From (Fundamental) Theory to Inclusive Signatures
(a primer in practical phenomenology)
Draft version 1.01

Jacob L. Bourjaily, Gordon L. Kane, Piyush Kumar, and Ting T. Wang
Michigan Center for Theoretical Physics,
University of Michigan, Ann Arbor, MI 48109, USA

Brent D. Nelson
Department of Physics and Astronomy,
University of Pennsylvania, Philadelphia, PA 19104, USA

Lian-Tao Wang
Jefferson Laboratory of Physics, Harvard University, Cambridge, MA 02138, USA

Contents

I. INTRODUCTION

One of the most important goals of particle physics is to learn, experimentally, what theory extends the Standard Model (SM) and strengthens its foundations. While some clues about the underlying theory may come from indirect information such as rare decays, magnetic dipole moments, electric dipole moments, proton decay, or detection of a weakly-interacting relic dark matter candidate, it will be crucial to *directly* observe (or not observe) any new particles in any extension of the SM. This could happen at the Tevatron collider at Fermilab and is very likely to happen at the LHC at CERN. Theories with low-energy supersymmetry are arguably the strongest candidates for physics beyond the SM. Low-energy supersymmetry is the focus of much experimental and theoretical progress, and there are strong reasons to expect that it will soon be confirmed by experiment. Therefore, for the purposes of this primer, we will assume that the new physics is supersymmetry—though much of our discussion would be relevant to the study of any other type of new physics which may be discovered. The experimental discovery of supersymmetry at these machines would be manifest by collider signals indicating the existence of superpartners of the SM particles. Since supersymmetry is a broken symmetry, the masses of the superpartners are not known; nevertheless, by requiring that broken supersymmetry does not re-introduce quadratic divergences, the form of the full (softly-broken) minimal supersymmetric Lagrangian, $\mathscr{L}_{\text{soft}}$, is known; and for a given Lagrangian all superpartner production and decays can be calculated.

The natural question which one would like to answer once there is data is: what is $\mathscr{L}_{\text{soft}}$? More importantly we would like to find out:

- how is supersymmetry broken?
- what is its mediation mechanism?
- what is the underlying theory giving rise to such a mechanism?

The soft supersymmetry-breaking Lagrangian $\mathscr{L}_{\text{soft}}$ encodes information that could point to the answers to those questions, if it could be measured.

The next decade will likely host some of the greatest challenges and greatest triumphs of particle physics. Even with ideal experimental data, however, the challenges involved will be enormous. It will take a lot of clever work to connect the data from a hadron collider to an underlying high-scale theory. There are many tools needed to interpret or predict data for hadron colliders, and these can be quite cumbersome to learn without active communication with the experimental and phenomenological communities that have created and mastered them. We have spent time trying to connect and integrate all of the (computational, phenomenological, theoretical) tools necessary to communicate between hadron collider phenomenology and underlying theory. From many questions at talks and in discussions with experimenters, string theorists, Standard Model theorists and others, we suspect that a ‘primer’ describing these tools would be useful to a wide audience of theoretical physicists. Although there exist devoted groups of experts in almost every field along the communication pathway, a global understanding of the entire problem and a proper appreciation for the many subtleties involved in addressing these issues is lacking in the present literature. A number of people have encouraged us to make a primer such as this available.

We would like to emphasize that although this primer contains a discussion of most relevant topics, it is *not* meant to be an expert analysis of any particular topic. It should be stressed that this primer is meant for interested non-experts. Another goal is to make different sections of the community aware of (and appreciate) each other's work and (more importantly) to encourage their working together.

Before giving an outline, a few words about the world-view which the primer assumes is in order. We assume that there exists an underlying microscopic theory which provides an ultra-violet (UV) or high energy scale completion to the effective theory constructed at low energies. This UV-complete theory is, in the back of our minds, taken to be string theory since string theory is capable of addressing all basic physics questions, and constructing concrete models based on string theories is a healthy area of activity. Even though we will try to obtain information about the low-energy Lagrangian from data, we maintain the viewpoint that the low-energy Lagrangian has a microscopic origin in terms of some underlying high-scale physics and that we ultimately want to learn about the underlying theory.

For most of the phenomenological analyses in various sections, it is assumed that the particle spectrum is that of the *Minimal Supersymmetric Standard Model* (MSSM).[1] This is a limitation in two respects: theoretical and phenomenological. From a theoretical point of view, most semi-realistic string constructions which give the MSSM matter content also give rise to other matter fields (exotics). Therefore, a proper analysis of these string constructions requires us to take these exotics into account; for example, they can affect RGE running. Phenomenologically, the MSSM appears to have what is known as the 'little fine-tuning problem,' which may indicate the full low scale theory is in an extended MSSM. Hence, one would like to analyze matter and perhaps gauge content beyond that of the MSSM. This lack of generality is a limitation of the existing phenomenological (computational tools), and it is quite non-trivial to generalize existing software packages. This problem exists, in varying degrees, for all existing RGE evolution, matrix-element computation, and collider simulation software, and will be discussed in more detail in section III A 1. It should be kept in mind, though, that all extensions of the MSSM will contain the MSSM, so most of the results remain useful.

From a conceptual point of view, one would like to start from an underlying theoretical construction (with a realistic spectrum), deduce its field theory effective action, incorporate a supersymmetry breaking mechanism while stabilizing the moduli, compute the low-scale Lagrangian and then derive and predict its phenomenological features. Of course with data one wants to do exactly the opposite, deduce the low-scale Lagrangian from the data and

---

[1] The MSSM is the supersymmetric extension of the SM, with a superpartner for every particle, with the $SU(3)_c \times SU(2)_L \times U(1)_Y$ gauge group, two Higgs doublets, and conserved R-parity. In general, masses cannot yet be computed from first principles, and the superpartner masses can be complex and non-flavor diagonal, and so it contains 105 new parameters all of which can be measured in principle—perhaps requiring a linear collider and rare decays.

then infer the theory from the structure of the Lagrangian. That is, solve the *inverse problem.* The primer is organized keeping this in mind.

Section II is an introduction to experimental and phenomenological particle physics (from a theorist's perspective). This is written for 'laymen theorists' with relatively little experience with experimental high-energy physics. In order to test any high-scale theoretical construction, its implications for low-scale phenomenology must be deduced in order to be compared with experiment. The details of connecting high-scale theory to low-scale phenomenology are spelled out theoretically, conceptually, and computationally in section III. The communication pathway between high-scale theory and hadron collider phenomenology is one that makes use of a wide variety of specialized (and esoteric) software tools. In section III we describe this process explicitly, including examples and detailed instructions for some of the existing software packages that must be employed; these include software for renormalization group (RG)-evolution to the low-scale, matrix-element calculation, and collider event generation. We introduce the reader to some software packages at a level that should be understood by even those uninitiated in computational high-energy physics.

It is increasingly clear that with hadron colliders alone [2] many or perhaps all Lagrangian parameters cannot be directly measured (unless one makes dangerous assumptions about knowing the correct theory and a number of mass parameter values in advance). Therefore, we follow a path of particular interest for us, and in section IV introduce the concept of 'inclusive signatures'—basically any quantity that can really be measured—explain their importance and list their advantages, followed by a general 'recipe' for constructing what is known as the 'inclusive signature footprint' for a given high-scale model. By exploring the entire range of possible low-scale phenomenology (encoded by the 'inclusive signature footprint') for a particular high-scale framework, we will be able to rule out entire classes of models when data becomes available. This will be very useful on the path to confirming other models.

This is the first version (1.01) of this primer and is quite preliminary. We intend to improve it in the near future. The purpose of the primer is to serve as a helpful introduction to important issues in high energy physics which need to be addressed once there is data. We include a lot of information that is known to many people, but not to all. We wish to give a broad overview of the various topics and have attempted to point out open questions. In doing so, we have tried to summarize results from several subfields of high-energy physics based on years of hard work; we hope that the experts in each field find our discussions amusing and acceptable (and not laughable). We welcome any help to improve this primer: further references, contributions, and insight would be greatly appreciated. We apologize in advance to all those authors whose work was not addressed or given the credit it deserves; we will try to incorporate suggestions in future versions. Everything in this primer is in response to questions we have been asked.

---

[2] Unfortunately, in practice linear collider data could not be available until over a decade from the expected initial LHC data.

II. A THEORIST'S INTRODUCTION TO EXPERIMENTAL COLLIDER PHYSICS

The only *direct* way to test theories beyond the Standard Model (BSM) of particle physics is by probing the physics of high energies with colliders. If low-scale supersymmetry is indeed a feature of Nature then the next decade of physics at the LHC and Tevatron may be filled with discoveries, insights, and mysteries. Buried within the data from these hadron colliders will be the keys to understanding supersymmetry, supersymmetry-breaking and perhaps even point us in the direction toward a particular class of high-scale theory (e.g. string theory).

Ultimately, no matter how beautiful or exotic or exciting a particular theory may be, it all comes down to jets, leptons, photons, and missing energy. For a theorist interested in probing a particular model, it is the distributions of these concrete objects that should be predicted because it is the distributions of jets, leptons, photons, and missing energy that will be directly observed at colliders.

