Falcon: towards an ultra fast non-parametric detector simulator

S. Gleyzer, R. D. Orlando, H. B. Prosper, S. Sekmen and O. A. Zapata

Abstract
We describe preliminary work towards a self-tuning non-parametric detector simulator that maps events at the generator level directly to events at the reconstruction level. The idea is not new. One such tool, `TurboSim`, was developed at D0 and CDF more than a decade ago. What is new is the scope of what is proposed and the opportunity to capitalize on new algorithms for creating the mapping. The ultimate goal is to increase substantially the rate at which events can be simulated relative to that offered by state-of-the-art programs such as `Delphes`, while eliminating the need to implement the mapping by hand.

1 INTRODUCTION

Until compelling evidence of new physics is found that focuses the scope of theoretical models, we shall continue to face the daunting task of comparing thousands of experimental results with the predictions of thousands of theoretical models, a challenge that is being addressed by a number of groups in a variety of ways (see, for example, Refs. [49, 52, 200, 511, 512]).

Broadly speaking, there are two approaches to compare experimental results and theoretical predictions. One can either unfold detector effects from experimental results and compare the unfolded results directly with the predictions or fold the theoretical predictions with detector effects and compare the folded predictions with experimental results. There are pros and cons for both approaches. On the whole, however, folding results is preferred if only because it is technically easier to fold than to unfold when experimental results are multidimensional. However, the price to be paid is computation time and the inconvenience of needing codes that are generally not publicly available. Moreover, even if the codes were readily available, their use typically requires knowledge and expertise not available to those outside the experimental collaborations.

The basic task in the folding approach is to approximate the multidimensional function

$$p(\text{r-particles}|\theta) = \int R(\text{r-particles}|\text{particles}) H(\text{particles}|\text{partons}) \times P(\text{partons}|\theta)\, d\text{particles}\, d\text{partons}, \tag{1}$$

the probability density to observe a collection of reconstructed particles (r-particles) given a point θ in the parameter space of the physics model under investigation. The probability density $P(\text{partons}|\theta)$ represents the theoretical prediction at the parton level for a given θ, $H(\text{particles}|\text{partons})$ represents the mapping from the parton to the particle level, that is, the hadronization, and $R(\text{r-particles}|\text{particles})$ represents the detector response to, and reconstruction of, the particles that enter the detector.

Sometimes it is computationally feasible to approximate Eq. (1) semi-analytically, in the so-called matrix element methods[1]. However, routine use of this method requires highly parallel computing systems [513]. Furthermore, current implementations approximate the detector response function with empirical functions that may not fully capture non-Gaussian effects. In practice, if an accurate rendering of detector effects is needed, the only feasible method is simulating the detector effects in detail using a Monte Carlo method. Unfortunately, the Monte Carlo approach can become prohibitive in terms of computation time if the detector response and event reconstruction must be simulated for tens to hundreds of thousands of events at thousands to hundreds of thousands of points in the parameter space of a multi-parameter model (see for example, Refs. [514, 515]). Moreover, as noted above, the required codes typically remain out of reach of physicists who are not members of the experimental collaborations.

These difficulties have spurred the development of fast, publicly available, detector simulators in which, as in the matrix element method, the detector response function R is approximated parametrically. But, in contrast to the matrix element method, the detector response function is used to create simulated *events* at the reconstruction level. The `Delphes` package [56] is generally regarded as the state-of-the-art in this approach. `Delphes`, as well as the fast simulators internal to the experimental collaborations, starts with simulated events at the particle level and replace the detailed time-consuming Monte Carlo simulation of the detector response by random sampling from R, which is a considerably faster procedure.

The principal difficulty with this approach is the need to hand-code the *form* of the detector response function. Should the detector change because of upgrades or changing experimental conditions, or if non-Gaussian effects become important, the response function will have to be re-coded to reflect these changes. Moreover, the form of the response function could differ from one experiment to another.

