Emulation
To do or Not to do

Gaurab Sarangi

University of Florida

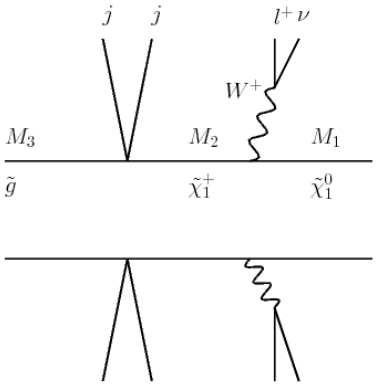
with

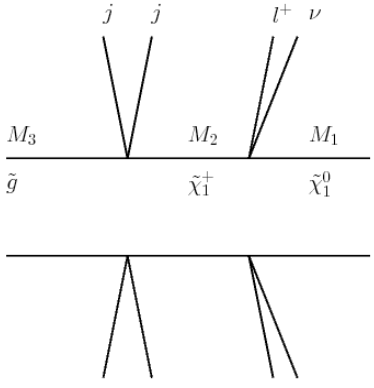
Konstantin Matchev and Myeonghun Park

Detector Simulation

- Full Detector Simulation (e.g. Geant, CMS-SW)
- Fast Sim (e.g. PGS, Delphes)
- Emulation

Simplified Model





Simulator or Emulator? What is the difference?

31

Emulation is the process of mimicking the outwardly observable behavior to match an existing target. The internal state of the emulation mechanism does not have to accurately reflect the internal state of the target which it is emulating.

Simulation, on the other hand, involves modeling the underlying state of the target. The end result of a good simulation is that the simulation model will emulate the target which it is simulating.

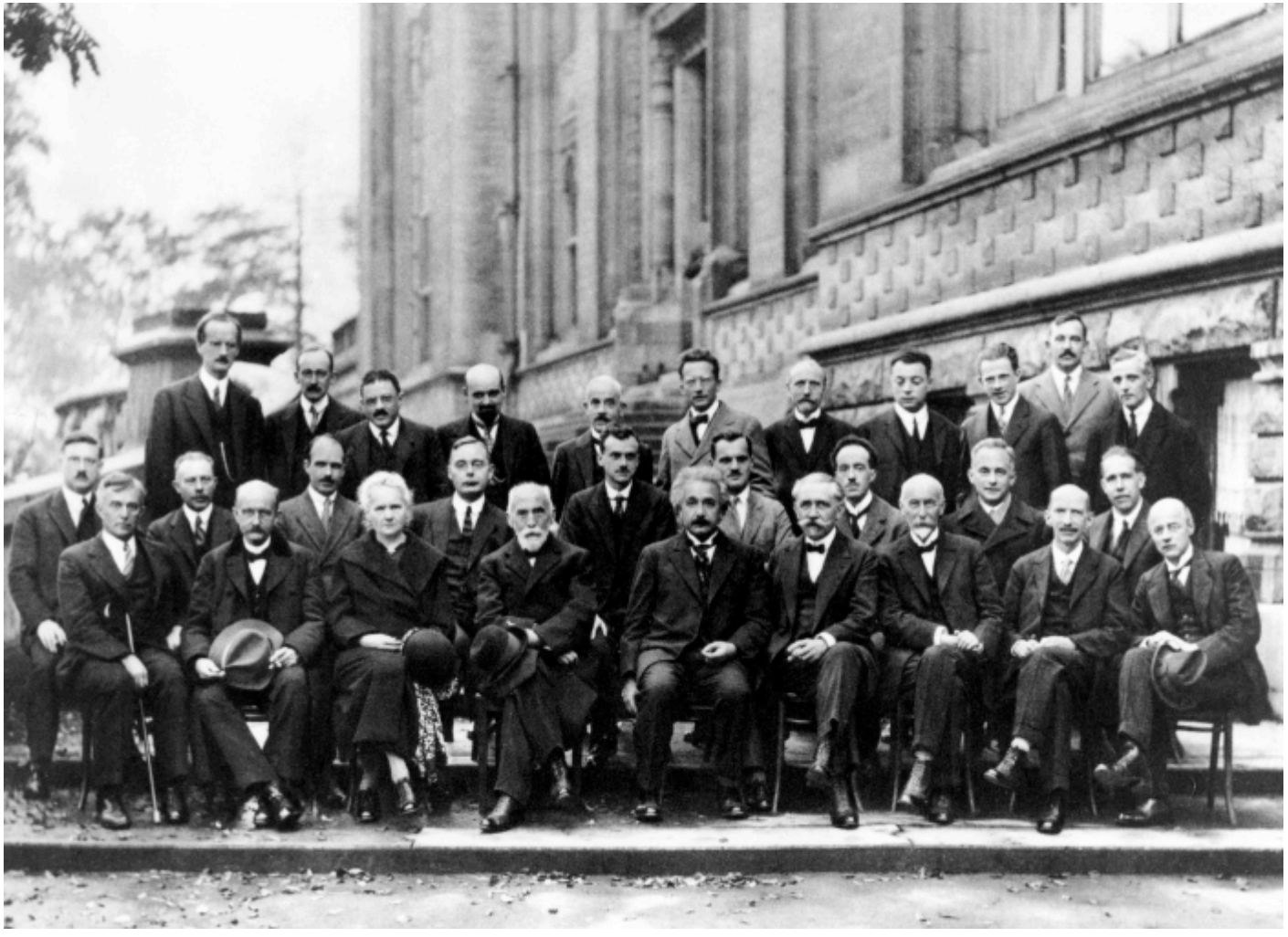
Ideally, you should be able to look into the simulation and observe properties that you would also see if you looked into the original target. In practice, there may some shortcuts to the simulation for performance reasons -- that is, some internal aspects of the simulation may actually be an emulation.