The technical and computational challenges involved in making these predictions are substantial for even the simplest models; but many of the necessary computational tools for such an analysis do exist and are publicly available. Indeed, there exist active, specialized communities devoted to the production and maintenance of software needed to evaluate renormalization group evolution, compute matrix elements, and simulate physics 'events' at hadron colliders. This edifice of tools, technologies, terminology, and techniques can be somewhat intimidating to someone beginning in the field. It is the purpose of this primer to better acquaint those interested in testing theories at hadron colliders with the tools and techniques necessary to do so.

A. Physics at Hadron Colliders

For at least a decade, the only direct probes of BSM physics will be hadron colliders[3]: the Tevatron at Fermilab and the LHC at CERN. Therefore, we will restrict our attention exclusively to the physics and tools relevant to hadron colliders.

At the outset, there are many challenges that must be overcome to deduce the underlying theory responsible for data at a hadron collider. Unlike leptons in a linear collider, hadrons are not fundamental particles: they are composed of quarks and gluons, each carrying an unknown fraction of the parent hadron's momentum. The objects which actually 'collide' producing an 'event,' therefore, are uncertain in type, momentum, polarization,[4] and energy. Therefore, very little can be deduced from any individual event; rather, one

---

[3] In this primer, we will focus exclusively on the phenomenology of hadron colliders. Indeed, a linear collider would be able to avoid many of the difficulties and statistical limitations of a hadron collider; but we do not expect the particle physics community to wait for a linear collider to deduce as many conclusions as possible form the LHC if new physics is observed.

[4] The polarization of incoming particles is not particularly relevant to hadron colliders not only because the polarization of the constituent partons is unknown, but also because there do not exist ways to achieve any high degree of polarization in a circular collider.

must marginalize over the statistical distributions of momentum and energy of the various components ('partons') of the incoming protons (or anti-protons). The relevant distributions are deduced from data and are called *parton distribution functions* (PDFs). They are incorporated in programs discussed below.

Therefore, although the LHC will collide protons with center-of-mass energy 14 TeV, only a fraction of this energy is available to the partons which actually interact, and the precise initial energy and momentum of any particular event is not known. This is a stringent limitation of hadron colliders. Indeed, virtually none of a theory's underlying parameters are directly 'observable' at hadron colliders, and many may not be deducible even in principle, i.e. independently of other model parameters or model-dependent assumptions [1] .

1. Cuts and Triggering

When the two proton beams of the LHC 'cross' each other, an average of 18 individual interactions will produce roughly 10-100 charged particle tracks in the detector. This happens about forty million times per second at relatively low luminosity[5]. It is not plausible (and in some sense not even physically possible) to record all of this information. Indeed, pushing the very limits of technology, only about one out of every ten million collisions can be recorded: the rest must be forgotten. This is itself not as bad as it first may sound: the vast majority of the collisions are merely confirmations of what we already know in physics—they are the 'uninteresting' SM background.

No human could decide on a case-by-case basis whether or not an event is sufficiently 'interesting.' What is and what is not recorded by particle detectors is decided beforehand and programmed into the threshold conditions of the triggering system which operates in real-time. It is broken into two primary stages[6]: the low-level ('level-one') trigger and the high-level ('level-two') trigger. The low-level trigger is often built into the hardware of the detector while the high-level trigger is often controlled by software. To be recorded (and thus be available for later analysis) the particles produced in the event—jets, leptons, or photons—must satisfy the conditions of both sets of triggers. Only those models which satisfy the cuts of a detector's trigger system can be studied. Therefore, a model can only be truly compared with experiment if its predictions include BSM signals that pass the triggers of particle detectors.

As an example, consider the triggering system for the CMS experiment to be run at CERN. Its level-one trigger decides which events to consider recording by first identifying isolated photons, electrons, muons, jets and taus via the imprint these leave in the detector's pixel layers, tracking chambers, calorimeters, and muon spectrometers. Electronic signals

---

[5] Luminosity is a measure of the beam intensity, or collisions per second. It is usually measured in $\text{fb}^{-1}/\text{sec}$; these units are particularly convenient for phenomenology: the (time) integrated luminosity then is a measure of the physics reach of the collider in fb^{-1}. To put this in perspective, at low luminosity—during the first year (nine months) of operation—the LHC expects to achieve approximately 10 fb^{-1} of data. At design luminosity, they anticipate $\mathcal{O}(100)\text{fb}^{-1}$ per year.

[6] Sometimes experimentalists discuss a third level of triggering, which is done by a physicist after the data has been stored.

Signature	Threshold (LVL1)
Muon (inclusive)	14 GeV
Two muons	3 GeV (each)
Electron or photon (inclusive)	29 GeV
Two electrons or photons	17 GeV (each)
Tau (inclusive)	86 GeV
Two taus	59 GeV (each)
1 Jet	177 GeV
3 Jets	86 GeV (each)
4 Jets	70 GeV (each)
Jet + $\not{E}_T$ (inclusive)	88 GeV + 46 GeV (respectively)
Jet + electron (inclusive)	45 GeV + 21 GeV (respectively)

TABLE I: Rough values of the CMS level-one trigger thresholds [2].

from these detector systems are passed to the high-level trigger only if the event satisfies the threshold conditions. A list of these threshold conditions is given in Table I.

Only about one out of every 50 thousand events is expected to satisfy the threshold conditions listed in Table I, or about 16 thousand events per second. This is still substantially more than could reasonably be stored, and so further cuts must be imposed by high-level triggers. The conditions to pass the higher-level triggers are much more stringent: for example, the threshold to pass the CMS high-level trigger for events involving at least one jet plus missing transverse energy is 180 GeV and 123 GeV, respectively; this should be compared with 88 GeV and 46 GeV in Table I. These cuts reduce the data rate recorded to tape by an additional factor of a hundred, to a modest one hundred events per second. Even after all of these severe cuts have been imposed the rate of data storage is still staggering: roughly 10^6 Gigabytes per year, one that required a quantum leap in computing to be possible.

2. Events*: Jets, Leptons, Photons and Missing Energy*

Quite literally, any model is observed by the number of leptons, jets, photons, etc. it produces, in what combinations, and with what distributions. Nothing else about a 'model' is ever observed. In order to understand and calculate what a particular model predicts for data at hadron colliders, it is worthwhile to review what is meant by "leptons," "jets," and "missing energy."

Of all the particles produced in an interaction, those originating from the primary event vertex or rapid decays are the most relevant to particle physics. These particles are the most energetic in the event and are called "hard" or "prompt" to distinguish them from those produced in secondary decays or radiative emission.

To understand how leptons and jets are defined, we introduce the coordinates used in experiments. The beam direction is usually chosen to be along the z-axis. At a hadron collider, the momentum of a particle is specified by (p_T, η, ϕ) instead of (p_x, p_y, p_z), where p_T

is the particle momentum projected to the transverse (x-y) plane, ϕ is the usual azimuthal angle and η is the pseudo-rapidity: $\sinh\eta = \cot\theta$ where θ is the angle between the particle moving direction and the z-axis.

Fragmentation of quarks and gluons[7] gives rise to *jets*. A "jet" is defined as a cluster of hadrons moving along a similar direction. To find a jet in an event algorithmically, first choose a direction (η_0, ϕ_0) as a candidate jet direction. Of course, η_0 must be in the range covered by hadron calorimeters—which are used to detect hadrons. Next, a cone around this direction is defined by specifying the radius $R \le \sqrt{\eta_0^2 + \phi_0^2}$. Then one finds all the hadrons moving in the same direction inside this cone and calculates the sum total of the traverse energy for these hadrons: $(E_T)_{\rm jet} = \sum_i E_T^i$ where i runs over the hadrons inside the cone of radius R. If E_T is greater or equal than a pre-specified value $E_T^0(\rm jet)$, this cluster of hadrons is then called a 'jet.' In practice, many subtleties arise from how to choose the starting direction, (η_0, ϕ_0), how big the radius R is taken to be, and what minimum transverse energy $E_T^0(\rm jet)$ should be used. In addition to these subtleties, it is highly non-trivial to calculate the momentum of the initial quark or gluon from the jet. Since some particles (e.g. neutrinos, K_L, etc) will always escape being detected, the observed energy of the jet is always less than the original quark or gluon energy. In the analysis of the data, corrections are applied to make the jet energy as close as possible to the original quark or gluon energy. This is the purpose of what is known as *jet reconstruction*.

The word "lepton" usually refers only to electrons and muons—the only leptons that can be observed directly in detectors: if a tau lepton is produced, it will decay immediately, with about a 65% chance to hadrons and a 35% chance to an electron or muon; thus, a tau lepton is not detected directly. For muons and electrons to be detectable, their pseudo-rapidity η_ℓ must be in the range covered by the detector. In addition, a minimum lepton transverse momentum is always required to activate the trigger, as described above. It is also quite common to require the visible activity around a lepton to be small—so that the lepton is 'isolated' from jets; this excludes many leptons arising from the semi-leptonic decays of b and c quarks. This can be done by summing over the transverse energy of particles inside a cone around the lepton momentum direction with a given radius $R = \sqrt{\eta^2 + \phi^2}$.