However, it is possible to create a program like `Delphes` that does not require the hand-coding of the detector response function. Such programs, `Falcon` and before that `TurboSim`, capitalize on the fact that the millions, and indeed billions, of events that are fully simulated by an experimental collaboration collectively encode the detector response function. The task is to extract a non-parametric representation of it.

2 FALCON

2.1 Introduction

The basic idea of the non-parametric approach is to represent the detector response function as a huge, highly optimized, lookup table that maps objects at the parton or particle level to objects at the reconstruction level. To the best of our knowledge, the first successful example of this general approach, which was used to speed up the simulation of particle showers in the D0 calorimeter, was pioneered by the late Rajendran Raja [516,517]. Similar approaches have been implemented in other experiments [518–520]. The first application of this approach, this time to the subject of this paper, namely the fast simulation of detector responses to particles, was pioneered by Bruce Knuteson [521] who developed a program called `TurboSim`. The program we propose to build, `Falcon`, can be viewed as an updated version of `TurboSim`.

[1] "So-called" because *all* our methods are matrix element methods!

2.2 The design of `Falcon`

The `Falcon` package comprises two components. The first, the `builder`, abstracts the detector response function from existing fully simulated events and creates a database containing a non-parametric representation of the function. The second, the *simulator*, uses this database to simulate events at the reconstruction level from events at the parton level; that is, the simulator approximates the product $R(\text{r-particles}|\text{particles}) \times H(\text{particles}|\text{partons})$. A key assumption, which underlies the matrix element method, all current fast simulators, as well as `Falcon`, is that the function which maps events from the parton level to the reconstruction level factorizes into a product of functions each of which map individual objects from one level to the other.

The first design question to be settled for `Falcon`, which was discussed at the Les Houches meeting, was whether it makes physical sense to map from partons directly to objects at the reconstruction level. The point is that the quantum nature of the particle interactions places a limit on the validity of the strictly classical notion of a well-defined parton-to-particle history. Nevertheless, as discussed at the meeting, it is possible to effect a mapping from partons to reconstructed particles provided that the partons are first clustered using any infra-red safe algorithm [522]. Once clustered, the parton jets can then be matched to jets at the reconstruction level. An important design feature of `Falcon` is that these jets can be of any flavor: electron, muon, tau, W and Z bosons, Higgs boson, top, bottom, charm, or light quark. The ability to map a parton jet of any flavor to its reconstructed counterpart will become increasingly important as more and more analyses at the LHC make use of boosted objects.

The second design question to be addressed is at what level should the parton jets be formed? Here the answer is clear: the jets should be formed at the pre-hadronization stage, but after the partons have been showered. However, a key design feature of the `Falcon` builder is that it should be agnostic with respect to the stage to which the event has been simulated. That is up to the user. What is key is that a jet algorithm must be run on the event in order to create a physically well-defined final state, which prompts a third design question. Should the execution of the jet algorithm be the responsibility of the `Falcon` simulator or of the program that generates the parton level events? We are inclined to argue that it should be the responsibility of the event generator to provide parton-level events with well-defined final states. After all, these final states together with the associated reconstructed events are the inputs to the `Falcon` builder.

The fourth design question is how are parton jets to be matched to their reconstruction level counterparts? For the first version of `Falcon`, we propose a simple proximity criterion: a parton level object and a reconstruction level object are matched if $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} < R_{\text{cut}}$, where $\Delta\eta$ and $\Delta\phi$ are the differences, respectively, between the pseudo-rapidities[2] and azimulthal angles of the parton and reconstruction level objects (e.g., jets) and R_{cut} is a cut-off that may be flavor dependent.

2.3 A proof of principle

`Falcon` does not yet exist as a useable program. However, we have exercised a prototype of the lookup table to reconfirm that the idea works and thus reprised the encouraging results obtained with `TurboSim` a decade ago.

