MadMiner vs MoMEMta

Comparing methods to determine the top mass in a fully leptonic decay

g g > t t~, t > b e+ ve, t~ > b~ mu- vm~

Context for the Comparison

MadMiner

- Train neural network on matrix element data to determine the joint likelihood ratio, no integration required
- Computation cost paid once during training, likelihood evaluations are cheap
- Like most machine learning problems, needs large amounts of data

MoMEMta

- Approximate showers and detectors with transfer functions and calculating likelihood ratios directly by integration
- Computations cost the same for each event and are expensive
- Has trouble approximating certain phenomena with transfer functions

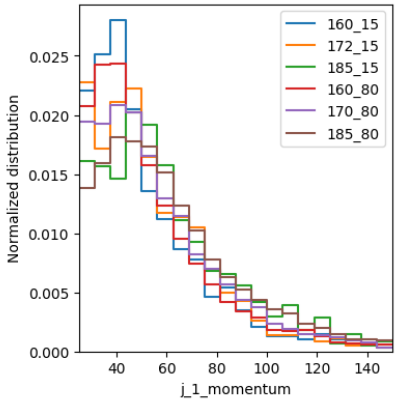
$$r(x,z|\theta) = \frac{p(x,z|\theta)}{p(x,z|\theta_{\text{ref}})} = \frac{p(z_p|\theta)}{p(z_p|\theta_{\text{ref}})} \sim \frac{|\mathcal{M}|^2(z_p|\theta)}{|\mathcal{M}|^2(z_p|\theta_{\text{ref}})} \, \frac{\sigma(\theta_{\text{ref}})}{\sigma(\theta)}$$

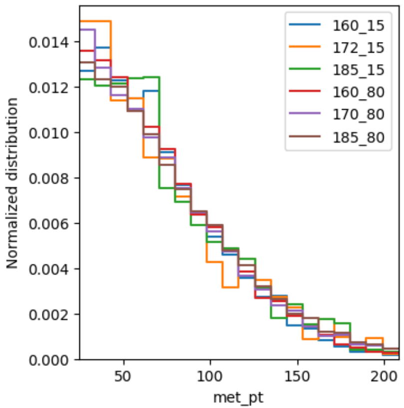
$$\text{NN} \rightarrow \arg\min_g L_r[g(x,\theta)] = r(x|\theta)$$

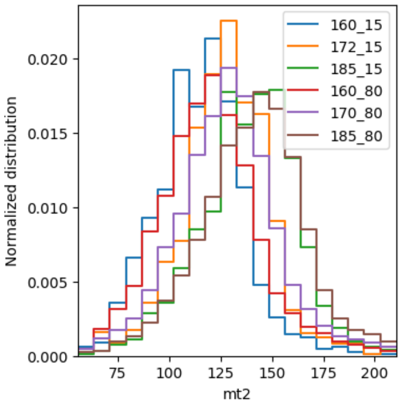
$$\hat{p}(x|\theta) = \int \mathrm{d}z_p \; \hat{p}(x|z_p) \, p(z_p|\theta) \sim \frac{1}{\sigma(\theta)} \int \mathrm{d}z_p \; \hat{p}(x|z_p) \, |\mathcal{M}(z_p|\theta)|^2$$

MadMiner Approach

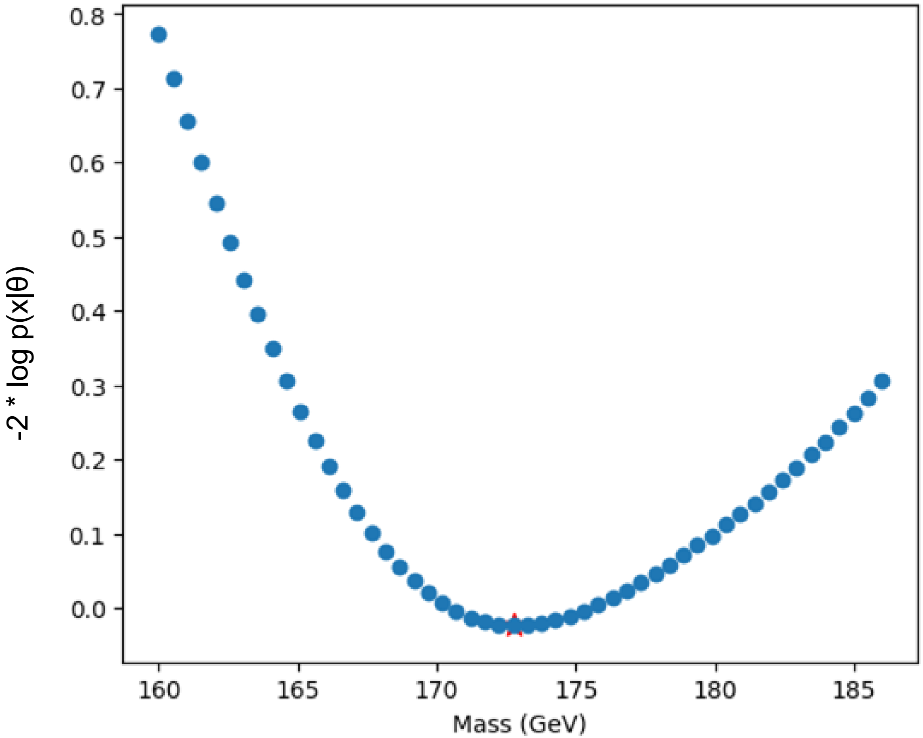
1. Sample events from MadGraph over different top mass hypotheses
2. Extract low-level observables and define high-level ones like mT2. Set a reference value.
3. Train neural network by inputting top mass hypotheses and the observables from the events. Output will be estimate of likelihood ratio between top mass and a reference value







MadMiner Results



Next steps

- Setup MoMEMta and see how it compares to MadMiner
- Why are low-level features not enough to train our network?
- Would trying different types of neural networks help?
- Could we avoid feature engineering if we get more data and better hardware?
- Currently running on a laptop with ~100,000 events taking ~8 hours