MAME is an arcade game emulator; Hyperterm is a (not very good) terminal emulator. There's no need to model the arcade machine or a terminal in detail to get the desired emulated behavior.

Flight Simulator is a simulator; SPICE is an electronics simulator. They model as much as possible every detail of the target to represent what the target does in reality.

EDIT: Other responses have pointed out that the goal of an emulation is to able to substitute for the object it is emulating. That's an important point. A simulation's focus is more on the modelling of the internal state of the target -- and the simulation does not necessarily lead to emulation. In particular, a simulation may run far slower than real time. SPICE, for example, cannot substitue for an actual electronics circuit (even if assuming there was some kind of magical device that perfectly interfaces electrical circuits to a SPICE simulation.) A simulation Simulation does not always lead to emulation --

Full Detector Simulation (e.g. Geant)



Emulation (e.g. CMS prescription)

Fast Sim (e.g. PGS)

Benefits of Emulation

Problems with Emulation

Interpreting Experimental Result

Experiment gives us N_{sig}

$N_{sig} = L \times \sigma \times BR \times \varepsilon$

L → from the experiment

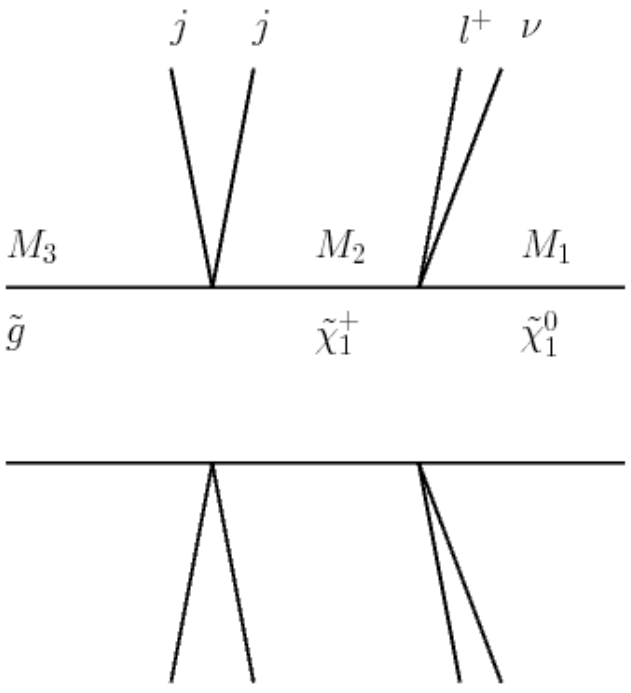
'σ' → easily with Monte-Carlo or Analytically

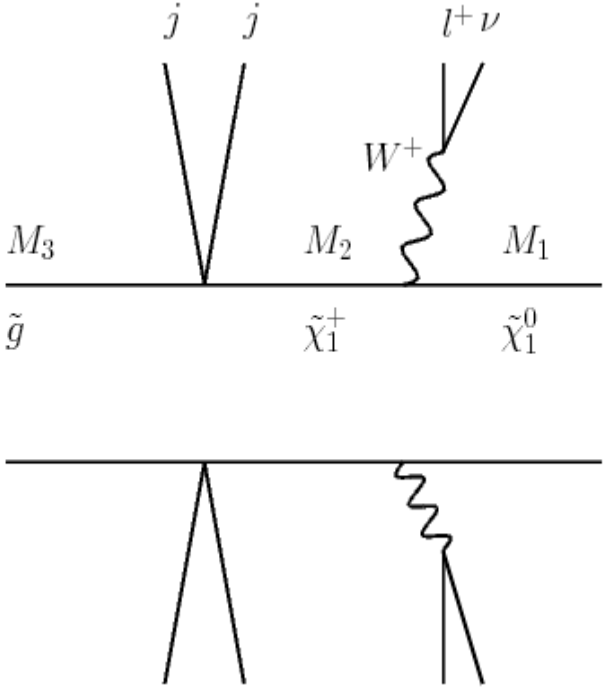
'BR' → easily with a formula from theory

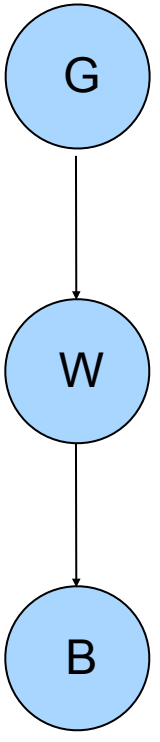
'ε' → hard → convoluted integral

Simplified Model

$$pp \to \tilde{g}\tilde{g}$$
$$\tilde{g} \to \tilde{\chi}_1^+ 2j$$
$$\tilde{\chi}_1^+ \to \tilde{\chi}_1^0 l\nu$$