Missing (transverse) energy is one of the most important signatures of SUSY. An event is said to have "missing transverse energy" if the vectorial sum of the transverse momenta of all the visible particles in the event is relatively 'large' in magnitude.[8] The missing transverse energy is then the negative of this vector. It accounts for all of the momentum carried by

---

[7] Quarks and gluons are confined by the color force of QCD. This means that they cannot be observed directly in the detector: any quark or gluon produced in a collision will immediately 'hadronize' into jets through a process known as *hadronization* or *fragmentation*. When a $q\bar{q}$-pair are produced with opposite momenta, the confining color field between them can be visualized as a 'flux-tube' or 'string' between the two quarks on the ends of the 'string.' It takes energy to stretch this color tube; and it eventually becomes energetically favorable to produce a new $q\bar{q}$-pair out of the vacuum in the middle of the string so that it 'breaks.' This phenomenological description of hadronization is made mathematically explicit in the framework of the Lund Model of hadronization (which, incidentally, makes use of classical string theory). This model is used, for example, in `PYTHIA` which is described in section III D.

[8] Recall that because the initial momenta of the interacting partons is unknown, so that it is not possible to absolutely determine $\not{E}_T$. However, this becomes relatively less important for large $\not{E}_T$.

particles that were not observed in the detector. If the MSSM with R-parity conservation describes Nature, then many supersymmetric events will display large $\not\!\!E_T$because two stable, lightest supersymmetric particles (LSPs) will escape the detector.

III. FROM A HIGH-SCALE FRAMEWORK TO COLLIDER PHENOMENOLOGY

A. Communication Between the High and Low-Scales—RG Evolution

We normally expect that models of supersymmetry breaking, both in explicit string constructions as well as in more phenomenological approaches, are described by an effective field theory at some high scale M_X. Nevertheless, there do exist supersymmetry breaking models in the literature which are specified only at the low scale. For such models, the question of communication between high and low scales does not arise. Even so, some of the computational techniques from subsections III C and III D can be applied to such models as well.

The high scale M_X is often assumed to be very large (such as near to, but less than, the Planck scale). In the absence of any top-down information it is common to take M_X to be near the scale at which gauge-coupling unification occurs, $M_U \sim 2 \times 10^{16}$ GeV. In more complete models, where the mechanism that transmits supersymmetry breaking from some hidden sector to the observable sector of the MSSM, one often uses the 'messenger scale.' This is typically the mass scale of some non-renormalizable operator in which the messenger couples to the MSSM fields (which may be near the string scale for string-derived models) or the mass of the messenger fields themselves. In many such models these scales are usually some orders of magnitude less than M_U. In any analysis, the precise scale chosen for M_X has phenomenologically relevant implications. The lack of knowledge of these scales is a major problem in going from data to theory. On the other hand, any theory implies definite scales.

Models are built with a few undetermined parameters and are typically classified by how supersymmetry is transmitted to the observable sector. Note that in these phenomenological frameworks the actual origin of supersymmetry breaking is often (but not always) left unspecified. At present, there are many such supersymmetry breaking models in the literature today. Some of the most popular parameterizations of the soft parameters at the high scale are minimal supergravity mediated supersymmetry breaking (mSUGRA), gauge mediated supersymmetry breaking (GMSB), anomaly mediated supersymmetry breaking (AMSB), gaugino-mediated supersymmetry breaking, etc.

No matter what framework in which one works, in order to obtain information about low energy phenomenology, one has to renormalization group (RG)-evolve the soft supersymmetry-breaking parameters from the high (unification) scale to low energies ($\sim$ TeV) and then obtain inclusive signatures. So once one obtains (or chooses) numerical values for the soft terms, the computational tools and techniques required to study supersymmetry-breaking for any particular string theory-inspired high-scale model are virtually the same as those needed for studying phenomenological supersymmetry-breaking parameterizations, even though the two are conceptually very different. Therefore, one should not hesitate to learn all the computational techniques necessary to study the phenomenology of such parameterizations: these tools need only slight modification to be generalized to any particular, microscopically inspired model of supersymmetry-breaking.

The form of the soft Lagrangian is the same at any scale and is described in detail in [3]. A well-known, simplified example of a high-scale parameterization of $\mathscr{L}_{\rm soft}$ is minimal supergravity (mSUGRA), where $\mathscr{L}_{\rm soft}$ is specified at a high-scale (typically the scale of gauge coupling unification) in terms of four unknown parameters and one sign:

- m_0, the universal scalar mass term at the high-scale;
- $m_{1/2}$, the universal gaugino mass term at the high-scale;
- A_0, the universal trilinear parameter at the high-scale;
- $\tan\beta$, the ratio of the two Higgs vacuum expectation values at the low-scale;
- $\text{sign}(\mu)$, where μ is the Higgs mixing parameter.

Minimal supergravity is by far the most actively studied framework of supersymmetric phenomenology and therefore is a convenient framework to test more general ideas. From the point of view of, say, string-based models, mSUGRA is an unlikely outcome.

1. *Computational Tools to Evolve a High-Scale Model to a Low-Scale Soft Lagrangian*

In principle, the connection between the high and low-scale parameters of a general softly-broken supersymmetry theory is a matter of simple computation: the 'hard work' of computing the β-functions for all the parameters in the general MSSM has already been done to second-loop order, (see e.g. [4]). In practice, however, the renormalization group equations require highly nontrivial computational techniques implemented in very specialized software packages[9]. We are extremely grateful for the efforts of those who have contributed to the development and continued maintenance of these software packages. The four most widely implemented software packages to run the parameters of the soft Lagrangian between the high and low-scales are:

1. `SuSpect` [5] available at: `http://w3.lpm.univ-montp2.fr/~kneur/Suspect/`
2. `Isajet` [6] available at: `http://www.phy.bnl.gov/~isajet/`
3. `SoftSUSY` [7] available at: `http://allanach.home.cern.ch/allanach/softsusy.html`
4. `Spheno` [8] available at: `http://www-theorie.physik.unizh.ch/~porod/SPheno.html`

---

[9] Recall the old adage, "In principle, there is no difference between principle and practice; in practice, there is a big difference."

Each of these software packages is capable of running soft parameters for the most commonly studied parameterizations of supersymmetry-breaking (mSUGRA, mAMSB, mGMSB, etc.).

In principle, the low-scale Lagrangian should depend only on the high-scale theory, and not on what software package is used. However, each of the packages use different algorithms to evaluate the renormalization group evolution; these include differences in resummation procedures, numerical approximations, differing levels of precision, etc. Because of these differences, the packages presently can disagree on the order of $\mathcal{O}(10\%)$ for even relatively 'nice' high-scale parameters [9]. For more computationally-challenging regions of parameter space, the discrepancies can be much larger [9]. These differences can have significant effects on the low-scale phenomenology—resulting in sometimes substantially different regions of parameter space that are 'excluded' by experiment. These (computational) theoretical uncertainties should improve with time; but theorists working with any particular package should be cognizant of the accuracy obtained. A more thorough, though slightly out of date comparison of the four software packages is given in [9].

Supposing all of the software packages equally capable of accurately calculating the low-scale Lagrangian for a high-scale theory, there are still important differences between them that one should keep in mind when selecting a particular package:

- Programming language: `SuSpect` and `Isajet` are written (mostly) in the `Fortran77` programming language, `Spheno` is written in `Fortran90`, and `SoftSUSY` is written in `C++`. The `Fortran77` programming language is perhaps the most familiar to the high energy physics community, but it is the oldest. Each of the languages have their shortcomings and advantages; these include availability of compilers, adaptability to more general frameworks, and ease of incorporation with other software packages (e.g. `micrOmegas` or `PYTHIA`). One's individual experience and background in computer programming may lead to a preference for one package over the others. It does not require any 'expertise' in computer programming to install and make use of these packages, although computer programming experience would certainly be advantageous.

- Portability: a phenomenological study of a particular model does not end when the low-scale soft Lagrangian has been computed: the soft parameters obtained with an RGE package almost always become the input of other software packages that compute matrix elements for the theory, its thermal contribution to the LSP relic density, its contribution to the anomalous magnetic moment of the muon and so on. Therefore, it may be preferable to use RGE software already 'built-in' to the other programs one may wish to use. A discussion of these programs, however, is beyond the present scope of discussion.

- Incorporation of particular model frameworks: each of the packages has the capability of restricting attention to the case of mSUGRA. In addition to mSUGRA, each offers other 'built-in' frameworks such as mGMSB, mAMSB, and sometimes even particular string-inspired constructions. The particular variety of built-in models varies between each software package, and it may be advantageous to select the package that already incorporates models of particular interest.

- Degree of generality: unfortunately, no existing RGE software package offers the capability of beginning with the most general soft supersymmetry-breaking Lagrangian at the high-scale. Flavor-non-diagonal couplings and complex phases are almost universally ignored. Until the general high-scale Lagrangian—with complex phases and non-flavor-diagonal couplings—is incorporated in one or more software packages, it will be very hard to study CP-violation or flavor physics in supersymmetry. Importantly, phases can have large effects on CP-conserving superpartner masses, production cross sections, and decay branching ratios. Because of the different computer languages used, these generalizations are easier to make in certain packages than in others. Even ignoring flavor and CP-violating physics, however, the four software packages are not equivalent in their degree of generality; it may be necessary to chose a software package based on its existing level of generality or the ease with which the software's code could be generalized.

