 Form factors

One of the main uncertainties in the model used as the baseline simulation (i.e. `mc.central`) stems from the form factors employed in the description of B decays. These form factors have previously been introduced in the calculation of the Standard Model prediction of $\mathcal{R}(D^{(*)})$ in Chapter 2. The models used in the generation of the simulated data in the *BABAR* framework are often outdated; either better models have been discovered or the parameters used in the models have been measured more precisely. Thus we first re-weight the initial MC data to transform it into the default baseline model.

In the rest of the section, we first describe the models used in the *BABAR* framework (i.e. the factory setting). Then, we show how those models were changed for generating the baseline simulation which we used to extract the central value. Lastly, we enumerate the various adjustments to the model that quantify theoretical and experimental uncertainties of the form factors.

It is worth noting that the form factors contribute to the differential decay rate of $\overline{B} \to D^{(*)}\tau\overline{\nu}_\ell$ through the angular distributions. This means that form factors influence the shapes of the component densities f_j and consequently $\hat{p}_j$. On the other hand, their effect on $\hat{\epsilon}_j$ falls under the detector effects systematic uncertainties, which will be treated separately in a later section.

The technique we use to fluctuate the form factor values is re-weighting. Given the baseline model i and the fluctuated model i', the recipe to calculate the re-weighting factor is as follows:

1. Define $\Gamma_i = \int d\Gamma_i/dx\, dx$ as the decay rate for model i, where x represents a set kinematic variables that the decay rate depends on (e.g. q^2). Similarly define $\Gamma_{i'}$ for model i'.

2. Given an event and its associated x, the weight is

$$w(x) = \frac{\frac{1}{\Gamma_{i'}}\frac{d\Gamma_{i'}}{dx}(x)}{\frac{1}{\Gamma_i}\frac{d\Gamma_i}{dx}(x)}. \tag{11.7}$$

3. Apply the weight to all events in the simulated dataset and recalculate the component densities f_j and extracted new proportions $\hat{p}_j^{i'}$.

11.2.1 Uncertainties due to $\overline{B} \to D^{(*)}\ell\overline{\nu}_\ell$ form factors

The kinematic variables of interest for the $\overline{B} \to D^{(*)}\ell\overline{\nu}_\ell$ decay rates are:

- $q^2 \equiv (p_B - p_M)^2$: Momentum transfer from the B meson to the virtual W boson.
- θ_ℓ: Angle of the ℓ in the W^* rest frame, see Figure 2.2.
- θ_V: Angle of the D in the D^*'s rest frame.
- χ: Angle between the decay planes of the W^* and the D^*.

We have previously derived the expressions for the relevant differential decay rates. We also have calculated the explicit branching fraction predictions based on a form factor model called CLN (see Chapter 2). There exist other form factor models, and in fact, the simulated data used in the analysis was generated using the ISGW2 and linear q^2 model for the case when $M = D$ and D^*, respectively. The exact settings and the values of the parameters are:

- Settings used to generate the simulated data (factory setting):
  - $B \to D\ell\nu$, $\ell = e, \mu, \tau$: ISGW2[30].
  - $B \to D^*\ell\nu$, $\ell = e, \mu$: Linear q^2[31]. The parameter settings are
    - $\rho^2 = 0.77$
    - $R_1 = 1.33$
    - $R_2 = 0.92$
  - $B \to D^*\tau\nu$: ISGW2.
- Settings used to obtain the central value (baseline setting):
  - $B \to D\ell\nu$, $\ell = e, \mu, \tau$: CLN[13]. The parameter settings are
    - $\rho^2 = 1.186$
    - $V_1 = 1.0816$
    - $\Delta = 1.0$
  - $B \to D^*\ell\nu$, $\ell = e, \mu, \tau$: CLN[13]. The parameter settings are
    - $F_1 = 0.921$
    - $\rho^2 = 1.207$
    - $R_0 = 1.14$
    - $R_1 = 1.401$
    - $R_2 = 0.854$
- Settings used to obtain the systematic uncertainty:

- $B \to D\ell\nu$, $\ell = e, \mu, \tau$: CLN[13]. The parameter settings are
  - $\rho^2 = 1.186 \pm 0.054$
  - $V_1 = 1.0816$
  - $\Delta = 1.0$
- $B \to D^*\ell\nu$, $\ell = e, \mu, \tau$: CLN[13]. The parameter settings are[1]
  - $F_1 = 0.921$
  - $\rho^2 = 1.207 \pm 0.026$
  - $R_0 = 1.14$
  - $R_1 = 1.401 \pm 0.033$
  - $R_2 = 0.854 \pm 0.02$

We can visualize the effects of the various form factor model and parameter assumptions by plotting the q^2 spectra of the differential decay rates. In Figures 11.1 and 11.2, we show the spectra predicted by the factory setting and the resulting simulated data. The slight differences in the low q^2 region for the light lepton cases can be attributed to the fact that the ISGW2 model does not take the widths of the B and D resonances into account. If we fix the respective masses to specific values, the differences disappear.

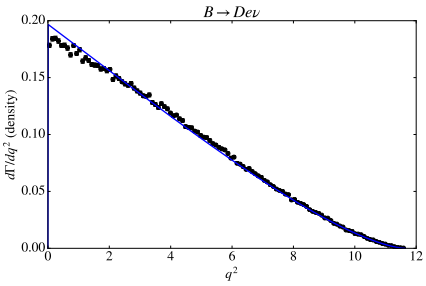
In Figures 11.3 and 11.4, we compare the spectra of the factory setting and the baseline setting. In Figures 11.5 and 11.6, we show the spectra for the variations of the $B \to D\ell\nu$ and $B \to D^*\ell\nu$ CLN parameters, respectively. The effects of the slight change in the shapes are assessed as the form factor systematics.

To obtain the systematic uncertainty due to these form factors, we extract the signal proportions using the data that we re-weight to the systematic settings above. The difference between the values obtained this way and the central values are then listed as the systematic. The results are shown in table 11.1 for both MC and detector data test sets.

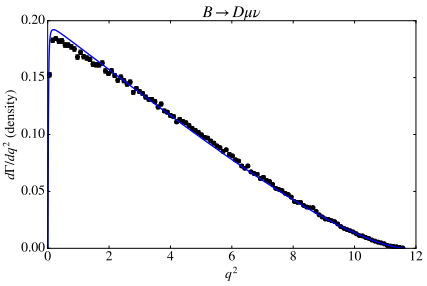
Signal Type	Simulated data	Detector data
$D\tau$	0.0003	0.0005
$D^*\tau$	0.0021	0.0013

Table 11.1: Systematic uncertainties on $\hat{p}_{sig}$ due to $B \to D^{(*)}$ form factors. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

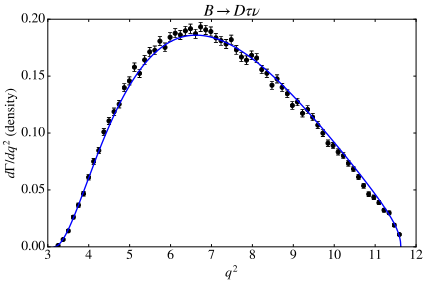
[1] Note that the variations of the $B \to D^*\ell\nu$ form factors were along the principal axis from taking the correlations into account.



(a) $B \to De\nu_e$

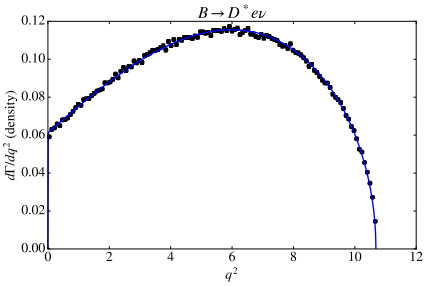


(b) $B \to D\mu\nu_\mu$

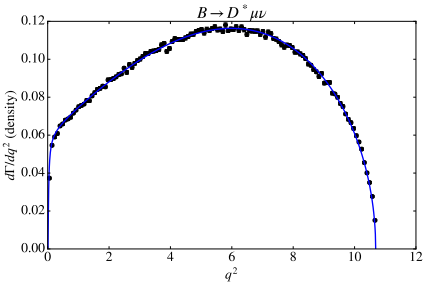


(c) $B \to D\tau\nu_\tau$

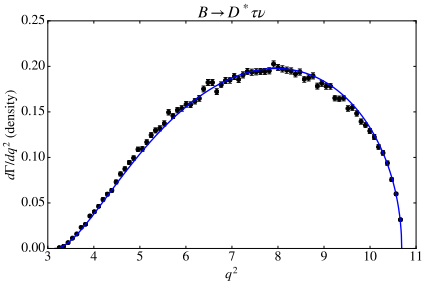
Figure 11.1: The default generated simulation data overlaid with the purported form factor model for $B \to D\ell\nu_\ell$.



(a) $B \to D^* e\nu_e$

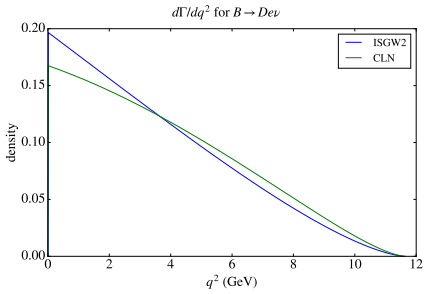


(b) $B \to D^* \mu\nu_\mu$

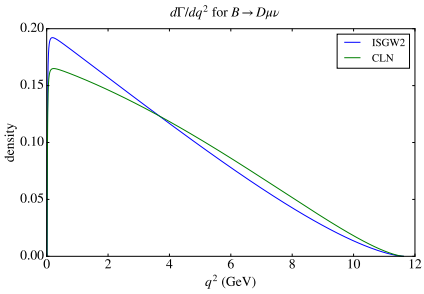


(c) $B \to D^* \tau\nu_\tau$

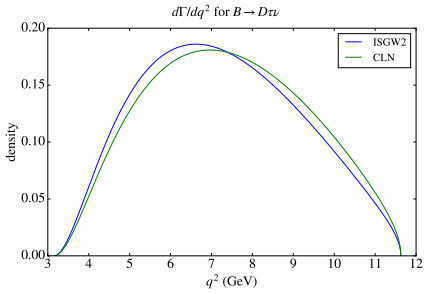
Figure 11.2: The default generated simulation data overlaid with the purported form factor model for $B \to D^*\ell\nu_\ell$.



(a) $B \to De\nu_e$

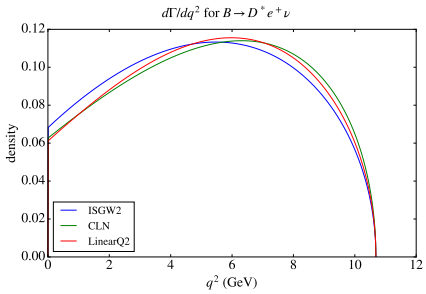


(b) $B \to D\mu\nu_\mu$

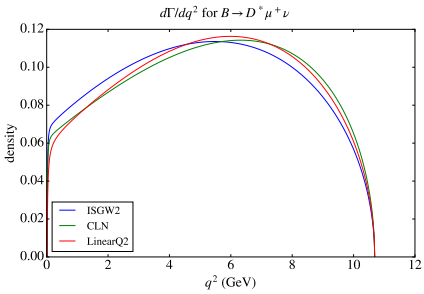


(c) $B \to D\tau\nu_\tau$

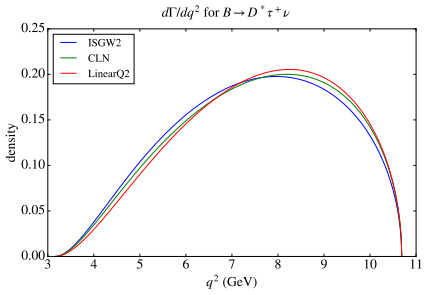
Figure 11.3: The differential decay rates for the $B \to D$ form factors considered in this analysis.