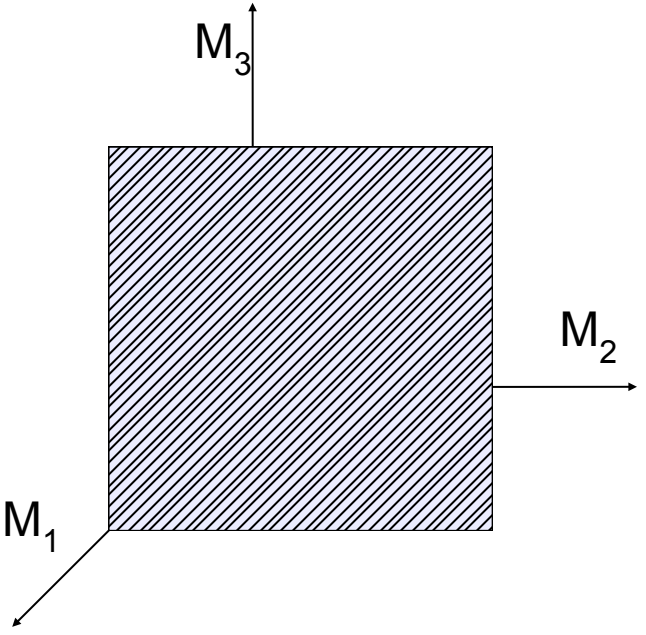
Parameter Space

- Run detector simulation in the chosen parameter space.

In our case:

we fix M_1 = 10 GeV

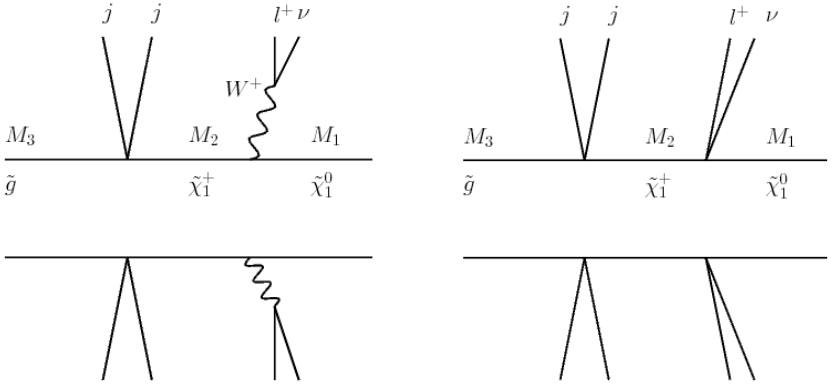
and vary M_2 and M_3

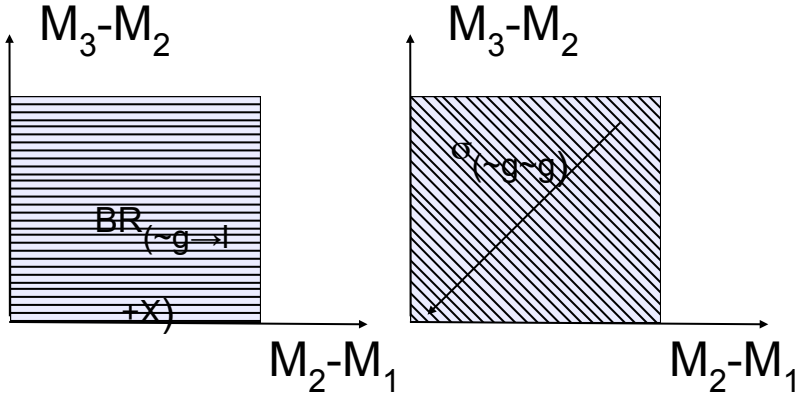


- Calculate efficiency (ε)=N_{cut} /N_{tot}
- Plot the limit on the cross-section using the formula:

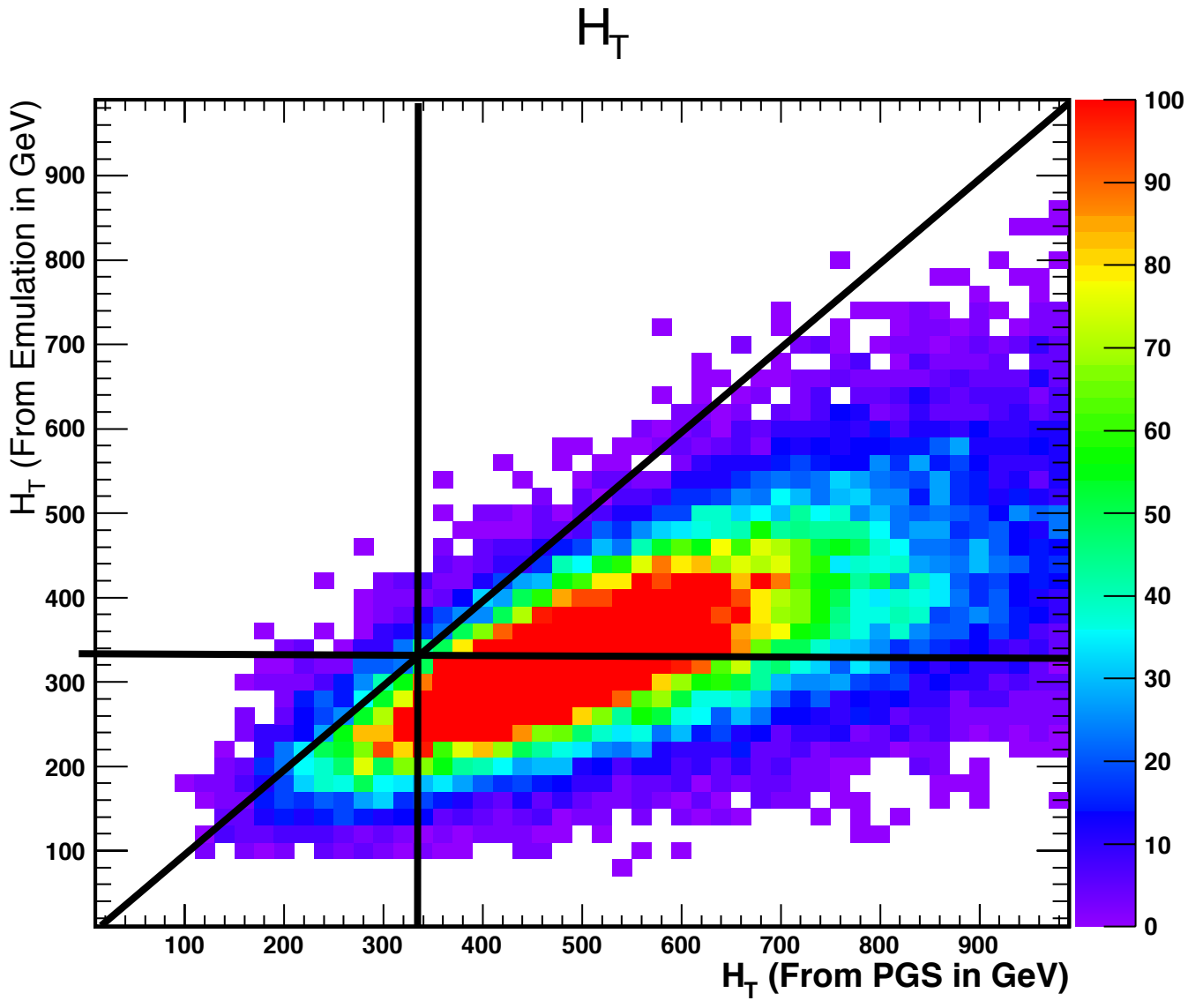
$$(\sigma.BR)_{max}(M_i) = \frac{S\sqrt{N_{bkg}}}{L.\epsilon(M_i)}$$

