

MoMEMta and Deep Neural Networks, a method to produce Matrix Element weights for LHC analyses

Florian Bury

IRN Terascale 2019 - Bruxelles



October 17, 2019

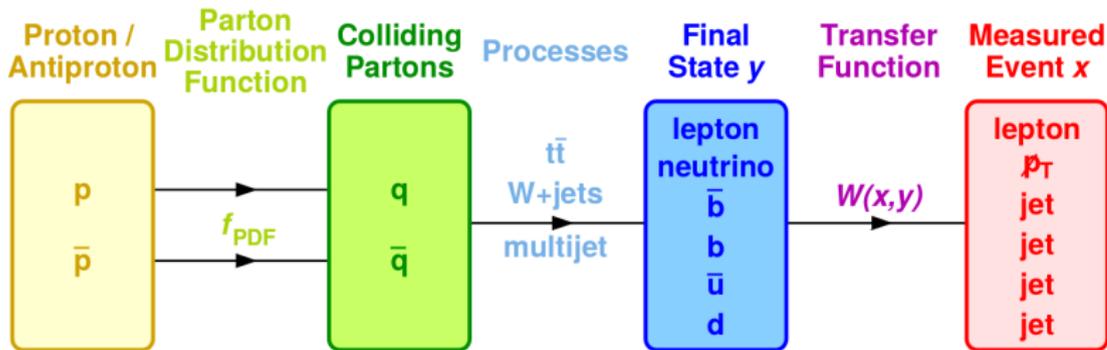
Matrix Element Method (MEM)

Matrix Element Method integral

$$P(x|\alpha) = \frac{1}{\sigma_{\alpha}^{\text{vis}}} \int_y d\phi(y) \int_{q_1, q_2} dq_1 dq_2 \sum_{a_1, a_2} f_{a_1}(q_1) f_{a_2}(q_2) |M_{\alpha}(q_1, q_2, y)|^2 W(x|y)$$

Phase space parameterization

$$d\phi(y) = \left(\prod_{i=3}^N \frac{d^3 P_i}{2E_i (2\pi)^3} \right) (2\pi)^4 \delta^4(P_1 + P_2 - \sum_{j=3}^N P_j)$$



▶ Link

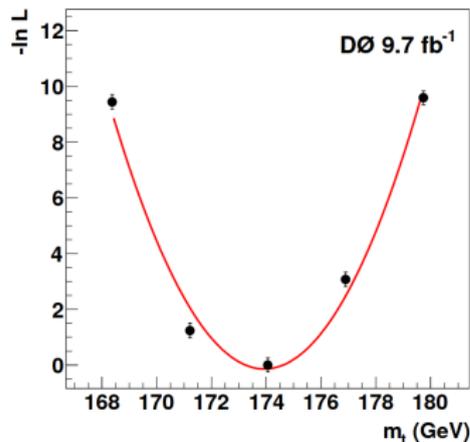
Matrix Element Method (MEM)

Matrix Element Method integral

$$P(x|\alpha) = \frac{1}{\sigma_{\alpha}^{\text{vis}}} \int_y d\phi(y) \int_{q_1, q_2} d^3q_1 d^3q_2 \sum_{a_1, a_2} f_{a_1}(q_1) f_{a_2}(q_2) |M_{\alpha}(q_1, q_2, y)|^2 W(x|y)$$

Phase space parameterization

$$d\phi(y) = \left(\prod_{i=3}^N \frac{d^3P_i}{2E_i(2\pi)^3} \right) (2\pi)^4 \delta^4(P_1 + P_2 - \sum_{j=3}^N P_j)$$



[▶ Link](#)

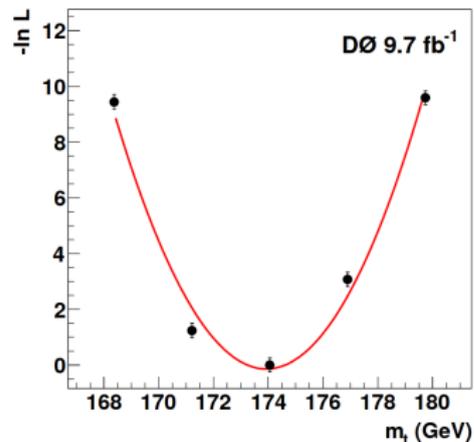
Matrix Element Method (MEM)