(a) $B \to D^* e\nu_e$

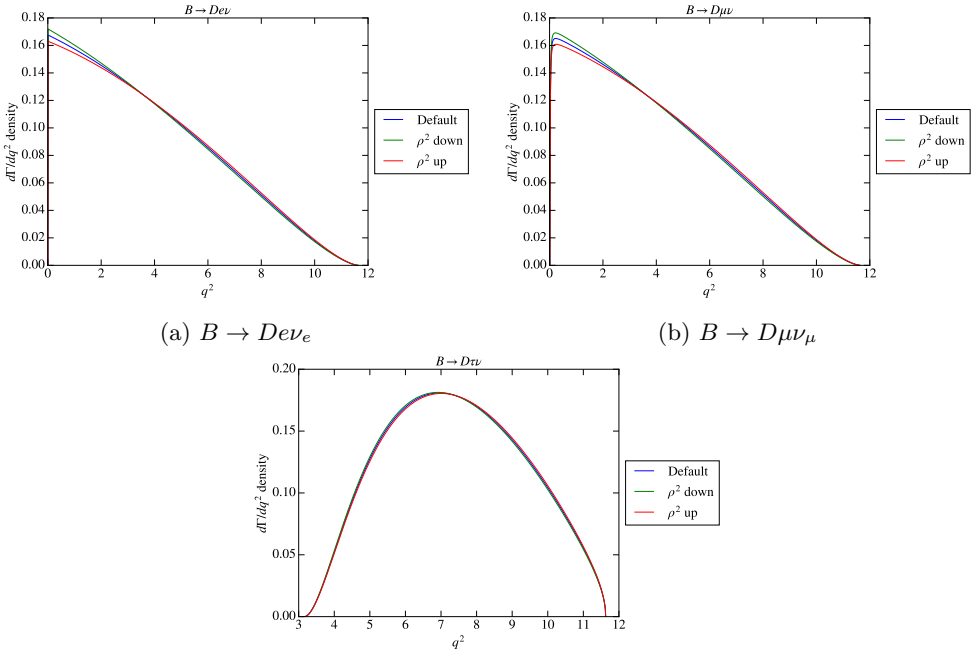


(b) $B \to D^* \mu\nu_\mu$



(c) $B \to D^* \tau\nu_\tau$

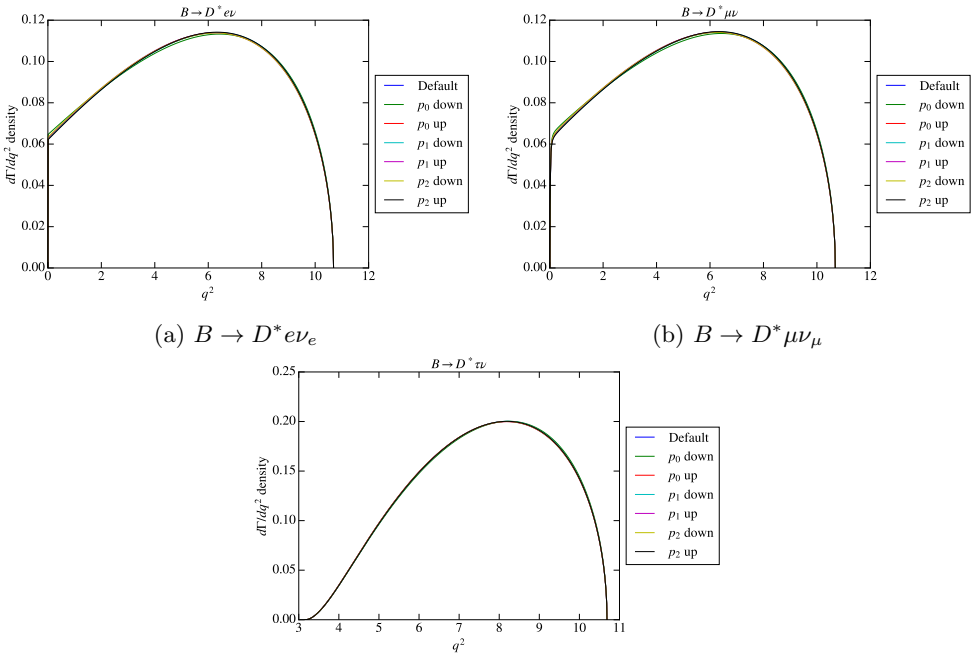
Figure 11.4: The differential decay rates for the $B \to D^*$ form factors considered in this analysis.



(a) $B \to De\nu_e$ (b) $B \to D\mu\nu_\mu$

(c) $B \to D\tau\nu_\tau$

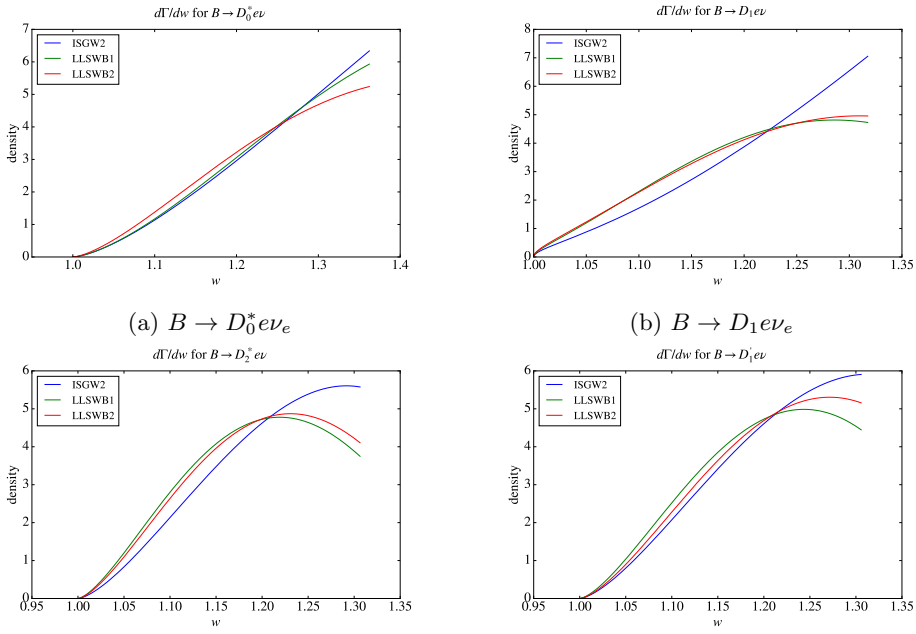
Figure 11.5: The differential decay rates for the variations of $B \to D$ form factors considered in this analysis.



(a) $B \to D^* e\nu_e$ (b) $B \to D^* \mu\nu_\mu$

(c) $B \to D^* \tau\nu_\tau$

Figure 11.6: The differential decay rates for the variations of $B \to D^*$ form factors considered in this analysis. p_i, where $i = 0, 1, 2$, refer to the principal axes of the covariance matrix for (ρ^2, R_1, R_2) along which the variations are performed.



(a) $B \to D_0^* e\nu_e$ (b) $B \to D_1 e\nu_e$

(c) $B \to D_2^* e\nu_e$ (d) $B \to D_1' e\nu_e$

Figure 11.7: The differential decay rate spectra for the $B \to D^{**}$ form factors considered in this analysis.

11.2.2 Uncertainties due to $B \to D^{**}\ell\nu_\ell$ form factors

We now quantify the systematic uncertainties due to the form factors describing semileptonic B decays involving D^{**}. D^{**} denotes excitations of the ground state D meson, which for our purposes refer to D_0^*, D_1, D_1', and D_2^*.

In this case, the relevant kinematic variables x are:

- $w = v \cdot v'$: $v(v')$ is the four velocity of the $B(D^{**})$.
- θ_ℓ: Angle of the ℓ in W^*'s rest frame.

As with the $D^{(*)}$ case, the form factor settings are (following BAD 1586):

- Factory setting:
  - $B \to D^{**}\ell\nu$, $\ell = e, \mu, \tau$: ISGW2[30].
- Baseline setting:
  - $B \to D^{**}\ell\nu$, $\ell = e, \mu, \tau$: LLSW B1 [32].
- Systematics setting:
  - $B \to D^{**}\ell\nu$, $\ell = e, \mu, \tau$: LLSW B2 [32].

Figures 11.7 show the q^2 spectra of the form factor models listed above.
The resulting systematic uncertainty estimates are shown in Table 11.2.

Signal Type	Simulated data	Detector data
$D\tau$	0.00001	0.0002
$D^*\tau$	0.00003	0.00002

Table 11.2: Systematic uncertainties on $\hat{p}_{sig}$ due to $B \to D^{**}$ form factors. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

11.3 Branching fractions

Branching fractions of various decays that occur in the detector are another set of model parameters that can cause mismatch between the MC and the data. More specifically, they influence the results in two ways:

1. The change in the relative abundance of the decays changes the learned $\hat{f}_j$, and in turn the $\hat{p}_j$.

2. Different decay modes generally have different efficiencies of being detected and passing the data filter. Thus the change of the relative compositions affects the overall event type efficiency $\hat{\epsilon}_j$.

The branching fraction factory setting was constructed using the best knowledge at the time of creation. Since then, the measurements of the branching fractions have improved considerably and we must incorporate the new knowledge into the simulation. In other words, we update the central values of the branching fractions and then, as we did with the form factors, vary the value according to the latest measurement uncertainties.

Since enumerating *all* possible decays is impractical, we proceed with the following approach of choosing the important decay modes:

1. B decays: rank the frequencies of B decay modes that occur in a sample of 10000 events. The B decays of interest will be the top k most frequent decays that comprises of 50% of all decays in our sample.

2. D decays: since the effects of D decay mode branching fractions are secondary to those of the B decays. We only perform a cross-check of all D decays based on the representative mode $D \to K\pi\pi$.

Given the modes of interest, we proceed to derive the re-weighting factor as follows:

1. Traverse the truth decay graph and check for modes of interest.

2. For each mode found, assign a weight $w = w_{\text{WA}}/w_{\texttt{DECAY.DEC}}$, where WA refers to the world average value and `DECAY.DEC` refers to the factory setting[2].

3. The correction for the event is the product of all such computed weights.

In addition, we correct for the asymmetry in production rate of B^+ vs. B^0 from decays of $\Upsilon(4S)$ based on the latest value from HFLAV by simply re-weighting the luminosity weights, which were initially calculated assuming equal production rate.

Let ω_+ be the initial luminosity weight of a $B^\pm$ event, which is defined as $\omega_+ = \mathcal{L}_{\text{data}}/\mathcal{L}_{\text{MC,B+}}$, where $\mathcal{L}$ refers to luminosity. Since initially $\mathcal{L}_{\text{MC,B+}} = 0.5 \times \mathcal{L}_{\text{MC,B}}$, where B refers to both charged and neutral B's, it is clear that the corrected luminosity weight for charged and neutral B events should be:

$$\omega'_+ = \frac{\omega_+}{2 \times 0.513} \tag{11.8}$$

[2] `DECAY.DEC` is a file used in EvtGen to store all model parameters

and

$$\omega_0' = \frac{\omega_0}{2 \times 0.487}, \tag{11.9}$$

where the production rate of neutral B's used is 0.487 ± 0.006[3].

The central values are obtained by applying the correction factor to all simulated events, then proceeding in the usual manner to obtain $\hat{p}_{sig}$ and $\hat{\epsilon}_{sig}$.

To quantify systematic uncertainties due to variations in the estimated densities, we proceed as follows:

1. Decide on a set of variations on the branching fractions, typically based on the standard deviation of the world average accounting for correlations when necessary.
2. For each event, re-compute the weights due to these variations and re-estimate the event type densities.
3. Use these densities to extract the result and quote its difference from the central value as a systematic uncertainty.

To quantify systematic uncertainties due to variations in the signal efficiencies, we proceed as follows:

1. Decide on a set of variations on the branching fractions.
2. For each event, re-compute the weights due to these variations.
3. Compute the efficiency as usual, this time applying the new weights. Quote the difference between this and the central value as a systematic uncertainty.

We now show the results of the systematic uncertainties due to branching fractions.

11.3.1 Uncertainties due to varying B branching fractions

The B decay modes of interests are grouped into five classes. The decays within a class have similar underlying physics process, allowing us to vary the branching fractions of the entire class at once rather than one at a time.

The five classes are as follows:

- Group A: semileptonic B decays.
- Group B: B decays to strange mesons.
- Group C: $B \to D\rho$.
- Group D: $B \to D^* a$.
- Group E: $B \to D^{**}\ell\nu$ and non-resonant $B \to D\ell\nu\pi$.

The branching fraction variations we consider are listed in Tables 11.3 and 11.4.

[3]It should be noted that there is indeed an uncertainty on the production rate, which provides another source of systematic uncertainty on the final result. Nevertheless, since the uncertainty is quite small we do not include it in our final set of systematics.

Decay Mode	DECAY.DEC Value	Corrected Value	Correction Uncertainty	Group
$B^+ \to \overline{D}^{*0}\mu^+\nu_\mu$	0.0617	0.0531	0.0012	A[4]
$B^+ \to \overline{D}^{*0}e^+\nu_e$	0.0617	0.0531	0.0012	A
$B^+ \to \overline{D}^{0}\mu^+\nu_\mu$	0.0224	0.0230	0.0010	A
$B^+ \to \overline{D}^{0}e^+\nu_e$	0.0224	0.0230	0.0010	A
$B^+ \to D_s^{*+}D^{*0}$	0.0278	0.0171	0.0024	B
$B^+ \to D_s^{+}D^{0}$	0.0129	0.009	0.0009	B
$B^+ \to D_s^{+}D^{*0}$	0.0124	0.0082	0.0017	B
$B^+ \to D^0\rho^+$	0.0134	0.0134	0.0018	C
$B^+ \to \overline{D}^{*0}a_1^+$	0.01597	0.019	0.005	D
$B^+ \to \overline{D}_1^0 e^+\nu_e$	0.0056	0.0096	0.001	E
$B^+ \to \overline{D}_0^{*0} e^+\nu_e$	0.0049	0.0044	0.0008	E
$B^+ \to \overline{D}_2^{*0} e^+\nu_e$	0.003	0.003	0.0004	E
$B^+ \to \overline{D}_1^{'0} e^+\nu_e$	0.009	0.002	0.0005	E
$B^+ \to D^{*-}\pi^+e^+\nu_e$	0.0006	0.006	0.0006	E
$B^+ \to D^{-}\pi^+e^+\nu_e$	0.0019	0.0042	0.0006	E
$B^+ \to \overline{D}_1^0 \mu^+\nu_\mu$	0.0056	0.0096	0.001	E
$B^+ \to \overline{D}_0^{*0} \mu^+\nu_\mu$	0.0049	0.0044	0.0008	E
$B^+ \to \overline{D}_2^{*0} \mu^+\nu_\mu$	0.003	0.003	0.0004	E
$B^+ \to \overline{D}_1^{'0} \mu^+\nu_\mu$	0.009	0.002	0.0005	E
$B^+ \to D^{*-}\pi^+\mu^+\nu_\mu$	0.0006	0.006	0.0006	E
$B^+ \to D^{-}\pi^+\mu^+\nu_\mu$	0.0019	0.0042	0.0005	E
$B^+ \to \overline{D}_1^0 \tau^+\nu_\tau$	0.0013	0.001	0.00014	E
$B^+ \to \overline{D}_0^{*0} \tau^+\nu_\tau$	0.0013	0.0004	0.00015	E
$B^+ \to \overline{D}_1^{'0} \tau^+\nu_\tau$	0.002	0.00012	0.00005	E
$B^+ \to \overline{D}_2^{*0} \tau^+\nu_\tau$	0.002	0.00021	0.00004	E

Table 11.3: Dominant B^+ decay modes of interest.

Uncertainties in $\hat{p}_j$ due to variations in $\hat{f}_j$

The resulting systematic uncertainties are listed in Table 11.5.

Uncertainties in $\hat{\epsilon}_j$

The resulting systematic uncertainties are listed in Table 11.6.

11.3.2 Uncertainties due to difference between exclusive and inclusive branching fractions of $B \to X_c\ell\nu$

It is well known [33] that there is a discrepancy of $\sim 1.5\%$ between the inclusive branching fraction of semileptonic B decays and the sum of exclusive branching fractions. This motivated a previous *BABAR* analysis [34] to "fill" the gap with B decays to $D^{**}\,(D\pi\pi)\,\ell\nu$.