Where, 'S' is the significance and L is the luminosity

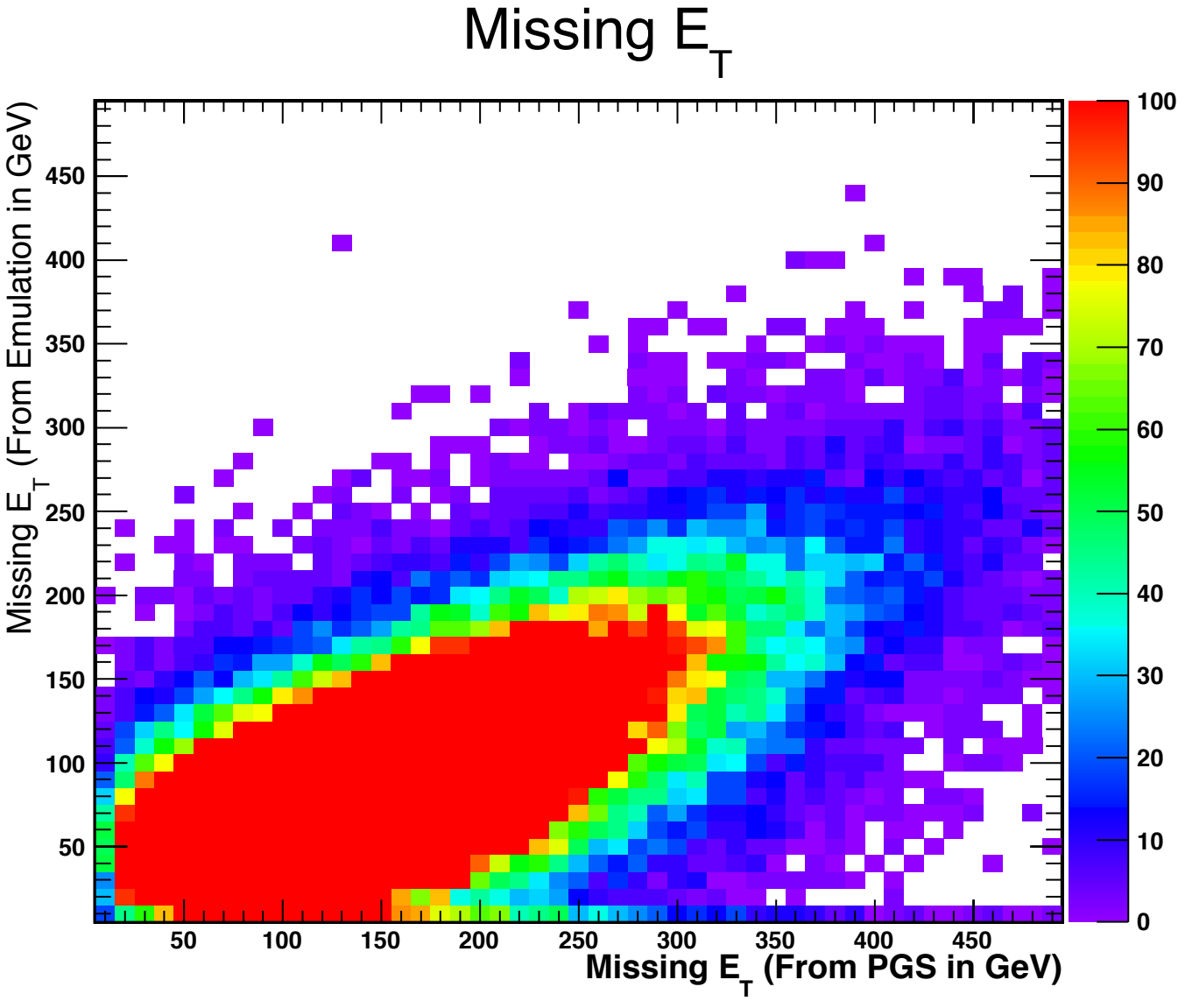




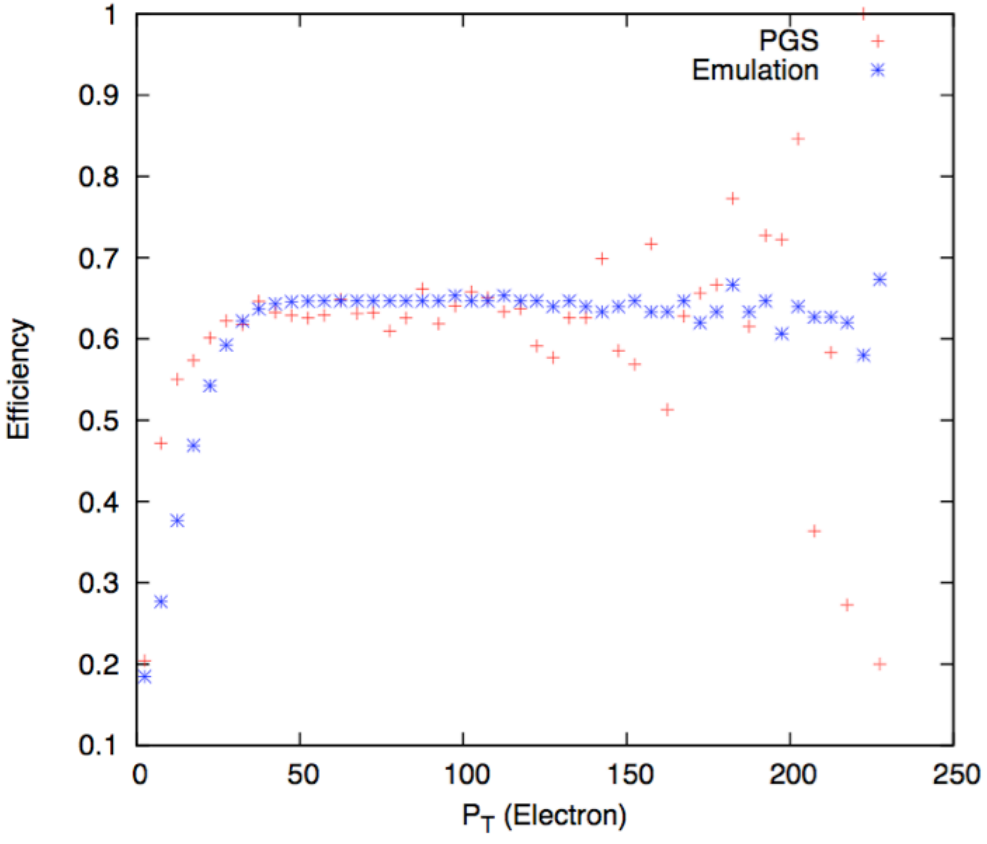
Montecarlo (Parton level emulation) vs Detector (PGS)