Considering that these software packages were written by different people at different times using different methods and different programming languages, it is somewhat amazing (and truly commendable) that their input and output is nearly standardized. This has been accomplished through the design, adaptation, and implementation of the standards specified in the SUSY Les Houches Accord (LHA) [10]. The LHA specifies a specific format for the high-scale input to be given to an RGE software package and the structure for its low-scale output. As more software packages implement the LHA, it becomes dramatically easier for 'external' programs such as those that calculate matrix elements to call upon any RGE package desired. This may soon eliminate any concerns about 'portability' described above.

However, while the LHA promises to make RGE calculations much easier and standardized, its attention is surprisingly restricted to the class of soft Lagrangians that are flavor-diagonal and real (no new CP-violation). We hope that any future Accord will include these fundamental aspects of the physics.

It is presently possible to use software packages for RGE running only for theories which reduce (or can be consistently truncated) to the MSSM. Although this limitation of existing software may seem ephemeral or easily ameliorated, there are at least two reasons why generalizing the MSSM will be very difficult from a computational point of view:

1. In the MSSM, there exist explicit algorithms to test electroweak symmetry breaking (EWSB) conditions; this is not true for more general models. It may not be hard to verify the EWSB conditions for a particular model by hand, but this cannot be implemented into RGE software for any arbitrary beyond the MSSM model.

2. Outside the MSSM, it is not generally possible to fix $\tan(\beta)$ (the ratio of the 'up' and 'down' Higgs' vacuum expectation values)—and hence the Yukawa couplings—*before* starting the RG evolution. Because this is possible in the MSSM, it is relatively easy to maintain consistency with existing experimental data—e.g. the mass of the top

quark. In models beyond the MSSM, there is no handle available to 'tune' these parameters.

These difficulties can usually be addressed by hand on a case-by-case basis; but an explicit, terminating algorithm would be needed for any software implementation. There may be hope that such algorithms could be developed, but there are not enough people working on generalizing existing software to expect these soon.

Because of these limitations, the particle spectrum obtained from (virtually all) string constructions must be truncated to the MSSM. This is done, for example, in the explicit string constructions alluded to earlier. However, we should be aware that this is an important issue that should be addressed in the near future.

2. Installing and Running RGE Software—`SuSpect 2.3`

Although good documentation is available for each of the RGE software packages discussed above, it can nevertheless be quite frustrating and difficult to use one of these tools for the first time. This of course depends much on the computer experience of the user, his or her familiarity with programming, and his or her experience with similar physics software. Some users will find no difficulty installing, compiling, editing, and utilizing these packages directly; others, however, may be discouraged from the beginning.

The purpose of this section is to provide an explicit, step-by-step walk-through of how to install, compile, run, test, and modify one of the RGE software packages, `SuSpect 2.3`, in a Linux environment[10]. This is not meant to be a guide to the use of `SuSpect` or as a substitution for the manual in [5]. Anyone interested in calculating the low-scale spectra for high-scale theories should read [5]. We merely present a '`SuSpect` for Dummies,' if you will, the purpose of which is to acquaint someone with little or no previous experience with `SuSpect` or similar programs with its basic installation and use.

As mentioned earlier, `SuSpect` is written in the `Fortran77` computer language. This computer language is somewhat 'old,' and `Fortran77` compilers[11] are sometimes not packaged with new computers or operating systems[12]. Therefore, it may be necessary to obtain and install a `Fortran77` compiler before you can run `SuSpect`. There are several free `Fortran77` compilers; we recommend the one which is part of the (free) GNU Compiler Collection and can be obtained at `http://gcc.gnu.org/`. It is outside the scope of this primer to discuss the details of obtaining and installing this compiler software.

---

[10] As far as this walk-though is concerned, the principle distinction between a Linux machine and a Windows/OSX machine is the syntax of the commands associated with the `Fortran77` compiler. The translated procedure for an OSX machine is nearly identical to that presented here for Linux.

[11] The `Fortran77` *code* of `SuSpect` must be translated into a language understandable to your computer's hardware—your microprocessor does not speak *Fortran*. This is accomplished with a *compiler*. Every program must be compiled to run, and a great deal of 'academic' (and open-source) software—including `SuSpect`—does not come ready to run out of the box; it must first be compiled.

[12] Many of the (esp. older) Linux machines not uncommon to many physics departments will already have a `Fortran77` compiler installed.

Once you have obtained and installed a `Fortran77` compiler, you are ready to download, compile, run, and check `SuSpect`. This can be done by the following steps (see [5] for details).

1. Visit the `Suspect` web site, `http://w3.lpm.univ-montp2.fr/~kneur/Suspect/`, and download the file `suspect2.tar.gz` to some place on your computer—preferably in a new directory.

2. Open a terminal and change directories to the place in which you downloaded the file `suspect2.tar.gz`. You must now extract the contents of the compressed archive. This can be done with the command:

   ```
   tar -xvf suspect2.tar.gz
   ```

3. Once the files have been extracted, you can compile `SuSpect` with its default settings as a test. This can be done with the following command[13]:

   ```
   g77 -o suspect2 suspect2_call.f suspect2.f twoloopHiggs.f bsg.f
   ```

   This compiles the `Fortran` code in each of the files ending in "`.f`" and creates the executable file `suspect2`.

4. If the program compiled without errors, then you can execute `SuSpect` with the following command:

   ```
   ./suspect2
   ```

The successful execution of `suspect2` will create two files: `suspect2.out` and `suspect2_lha.out`. These two files contain essentially the same information: both include the soft parameters of the low-scale Lagrangian, the contribution to $(g_\mu - 2)$, $BR(b \to s\gamma)$, and an estimate of the fine tuning of the theory. In addition to this information, `suspect2.out` contains warning messages. The only substantial difference between the two output files is that `suspect2.out` is written in a format native to `SuSpect` whereas `suspect2_lha.out` is written in a format in accordance with the LHA. The input data used to generate the output files in this trial run are contained in the file `suspect2_lha.in` by default[14]. It will be helpful to open this file and become acquainted with the way this input information is communicated to the program `suspect2`. You will notice in the 8th line of `suspect2_lha.in` that the input data are "arbitrary soft-terms at low-scale." This implies that little work is done involving RGEs for this trial run. You could compute the spectrum for a mSUGRA model, for example, by changing the entry on the 8th line to read:

```
1        10    ...
```

---

[13] Depending on the compiler used, '`g77`' may need to be replaced with '`f77`' or '`gfortran`' or otherwise.

[14] The file `suspect2_call.f` can be modified so that the program `suspect2` either reads from the input file `suspect2.in` or generates its own input. The sample file `suspect2.in` is completely ignored by `SuSpect` by default—it is the old input format that is not in accordance with the LHA. Of course, any changes made to `suspect2_call.f` must be implemented by recompiling the `SuSpect` code files, with the same command as above.

The '1' in the first column corresponds to the variable 'name'—see the Les Houches Accord [10]—and the second column is its value. The value of '10' in the '1' variable of the '`MODSEL`' block tells `SuSpect` to consider an mSUGRA model. If you look further down in `suspect2_lha.in` you will find that the default mSUGRA model is the benchmark point SPS 1a [11]. If you save the modified input file and re-run `suspect2` with the same command as above, namely,

```
./suspect2
```

then the two output files will be overwritten with the spectrum of the mSUGRA model SPS point 1a. If you wished to explore large regions of parameter space, this could be done by directly modifying the `Fortran77` code in the file `suspect2_call.f`—examples and explanations are provided therein.

Let us now use `SuSpect` to compute the low-scale spectrum of a model specifically related to a string construction. Although this model is completely specified in terms of only a few high-scale parameters, it must be input into `SuSpect` using its "arbitrary soft-terms at high-scale" option (in the `MODSEL` block); this is done setting the variable '1' to the value '1' in the 8^{th}-line of `suspect2_lha.in`. After this is done, the high-sale parameters need to be set. This is done by following the changes specified in Table II. After running `suspect2`, new output will be (over-)written to the file with the name `suspect2_lha.out`. By studying the syntax and format used in this example, you should be able to make all the necessary changes for a model of your own.

TABLE II: This table lists all of the changes that should be made to the file `suspect2_lha.in` in order to compute the low-scale Lagrangian of the intersecting D-brane benchmark model described in section ?? using the `SuSpect` program. The first column is not part of `suspect2_lha.in`, these are listed for the reader's convenience. No changes need to be made to lines not listed here.