To assess the uncertainty due to the discrepancy, the $D^{**}\,(D\pi\pi)\,\ell\nu$ signal MC samples generated for the previous analysis are first re-weighted to make up 1.5% of the training set *prior* to any reconstruction and data filtering has been applied. Then the records go through the same pipeline as all the other events and incorporated into the final training set, which are again re-weighted to keep the total number of training data points the same. The samples used are summarized in Table 11.7, where we mix the three D^{**} decays modes in equal proportions.

[4] Group A is varied along the principal axis based on correlations used for the world average. Thus two variations are performed for this group, rather than just one for other groups.

Decay Mode	`DECAY.DEC` Value	Corrected Value	Correction Uncertainty	Group
$B^0 \to D^{*-}\mu^+\nu_\mu$	0.057	0.0493	0.0011	A
$B^0 \to D^{*-}e^+\nu_e$	0.057	0.0493	0.0011	A
$B^0 \to D^-\mu^+\nu_\mu$	0.0207	0.0213	0.0010	A
$B^0 \to D^-e^+\nu_e$	0.0207	0.0213	0.0010	A
$B^0 \to D_s^{*+}D^{*-}$	0.024	0.0177	0.0014	B
$B^0 \to D_s^{+}D^{*-}$	0.0126	0.008	0.0011	B
$B^0 \to D^{*-}D^{*0}K^+$	0.01	0.0106	0.0009	B
$B^0 \to D_s^{*+}D^{-}$	0.009	0.0074	0.0016	B
$B^0 \to D_s^{+}D^{-}$	0.009	0.0072	0.0008	B
$B^0 \to D^{-}D^{*0}K^+$	0.0049	0.0035	0.0004	B
$B^0 \to D^{*-}D^{*+}K^0$	0.007	0.0081	0.0007	B
$B^0 \to D_{s1}^{+}D^{-}$	0.0098	0.0005	0.00014	B
$B^0 \to D^+\rho^-$	0.0077	0.0078	0.0013	C
$B^0 \to D^{*+}a_1^-$	0.012	0.013	0.0027	D
$B^0 \to D_2^{*-}e^+\nu_e$	0.0023	0.0028	0.0004	E
$B^0 \to D_1^{'-}e^+\nu_e$	0.0083	0.0019	0.00046	E
$B^0 \to D_0^{*-}e^+\nu_e$	0.0045	0.00408	0.00074	E
$B^0 \to D_1^{-}e^+\nu_e$	0.0052	0.0089	0.000911	E
$B^0 \to \overline{D}^{*0}\pi^-e^+\nu_e$	0.0007	0.0048	0.0008	E
$B^0 \to \overline{D}^{0}\pi^-e^+\nu_e$	0.002	0.0042	0.0006	E
$B^0 \to D_1^{-}\mu^+\nu_\mu$	0.0052	0.0089	0.000911	E
$B^0 \to D_0^{*-}\mu^+\nu_\mu$	0.0045	0.00408	0.00074	E
$B^0 \to D_1^{'-}\mu^+\nu_\mu$	0.0083	0.0019	0.00046	E
$B^0 \to D_2^{*-}\mu^+\nu_\mu$	0.0023	0.0028	0.0004	E
$B^0 \to \overline{D}^{*0}\pi^-\mu^+\nu_\mu$	0.0007	0.0048	0.0008	E
$B^0 \to \overline{D}^{0}\pi^-\mu^+\nu_\mu$	0.002	0.0042	0.0006	E
$B^0 \to D_1^{-}\tau^+\nu_\tau$	0.0013	0.0009	0.00013	E
$B^0 \to D_0^{*-}\tau^+\nu_\tau$	0.0013	0.0003	0.00014	E
$B^0 \to D_1^{'-}\tau^+\nu_\tau$	0.002	0.00017	0.00005	E
$B^0 \to D_2^{*-}\tau^+\nu_\tau$	0.002	0.00013	0.00004	E

Table 11.4: Dominant B^0 decay modes of interest.

Signal Type	BF variation type	Simulated data	Detector data
$D\tau$	$A_0 + 1\sigma$	-0.00016	0.00021
$D^*\tau$	$A_0 + 1\sigma$	0.00011	0.00029
$D\tau$	$A_0 - 1\sigma$	0.00021	0.00042
$D^*\tau$	$A_0 - 1\sigma$	-0.00015	0.00032
$D\tau$	$A_1 + 1\sigma$	0.00019	0.00032
$D^*\tau$	$A_1 + 1\sigma$	-0.00085	-0.00035
$D\tau$	$A_1 - 1\sigma$	-0.00005	0.00027
$D^*\tau$	$A_1 - 1\sigma$	0.00065	-0.00033
$D\tau$	B $+1\sigma$	0.00008	-0.00014
$D^*\tau$	B $+1\sigma$	0.00004	0.00022
$D\tau$	B -1σ	-0.00001	0.00024
$D^*\tau$	B -1σ	-0.00011	-0.00076
$D\tau$	C $+1\sigma$	0.00010	-0.00004
$D^*\tau$	C $+1\sigma$	0.00009	0.00026
$D\tau$	C -1σ	-0.00002	0.00026
$D^*\tau$	C -1σ	-0.00020	0.00016
$D\tau$	D $+1\sigma$	0.00028	0.00037
$D^*\tau$	D $+1\sigma$	0.00065	0.00063
$D\tau$	D -1σ	-0.00015	-0.00031
$D^*\tau$	D -1σ	-0.00069	-0.00041
$D\tau$	E $+1\sigma$	-0.00022	-0.00026
$D^*\tau$	E $+1\sigma$	-0.00020	-0.00015
$D\tau$	E -1σ	0.00029	0.00007
$D^*\tau$	E -1σ	0.00018	0.00031

Table 11.5: Systematic uncertainties on $\hat{p}_j$ due to varying well- determined B decay branching fractions. The first column is the event type, the second column indicates the kind of branching fraction variation, the third column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data. Note that groups A_0 and A_1 roughly correspond to $D\ell\nu$ and $D^*\ell\nu$ branching fractions, respectively.

The new dataset with these decays added are then used to assess the change in the resulting $\hat{p}_j$ in the same procedure as the assessments of the B branching fraction systematic uncertainties: Table 11.8 shows the results.

Signal Type	BF variation type	Simulated data (10^{-5})	Detector data (10^{-5})
$D\tau$	$A_0 + 1\sigma$	0.1	same as simulation
$D^*\tau$	$A_0 + 1\sigma$	0.3	
$D\tau$	$A_0 - 1\sigma$	0.1	
$D^*\tau$	$A_0 - 1\sigma$	0.3	
$D\tau$	$A_1 + 1\sigma$	0.4	
$D^*\tau$	$A_1 + 1\sigma$	1.1	
$D\tau$	$A_1 - 1\sigma$	0.4	
$D^*\tau$	$A_1 - 1\sigma$	1.1	
$D\tau$	B $+1\sigma$	0.5	
$D^*\tau$	B $+1\sigma$	0.3	
$D\tau$	B -1σ	0.5	
$D^*\tau$	B -1σ	0.3	
$D\tau$	C $+1\sigma$	0.3	
$D^*\tau$	C $+1\sigma$	0.3	
$D\tau$	C -1σ	0.3	
$D^*\tau$	C -1σ	0.3	
$D\tau$	D $+1\sigma$	0.5	
$D^*\tau$	D $+1\sigma$	0.7	
$D\tau$	D -1σ	0.5	
$D^*\tau$	D -1σ	0.8	
$D\tau$	E $+1\sigma$	1.1	
$D^*\tau$	E $+1\sigma$	2.1	
$D\tau$	E -1σ	1.1	
$D^*\tau$	E -1σ	2.1	

Table 11.6: Systematic uncertainties in $\hat{\epsilon}_j$ due to varying poorly determined B decay branching fractions. The first column is the event type, the second column indicates the kind of branching fraction variation, the third column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

11.3.3 Uncertainties due to D branching fractions

The effects of the branching fractions of D decay modes are also studied. Rather than performing fluctuations on a set of dominant D decays, we simply perform a cross-check by making a correction on the most common decay $D \to K\pi\pi$.

More specifically, for each event we multiply a factor of $w = w_{\text{WA}}/w_{\texttt{DECAY.DEC}} = 0.992$ for every $D \to K\pi\pi$ decay. Since the branching fractions of D decays are well-measured, we deemed fluctuation by its measurement error unnecessary.

Table 11.9 shows the resulting systematic uncertainties on $\hat{p}_j$.

11.4 $B\overline{B}$ background validation

The form factor and branching fraction uncertainties affect the shapes of the $\hat{f}_j$ of *all* signal and background event types simultaneously. In many analyses, attempts are made to pin down the exact shapes of the background densities, which can then be fixed when fitting for the signal components. The key ingredient for such procedure is control samples. Control samples are *detector data* samples that are known to be a very pure sample of a particular type.

In this analysis, out of the three the background event types, one of them has such control sample: the off-peak detector data is a control sample for the *Cont* event type. In fact, as stated in Chapter 9, $\hat{f}_{Cont}$ is learned from the off-peak data sample.

Decay Type	# event [10^6]	*BABAR* Dataset Name
$B^+ \to D_1(D\pi\pi)\ell\nu$	6.642	SP-11459-R24
$B^0 \to D_1(D\pi\pi)\ell\nu$	7.100	SP-11465-R24
$B^+ \to D_1(D^*\pi\pi)\ell\nu$	6.480	SP-11460-R24
$B^0 \to D_1(D^*\pi\pi)\ell\nu$	6.870	SP-11466-R24
$B^+ \to D(2S)(D\pi\pi)\ell\nu$	6.776	SP-11461-R24
$B^0 \to D(2S)(D\pi\pi)\ell\nu$	6.826	SP-11467-R24
$B^+ \to D(2S)(D^*\pi\pi)\ell\nu$	6.530	SP-11462-R24
$B^0 \to D(2S)(D^*\pi\pi)\ell\nu$	6.769	SP-11468-R24
$B^+ \to D(2S)^*(D\pi\pi)\ell\nu$	6.369	SP-11463-R24
$B^0 \to D(2S)^*(D\pi\pi)\ell\nu$	6.552	SP-11469-R24
$B^+ \to D(2S)^*(D^*\pi\pi)\ell\nu$	6.425	SP-11464-R24
$B^0 \to D(2S)^*(D^*\pi\pi)\ell\nu$	6.616	SP-11470-R24

Table 11.7: Signal MC samples used for assessing the gap between inclusive and sum of exclusive $B \to X_c\ell\nu$ branching fractions.

Signal Type	Simulated data	Detector data
$D\tau$	0.00001	0.00010
$D^*\tau$	0.00017	0.00046

Table 11.8: Systematic uncertainties in $\hat{p}_j$ due to the discrepancy between inclusive and the sum of exclusive branching fractions of B decays to $X_c\ell\nu$. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

The other two background event types, which we will collectively denote as the $B\overline{B}$ background, has no direct control sample available from the experiment. We therefore attempt to bound the possible discrepancies in the $B\overline{B}$ background density shapes using a *sideband* control sample.

The idea is as follows: find a set of selection criteria that filters out most signal events, leaving us a detector data sample consisting of only background events. This detector sideband sample consists of three event types, one of which, the continuum, can be fixed as known. If we subtract out the continuum contribution using the off-peak data, we should be left with a $B\overline{B}$ background detector data. We can now compare the this sample with the sideband sample of the MC $B\overline{B}$ background to capture the discrepancy. More specifically, we can learn correction factors for the MC $B\overline{B}$ background events to make them seem more like the detector data. The learned correction factors is then used to calculate the systematic uncertainty.

The above procedure relies on the following assumptions:

A1. The continuum MC and the off-peak data are indistinguishable in distribution.

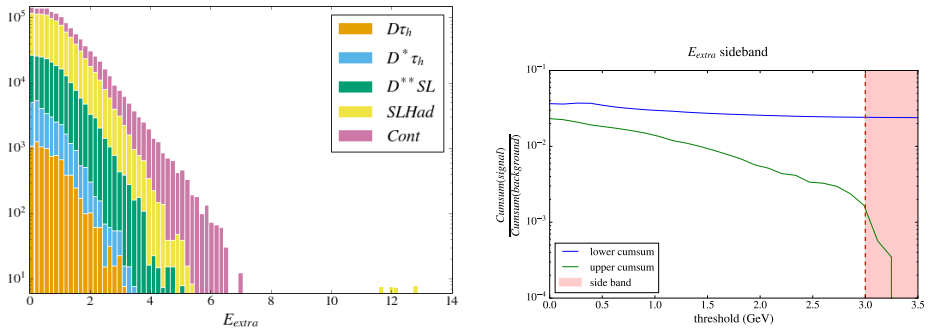
A2. The signal event types are negligible in the sideband.

A3. The discrepancy in Z_1 and Z_2 is the same independent of whether an event is in the sideband or not.

We have verified Assumption **A1.** by swapping out the continuum MC with the off-peak data in extraction of the central values, which showed no change. Assumption **A2.** will be shown to be valid when we define the sideband region and studying the signal contamination. Assumption **A3.** is the only one that we must take in faith, though the collective wisdom from previous *BABAR* analyses seems to vouch for its validity.

Signal Type	Simulated data	Detector data
$D\tau$	0.00007	0.00022
$D^*\tau$	0.00003	0.00021

Table 11.9: Systematic uncertainties in $\hat{p}_j$ due to corrections to branching fraction of $D \to K\pi\pi$. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.



(a) Stacked log counts.

(b) Cumulative sum ratios.

Figure 11.8: Sideband sample in E_{extra}.

11.4.1 Qualitative inspection of sideband sample

The criteria for an event being in the sideband are:

S1. $E_{extra} \geq 3$ GeV.