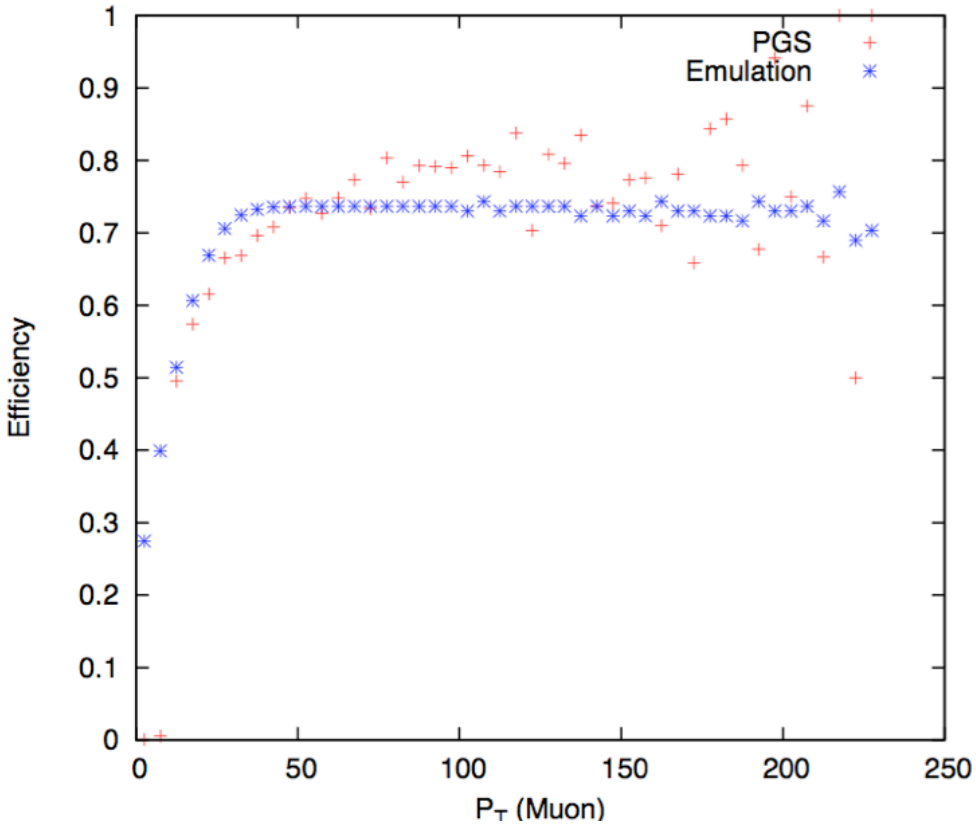


PGS vs Emulation (MET)



Comparision PGS vs Emulation

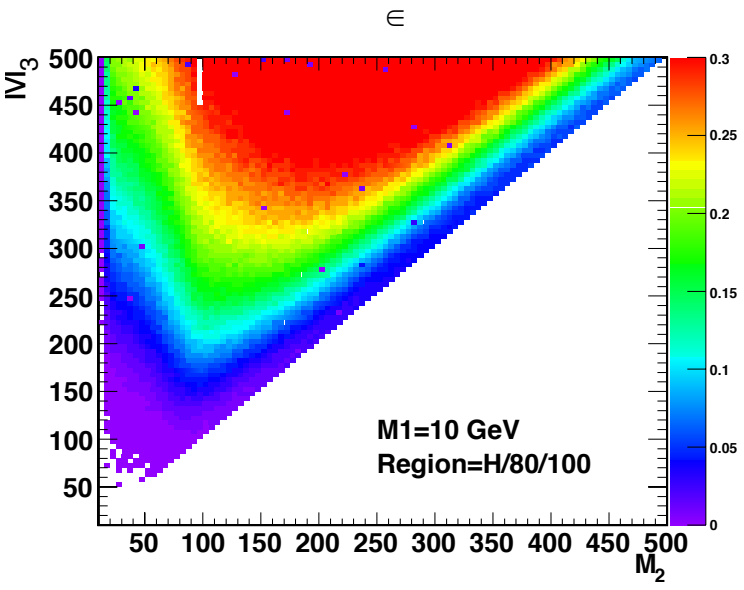


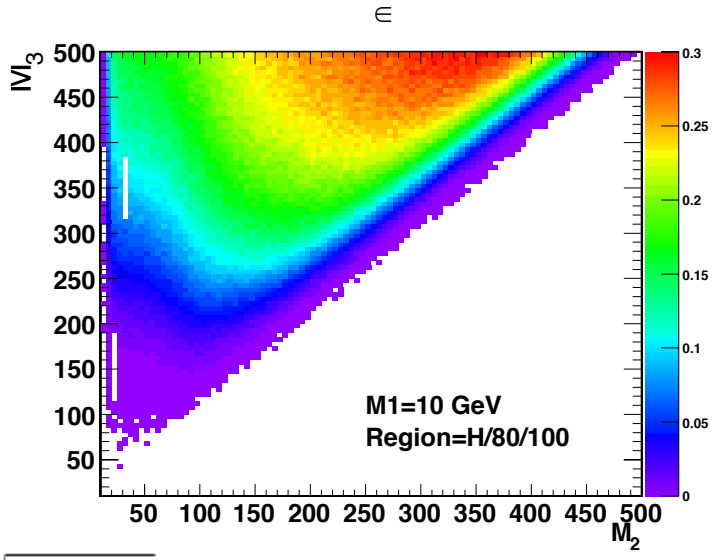


$$\epsilon_{lepton}(P_T) = \epsilon_{max,lepton} + A_{lepton}\, erf\left(\frac{P_T - P_{Tcut}}{B_{lepton}} - 1\right) + 0.1\frac{25 - N_{track}}{15}$$