Matrix Element Method integral

$$P(x|\alpha) = \frac{1}{\sigma_\alpha^{\text{vis}}} \int_y d\phi(y) \int_{q_1, q_2} dq_1 dq_2 \sum_{a_1, a_2} f_{a_1}(q_1) f_{a_2}(q_2) |M_\alpha(q_1, q_2, y)|^2 W(x|y)$$

Phase space parameterization

$$d\phi(y) = \left(\prod_{i=3}^N \frac{d^3 P_i}{2E_i (2\pi)^3} \right) (2\pi)^4 \delta^4(P_1 + P_2 - \sum_{j=3}^N P_j)$$



Advantages

- Exploits directly our knowledge of the SM
- Includes all detector effects (parametric way)
- No need for training ($><$ multivariate methods)

Drawbacks

- Complex integration \rightarrow MoMEMta
- Computation time \rightarrow DNN

[▶ Link](#)

Numerical integration

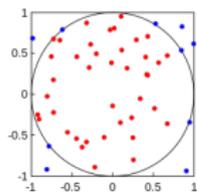
A dummy example

The game : Integrate $f(x, y) = \begin{cases} 1 & x^2 + y^2 < 1 \\ 0 & x^2 + y^2 \geq 1 \end{cases}$

Numerical integration

A dummy example

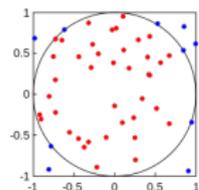
The game : Integrate $f(x,y) = \begin{cases} 1 & x^2 + y^2 < 1 \\ 0 & x^2 + y^2 \geq 1 \end{cases}$



$$\begin{aligned} I \simeq S &= \frac{V}{N} \sum_i^N f(\vec{x}_i) \\ &= A_{square} \frac{N_{in}}{N_{total}} \end{aligned}$$

Numerical integration

A dummy example



The game : Integrate $f(x, y) = \begin{cases} 1 & x^2 + y^2 < 1 \\ 0 & x^2 + y^2 \geq 1 \end{cases}$

$$\begin{aligned} I \simeq S &= \frac{V}{N} \sum_i^N f(\vec{x}_i) \\ &= A_{square} \frac{N_{in}}{N_{total}} \end{aligned}$$

Empty space phenomenon :

$$\Phi(N) \propto N^d$$

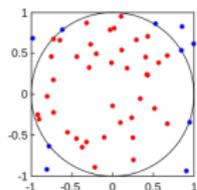
In high dimension : Impossible !

Solution : $\Phi(N) \propto \text{Var}(f)$

Numerical integration

A dummy example

The game : Integrate $f(x, y) = \begin{cases} 1 & x^2 + y^2 < 1 \\ 0 & x^2 + y^2 \geq 1 \end{cases}$



Stratified sampling

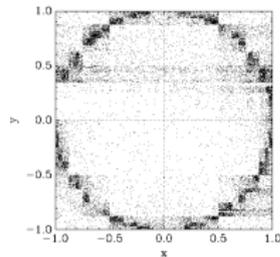
$$I \simeq S = \frac{V}{N} \sum_i^N f(\vec{x}_i) \\ = A_{square} \frac{N_{in}}{N_{total}}$$

Empty space phenomenon :

$$\Phi(N) \propto N^d$$

In high dimension : Impossible !