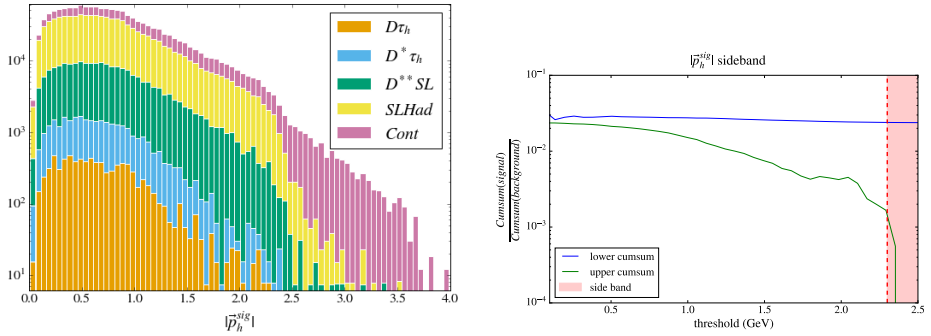
S2. $|\vec{p}_h^{sig}| \geq 2.3$ GeV.

Figures 11.8, and 11.9 show that this sample consists almost entirely of event types $B\overline{B}$ and continuum, verifying Assumption **A2.**.

Figure 11.10 shows the Z_1 and Z_2 comparisons for data points belonging to the sideband; the comparisons for all features are shown in Appendix C.

11.4.2 Method

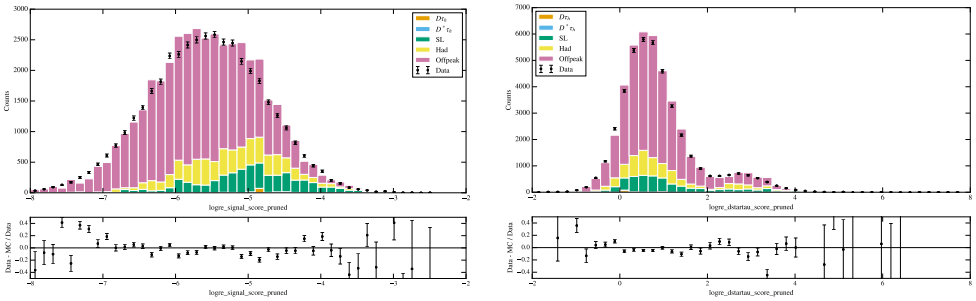
The correction factors for the $B\overline{B}$ background shapes are calculated as follows:



(a) Stacked log counts.

(b) Cumulative sum ratios.

Figure 11.9: Sideband sample in $|\vec{p}_h^{sig}|$.



(a) Sideband signal score (Z_1). (b) Sideband $D^*\tau$ score (Z_2).

Figure 11.10: Sideband comparison for Z_1 and Z_2. All data points used to produce these figures belong to the sideband sample. The black points are on-peak detector data, the pink filled histograms are luminosity-scaled off-peak data, and the yellow and green filled histograms are the simulated $B\overline{B}$ background.

1. Let $g(z_1, z_2)$ be the density function in Z_1 and Z_2 for the on-peak sideband. By assumption **A2.**, we can decompose this into two components: the continuum and the from $B\overline{B}$ background. We write this as:

$$g(z_1, z_2) = (1 - p_{cont})g_{B\overline{B}}(z_1, z_2) + p_{cont}g_{cont}(z_1, z_2). \tag{11.10}$$

2. Let $f_{B\overline{B}}(z_1, z_2)$ be the density function of the sideband MC $B\overline{B}$ background.

3. Then the correction factors are:

$$w(z_1, z_2) = \frac{g(z_1, z_2) - p_{cont}g_{cont}(z_1, z_2)}{(1 - p_{cont})f_{B\overline{B}}(z_1, z_2)}. \tag{11.11}$$

Each quantity in (11.11) can be estimated as follows:

- Estimate $g(z_1, z_2)$ using the on-peak sideband sample.
- By Assumptions **A1.** and **A3.**, estimate $g_{cont}(z_1, z_2)$ using the off-peak sample (includes sideband and non-sideband).
- Estimate $f_{B\overline{B}}(z_1, z_2)$ using the sideband MC $B\overline{B}$ background.
- Estimate p_{cont} using its value from the simulation.

By Assumption **A3.**, we use $\hat{w}(z_1, z_2)$ to quantify a systematic uncertainty as follows:

1. Apply a weight $w(z_1, z_2)$ for each simulated $B\overline{B}$ background data point.
2. Use these weighted points to update the estimated density functions input into the signal extraction.
3. Quote the difference between the updated optimized value and the central value as the systematic uncertainty.

11.4.3 Results

The results shown in Table 11.10 are a bit surprising. This particular systematic turns out to be the most dominant source of error for this analysis by a large factor. The possible explanations are discussed in further detail in Section 11.8.

Signal Type	Simulated data	Detector data
$D\tau$	0.02188	0.02686
$D^*\tau$	0.10291	0.08394

Table 11.10: Systematic uncertainties on $\hat{p}_j$ due possible misrepresentations of $\hat{f}_j$ by the simulated $B\overline{B}$ background. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

11.5 Detector efficiencies

In this section, we quantify the effects of mismodeling of the detector in simulation. More specifically, the shortcomings of the simulated detector manifest themselves in our result during the estimations of $\hat{\epsilon}_j$. We account for such shortcomings for the following two quantities:

1. Tracking efficiency: given a track created by a final state particle in the detector, the probability of the track being recognized as such.

2. Particle identification (PID) efficiency: given a PID algorithm classifying a particle as a specific type, the probability of it being correct.

11.5.1 Tracking efficiency

We follow the recipe provided by the tracking group of *BABAR*, which is to simply bound the mistakes of the simulated detector by assigning a relative uncertainty of 0.2% per track. Since this is quantity is per event, we perform a weighted average over each event type sample to extract its effect on $\hat{\epsilon}_j$ as follows:

$$\sigma^{trk}_{\hat{\epsilon}_j} = \frac{\sum_{i=1}^{N^{(j)}} w_i^{(j)} n_i^{(j)} 0.2\%}{\sum_{i=1}^{N^{(j)}} w_i^{(j)}}, \tag{11.12}$$

where $N^{(j)}$ is the number of events of event type j, $w_i^{(j)}$ is the weight of the event i, and $n_i^{(j)}$ is the number of tracks for event i.

This results in a relative uncertainty of $\boxed{2.3\%}$ for $\hat{\epsilon}_j$ for both signal event types.

11.5.2 PID efficiency

Similar to the tracking efficiency systematic, we follow the recipe provided by the PID group which simply assigns 0.7%/1.3%/1.1% relative uncertainty per $e/\mu/K$ in an event.

Rather than identifying all such particles in our dataset, the reconstruction criteria for the signal event types give us a simple way to bound this uncertainty:

- Two light leptons per signal event: one from B_{tag} and the other from having to veto the leptonic decays of the τ on the signal side. Since $m_e, m_\mu \ll \Lambda_{QCD}$, we can average the electron and the muon case, resulting in 2.0% relative uncertainty per signal event.

- Two K's per signal event: one from each $D^{(*)}$, giving us 2.2% relative uncertainty per signal event.

Thus we assign a relative uncertainty of $\boxed{4.2\%}$ for $\hat{\epsilon}_j$ for both signal event types.

11.6 Systematic uncertainty on the bias correction

The bias correction procedure as outlined in Section 9.3 has an associated systematic uncertainty stemming from our limited knowledge of the background proportions (i.e. p_i where $i = D^{**}SL, Comb, Cont$).

We start by making the following assumptions:

1. The uncertainties in the background proportions are dominated by branching fraction measurement uncertainty.
2. Uncertainty of the continuum component proportion is negligible since we use the off-peak data, which can be assumed to be identical to the continuum component of the real data.

The systematic uncertainty due to the branching fractions can be quantified as before by fluctuating the background proportion of $D^{**}SL$ and $Comb$, and measuring the bias as before. The changes in the bias from these fluctuations are used as upper bounds of the systematic uncertainty. The next few sections aim to quantify how much we trust the values generated by the `mc.central` model.

11.6.1 Branching fraction uncertainty in background components

To quantify the total uncertainty in the $B\overline{B}$ background components due to branching fraction measurement errors, we rank the most common decay modes for each event type. More specifically, we draw a random sample of 10000 records for each event type, then categorize each B into one of 9 possible `McBTypes`, or categories:

- `NoB`: Not a B decay.
- `Dtau`: $B \to D\tau\nu$.
- `Dstartau`: $B \to D^*\tau\nu$.
- `Dl`: $B \to D\ell\nu$.
- `Dstarl`: $B \to D^*\ell\nu$.
- `Dstarstar_res`: $B \to D^{**}\ell\nu$.
- `Dstarstar_nonres`: $B \to D^{**}\ell n\pi\nu$.
- `SL`: All other semileptonic decays of B.
- `Had`: Hadronic decays of B.

Figure 11.11 shows the breakdown of the background event types and Table 11.11 shows the upper bounds of relative measurement errors for each `McBType` based on Tables 11.3 and 11.4.

From these values, we can proceed to bound the relative uncertainties on $p_{D^{**}SL}$ and p_{Comb} by considering them as weighted sums of the 9 `McBTypes`. This results in 6.3% and 5.7% uncertainty on $p_{D^{**}SL}$ and p_{Comb}, respectively.

The following recipe is used to assign systematic uncertainties due to the uncertainties in background proportions:

1. Calculate the biases for a grid of points in true ($D\tau$, $D^*\tau$) space.
2. For each grid point (i.e. choice of true signal proportions), vary the background proportions up and down based on the respective uncertainties given above.

[5] It should be noted that other analyses, for instance [4], assign much more conservative uncertainties on the branching fractions of B into D^{**} decay modes, up to 100%.

McBType	w	Fraction
Dstarstar_res	7062.35	0.34
Had	6879.70	0.33
Dstarstar_nonres	3799.62	0.18
Dstarl	1895.73	0.09
Dl	836.97	0.04
Dstartau	116.14	0.01
Dtau	46.43	0.00
SL	30.95	0.00

(a) Event type 3, $D^{**}SL$.

McBType	w	Fraction
Had	10164.59	0.57
Dstarl	4794.70	0.27
Dl	2297.09	0.13
Dstartau	223.51	0.01
Dtau	137.01	0.01
SL	126.56	0.01

(b) Event type 4, $Comb$.

Figure 11.11: Ranking of the `McBTypes` in 10000 records of event types 3 and 4, or $D^{**}SL$ and $Comb$.

McBType	Relative Measurement Error
Dl	1%
Dstarl	1%
Dstarstar_res [5]	15%
Dstarstar_nonres	10%
SL	2%
Had	10%

Table 11.11: Summary of relative uncertainties on the most common decay modes in the background event types.

3. Assign the systematic uncertainty as the maximum differences between the biases calculated from the four variations and the default bias.

When we performed the above procedure for 25 grid points ranging from $(0, 0)$ to $(0.02, 0.02)$ in step sizes of 0.005, we found that the systematic uncertainty is rather uniform as a function of the true proportions. Thus, we assign as the systematic uncertainty that of the grid point closest to the extracted proportions.

Signal Type	Systematic Uncertainty
$D\tau$	0.00022
$D^*\tau$	0.00054

Table 11.12: Systematic uncertainties on the bias correction. The first column is the event type, the second column is the systematic uncertainty. There is no differentiating between simulated and detector data since all estimations are performed in simulation.

11.7 Additional sources of systematic uncertainty

11.7.1 Possible discrepancy of $R(D^{**})$

The main motivation behind this analysis is the tension between measured and predicted value of $R(D^{(*)})$ of around 30%. We assess a possible systematic uncertainty arising from discrepancy in the $R(D^{**})$ system of a similar magnitude.

We increase the branching fractions of semitauonic decays of B to D^{**} (i.e. the last four decays listed in Tables 11.3 and 11.4) by 30%. This results in $\boxed{0.0002\ (0.0003)}$ and $\boxed{0.0003\ (0.0001)}$ uncertainty on $\hat{p}_{D\tau}$ and $\hat{p}_{D^*\tau}$, respectively, for simulated (detector) test dataset.

11.8 Summary

Table 11.13 collects all the systematic uncertainties derived in this chapter.

		Simulation		Detector Data	
		$D\tau$	$D^*\tau$	$D\tau$	$D^*\tau$
$\hat{p}$ uncertainties	$D^{(*)}\tau$ form factors	0.00029	0.00208	0.00053	0.00133
	$D^{**}\tau$ form factors	0.00001	0.00003	0.00017	0.00002
	Semileptonic D^{**} branching fractions	0.00029	0.00020	0.00026	0.00031
	Semileptonic $D^{(*)}$ branching fractions	0.00034	0.00079	0.00074	0.00067
	Strange decay branching fractions	0.00008	0.00011	0.00024	0.00076
	$B \to D^{(*)}\rho$ branching fractions	0.00010	0.00020	0.00026	0.00026
	$B \to D^{(*)}a_1$ branching fractions	0.00028	0.00069	0.00037	0.00063
	$B\overline{B}$ background shape validation	0.02188	0.10291	0.02686	0.08394
	D decay branching fraction cross-check	0.00007	0.00003	0.00022	0.00021
	Gap sample for $B \to X_c\ell\nu$ branching fractions	0.00001	0.00017	0.00010	0.00046
	Background proportions on bias correction	0.00022	0.00054	same as simulation	
$\hat{\epsilon}$ uncertainties (10^{-5})	Semileptonic D^{**} branching fractions	1.1	2.1	same as simulation	
	Semileptonic $D^{(*)}$ branching fractions	0.4	1.1		
	Strange decay branching fractions	0.5	0.3		
	$B \to D^{(*)}\rho$ branching fractions	0.3	0.3		
	$B \to D^{(*)}a_1$ branching fractions	0.5	0.8		
	Tracking efficiency	4.2	5.2		
	PID efficiency	7.6	9.5		

Table 11.13: Summary of systematic uncertainties on $\hat{p}_j$ and $\hat{\epsilon}_j$.

First, we note the consistency between the estimations on between MC and detector data test set. This can be interpreted as a validation of the quality of the simulation. With the exception of the $B\overline{B}$ background shape systematic, the sizes of all other systematic uncertainties align with the expectations based on the relative sizes of the parameter fluctuations.