Line	Variable	Input	
8	1	1	
⋮	⋮	⋮	
15	6	1	`# 1: M_Hu, M_Hd input (default in ...`
⋮	⋮	⋮	
64	10	2.d16	`# HIGH scale (RGE end) (!only relevant if ...`
65	22	1.4896d6	`# M^2_Hu`
66	21	1.4896d6	`# M^2_Hd`
67	25	25.0d0	`# tbeta`
68	26	1	`# sign(mu)`
69	1	-1281.849962	`# M1 U(1)_Y Bino mass`
70	2	242.4871130	`# M2 SU(2)_L Wino mass`
71	3	109.1192008	`# M3 SU(3)_c gluino mass`
72	33	2438.73	`# M_tau_L`
73	36	1722.47	`# M_tau_R`
74	43	2438.73	`# M_Q_L (3rd gen.)`
75	46	1887.35	`# M_t_R`
76	49	1722.47	`# M_b_R`
77	31	2438.73	`# M_e_L (1st, 2d gen.)`
78	34	1722.47	`# M_e_R (1st, 2d gen.)`
79	41	2438.73	`# M_qu_L (1st, 2d gen.)`
80	44	1887.35	`# M_u_R (1st, 2d gen.)`
81	47	1722.47	`# M_d_R (1st, 2d gen.)`
82	13	-679.34521180	`# A_tau`
83	11	-801.72598321	`# A_t`
84	12	-679.34521180	`# A_b`
85	16	-679.34521180	`# A_e (1st, 2d gen.)`
86	14	-801.72598321	`# A_u (1st, 2d gen.)`
87	15	-679.34521180	`# A_d (1st, 2d gen.)`

B. Checking Experimental Constraints

There are many specific and sometimes stringent limits on any physics beyond the Standard Model. Once the low-scale Lagrangian is known, the model should be checked against all of these experimental limits. The constraints most relevant to low-scale SUSY include the following[15]:

1. The thermal production contribution to the LSP's relic density cannot exceed that of dark matter, which is presently bounded by $\Omega_\chi h^2 \leq 0.129$ at the 2σ level [12]. This calculation can be quite complicated; indeed, there exist quite specialized software packages devoted to this calculation such as `micrOmegas` and `DarkSUSY`. These programs are relatively easy to use and incorporate into a wider framework. We intend to write a detailed walk-through for at least one of these packages in the near future—which will be incorporated into a later version of this review.

   It should be emphasized that *no lower bound on the relic density of the LSP exists experimentally.* (Recall that while cosmological data imply a lower bound on the relic density of cold dark matter, this is indifferent to what fraction of the dark matter is the LSP.)

2. Rare decays, e.g. the branching ratio of $b \to s\gamma$ which is bounded experimentally by $BR(b \to s\gamma) = (3.25 \pm 0.37) \times 10^{-4}$ [13][16].

3. The BSM contribution to the anomalous magnetic moment of the muon should be no more than $\Delta a_\mu \equiv \Delta(g_\mu - 2)/2 = (27 \pm 10) \times 10^{-10}$ from the SM prediction [14].

4. Direct searches for particles: the masses of superparticles should be consistent with the null-results from existing collider searches. The specific bounds are given in the Particle Data Tables [15]. It should be mentioned that these limits should be used with some care: many of the explicit bounds are model-dependent (most of these bounds apply specifically for the case of mSUGRA). The bounds on light superparticles may not be as strong for more general models. For example, there are no general bounds on any of the neutralino masses.

5. Other experimental limits and bounds including: electric dipole moments, flavor-changing neutral currents, $\Delta\rho$, CP-violation, &c. Relatively few of these constraints are checked by default in existing software packages. This reflects the fact that most phenomenological studies—and hence most software implementations—have been done under the assumptions of flavor-diagonal superparticle mass matrices and real Lagrangian parameters; these assumptions automatically ignore all flavor physics and CP-violation so that these experimental limits are not usually checked.

---

[15] The numerical bounds listed below will certainly improve with future experiments; the values given here, while the strongest bounds at present, are only used for illustration. The most up-to-date bounds should be used.

[16] This bound represents the combined data from several different experiments.

Often there are significant theoretical uncertainties involved in the experimental bounds listed above.

Some experimental constraints are built into software packages. For example, the output of the `SuSpect` example in section III A 2 contains the contributions to $\Delta a_\mu, \Delta\rho$, and $BR(b \to s\gamma)$. The spectrum can be directly compared with the bounds listed in the PDG review in [15], although some of these bounds may be somewhat model dependent.

C. From a Low-Scale Lagrangian to Matrix Elements

Once the complete low-scale Lagrangian is known, the next step is to compute all the matrix elements relevant to physics at hadron colliders, applying Feynman rules to calculate cross sections and decay widths. However, there are an enormous number of such calculations required, and each of these involves a large-number of diagrams. Furthermore, these calculations may need to be performed for many choices of parameters. Needless to say, these computations cannot be performed practically by hand. Luckily, there are now very good computational tools that can do this automatically for particular models.

Our cursory discussion here is covered in much greater detail in the appendices of [3, 16]. Any student wishing to study this physics in more detail is encouraged to see the recently published or forthcoming texts [17–19]; in particular, [19] may be quite helpful as it focuses on the connections of theory to colliders. Here we just give a brief overview.

1. Computing Matrix Elements for a Low-Scale MSSM

Again, we will restrict our attention to the MSSM for the same reasons as before. The Lagrangian is often written in terms of family and gauge eigenstates. These will be mixed amongst themselves after SUSY and electroweak symmetry breaking. Therefore, the first step is to find the mass eigenstates. This is done by diagonalizing the following six mass matrices:

- One 4×4 mass matrix of neutralino mass matrix and one 2×2 chargino mass matrix.
- One 3×3 mass matrix of neutral Higgs particles. When CP is conserved (which can only be guaranteed at tree level), the two CP-even and one CP-odd states decouple such that there is only a 2×2 mass matrix.
- Two 6×6 mass matrices for squarks: one for the up-type and one for the down-type.
- One 6×6 mass matrix for charged sleptons.
- One 3×3 mass matrix for sneutrinos.

After getting the mass eigenstates, the effective couplings and Feynman rules and derived, and are given in Refs. [3, 16–19]. If observables from precision measurements are in the inclusive signature study, loop corrections to relevant couplings and masses are also required. With the superpartner spectrum and Feynman rules at hand, we can start calculating matrix elements.

Often, many matrix elements need to be computed, which makes the task quite involved. One reason for this complication is that a superpartner may have many decay channels. To get the total decay width, which is needed to determine the branching fractions, all the decay Feynman diagrams need to be calculated. In addition, several production processes may contribute to an inclusive signature so that all of them must be calculated in order to study the inclusive signatures.

2. Installing and Running Matrix Element Calculation Software—`CompHEP 4.4.0`

We intend to expand on this section for a future version of this primer. Detailed information about `CompHEP` can be found in the `CompHEP` manual [20] and at the website:

`http://theory.sinp.msu.ru/parser/parser.php?file=/phpcms/comphep/`

In brief, `CompHEP` is a program that can be used to calculate matrix elements from an input model, both numerically and analytically. It can also do event simulations so that cross sections and decay width can be obtained.

To input a model in `CompHEP`, one needs to define the following four parts:

- Fundamental parameters in the model: any given model consists a set of fundamental parameters. For example, in QED there are two parameters: electron mass and fine structure constant.
- Constraints: using constraints, a set of parameters which depend on the fundamental parameters can be defined.
- Particles: each particles should be specified by its name, spin, mass, width, etc.
- Vertices: all interactions should be defined in this part. They are essentially the Feynman rules of the model.

Once a model is defined as specified above, `CompHEP` can calculate the matrix elements for almost any user-specified process; this can be done both analytically, with output in `REDUCE` or `MATHEMATICA` format, or numerically, with output in `C` or `Fortran77` format.

Monte Carlo simulation is also implemented in `CompHEP`. This is used, for example, in the calculation of 3-body decay widths. The incoming particles can be either electron-positron or proton-proton. For hadron collisions, `CompHEP` makes use of the experimentally known parton distribution functions. All the final states are fundamental particles which means that quarks or gluons are not fragmented. For many studies, this is already good enough. One can also use `CompHEP` as an event generator. Feeding these events into `PYTHIA` will fragment quarks and gluons.

D. From Matrix Elements to Collider Physics

Once all of the relevant matrix elements have been computed, inclusive signatures can be determined. At first glance, it may appear that once one has determined all of the production cross sections for processes that have particular final event configurations, then it is easy to compute all the inclusive signatures; i.e. if the total cross section for producing SS dileptons is say X fb (as computed by summing all of the possible sub-process-cross sections from `CompHEP`), then there would be $10X$ such events in 10 fb^{-1} of data at the LHC. This naïve analysis is quite insufficient for a number of important reasons.

At the outset, the matrix elements are specified in terms of final states with quarks, gluons, the LSP, neutrinos, tau leptons, etc.—none of which are actually observed in a detector. As described in section II, gluons and quarks decay or fragment into jets; the LSP and neutrinos escape the detector as missing (transverse) energy; and tau leptons immediately decay either hadronically into jets or leptons[17]. Furthermore, these production cross sections and decay widths are not (directly) connected to the angular or kinematical distributions that would be observed at detectors—which are important signatures.

Furthermore, recall that for any event at a hadron collider, the initial momenta and identities of the interacting particles are always unknown. Only the statistical distributions of the particle content of the incoming protons (or antiprotons) are known experimentally; these are called the parton distributions functions.

And perhaps most importantly, one must take into account the triggering conditions and cuts used by actual particle detectors. In order for any event to be observed at a detector—i.e. recorded on tape—all of the triggering conditions must be satisfied. These conditions include lower bounds on the transverse momenta of various particles, phase-space isolation, missing energy estimates[18], etc.; and these conditions are highly correlated to one another and change depending on the configuration of the particles in the event. For example, having energetic muons in the final state can reduce the threshold conditions on other types of particles in the event. Therefore, detectors only observe an extremely biased selection of all possible events—and this bias can only be accounted for if 'triggers' are imposed on theoretical predictions too.