Solution : $\Phi(N) \propto \text{Var}(f)$



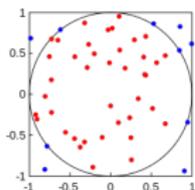
$$I \simeq S = \sum_{box_i} \frac{V(box_i)}{N_i} \sum_{n=1}^{N_i} f(\vec{x}_{in})$$

Iteratively divide boxes in region where f fluctuates the most (high variance) and generate the same number of points in each box
 \Rightarrow Becomes slow in very high dimension space

Numerical integration

A dummy example

The game : Integrate $f(x, y) = \begin{cases} 1 & x^2 + y^2 < 1 \\ 0 & x^2 + y^2 \geq 1 \end{cases}$



$$I \simeq S = \frac{V}{N} \sum_i^N f(\vec{x}_i) \\ = A_{square} \frac{N_{in}}{N_{total}}$$

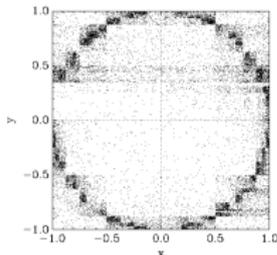
Empty space phenomenon :

$$\Phi(N) \propto N^d$$

In high dimension : Impossible !

Solution : $\Phi(N) \propto \text{Var}(f)$

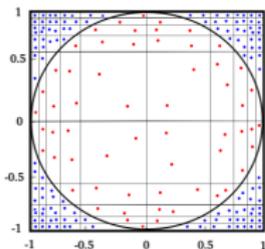
Stratified sampling



$$I \simeq S = \sum_{box_i} \frac{V(box_i)}{N_i} \sum_{n=1}^{N_i} f(\vec{x}_{in})$$

Iteratively divide boxes in region where f fluctuates the most (high variance) and generate the same number of points in each box
 \Rightarrow Becomes slow in very high dimension space

Importance sampling



$$I \simeq S = \frac{1}{N} \sum_i^N \frac{f(\vec{x})}{p(\vec{x})}$$

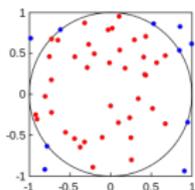
Separable sampling function

$$p(\vec{x}) = p_1(x_1)p_2(x_2)\dots p_N(x_N)$$

Numerical integration

A dummy example

The game : Integrate $f(x, y) = \begin{cases} 1 & x^2 + y^2 < 1 \\ 0 & x^2 + y^2 \geq 1 \end{cases}$



$$I \simeq S = \frac{V}{N} \sum_i^N f(\vec{x}_i) \\ = A_{square} \frac{N_{in}}{N_{total}}$$

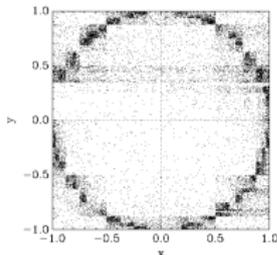
Empty space phenomenon :

$$\Phi(N) \propto N^d$$

In high dimension : Impossible !

Solution : $\Phi(N) \propto \text{Var}(f)$

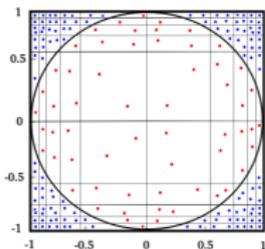
Stratified sampling



$$I \simeq S = \sum_{\text{box}_i} \frac{V(\text{box}_i)}{N_i} \sum_{n=1}^{N_i} f(\vec{x}_{in})$$

Iteratively divide boxes in region where f fluctuates the most (high variance) and generate the same number of points in each box
 \Rightarrow Becomes slow in very high dimension space

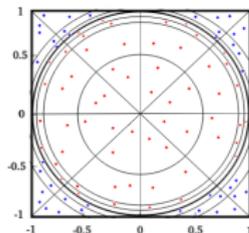
Importance sampling



$$I \simeq S = \frac{1}{N} \sum_i^N \frac{f(\vec{x})}{p(\vec{x})}$$

Separable sampling function
 $p(\vec{x}) = p_1(x_1)p_2(x_2)\dots p_N(x_N)$

Peak remapping



Numerical integration

Back to the MEM

Matrix Element Method integral

$$P(x|\alpha) = \frac{1}{\sigma_\alpha^{\text{vis}}} \int_y d\phi(y) \int_{q_1, q_2} dq_1 dq_2 \sum_{a_1, a_2} f_{a_1}(q_1) f_{a_2}(q_2) |M_\alpha(q_1, q_2, y)|^2 W(x|y)$$

Phase space parameterization

$$d\phi(y) = \left(\prod_{i=3}^N \frac{d^3 P_i}{2E_i (2\pi)^3} \right) (2\pi)^4 \delta^4(P_1 + P_2 - \sum_{j=3}^N P_j)$$

Integration rule : Map every shark peak to one variable of integration

Where do they come from ?