Now, let us discuss the $B\overline{B}$ background shape systematic a bit more carefully. There are two points of discussion that arise:

1. Are we sure that the estimation is being performed correctly? Is it possible that some of the assumptions made in Section 11.4 are false?

2. Is there an alternate way of estimating the systematic without relying on the sideband sample?

It is possible to attribute the large uncertainty to the limited statistics in the sideband sample. More specifically, the off-peak sideband sample has very few events. We reject this possibility by performing the following experiment: estimate the systematic using two off-peak samples, sideband-only and full-region. We used the full-region of the off-peak sample by Assumption **A3.**, and if the effect was indeed due to limited statistics, we expect an *increase* in the uncertainty with lower sample size. It turns out that we actually observe a decrease in the systematic, rejecting the hypothesis.

We also reject the possibility of any bugs in the software performing the kernel density estimation and the calculation of the weights as described in (11.11). This is due to the fact that in the previous iteration of the analysis [9], this particular systematic was large but not dominant over other systematics such as those stemming from the branching fraction uncertainties.

Let us now examine the assumptions. Assumptions **A1.** and **A2.** have been demonstrated to be quite good as previously discussed. Suppose Assumption **A3.** is false, then the whole methodology falls apart. In discussions with collaborators and reviewers, we have not been able to come up with an alternative method that does not rely on the sideband assumption. If we assume it to be true, then we are left with the conclusion that our estimate is indeed valid, and that in this analysis the extracted signal proportions are very sensitive

to the fluctuations of the $B\overline{B}$ background shapes. This notion is very reasonable given that the two background event types make up 71% of the dataset. Any future analyses should make great efforts to reduce the relative size of the background, which is an optimization problem that should also attempt to keep as many signal events as possible. Another alternative is to better separate the background densities from the signal densities, in which case the relative sizes of the event types matter less.

Chapter 12

Results and Conclusion

12.1 Results on detector data

As shown in (4.3) and (4.5), the value of $\mathcal{R}(D^{(*)})$ can be calculated from the estimations of $\hat{p}_{D^{(*)}\tau}$ and $\hat{\epsilon}_{D^{(*)}\tau}$ using simple arithmetic.

To combine various sources of systematic uncertainties, we assign a correlation of 0 or ± 1 between the extracted signal proportions based on the signs of the differences. More specifically, the estimated systematic uncertainties as shown in Table 11.13 are simply absolute differences of the results from two model assumptions: the baseline model and a fluctuated model. When comparing just two results, the only possible correlations are 0 and ± 1. Once the covariance matrices of all systematics have been constructed, the sum of all but one systematic is used to quote the overall systematic uncertainties which can be visualized as an ellipse in the $\mathcal{R}(D) - \mathcal{R}(D^*)$ plane.

The one systematic that is not part of the sum is the systematic due to the $B\overline{B}$ background shape validation. This particular source of systematic uncertainty differs from all others in the fact that there is no single parameter that varied. It is a data-driven validation of the MC using the correction factors for the two $B\overline{B}$ background density shapes learned from the sideband detector data.

In this case, drawing an ellipse based on the absolute differences would be misleading. This is due to the fact that this model parameter is binary modification (correction factors applied or not applied); we cannot interpret the differences as the half-width of an ellipse that extends symmetrically in the other direction. A line between the two result is a better interpretation of the uncertainty on our final result due to this systematic.

Thus, we propose the following way of visualizing our result:

1. Plot an ellipse formed by the central value (model A) and the statistical and all systematic uncertainties except that due to the $B\overline{B}$ background shape validation systematic (model B).

2. Plot a second ellipse with the same shape (uncertainty) as the first ellipse centered at the extracted $\mathcal{R}(D^{(*)})$ assuming model B.

3. Connect the two ellipses by drawing the tangent lines.

The above recipe puts both models in equal footing; neither model is more valid than the other, it is simply different ways of modeling the $B\overline{B}$ background densities. In a sense, we have two significantly different results based on the assumed background model.

Thus, the result of this analysis contains two measurements of $\mathcal{R}(D^{(*)})$: one extracted using the baseline model (model A) and one extracted using the model where the $B\overline{B}$ background density correction factors have been applied (model B).

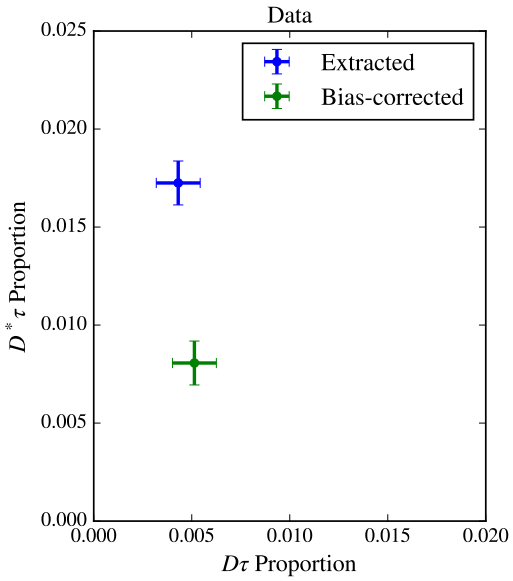


Figure 12.1: Bias correction applied to the extracted signal proportions from the detector data assuming the baseline model (model A). Errors are statistical only.

12.1.1 $\hat{p}_{D^{(*)}\tau}$

The extracted signal proportions *after* bias correction assuming model A are as follows:

$$\begin{aligned} \hat{p}_{D\tau} &= 0.0051 \pm 0.0011(\text{stat.}) \pm 0.0010(\text{sys.}), \\ \hat{p}_{D^*\tau} &= 0.0081 \pm 0.0012(\text{stat.}) \pm 0.0018(\text{sys.}), \qquad (12.1) \\ \rho_{\hat{p}_{D\tau},\hat{p}_{D^*\tau}} &= 0.07, \qquad (12.2) \end{aligned}$$

where the statistical component of the correlation is -0.41.

The bias-corrected signal proportions assuming model B are:

$$\begin{aligned} \hat{p}_{D\tau} &= 0.0320 \pm 0.0011(\text{stat.}) \pm 0.0010(\text{sys.}), \\ \hat{p}_{D^*\tau} &= 0.0920 \pm 0.0012(\text{stat.}) \pm 0.0018(\text{sys.}), \qquad (12.3) \\ \rho_{\hat{p}_{D\tau},\hat{p}_{D^*\tau}} &= 0.07, \qquad (12.4) \end{aligned}$$

where the uncertainties and the correlation are the same as those when using model A, only the central value has changed.

Figure 12.1 shows the extracted signal proportions of model A before and after the bias correction.

12.1.2 $\hat{\epsilon}_{D^{(*)}\tau}$

The signal type efficiencies are estimated using simulation, and since the $B\overline{B}$ background validation shapes do not affect the overall normalization, we have a single result for both model A and model B:

$$\begin{aligned} \epsilon_{D\tau} &= 0.00182 \pm 0.00009(\text{sys.}), \\ \epsilon_{D^*\tau} &= 0.00226 \pm 0.00011(\text{sys.}). \qquad (12.5) \end{aligned}$$

12.1.3 $\mathcal{R}(D^{(*)})$

The measurements of $\mathcal{R}(D^{(*)})$ based on the two model assumptions are:

- Model A (baseline):

$$\begin{aligned} \mathcal{R}(D) &= 0.231 \pm 0.028 \pm 0.028, \\ \mathcal{R}(D^*) &= 0.127 \pm 0.019 \pm 0.031, \\ \rho_{\mathcal{R}(D),\mathcal{R}(D^*)} &= 0.06. \end{aligned} \tag{12.6}$$

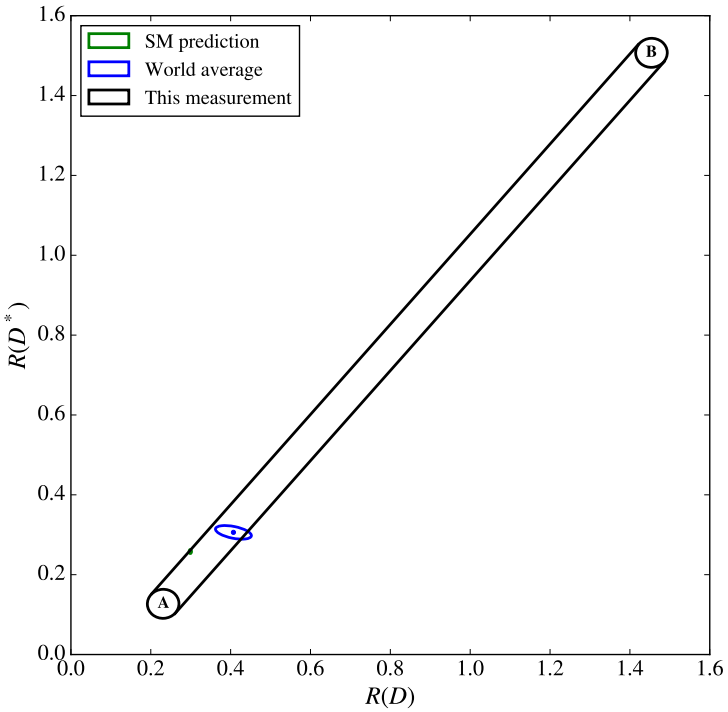
- Model B (baseline model with $B\overline{B}$ background shape correction factors applied):

$$\begin{aligned} \mathcal{R}(D) &= 1.454 \pm 0.028 \pm 0.028, \\ \mathcal{R}(D^*) &= 1.507 \pm 0.019 \pm 0.031, \\ \rho_{\mathcal{R}(D),\mathcal{R}(D^*)} &= 0.06, \end{aligned} \tag{12.7}$$

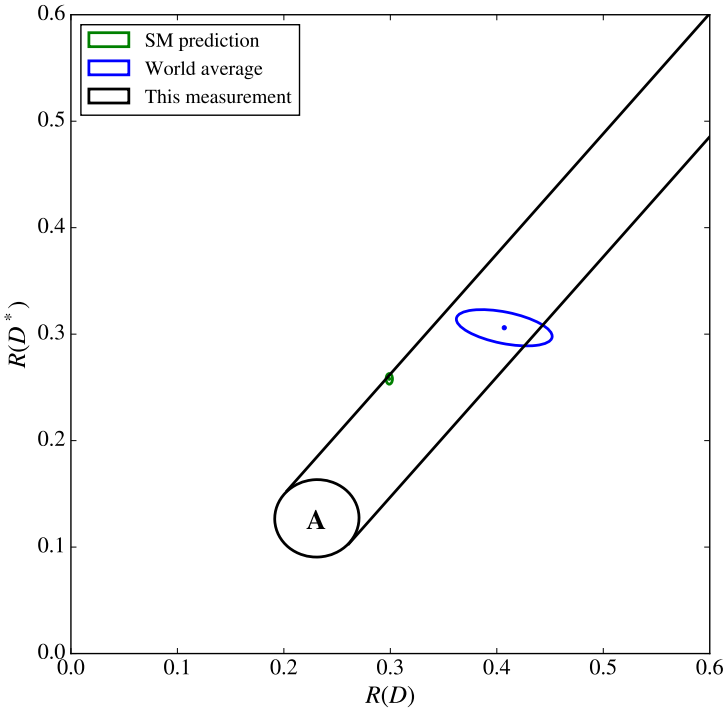
  where the uncertainties and the correlation are the same as those extracted using model A.

Figure 12.2 shows our results when visualized using the recipe from Section 12.1. Given how our result was constructed, we do not have a way to numerically state our result. One possible way would be to construct a single ellipse based on the average of the two results with a confidence contour extending from the model A result to the model B result, but it would obscure the fact that we have two distinct results based on the method used to model the $B\overline{B}$ background density shapes.

While this procedure is somewhat non-traditional, this result is consistent with both the world average and the SM prediction, and we believe it more faithfully shows the true uncertainties on this measurement of $\mathcal{R}(D^{(*)})$.

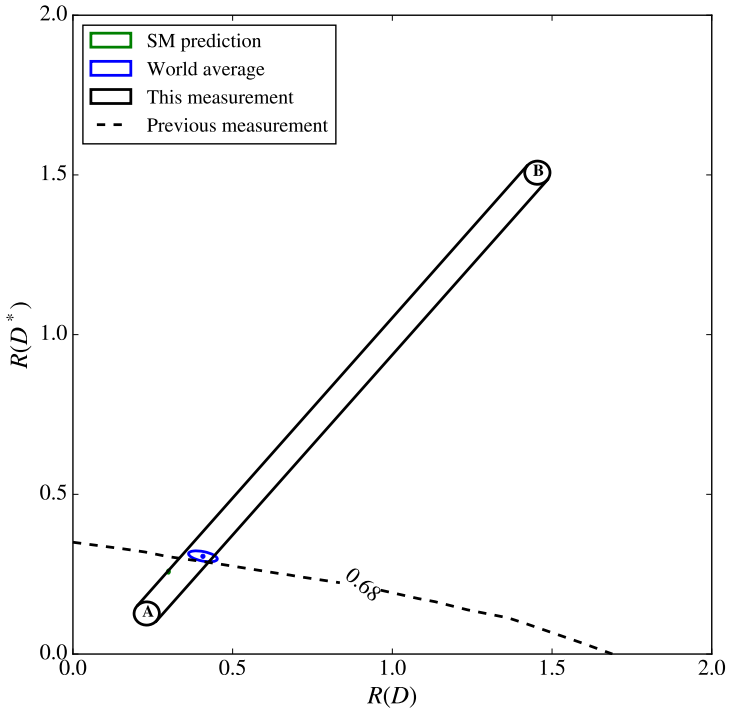


(a) SM expectation, world average, and this result. **A** denotes the central value of our baseline model and **B** denotes the central value of the model with the $B\overline{B}$ background shape validation factors applied.



(b) Same as (a), but zoomed into the baseline model (**A**) result to show the consistency with the SM prediction and the world average.

Figure 12.2: Results of this analysis along with the SM expectation, the world average, and the result of the previous iteration of this analysis in the same channel [9].



(c) Same as (a), but with the result from the previous measurement added.