Any fast simulator for the LHC is expected to do a good job simulating electrons and muons since these particles are measured with high precision at CMS and ATLAS. Therefore,

[2] $\eta = -\ln\tan\theta/2$

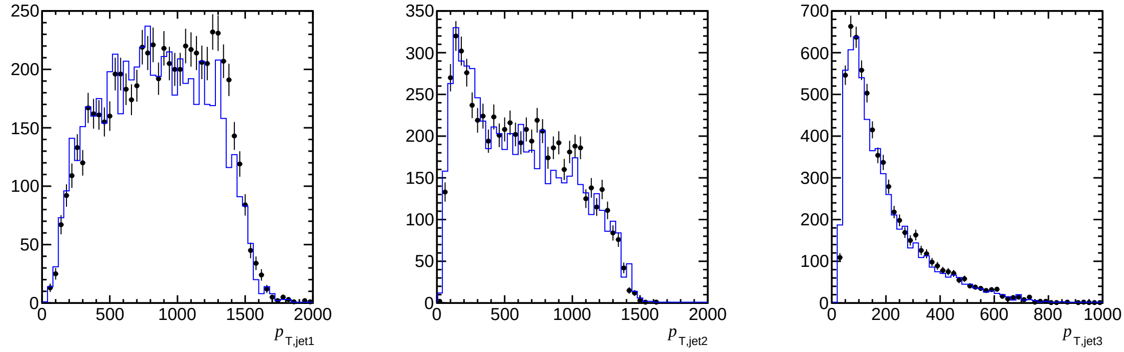

Figure 1: Transverse momentum (p_T) distributions of the three highest p_T jets in $p + p \to H \to f\bar{f}$, where H is the heavy neutral scalar Higgs boson (with mass 2.9 TeV) and f, more than 60% of the time, is either a tau or a bottom quark. As expected from kinematic considerations, the transverse momentum cuts off at approximately half the mass of the parent particle. The distribution depicted with the points is obtained with the full-blown simulation (mimicked using `Delphes`), while the histogram is obtained using `Falcon`.

in our preliminary study, we focus on jets; in particular, on bottom and tau jets. We consider a heavy neutral scalar Higgs boson of mass of 2.9 TeV created in proton-proton collisions at 13 TeV, which subsequently decays to bottom quarks 50% of the time and to taus 12% of the time. The goal of the exercise is to reproduce the transverse momentum (p_T) spectra of the three highest p_T jets using `Falcon`.

Three sets of events are generated (without pileup) at 13 TeV: 10,000 $p + p \to t\bar{t}$ events and two sets of 10,000 $p + p \to H \to f\bar{f}$ using `Pythia 8.2.09` [61] and its default settings. We use `Delphes 3.3.0` [56] to mimic a full-scale Monte Carlo simulation of the response of the CMS detector. The $t\bar{t}$ sample and one heavy Higgs sample are used to create a map between the `Delphes` objects `GenJets` and `Jets`[3]. Different samples are used in an attempt to populate a large range of jet transverse momenta. A `GenJet` is matched to a `Jet` if $\Delta R < R_\mathrm{cut} = 0.35$. This results in a table with approximately 100000 `GenJet` objects most of which are matched to `Jets`. In a realistic application, such a table would be populated with millions of jets.

The core of Falcon is one or more lookup tables. The lookup table in our exercise comprises two components. The first is a k-d tree [523], binned in the `GenJet` quantities (p_T, η, ϕ), which associates a unique index to every `GenJet`. The second component is a map, which given a `GenJet` index maps a `GenJet` to the associated `Jet`. We assess how well the lookup table works by running a mockup of the `Falcon` simulator on the second sample of heavy Higgs boson events. The detector response is simulated as follows. For each `GenJet`, its closest match is found in the k-d tree together with its index. Given the index of the `GenJet`, we retrieve the associated reconstruction level jet from the map.

Figure 1 shows the result of this exercise. This primitive version of `Falcon` is seen to do a reasonable job of reproducing the reconstruction level transverse momentum distributions of the three leading jets. As expected, the transverse momentum of the jets from the heavy Higgs boson cuts off at roughly half the boson mass.

[3] A `GenJet` is a jet constructed at the particle level, while a `Jet` is a jet at the reconstruction level.