- Transfer function resolution : ✓

$$W(x|y) = \prod_{i=1}^n W^E(x^i|y^i) W^\eta(x^i|y^i) W^\phi(x^i|y^i)$$

- Propagator enhancements $|M_\alpha(q_1, q_2, y)|^2$: ✗

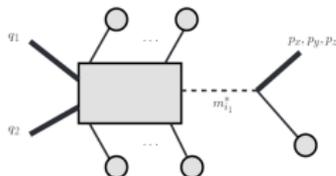
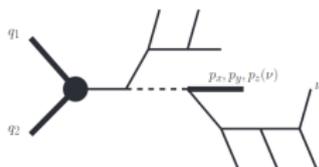
Example : Breit-Wigner resonances

In addition, need to integrate out the δ of the momentum conservation

MoMEMta can perform the MEM integration almost out of the box

- C++ classes for each step of the MEM
- Configured via a Lua script
- Matrix element provided by MadGraph (via an included exporter [▶ Link](#))
- PDF from LHAPDF [▶ Link](#)
- Integration with Cuba [▶ Link](#)

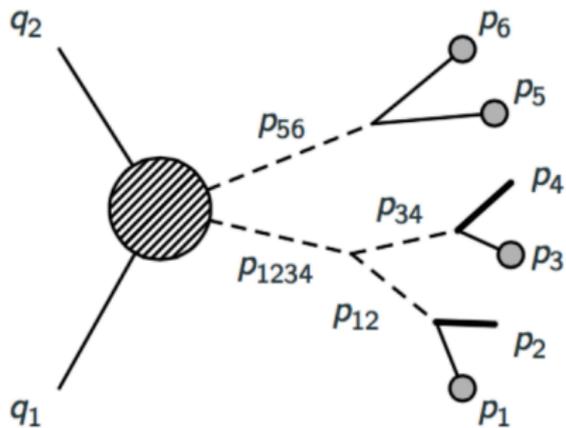
Keyword : **Modular** → uses a structure of blocks [▶ Link](#)



- Integrates out the delta
- Removes the proton momentum fractions
- Removes one particle momentum (invisible)
- Integrates over the resonance

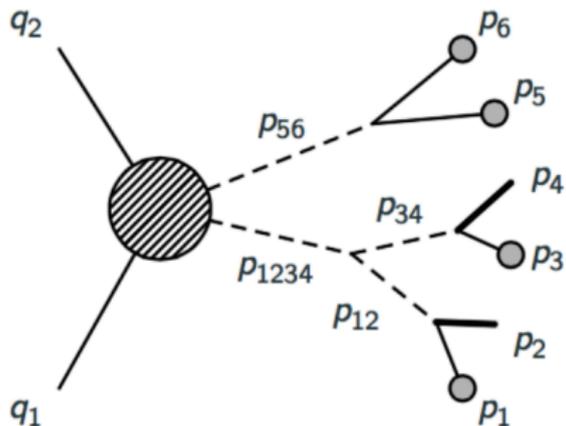
Block structure example [▶ Link](#)

Di-Higgs production: $H(b\bar{b})H(W(l\nu)W(l\nu))$



Block structure example [▶ Link](#)