Because decay chains, jet fragmentation, final state distributions, and parton distribution functions are inherently stochastic, and all of these processes combine in a complicated way to determine whether or not an event satisfies triggering conditions and hence observed at all, the best way to predict the physics to be observed at colliders is to simulate it directly, taking all of these stochastic processes and realistic triggers into account. It should not be a surprise that any such calculation is inherently complicated and can only be done using specialized simulation software. This is accomplished using Monte Carlo techniques in software like `PYTHIA`.

---

[17] As elsewhere in this primer, the word "lepton" includes only electrons and muons unless explicitly stated.

[18] There is not enough time for low-level triggers to actually determine the amount of missing energy for a particular event—the computation is too time-consuming. Rather, low-level triggers merely estimate the missing energy using imperfect algorithms. These imperfections should be taken into account to produce realistic data.

1. *Generating Events*

Given all the matrix elements relevant to collider physics—i.e. production cross sections and decay widths—one can simulate the physics that would be observed at hadron colliders. Several good programs exist for this purpose — we will illustrate with PYTHIA.

2. *Installing and Running Event Generation Software*—`PYTHIA`

We intend to expand on this section for a future version of this primer. Detailed information about `PYTHIA` can be found in the `PYTHHIA` manual [21] and on their website:

`http://www.thep.lu.se/∼torbjorn/Pythia.html`

`PYTHIA` [21] is a program for generating high-energy physics events of collisions between elementary particles such as electron, positron, proton and anti-proton in various combinations. The current version is written in `Fortran77`. To use `PYTHIA`, there are three steps:

- Install a `Fortran77` compiler. For those who are using Unix or Linux, this compiler is almost certainly installed. For those who use Windows, free Fortran compilers can be downloaded from the web.
- Download the `PYTHIA` Fortran program. The program is in a single Fortran file. This makes `PYTHIA` installation relatively easier comparing to other programs. There are many subroutines in the `PYTHIA` file.
- Make your own Fortran main program which calls subroutines of `PYTHIA`.

Of course, the main step is the third one. The main program typically has the following parts in sequence:

1. Set the parameters that specify which processes `PYTHIA` should simulate. They are mainly $2 \to 2$ processes. If one studies the MSSM model, the MSSM parameters should be assigned in this part of the program. Final particles can be from the Standard Model, or any superpartner or MSSM Higgs. One can also turns off particular decay modes of particles the speed up the simulation. Parameters characterizing the detector should also be specified here if one wants to use `PYTHIA` subroutines to reconstruct jets from final state particles.
2. Specify the beam properties: the particles of the beam and the center of mass energy.
3. The main loop of generating events. Usually a large number of events should be generated to get a good accuracy. For each event, a list of final state particles is given by `PYTHIA`. Each particle has its name, parent particle and momentum. Based on this list, one can test whether the current event satisfies the cuts used to define a certain event configuration. Counters are usually used to count how many events

pass the cuts. One can look at a complete list of all decays (with their respective branching ratios) for any given superpartner—this feature is very useful for theorists to understand a model better.

4. Get the total cross section and cross sections for each individual event configuration. The total cross section for all the simulated processes are directly given by `PYTHIA`. The cross section of a individual event configuration defined by a set of cuts is

$$\sigma_{\text{an event configuration}} = \sigma_{\text{total}} \times \frac{\text{Number of events that pass the corresponding cuts}}{\text{Total number of events simulated}} \tag{1}$$

A main program outlined as above can be used to compute inclusive signatures using `PYTHIA`. An example of such a main program is given in Appendix A.

IV. INCLUSIVE SIGNATURES

From the discussion in section II, it is clear that it is virtually impossible to deduce lagrangian parameters from hadron collider data independently of other model parameters or model-dependent assumptions. This implies that predictions for hadron colliders should be made in terms of what is actually observed in experiments: jets, leptons, photons, and missing energy. It is meaningful and practical to make predictions about the distributions and production rates of particular event configurations (e.g. events with two isolated, prompt, oppositely-charged leptons, large missing transverse energy, and a pair of jets). The rate of production and kinematical or angular distributions of a particular event configuration is easily observed at hadron colliders, and can be reported with no model-dependent assumptions. If theorists made predictions in terms of these types of signatures, then it would be much simpler to exclude or confirm various types of models. In this mind-set, an "inclusive signature" is anything that can be directly observed at experiments. These types of signatures are *inclusive* because they are naturally include all possible production mechanisms and are indifferent to any particular model-framework.

A particularly robust example of an inclusive signature would be the production rate of a specific event configuration. It is useful to comment that while the *rate* is sensitive to beam luminosity and other experimental uncertainties, these uncertainties could be significantly reduced by considering the ratios of two such inclusive signature rates. A generic event configuration could be characterized as $(m\text{-jet}) + (n\text{-lepton}) + (k\text{-photon}) + \not{E}_T$. When there are two or more (prompt) leptons in the final state, it is useful to notice their relative charges. This is particularly true for the case of dilepton events: same-sign (SS) dilepton events can and should be distinguished from opposite-sign (OS) dilepton events. The total number of events for a given configuration is certainly an inclusive signature. Notice that the theoretical prediction for the production rate of events of a particular configuration is also inherently 'inclusive:' for any given final state configuration, there are typically many different processes which contribute to its production. Some of these will be Standard Model events and may be considered 'background' to some BSM process of interest. *But often multiple BSM process also give rise to the same final state configuration also.* As an example, consider the case of events in which there is missing transverse energy, two isolated leptons (each with sufficient transverse momentum to pass the trigger requirements) and some number of jets. This could be a signal for supersymmetry, caused by the production and decay of a neutralino pair. But this configuration could also have arisen from the production of a pair of charginos, or even of a squark and a gluino; alternatively, it could signal the production of a pair of scalar leptoquarks, and these might be present in a model that also contains the MSSM—as in many explicit string constructions [22]. In fact, *all* of these processes together with Standard Model contributions like $t\bar{t}$-production and electroweak gauge boson production with initial state QCD radiation, may contribute to the observed events in unknown proportions. While judicious cuts may help restrict the sample to just those events coming from a particular source of interest, there is never a perfectly clean sample.

Another type of inclusive signature would be the distribution of some kinematic variable for all events with the same configuration. A good example is the distribution of invariant (transverse) mass of two oppositely-charged leptons for events of opposite-sign dilepton signature. This event sample is easy to collect and the distribution easy to plot. There will be an endpoint to the distribution at some large invariant transverse mass. So far there is no model-dependence. However, if the event sample comes from the decays of the second lightest neutralino $\widetilde{N}_2$ to the lightest neutralino $\widetilde{N}_1$ via a virtual slepton $\widetilde{N}_2 \to \widetilde{N}_1 \ell^+ \ell^-$ then this endpoint gives the mass difference between $\widetilde{N}_2$ and $\widetilde{N}_1$. But if the slepton is light enough to be on-shell, the end-point measures a more complicated function of the three masses involved. Ultimately the best way to test a model against these distributions is to construct them directly from the model using event simulators, factor in the detector efficiencies, add in the background contributions, and then compare directly to the data. This will eventually help to constrain the parameter space theory beyond the Standard Model. It is important to note the distinction being made between inclusive signatures, like the number of events produced of a given configuration, and model parameters, like mass eigenstates of superparticles: the former being a directly observable quantity in the data and the latter being a derived quantity that may not be observable at all. To illustrate the distinction that should be kept in mind, let us consider the case of a “measurement” of the Z boson mass at an e^+e^- collider. From an inclusive signature point of view, the total number of OS dilepton events increases (dramatically!) when the center of mass energy is near a certain value—in this case the Z-boson mass. This is an observation. The *measurement* comes when we try to fit this dramatic increase to a functional form that represents a particle of particular mass and decay width. In this example the relationship between observation and measurement is particularly simple and robust—there is very little model-dependence involved. With hadron collider measurements involving superpartners this is rarely the case.

There are two very important distinctions between the case of Z-boson production at a lepton collider and superparticle production at a hadron collider. First, as discussed earlier, because the quarks or gluons involved in a particular collision posses an unknown fraction of their hadron’s momentum, there is no ‘resonant’ beam energy observed by which one could infer a particle’s (mass eigenstate) mass. Secondly, if the world conserves R-parity, then final states of superparticle production will contain at least two lightest supersymmetric particles (LSPs) so that there is always missing energy, and this missing energy is not associated with a single particle. Thus, in general cases, the masses of superpartners cannot be directly measured and should not be considered inclusive signatures. Certain care must be taken when considering kinematic distributions. In particular, a distribution itself may be an inclusive signature but not the numbers that may be derived from it. Here it is good to distinguish between observations and measurements. Observations are the kinematic distributions—measurements are the parameters that fit it, such as the mass difference between two postulated mass eigenstates. The latter must necessarily be a derived quantity with some degree of model dependence inherent in its extraction. The degree of that model dependence may vary, however, from one 'measurement' to another. Ultimately the best way to test a model against these distributions is to construct them directly from the model using event simulators, factor in the detector efficiencies, add in the background contributions, and then compare directly to the data.