Figure 12.2: Results of this analysis along with the SM expectation, the world average, and the result of the previous iteration of this analysis in the same channel [9] (Cont.).

12.2 Conclusion

We perform a measurement of $\mathcal{R}(D^{(*)})$ using the data collected from the *BABAR* experiment with semileptonic tagging and hadronic reconstruction of the τ. The analysis procedure is tuned to maximize the statistical sensitivity on the dataset in which the signal events are scarce and noisy. In the first iteration of the analysis, we obtain a result with excellent statistical precision but also large systematic uncertainties. Moreover, the systematic uncertainties estimated on the MC are not consistent with those estimated on the detector data.

We attempt to reduce this discrepancy in the second iteration of the analysis, which is the topic of this thesis, by improving the reconstruction selection criteria. The result of this analysis maintains the competitive statistical precision while restoring the consistency between the systematic uncertainties estimated on the MC and the detector data. The large size of the systematic uncertainties is dominated by the systematic quantifying the validity of $B\overline{B}$ background density shapes, which we classify as a characteristic of the dataset rather than a shortcoming of the methods.

Any future analyses of similar nature should consider the benefits of our method which results in large signal yields and higher statistical sensitivity.

In summary, we believe the most notable contributions of this analysis are:

1. Bias correction of extracted results as a function of signal proportions, see Section 9.3.
2. GPU-compatible fast kernel density estimation[1].
3. Truth matching algorithm, see Chapter 6.
4. PostgreSQL adapter for ROOT files[2].
5. Employing domain adaptation algorithm for MC training set and detector data test set, see Appendix A.
6. Estimation of background density shapes using sideband data, see Section 11.4.

[1] https://github.com/dchao34/bbrcit_kde
[2] https://github.com/jkim-/root2postgres

Appendices

Appendix A

Unsupervised Domain Adaptation

A.1 Introduction

In a standard supervised learning setting, we deal with two types of data: labeled and unlabeled. The labeled data is typically used for training and validating the model while the unlabeled data is the test set on which the model will be deployed.

One of the fundamental assumptions is that the two datasets are generated from the same distribution. It makes logical sense to require a consistency between the training and the test data. Typically, the data collection for a learning task involves collection of some initial set of data that is manually labeled by humans. This training set is used to train the model, and when the learning is complete, any new observations from the same source is predicted by the model. It is the goal of the model to minimize the error of the predictions of the test set, or the generalization error.

Let us now focus on the learning task at hand. The training set consists of simulated labeled data whereas the test set is the detector data. In this case, we do not have any guarantees of the consistency of the distributions between the two samples. In fact, it is very unlikely for the simulation to model the truth exactly, and we quantify such differences in the form of systematic uncertainties.

In this analysis, we use machine learning algorithms for the following three tasks:

1. Score each candidate based on its probability to be truth matched.
2. Score each event based on its probability to be one of the two signal event types.
3. Score each event based on its probability to be the $D^*\tau$ event type.

The exact learning algorithm (e.g. random forest, logistic regression) is not important. Rather, the concern at hand is the ability of each model to generalize to a dataset drawn from a distribution slightly different from that of the simulation (training set).

The test set, or the detector data, has no labels in all three contexts. For example, given a reconstructed $\Upsilon(4S)$ candidate of a detector event, we have no way to know whether it is truth matched or not. It is not a matter of difficulty, it is simply not available to us. This leads to a situation where we have no way of knowing how well our model is performing in the test set.

This is a problem that exists for all high energy physics analyses, and we attempt to learn more robust models that provide performance guarantees on an unlabeled test set that might differ in distribution to the labeled training set.

A.2 Unsupervised domain adaptation

The class of unsupervised domain adaptation algorithms is first developed in the context of computer vision. The standard problem is the digit recognition task with an added complexity: instead of having solid color backgrounds for the digits (MNIST), the backgrounds random patches from real photos (MNIST-M), where the terms in the parentheses are names of the standard dataset. This added noise turns out to be a great obstacle for models trained only on the former dataset.

In other words, we have a labeled training set drawn from the *source* distribution (digits with black background) and an unlabeled test set drawn from the *target* distribution (digits with colorful background). The term unsupervised refers to the fact that the test set has no labels.

The similarities between the settings of MNIST-M and our problem is apparent: the MC represents data drawn from the source distribution and the detector data represents the data from the target distribution.

We implement two unsupervised domain adaptation algorithms, domain-adversarial neural network (DANN) [35] and PixelDA [36], and quantify their performances on the test set based on the reverse validation score, which will be discussed in detail in later sections.

A.2.1 Overview

The central idea of domain adaptation algorithms is learning a common representation of both source and target domains. DANN learns a mapping of both source and target domains to a new common representation while PixelDA learns a mapping of the source domain to the target domain. We will focus on DANN for the rest of the appendix, but it also applies to PixelDA without much loss in generality.

The feature learning step draws inspiration from the generative adversarial network (GAN), which is a generative model with two components: the generator that produces fake data and the discriminator that tries to tell apart the fake data from the real data. In the domain adaptation setting, the generator transforms both source and target domain data into the same representation while the discriminator, or domain classifier, attempts to classify the domain of each transformed point. The training of this step is considered to be converged when the discriminator always outputs 0.5, i.e. it cannot tell apart the domain of the transformed points.

It is trivial for the generator to fool the discriminator by throwing away all information. For example, it can always output 0 regardless of the input, and the discriminator can never be sure of its origin. Thus, domain adaptation algorithms contain a third component, the task classifier, that uses the transformed source domain points with labels as the training set. The idea is that given a new target domain point, after the transformation, the task classifier can predict its class with the same confidence had it been given a transformed source domain point.

A.2.2 Domain adversarial neural network

We can now describe the DANN algorithm in a more formal manner. Let $x_s = (x_{1,s}, x_{2,s}, \ldots, x_{n_s,s})$ be the $n_s \times d$ design matrix containing source domain data, for which we have the corresponding labels $y_s = (y_{1,s}, y_{2,s}, \ldots, y_{n_s,s})$. Similarly define x_t to be the target domain data.

The generator is an autoencoder that learns a function $G_f : \mathbb{R}^d \mapsto \mathbb{R}^d$ that performs the feature transformation. Let $z_i = G_f(x_i; \theta_f)$, where $i \in \{s, t\}$, denote the output of such transformation. The transformed data is then used as input for both the domain classifier $G_d(\cdot; \theta_d)$ and the task classifier $G_t(\cdot; \theta_t)$.

Let $z = z_s \bigcup z_t$ be the entirety of our data. G_d can be any supervised learning algorithm that minimizes the loss of z given its *domain labels* $y_d = 0$ for source domain points and 1

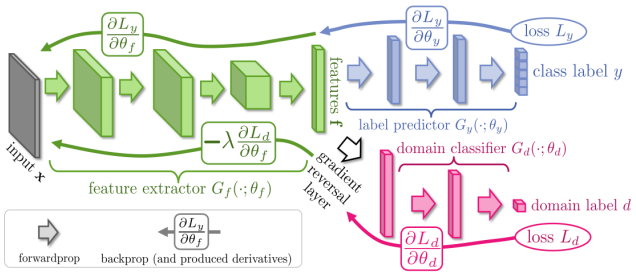


Figure A.1: Diagram of the DANN algorithm. [35]

for target domain points. The task classifier G_t is also a supervised learning algorithm, but uses (z_s, y_s) as its training data instead.

We denote the domain loss and the prediction loss as

$$\mathcal{L}_d(\theta_f, \theta_d) = \mathcal{L}_d(G_d(G_f(x)), y_d), \tag{A.1}$$

$$\mathcal{L}_t(\theta_f, \theta_t) = \mathcal{L}_t(G_t(G_f(x_s)), y_s), \tag{A.2}$$

where $\mathcal{L}$ can be any loss function such as the cross-entropy loss.

Then, the overall loss function of the algorithm is

$$\mathcal{L} = \mathcal{L}_t - \lambda \mathcal{L}_d, \tag{A.3}$$

where λ is a hyperparameter controlling the amount of adaptation. If $\lambda = 0$, it is as if we do not perform any feature transformation; the optimizer solely focuses on the task classifier performance. On the other hand, as λ becomes large, the domain adaptation will be the main focus, disregarding the task classifier performance.

The optimal parameters are the saddle point solution to the following optimization problem:

$$\min_{\theta_f, \theta_t} \max_{\theta_d} \mathcal{L}(x_s, x_t, y_s; \theta_f, \theta_d, \theta_t). \tag{A.4}$$

Figure A.1 shows the diagram of the DANN algorithm.

A.3 Reverse validation

Given that the test set has no labels, how can we quantify the performance of the task classifier? This is a problem that the vanilla supervised learning algorithms also face given the dataset of this analysis. Model parameters selected based on the cross validated metrics do not provide any theoretical guarantees on the out-of-sample performance.

We use the metric called the reverse validation, which can be calculated as follows [37]:

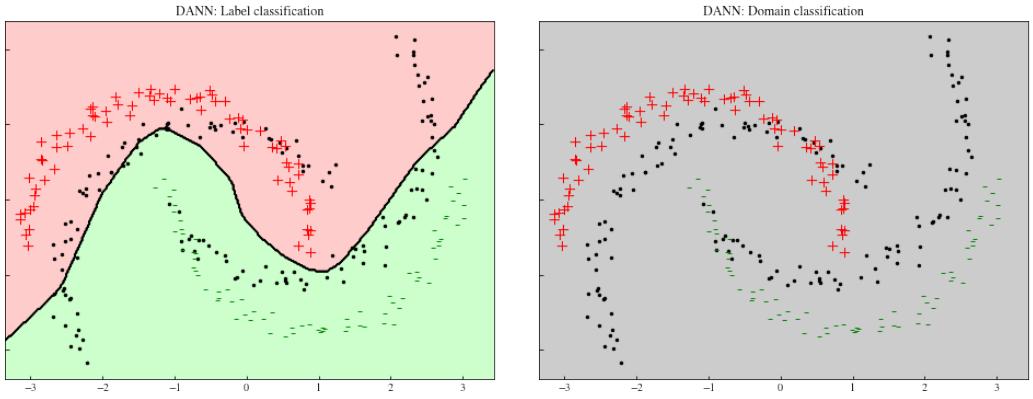
1. Using the training set, which contains both source and target domain data, train the model and extract the optimal set of parameters θ, which does not include the hyperparameters such as learning rate that remains fixed throughout the estimation of reverse validation.

2. Using the model defined by θ to predict the classes of the target domain portion of the training set, which gives us a pseudo-label of the target data set $(x_t, \hat{y}_t)$.

3. Now, swap the roles of the source and target domains: treat $(x_t, \hat{y}_t)$ as the labeled source domain data and x_s as the unlabeled target domain data.

4. Train a new model with parameters θ' using the swapped dataset.

5. The reverse validation error is the error of the new model with θ' predicting on x_s, which has known labels y_s.



(a) The decision boundary contour map of the task classifier.

(b) The decision boundary contour map of the domain classifier.

Figure A.2: Results of the DANN algorithm on a toy dataset. Red plus and green minus signs represent the source domain data and the black dots represent the unlabeled target domain data.

It is important to note that the reverse validation is a *heuristic* and has no provable performance guarantees. In fact, it has been demonstrated that the reverse validation heuristic can result in suboptimal hyperparameter selection [36]. The ideal method would be to be able to label some portion of the target domain data, and use it to quantify the out-of-sample performance. Since such a method is not available for us, we use reverse validation for the purpose of comparing various sets hyperparameters for model selection, and do not quote it as a performance metric on the target test set.

A.4 Results and summary

We gauge the performances of our models by comparing the task classifier AUC on a source domain validation set that was reserved for this purpose from the start. In addition, we estimate its domain adaptation performance using the p-value from the Kolmogorov-Smirnov (KS) test on equal-sized source and target domain validation sets under the null hypothesis that they are both drawn from the same distribution.

Both the DANN and PixelDA algorithms were implemented in Tensorflow and trained on a NVIDIA 1080Ti GPU. The training of deeper networks such as ours was expected to be finicky and unstable, but we were eventually able to tune the hyperparameters to result in training convergence using techniques such as batch normalization and dropout.

We use an artificial dataset (`make_moons` [24]) to validate our algorithm implementations. The source domain data is generated from two semicircle arcs, each representing a class. The target domain data is generated from the same arcs but rotated by 30 degrees with the labels thrown out. Figure A.2 shows the results of the DANN algorithm.

As we can see, the task classifier successfully captures the target domain data even with no labels. Furthermore, the domain classifier classifies the entire region as the source domain due to its lack of ability to distinguish the two domains.

The final model showed slightly reduced task classification AUC compared to the simpler logistic regression model, which was expected due to the loss of information from the feature transformation. However, the KS test showed evidence that the two validation sets were still from different distributions, i.e. large p-value. This behavior persisted even at very large values of λ, leading us to conclude that the domain adaption was not performing well. This may be due to the lack of power of the KS test for sharply peaked distributions. Another possible reason would be that the domain adaptation algorithms do not lend well to our

dataset.

While we were not able to extract a satisfactory result, we have explored the possibility of applying a new type of learning algorithm in high energy physics applications.

Appendix B

Consistency Test

In order to validate the analysis procedure, especially the home-brewed software like the kernel density estimation package, we perform a self-consistency test.

Suppose we have fixed component densities for the five event types and a set of proportions. While the exact values of the proportions and bandwidths used for the KDE's are not important, we use the proportions expected from SM prediction.

We can generate test datasets of the same size ($N = 8.7$ million) as follows:

1. Draw $x \sim unif(0, 1)$.
2. Determine the event type x correpsonds to based on $\vec{p}$, then generate a point from the corresponding event type KDE.
3. Repeat until we have N points.

This effectively removes all possible external discrepancies between the training data, used to construct the KDE's, and the test data, from which we attempt to recover its true proportions.

We generate 300 such test datasets and estimate the signal proportions for all datasets. Figure B.1 shows the distribution of difference between expected and extracted signal proportions where we observe the clear lack of bias in our estimated signal proportions.

This test also demonstrates the validity of using the bootstrap to estimate the variance of the extracted signal proportions: the standard deviation estimated based on the sample variance of the 300 test datasets agree well with the bootstrap estimation of the standard deviation (Table B.1).