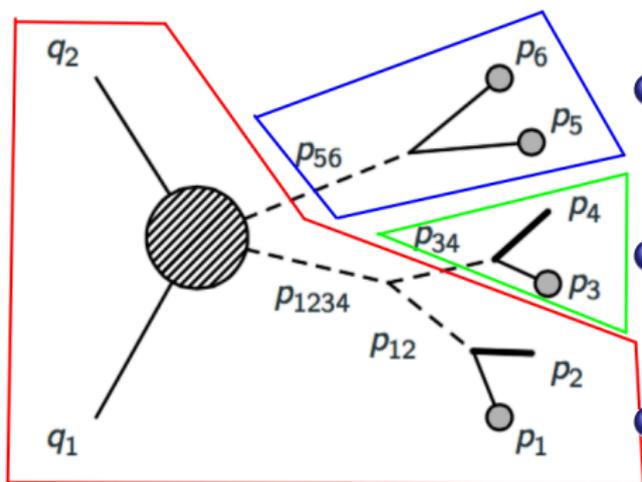
Di-Higgs production: $H(b\bar{b})H(W(l\nu)W(l\nu))$



- 1 Use the transfer function
 x (experimental) $\rightarrow y$ (parton level)

Block structure example [▶ Link](#)

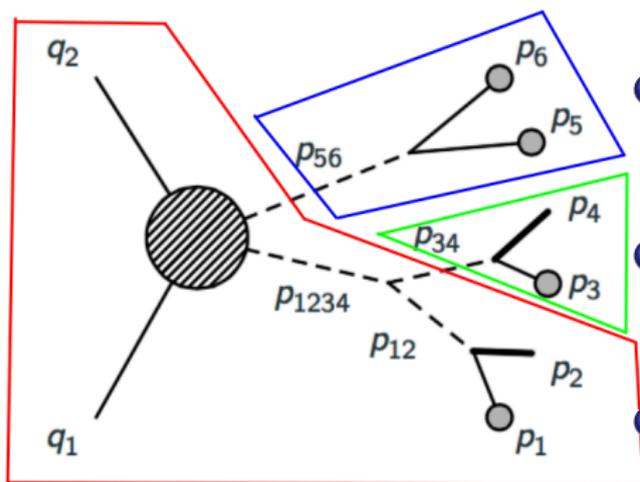
Di-Higgs production: $H(b\bar{b})H(W(l\nu)W(l\nu))$



- 1 Use the transfer function x (experimental) \rightarrow y (parton level)
- 2 Apply **Secondary Block D**
Removes $|p_5|$ or $|p_6|$
Integrates over p_{56} (BW or NWA)
- 3 Apply **Secondary Block C**
Removes $|p_4|$ (because neutrino)
Integrates over p_{34} (BW at m_W)
- 4 Apply **Main Block B**
Removes q_1, q_2, \vec{P}_2
Integrates over p_{1234} (BW or NWA)
 δ^4 has been integrated out

Block structure example [▶ Link](#)

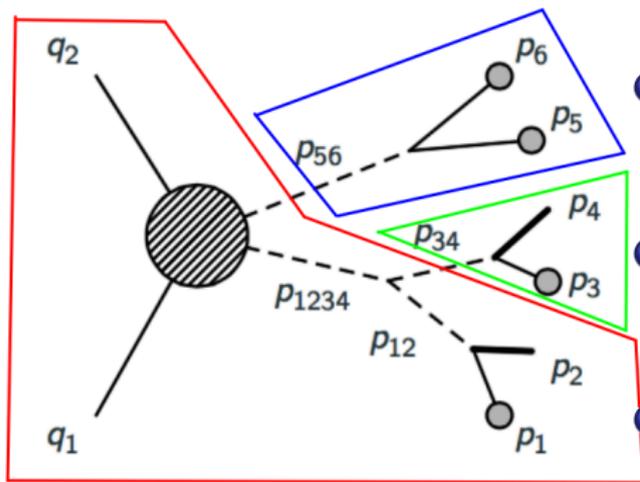
Di-Higgs production: $H(b\bar{b})H(W(l\nu)W(l\nu))$



- 1 Use the transfer function x (experimental) \rightarrow y (parton level)
- 2 Apply **Secondary Block D**
Removes $|p_5|$ or $|p_6|$
Integrates over p_{56} (BW or NWA)
- 3 Apply **Secondary Block C**
Removes $|p_4|$ (because neutrino)
Integrates over p_{34} (BW at m_W)
- 4 Apply **Main Block B**
Removes q_1, q_2, \vec{P}_2
Integrates over p_{1234} (BW or NWA)
 δ^4 has been integrated out
- 5 Specify the PDF scale (here M_H)
- 6 Link the ME
Needs a prior use of the ME exporter