Therefore, inclusive signatures have the advantage of being directly comparable with experiment and suffering from very few model-dependent assumptions. They also generally avoid many of the difficulties that arise from experimental and theoretical uncertainties. And this is important because no measurement (or prediction!) is made without uncertainties: measurements at hadron colliders suffer from both statistical and systematic errors in event reconstruction, energy measurements, parton distribution function calculations, finite statistics, and so on. Many of these errors are reduced or eliminated if: (a) we stick to inclusive signatures (i.e. counting—not kinematics) and (b) we measure ratios of signatures instead of absolute rates. Let us not forget that there are theoretical errors as well that will affect how good our 'predictions' are, against which we compare the imperfect data. While the generators are usually better than the RGE programs, the latter are usually better than the accuracy with which theorists developed the model in the first place!

A. Generating an Inclusive Signature 'Footprint'

In this section we take the liberty of focusing on the approach which we think will be particularly fruitful for learning physics implications of hadron collider data if Nature is indeed supersymmetric. To generate the inclusive signature predictions for any realistic high-scale theory— one needs to evolve the high-scale Lagrangian to the TeV-scale using the renormalization group equations, compute all the matrix elements relevant to phenomenology, and use these to generate a sample of 'events' that would simulate (predict) the phenomenology at hadron colliders (often with a collider and even a particular detector in mind). These predictions could be used directly to either support or rule out any theory. This is the ideal situation.

Since our theoretical understanding of the fundamental, theoretical description of supersymmetry-breaking is still quite incomplete, and a completely realistic model of supersymmetry breaking is not known at present. For example, in string-theoretic constructions, the 'stringy' parameters on which the soft terms depend are not generically known, although the precise functional form of the dependence is known for some explicit constructions. Therefore, in string-based models, one typically obtains a ‘parameter space’ for a given string construction. It is important to understand that the parameters are not *free* in the usual sense, since they must only take values that are consistent with the regime of validity of the approximations involved. In principle these numbers will take on explicit values as determined by a better understanding of string theory.

In more traditional and phenomenological approaches to supersymmetry breaking, the soft terms at the high scale are specified in terms of a few free tunable parameters, which need to be RG evolved to the TeV scale to deduce their low energy phenomenology. Thus, one obtains a parameter space for each model of the above type as well.

When varying the parameters, it is very likely that many combinations of parameters will yield models which are inconsistent with low-energy physics. For example, it is often difficult to find points in parameter space for explicit string constructions that satisfy electroweak symmetry breaking (EWSB) conditions or that get the correct value for the top Yukawa coupling. Even sets of parameters for which these constraints are satisfied may contribute too much to the anomalous magnetic moment of the muon, or produce too much cold dark matter, or predict a too large a branching fraction for $b \to s\gamma$, or too much CP-violation, etc.

Therefore, whatever the ‘high-scale parameter space,’ only a bounded region (or regions) should be compatible with existing experiments. If the low-scale phenomenology of a given set of parameters is consistent with all known experimental constraints, then it is possible that it may describe Nature. In this case, its inclusive signatures should be calculated to be eventually compared with experiment.

The set of all high-scale models for a given construction, or ‘points’ in parameter space, that are consistent with experimental constraints will translate into a set of ‘points’ in inclusive signature space. The union of all of these points is called the *inclusive signature footprint* for the particular high-scale framework being considered. The recipe for obtaining such a footprint is outlined schematically in Figure 1. The inclusive signature footprint—if sufficiently bounded—*completely classifies the range of low-scale phenomenology that could be produced by a particular framework.* This has been done for the case of mSUGRA[19], for example in [23].

The collection of inclusive signatures actually observed at hadron colliders will translate into a single point in *inclusive signature space.* If this point lies outside the footprint for a particular class of models, then that entire class can be immediately ruled out. This shows one significant way to use inclusive signature footprints. They may also give a very powerful theoretical way of studying physics beyond the Standard Model.

To generate the inclusive signature footprint for a particular high-scale framework, one must write a computer program that includes all of the diverse steps connecting high-scale theory to low-scale phenomenology. This can be done by implementing the following general outline.

---

[19] This phenomenological parameterization of the high-scale soft supersymmetry-breaking Lagrangian is described briefly in section III A.

FIG. 1: Recipe, or algorithm, for calculating inclusive signatures for any particular high-scale supersymmetric theoretical framework.

1. Choose a high–scale soft-breaking Lagrangian. This could range from that of the full MSSM to one implied by a particular theory[20]. If all parameters are fixed by the theory, the programs described below allow one to calculate the predictions of that

---

[20] As described briefly in the introduction and in more detail in section III A, it is a limitation of existing software that inclusive signatures can only be studied directly within the framework of the MSSM.

theory, and the footprint is a single point. If some parameters are not determined (or known beforehand), they could be varied over their allowed ranges.

(When first beginning to develop the 'inclusive signature footprint,' i.e. when a relatively small number of (experimentally consistent) high-scale models have been found, the parameters can be chosen either randomly or over a discrete lattice.

After inclusive signatures have been calculated for several thousand models as in step (6) below, new models should be chosen near those on the 'boundary' of the inclusive signature footprint: by perturbing around those models that produce inclusive signatures on the boundary of the footprint, the boundary will become much better resolved.)

2. Run the parameters of the high-scale Lagrangian to the TeV-scale using RGE-software like those described in section III A.

   We again comment that it is not possible today to evolve all of the soft Lagrangian parameters using existing software because there does not exist any RGE software package that has the capability of considering the most general MSSM soft Lagrangian; in particular, phases and flavor physics are almost universally ignored, and some mass degeneracies may be implicitly assumed.

3. Check all relevant experimental constraints. This step is described in section III B. Most experimental constraints can be checked directly using the RGE software of one's choice. However, because existing RGE software implicitly or explicitly ignores flavor physics and complex phases, it may be necessary to implement experimental constraints on CP-violation and flavor changing neutral currents (FCNCs) separately. Also, the effect of the constraints changes when phases are included even for CP conserving quantities. And some constraints such as the LSP's contribution to the relic density of cold dark matter may need to be checked using specialized software such as `micrOmegas` [24] or `DarkSUSY` [25].

   While in a certain sense some of the existing experimental constraints could be viewed as inclusive signatures, this step should be considered somewhat distinct from hadron collider signatures. In particular, many seeming general 'experimental constraints' are actually quite model dependent and therefore not inclusive signatures; this is the case for sparticle masses.

   Furthermore, imposing experimental constraints is qualitatively different from generating hadron collider inclusive signatures: at this stage in the algorithm regions of the high-scale parameter space are cut away from further consideration. This itself can be quite interesting theoretically. Furthermore, these constraints can usually be checked within a relatively short period of time (computationally) as compared to generating hadron collider events.

4. Calculate all the matrix elements relevant to collider physics (e.g. cross sections, decay widths, and branching fractions). Because of the enormous number of relevant

channels and contributing Feynman diagrams, this must be performed with specialized software such as `CompHEP` or `CalcHEP`. These packages are described in section III C. The matrix elements must then be passed to collider simulation software described below.

5. Simulate collider physics phenomenology by generating 'events' that would be seen at a particular collider (and perhaps with a particular detector in mind). The more events generated for a model, the less statistical uncertainty there is in the predictions for hadron collider phenomenology. Therefore, many events—on the order of a few hundred thousand—should be generated using cuts and trigger conditions similar to those used by actual experiments.

   This is perhaps the most computationally-intensive step in the algorithm, requiring up to several minutes for each model. It should not be surprising that this is also one of the most critical steps in the algorithm. Because cuts and triggering conditions can have significant effects on the phenomenology actually observed at colliders, it is important to simulate events using cuts as close to 'actual experiments' as possible, so that the comparison between theoretical predictions and experimental data is can be done robustly.

   In section III D, we describe how this step can be implemented in the framework of the program `PYTHIA`.

6. Calculate and tabulate inclusive signatures. Using the events simulated with `PYTHIA`, inclusive signatures can be directly computed. Because an 'event' in the output of `PYTHIA` contains all the relevant data that would be seen at a real collider (and sometimes a bit more[21]), the events can be analyzed to determine any particular inclusive signature of interest: e.g. the number of SS dilepton events with 3 jets and missing energy, the distribution of the invariant mass of two SS dileptons in events with any number of jets and missing energy, etc.

   All of the inclusive signatures should be saved together with its high-scale parameters. This set of inclusive signatures then contributes to the footprint of the entire class of models.

7. Repeat.

---

[21] While simulating data, the computer 'knows' all of the details that produced a particular event. This type of information should *not* be used when calculating inclusive signatures: if an experiment cannot remove certain background sub-processes, then neither should the theorist.

V. ACKNOWLEDGEMENTS

The research of JB, GLK, PK, BN, LW and TTW are supported in part by the US Department of Energy.

[1] M. Brhlik and G. L. Kane, Phys. Lett. B **437**, 331 (1998) [arXiv:hep-ph/9803391]; G. Kane, arXiv:hep-ph/0504257.

[2] F. Gianotti and M. L. Mangano, [arXiv:hep-ph/0504221].

[3] D. J. H. Chung, L. L. Everett, G. L. Kane, S. F. King, J. Lykken and L. T. Wang, Phys. Rept. **407**, 1 (2005) [arXiv:hep-ph/0312378].