Model independent limit

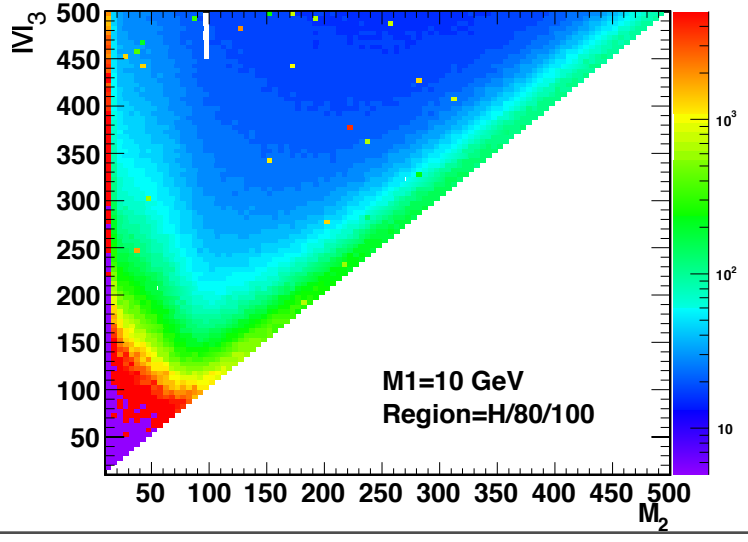
PGS vs Emulation



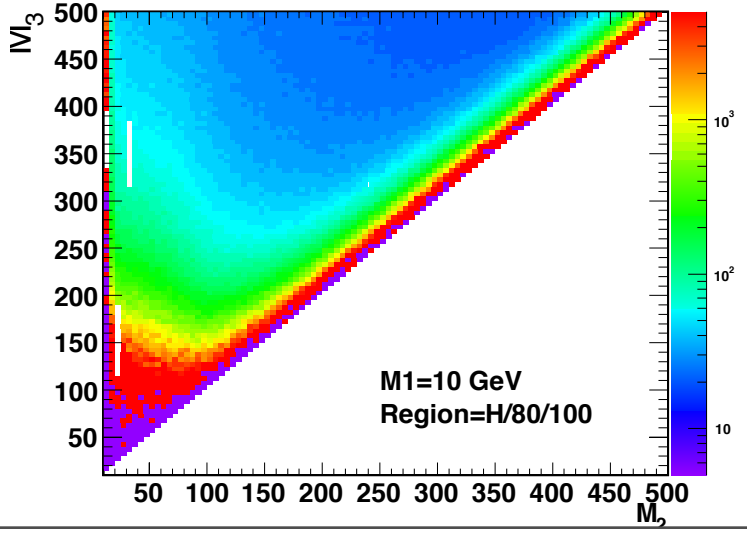


$$(\sigma.BR)_{max}(M_i) = \frac{S\sqrt{N_{bkg}}}{L.\epsilon(M_i)}$$

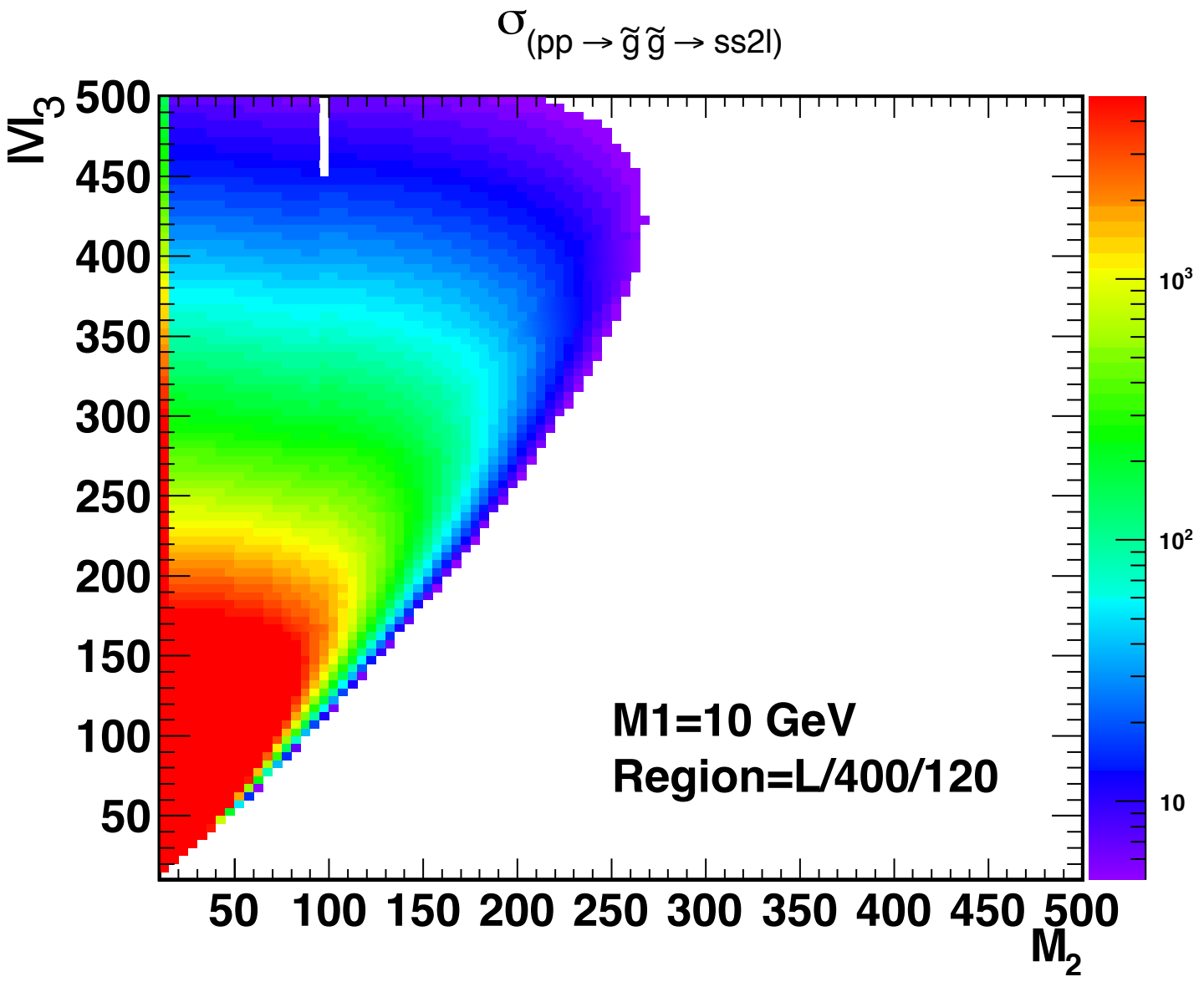
3σ model independent limit on cross-section