Block structure example [▶ Link](#)

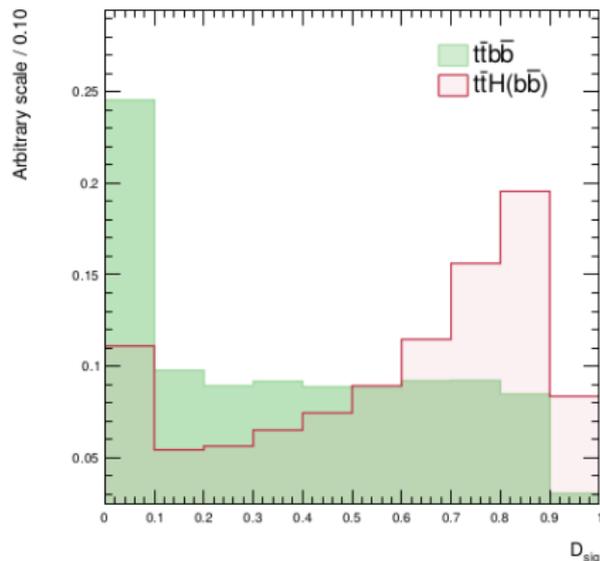
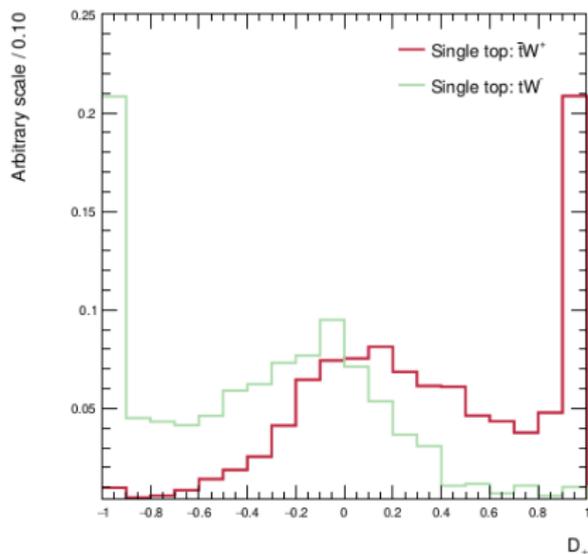
Di-Higgs production: $H(b\bar{b})H(W(l\nu)W(l\nu))$



Each block only requires a few lines of Lua code from the user !

- 1 Use the transfer function x (experimental) \rightarrow y (parton level)
- 2 Apply **Secondary Block D**
Removes $|p_5|$ or $|p_6|$
Integrates over p_{56} (BW or NWA)
- 3 Apply **Secondary Block C**
Removes $|p_4|$ (because neutrino)
Integrates over p_{34} (BW at m_W)
- 4 Apply **Main Block B**
Removes q_1, q_2, \vec{P}_2
Integrates over p_{1234} (BW or NWA)
 δ^4 has been integrated out
- 5 Specify the PDF scale (here M_H)
- 6 Link the ME
Needs a prior use of the ME exporter

Application : discrimination [▶ Link](#)



$$\mathcal{D}_{\pm}(x) = \frac{W(x|\bar{t}W^+) - W(x|tW^-)}{W(x|\bar{t}W^+) + W(x|tW^-)}.$$

$$\mathcal{D}_{\text{sig}}(x) = \left(1 + \frac{P(x|t\bar{t}b\bar{b})}{P(x|t\bar{t}H)} \right)^{-1}$$

Remaining obstacle

Gain from MoMEMta

- Complexity : **Solved**
- Computation time : **Still expensive**
(LHC data analysis sizes, parameter scans, up and down fluctuations ...)