	$D\tau$	$D^*\tau$
Bootstrap S.D.	0.000895	0.000812
Sample S.D.	0.000887	0.000826

Table B.1: Validation of bootstrap estimation of the variance by comparison to the sample variance of the results of the 300 simulated datasets.

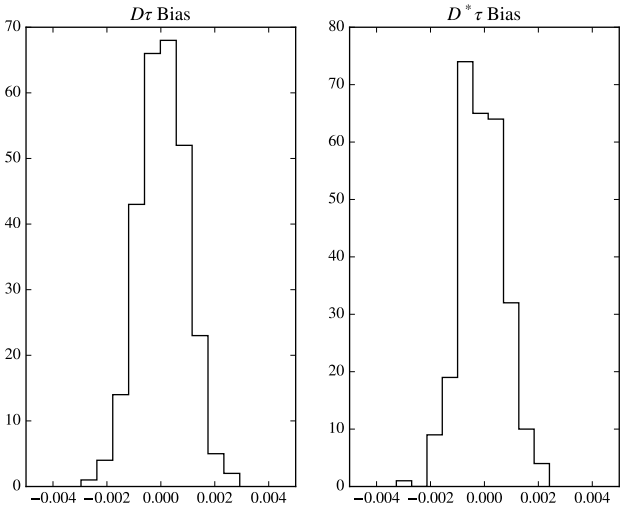
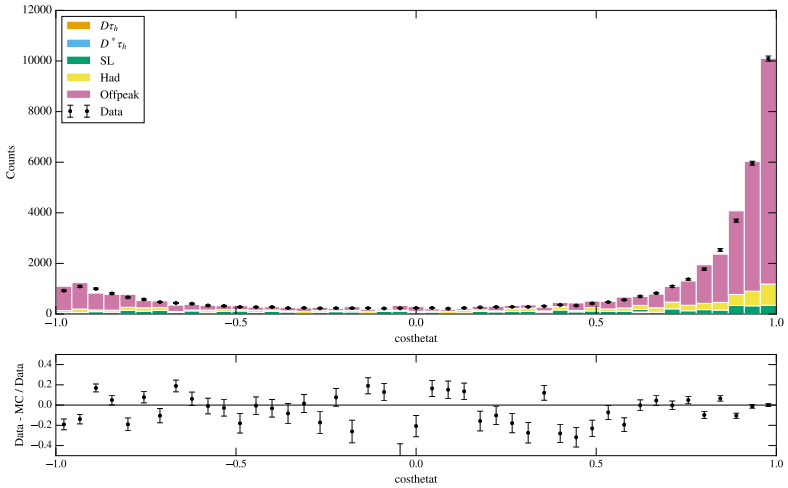


Figure B.1: Distributions of the biases of the extracted signal proportions.

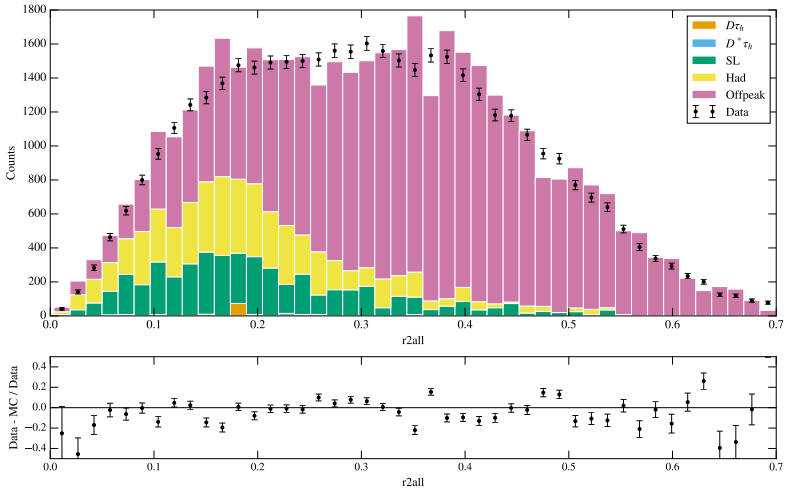
Appendix C

Sideband comparisons

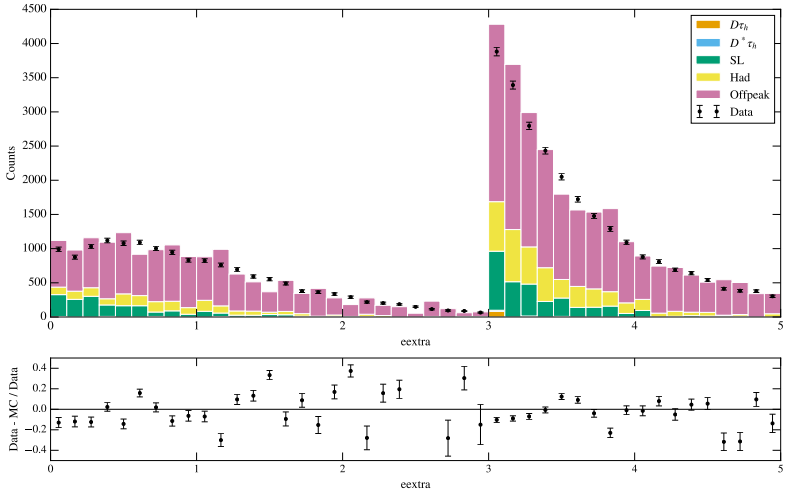
We show data and simulation comparisons in the sideband region. This is an extension of the discussion in section 11.4.



(a) $\cos\theta_T$.

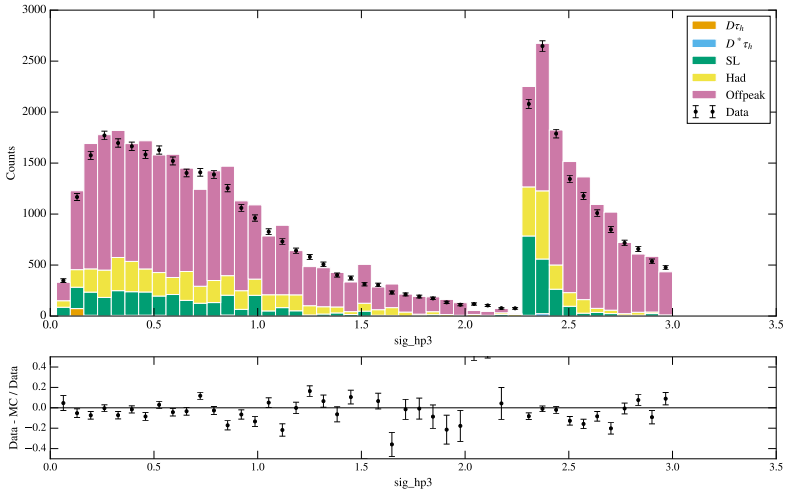


(b) R_2.

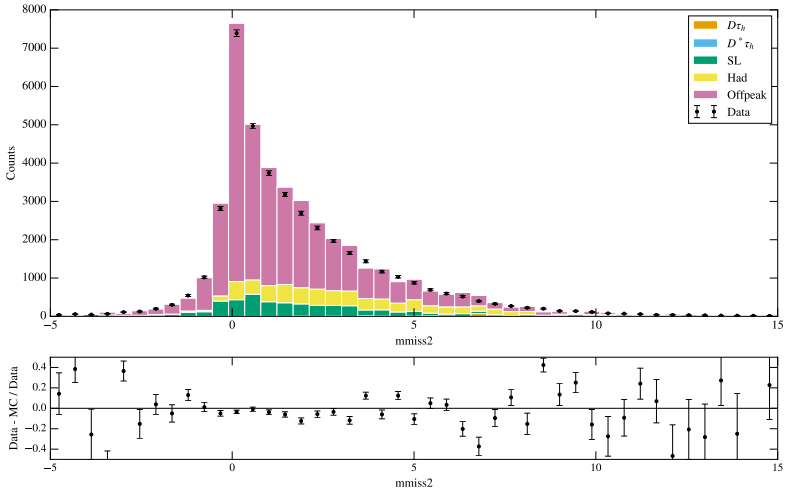


(c) E_{extra}. Left of the cliff belongs to the sideband sample where $|\vec{p}_h^{\,sig}| > 2.3$ GeV.

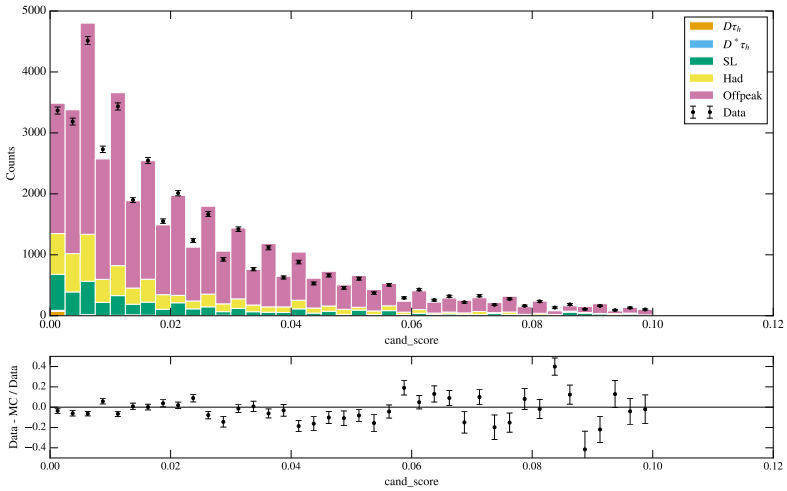
Figure C.1: Comparisons between data and MC for each event type in the sideband.



(d) $|\vec{p}_h^{sig}|$. Similar comments to those for E_{eextra} apply.

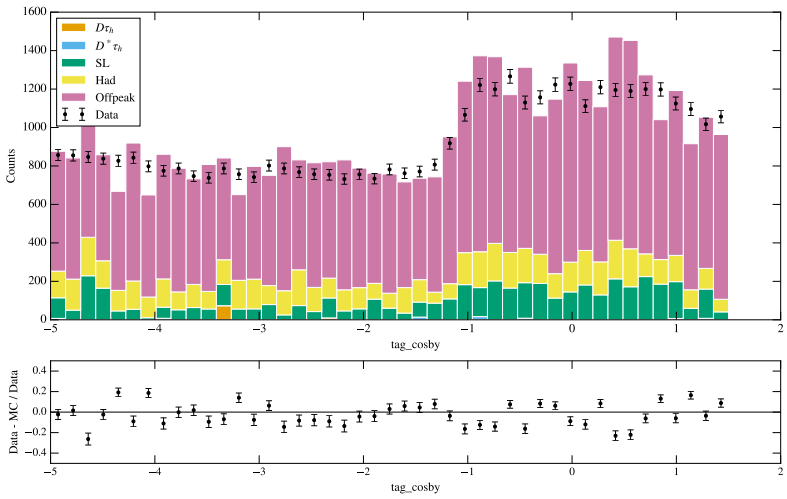


(e) M^2_{miss}.

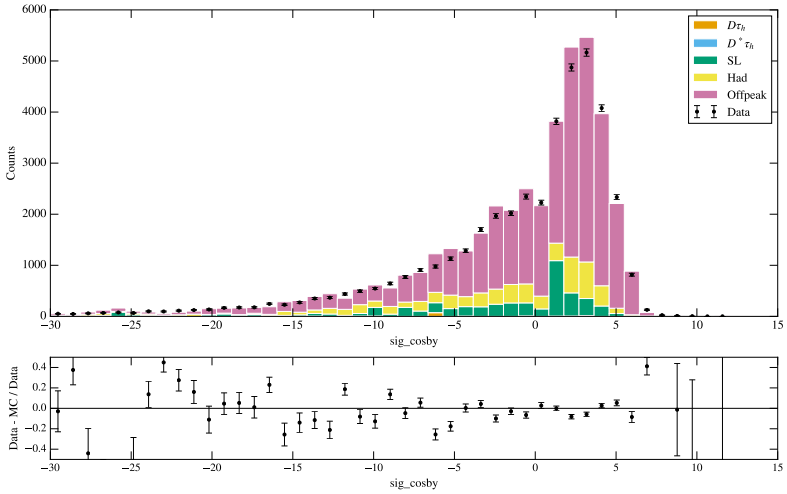


(f) Candidate score.

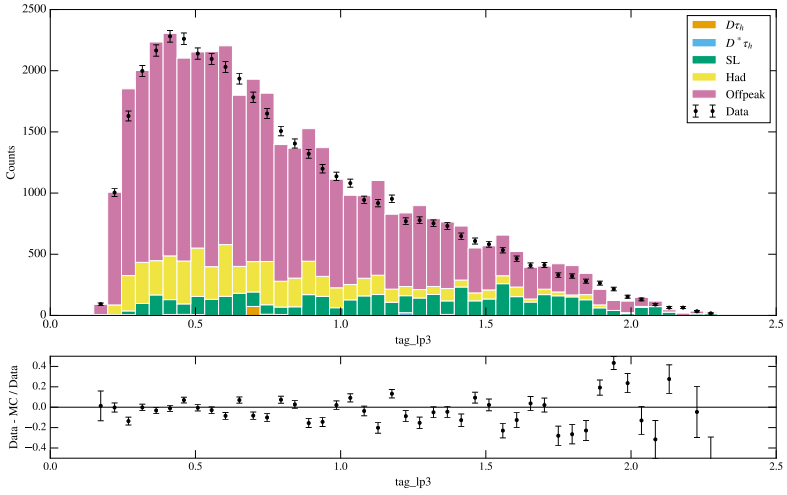
Figure C.1: Comparisons between data and MC for each event type in the sideband (Cont.).



(g) $\cos\theta_{BY}^{tag}$.