3σ model independent limit on cross-section

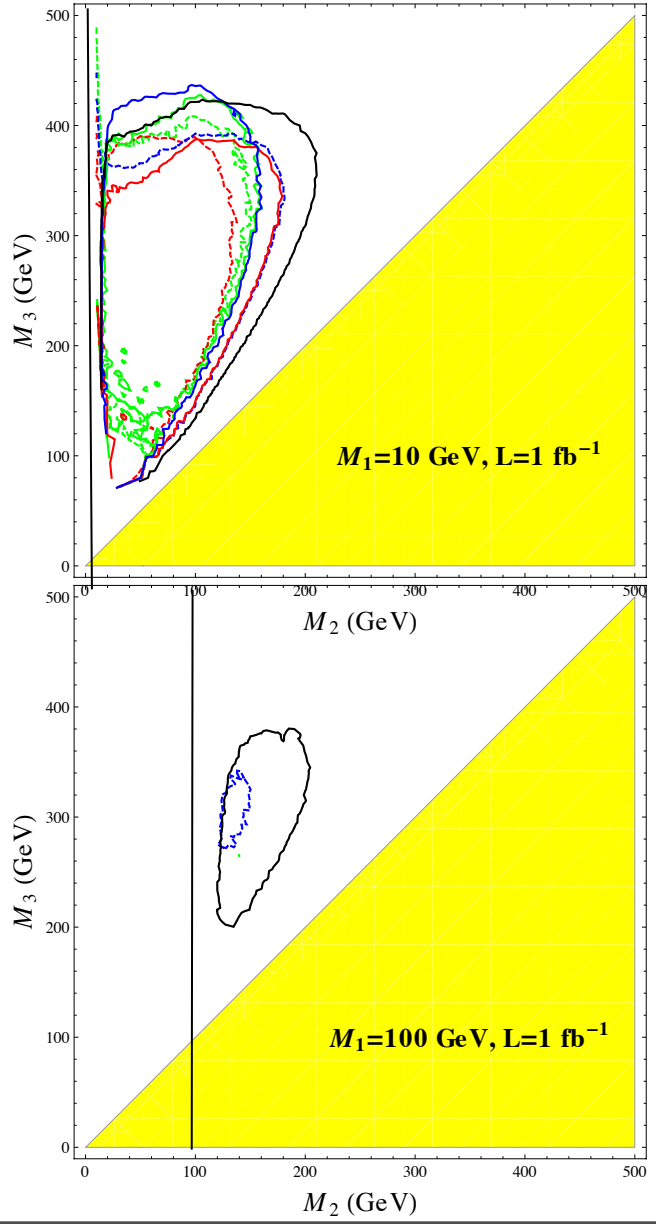


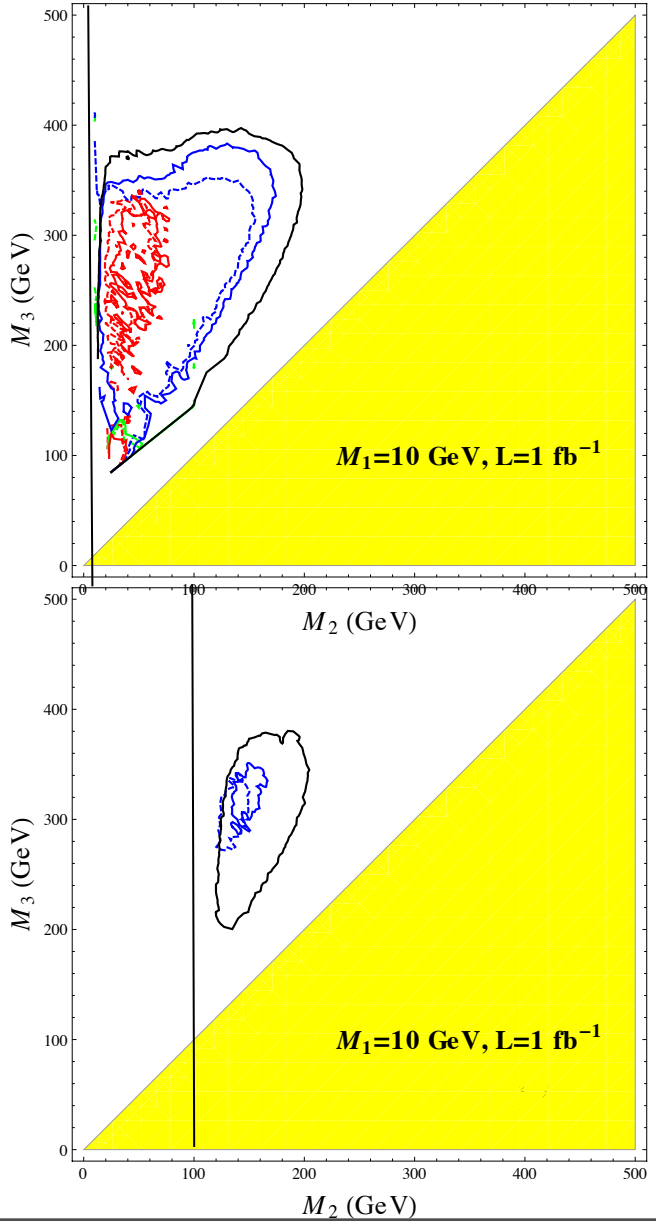
Model dependent production cross-section



Reach

PGS vs Emulation





Conclusion