Remaining obstacle

Gain from MoMEMta

- Complexity : **Solved**
- Computation time : **Still expensive**
(LHC data analysis sizes, parameter scans, up and down fluctuations ...)

Idea



$$P(x|\alpha) = \frac{1}{\sigma_{\alpha}^{vis}} \int_y d\phi(y) \int_{q_1, q_2} dq_1 dq_2 \sum_{a_1, a_2} f_{a_1}(q_1) f_{a_2}(q_2) |M_{\alpha}(q_1, q_2, y)|^2 W(x|y)$$

Is a function of $x = P_1, P_2, P_3, \dots$ that can be learned by a DNN

Remaining obstacle

Gain from MoMEMta

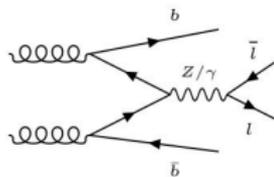
- Complexity : **Solved**
- Computation time : **Still expensive**
(LHC data analysis sizes, parameter scans, up and down fluctuations ...)

Idea

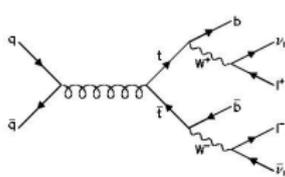
$$P(x|\alpha) = \frac{1}{\sigma_{\alpha}^{vis}} \int_y d\phi(y) \int_{q_1, q_2} dq_1 dq_2 \sum_{a_1, a_2} f_{a_1}(q_1) f_{a_2}(q_2) |M_{\alpha}(q_1, q_2, y)|^2 W(x|y)$$

Is a function of $x = P_1, P_2, P_3, \dots$ that can be learned by a DNN

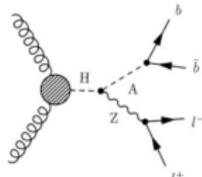
Case study : $H \rightarrow ZA \rightarrow llbb$ analysis, see [Alessia's talk](#)



~ 4s / event

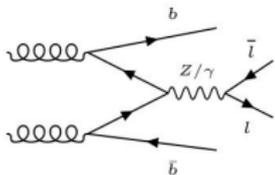


~ 15s / event

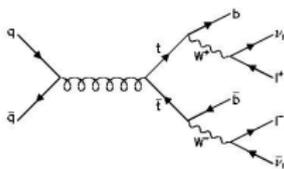
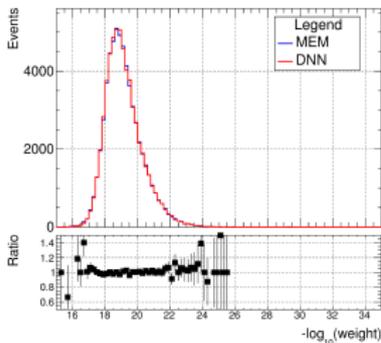


~ 10min / event / parameter

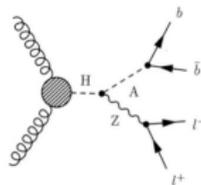
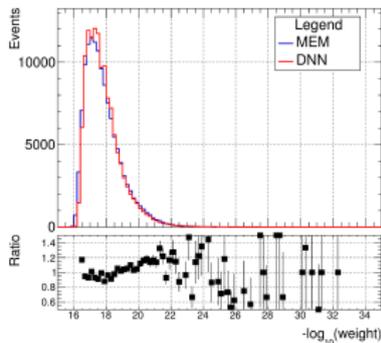
DNN regression results



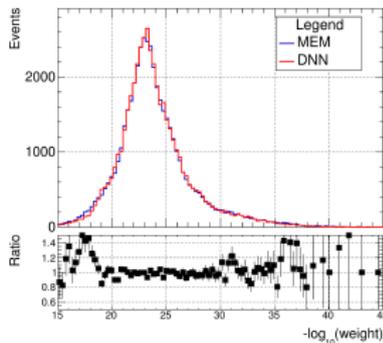
Drell-Yann sample : Ratio MEM/DNN weight Drell-Yann