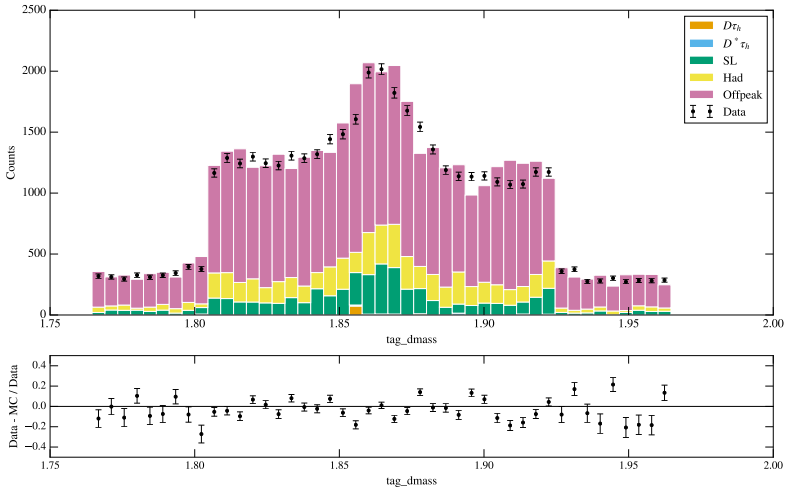


(h) $\cos\theta_{BY}^{sig}$.

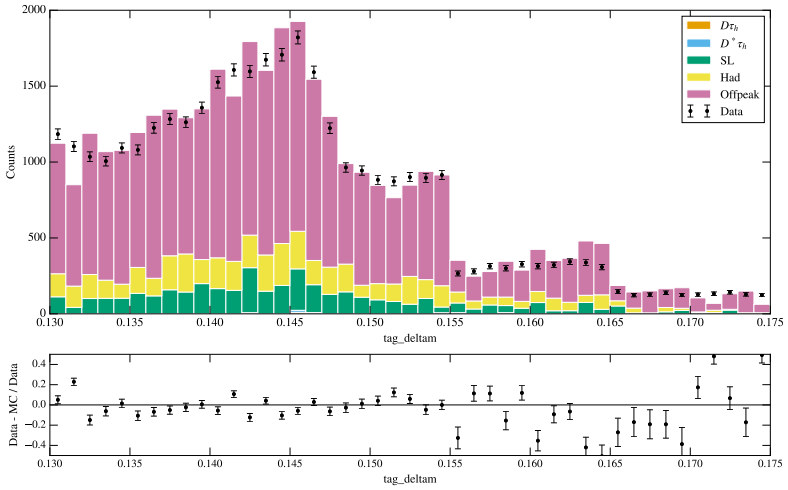


(i) $|\vec{p}_\ell^{tag}|$.

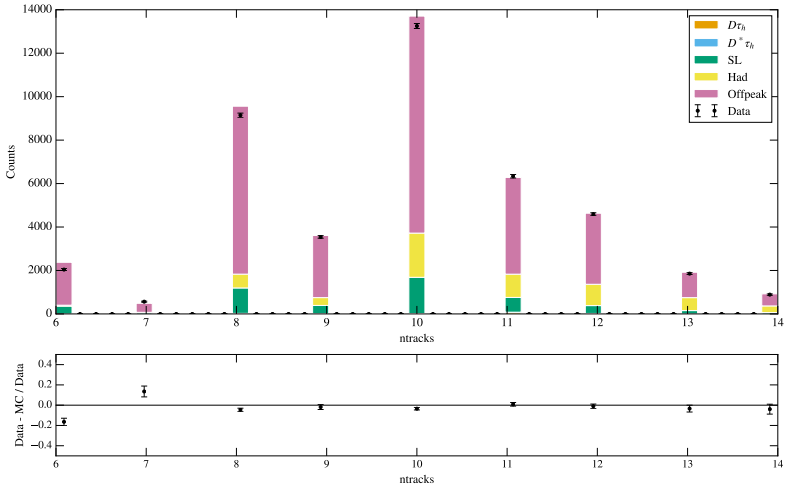
Figure C.1: Comparisons between data and MC for each event type in the sideband (Cont.).



(j) m_D^{tag}.

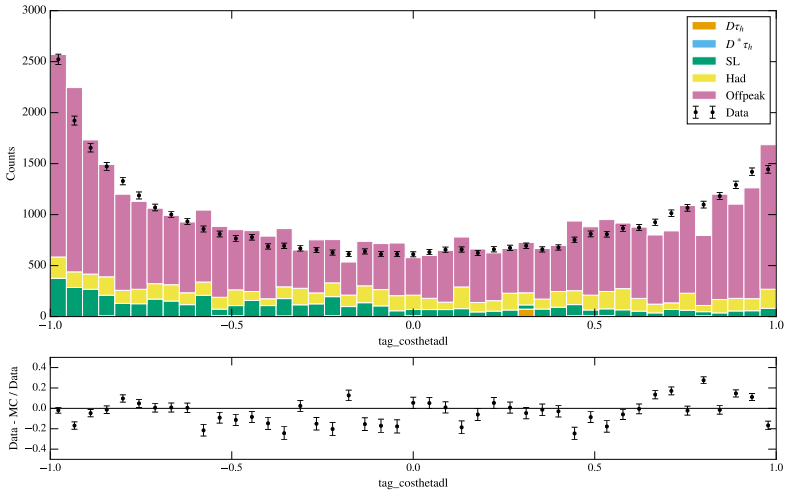


(k) Δm^{tag}.

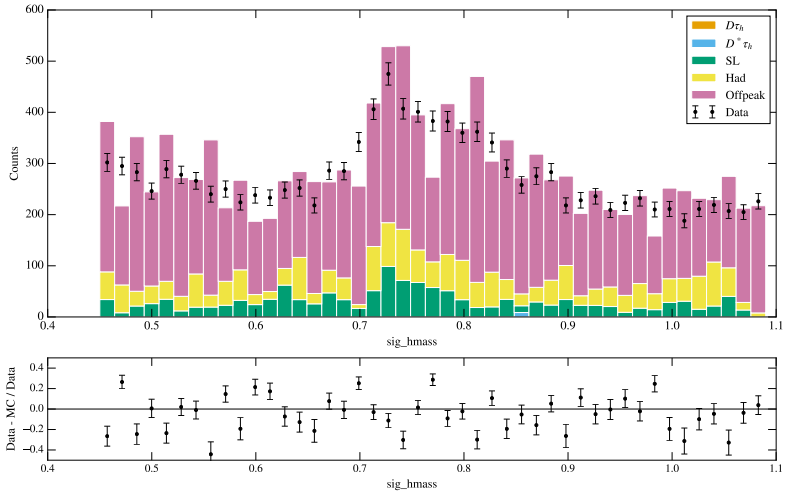


(l) N tracks.

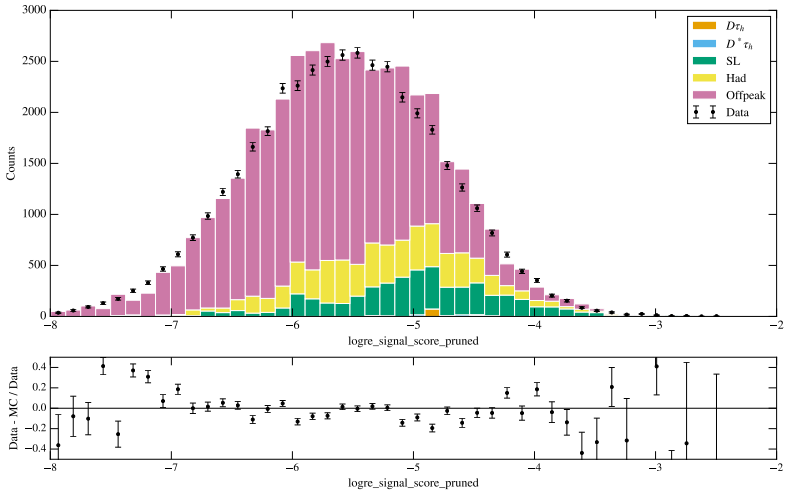
Figure C.1: Comparisons between data and MC for each event type in the sideband (Cont.).



(m) $\cos\theta_{D\ell}$.

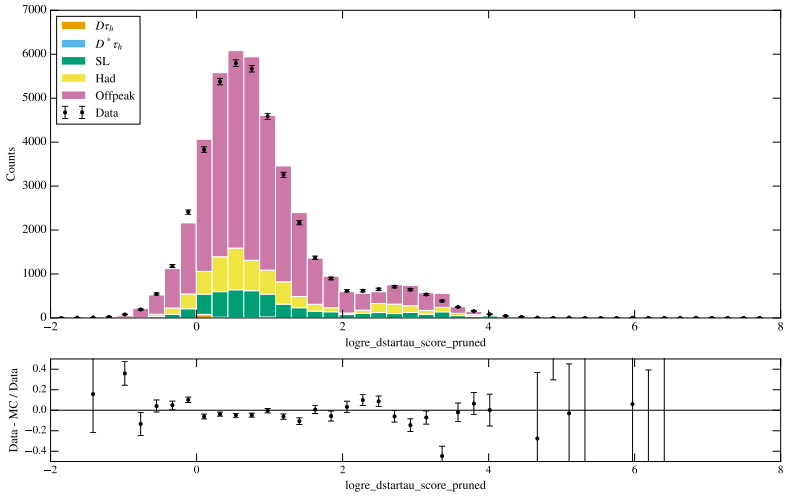


(n) m_h^{sig}.



(o) Z_1.

Figure C.1: Comparisons between data and MC for each event type in the sideband (Cont.).



(p) Z_2.

Figure C.1: Comparisons between data and MC for each event type in the sideband (Cont.).

Bibliography

[1] P. W. Anderson, Science **177**, 393 (1972).

[2] Heavy Flavor Averaging Group, Y. Amhis *et al.*, Eur. Phys. J. **C77**, 895 (2017), 1612.07233, updated results and plots available at https://hflav.web.cern.ch.

[3] *BABAR* Collaboration, J. P. Lees *et al.*, Phys. Rev. D **88**, 072012 (2013).

[4] Belle Collaboration, M. Huschle, *et al.*, Phys. Rev. D **92**, 072014 (2015).

[5] LHCb Collaboration, R. Aaij *et al.*, Phys. Rev. Lett. **115**, 111803 (2015).

[6] Belle Collaboration, A. Abdesselam, *et al.*, (2016), arXiv:1603.06711 [hep-ex].

[7] Belle Collaboration, S. Hirose *et al.*, Phys. Rev. Lett. **118**, 211801 (2017).

[8] LHCb Collaboration, R. Aaij *et al.*, Phys. Rev. Lett. **120**, 171802 (2018).

[9] D. S. Chao, *Measuring $\mathcal{R}(D^{(*)})$ for $\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)$ using Semileptonic Tags and τ Decays to Hadrons*, PhD thesis, California Institute of Technology, 2018.

[10] N. Isgur and M. B. Wise, Physics Letters B **232**, 113 (1989).

[11] M. Neubert, Physics Reports **245**, 259 (1994).

[12] K. Hagiwara, A. D. Martin, and M. F. Wade, Nucl. Phys. **B327**, 569 (1989).

[13] I. Caprini, L. Lellouch, and M. Neubert, Nucl. Phys. **B530**, 153 (1998), hep-ph/9712417.

[14] Particle Data Group, K. A. Olive *et al.*, Chin. Phys. **C38**, 090001 (2014).

[15] J. G. Körner and G. A. Schuler, Zeitschrift für Physik C Particles and Fields **46**, 93 (1990).

[16] F. U. Bernlochner, Z. Ligeti, M. Papucci, and D. J. Robinson, Phys. Rev. D **95**, 115008 (2017).

[17] T. D. Lee, Phys. Rev. D **8**, 1226 (1973).

[18] G. Branco *et al.*, Physics Reports **516**, 1 (2012), Theory and phenomenology of two-Higgs-doublet models.

[19] S. Fajfer, J. F. Kamenik, I. Nišandžić, and J. Zupan, Phys. Rev. Lett. **109**, 161801 (2012).

[20] *BABAR* Collaboration, B. Aubert *et al.*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **729**, 615 (2013).

[21] *BABAR*, D. Boutigny *et al.*, The *BABAR* physics book: Physics at an asymmetric B factory, in *Workshop on Physics at an Asymmetric B Factory (BABAR Collaboration Meeting) Pasadena, California, September 22-24, 1997*, 1998.

[22] J. Siek, L.-Q. Lee, and A. Lumsdaine, *The boost graph library: user guide and reference manual* (Addison-Wesley, 2002).

[23] L. Breiman, Mach. Learn. **45**, 5 (2001).

[24] F. Pedregosa *et al.*, Journal of Machine Learning Research **12**, 2825 (2011).

[25] M. S. Andersen, J. Dahl, and L. Vandenberghe, CVXOPT: A Python package for convex optimization, http://cvxopt.org/index.html (2015).

[26] B. W. Silverman, Chapman & Hall, London (1986).

[27] C. J. Stone, Ann. Statist. **12**, 1285 (1984).

[28] A. G. Gray and A. W. Moore, Proceedings of the third SIAM international conference on data mining. (2003), http://dx.doi.org/10.1137/1.9781611972733.19.

[29] J. Bentley, Commun. ACM **18**, 509 (1975).

[30] D. Scora and N. Isgur, Phys. Rev. D **52**, 2783 (1995).

[31] CLEO Collaboration, J. E. Duboscq *et al.*, Phys. Rev. Lett. **76**, 3898 (1996).

[32] A. K. Leibovich, Z. Ligeti, I. W. Stewart, and M. B. Wise, Phys. Rev. D **57**, 308 (1998).

[33] F. U. Bernlochner, Z. Ligeti, and S. Turczyk, Phys. Rev. D **85**, 094033 (2012).

[34] BABAR Collaboration, J. P. Lees *et al.*, Phys. Rev. Lett. **116**, 041801 (2016).

[35] Y. Ganin *et al.*, J. Mach. Learn. Res. **17**, 2096 (2016).

[36] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, CoRR **abs/1612.05424** (2016), 1612.05424.

[37] E. Zhong, W. Fan, Q. Yang, O. Verscheure, and J. Ren, Cross validation framework to choose amongst models and datasets for transfer learning, in *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III*, ECML PKDD'10, pp. 547–562, Berlin, Heidelberg, 2010, Springer-Verlag.