[4] S. P. Martin and M. T. Vaughn, Phys. Rev. D **50**, 2282 (1994) [arXiv:hep-ph/9311340].

[5] A. Djouadi, J. L. Kneur and G. Moultaka, [arXiv:hep-ph/0211331].

[6] F. E. Paige, S. D. Protopescu, H. Baer and X. Tata, [arXiv:hep-ph/0312045].

[7] B. C. Allanach, Comput. Phys. Commun. **143**, 305 (2002) [arXiv:hep-ph/0104145].

[8] W. Porod, Comput. Phys. Commun. **153**, 275 (2003) [arXiv:hep-ph/0301101].

[9] G. Belanger, S. Kraml and A. Pukhov, [arXiv:hep-ph/0502079]. B. Allanach, S. Kraml and W. Porod, [arXiv:hep-ph/0207314].

[10] P. Skands *et al.*, JHEP **0407**, 036 (2004) [arXiv:hep-ph/0311123].

[11] B. C. Allanach *et al.*, in *Proc. of the APS/DPF/DPB Summer Study on the Future of Particle Physics (Snowmass 2001)* ed. N. Graf, Eur. Phys. J. C **25**, 113 (2002) [eConf **C010630**, P125 (2001)] [arXiv:hep-ph/0202233].

[12] C. L. Bennett *et al.*, Astrophys. J. Suppl. **148**, 1 (2003) [arXiv:astro-ph/0302207]; D. N. Spergel *et al.* [WMAP Collaboration], Astrophys. J. Suppl. **148**, 175 (2003) [arXiv:astro-ph/0302209].

[13] K. Abe *et al.* [Belle Collaboration], Phys. Lett. B **511**, 151 (2001) [arXiv:hep-ex/0103042]; D. Cronin-Hennessy *et al.* [CLEO Collaboration], Phys. Rev. Lett. **87**, 251808 (2001) [arXiv:hep-ex/0108033]; R. Barate *et al.* [ALEPH Collaboration], Phys. Lett. B **429**, 169 (1998).

[14] G. W. Bennett *et al.* [Muon g-2 Collaboration], Phys. Rev. Lett. **89**, 101804 (2002) [Erratum-ibid. **89**, 129903 (2002)] [arXiv:hep-ex/0208001]; G. W. Bennett *et al.* [Muon g-2 Collaboration], Phys. Rev. Lett. **92**, 161802 (2004) [arXiv:hep-ex/0401008].

[15] S. Eidelman *et al.* [Particle Data Group], Phys. Lett. B **592**, 1 (2004).

[16] H. E. Haber and G. L. Kane, Phys. Rept. **117**, 75 (1985).

[17] M. Drees, R. Godbole, and P. Roy, “Theory and Phenomenology of Sparticles,” World Scientific, 2005.

[18] P. Binetruy, “Supersymmetry: Theory, Experiment, and Cosmology,” Oxford University Press, 2005.

[19] H. Baer and X. Tata, “Weak Scale Supersymmetry” Cambridge University Press, 2005.

[20] E. Boos *et al.* [CompHEP Collaboration], Nucl. Instrum. Meth. A **534**, 250 (2004) [arXiv:hep-ph/0403113].

[21] T. Sjostrand, L. Lonnblad, S. Mrenna and P. Skands, [arXiv:hep-ph/0308153].

[22] See, for example, O. J. P. Eboli, R. Zukanovich Funchal and T. L. Lungov, Phys. Rev. D **59**, 035002 (1999) [arXiv:hep-ph/9808288].

[23] J. L. Bourjaily, G. L. Kane, P. Kumar and T. T. Wang, [arXiv:hep-ph/0504170].

[24] G. Belanger, F. Boudjema, A. Pukhov and A. Semenov, [arXiv:hep-ph/0405253].

[25] P. Gondolo, J. Edsjo, P. Ullio, L. Bergstrom, M. Schelke and E. A. Baltz, JCAP **0407**, 008 (2004) [arXiv:astro-ph/0406204].

APPENDIX A: A PYTHIA EXAMPLE

Here we provide a sample 'main program' which calls `PYTHIA` in order to simulate gluino pair production at the LHC. In the MSSM, the gluino pair production are mainly from the following two processes:

$$gg \to \widetilde{g}\widetilde{g} \quad \text{and} \quad f\bar{f} \to \widetilde{g}\widetilde{g} \tag{A1}$$

We turn on both of them in this example.

To run this simulation, the following code can be used. We have included many comments in the program it make it more readable, but we apologize if it appears slightly esoteric. It may be helpful to compare this code with that outlined for a general `PYTHIA` program in section III D. As one can see, we first set some parameters: parameters that tell `PYTHIA` to turn off initial and final state radiations; the parameters of the MSSM at the weak scale; and the parameters specifying which processes to turn on. We then specify the properties of the beam. After this 'initialization,' we begin the simulation, and finally print out cross sections and a list of particle masses and decay modes.

```
C...Supersymmetry at a hadron collider.

C-----------------------------------------------------------------

C...Preamble: declarations.

C...All real arithmetic in double precision.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)
C...Three Pythia functions return integers, so need declaring.
      INTEGER PYK,PYCHGE,PYCOMP
C...EXTERNAL statement links PYDATA on most machines.
      EXTERNAL PYDATA

C..........................................
C...Commonblocks.

C...Here is where the code starts in Appendix A.....
C...The event record.
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
      COMMON/PYDATR/MRPY(6),RRPY(100)
C...Parameters.
      COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
C...Particle properties + some flavour parameters.
      COMMON/PYDAT2/KCHG(500,4),PMAS(500,4),PARF(2000),VCKM(4,4)
C...Decay information.
```

```
      COMMON/PYDAT3/MDCY(500,3),MDME(8000,2),BRAT(8000),KFDP(8000,5)
C...Selection of hard scattering subprocesses.
      COMMON/PYSUBS/MSEL,MSELPD,MSUB(500),KFIN(2,-40:40),CKIN(200)
C...Parameters.
      COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
C...Supersymmetry parameters.
      COMMON/PYMSSM/IMSS(0:99),RMSS(0:99)
C...Cross section
      COMMON/PYINT5/NGENPD,NGEN(0:500,3),XSEC(0:500,3)
C...Histograms
      COMMON/PYBINS/IHIST(4),INDX(1000),BIN(20000)
C-----------------------------------------------------------------
C...Set the seed for the random number generator
      mrpy(1)=711753
C...User control mode
      MSEL=0
C...Master switch for initial-state QCD and QED radiation(D=1)
      MSTP(61)=0
c...Master switch for final-state QCD and QED radiation
      MSTP(71)=0
C.....Master switch for multiple interactions
      MSTP(81)=0
C...Switch for particle decay , fragmentation.
      MSTP(111)=0
C....2->2 process Particle put on-shell(0: on-shell)
      MSTP(42)=0


C-----------------------------------------------------------------
C...Start setting SUSY parameters
      IMSS(3)=1                                !gluino parameter is pole mass
c...General MSSM
      IMSS(1)=1
      xmsq=1000.
      xma=1000.
      atri=1000.
C...Gaugino mass
      RMSS(1)=100.                             !M_1
      RMSS(2)=300.                             !M_2
      RMSS(3)=600.                             !gluino pole mass
C...mu parameter
      RMSS(4)=500.
C...Tan beta
```

```
      RMSS(5)=10.
C...Slepton L and R mass, physical mass.
      RMSS(6)=400.                              !slepton_L mass
      RMSS(7)=400.                              !slepton_R mass
C...Squark L and R mass
      RMSS(8)=xmsq                              !squark_L mass
      RMSS(9)=xmsq                              !squark_down_R mass
C...third family
      IMSS(8)=0
      IMSS(5)=0
      RMSS(10)=xmsq                             !Q_L
      RMSS(11)=xmsq                             !b_R
      RMSS(12)=xmsq                             !t_R
      RMSS(13)=xmsq                             !stau_L
      RMSS(14)=xmsq                             !stau_R
      RMSS(15)=atri                             !A_b
      RMSS(16)=atri                             !A_t
      RMSS(17)=atri                             !A_tau
      IMSS(9)=1
      RMSS(22)=xmsq                             !~u_R mass
      RMSS(19)=xma                              !Pseudoscalar higgs mass
      RMSS(32)=0.
c...Turn on the process that we want to simulate
      MSUB(243)=1                               !
      MSUB(244)=1                               !
C-----------------------------------------------------------------
c...Specify the beam. ECM is the center of mass energy
      ECM=14000.d0
      CALL PYINIT(’CMS’,’p+’,’p+’,ECM)
C-----------------------------------------------------------------
c...Start simulation.  NEV: how many events to generate
      NEV=1000
      DO 200 IEV=1, NEV
c...Generate one evnet
       CALL PYEVNT
c...Print out the first ten events
       IF (IEV.LE.10) THEN
        CALL PYLIST(1)
       ENDIF
 200  CONTINUE
C-----------------------------------------------------------------
c...Print out the cross sections
      CALL PYSTAT(1)
```

```
c...Print out the masses and decay modes for all particles
c      CALL PYLIST(12)
      END
```

Suppose the name of the above code is saved in the file `lhc.f`. To generate the executable file, one needs the following two commands:

```
g77 -c pythia6321.f
g77 -o lhc lhc.f pythia6321.o
```