$t\bar{t}$ sample : Ratio MEM/DNN weight $t\bar{t}$



Signal sample : Ratio MEM/DNN weight signal ($M_H = 800$ GeV, $M_A = 400$ GeV)

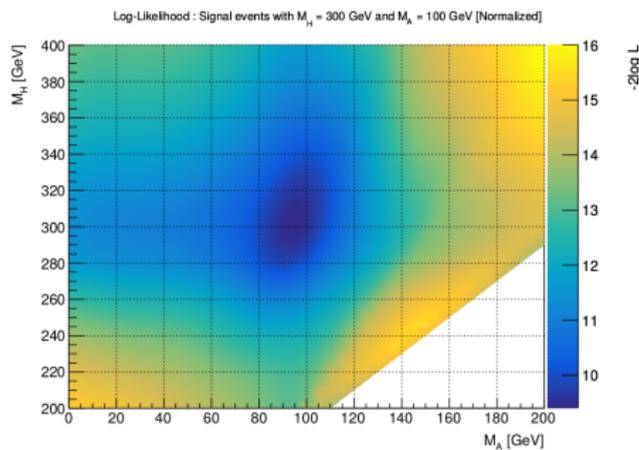


On average, with a DNN a weight can be computed in $150 \mu\text{s}$ on CPU

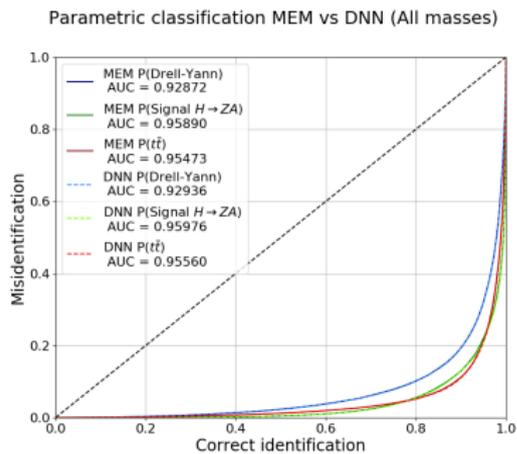
Note that the signal ($H \rightarrow ZA$) DNN is parametric in M_H and M_A

New available applications

Likelihood with parameter scan

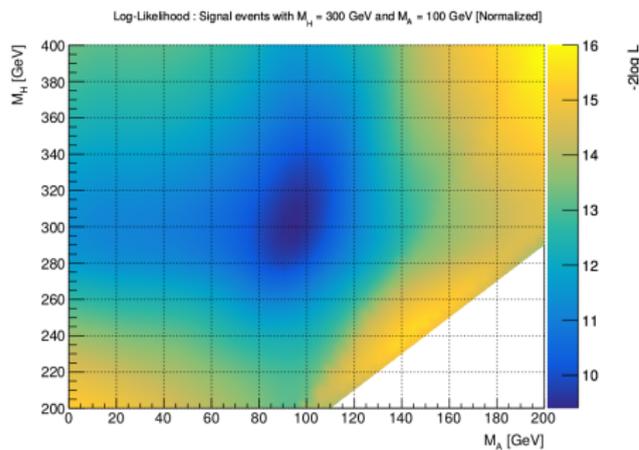


Classification

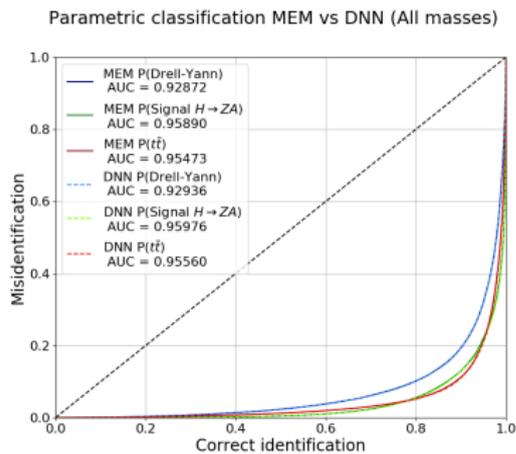


New available applications

Likelihood with parameter scan



Classification

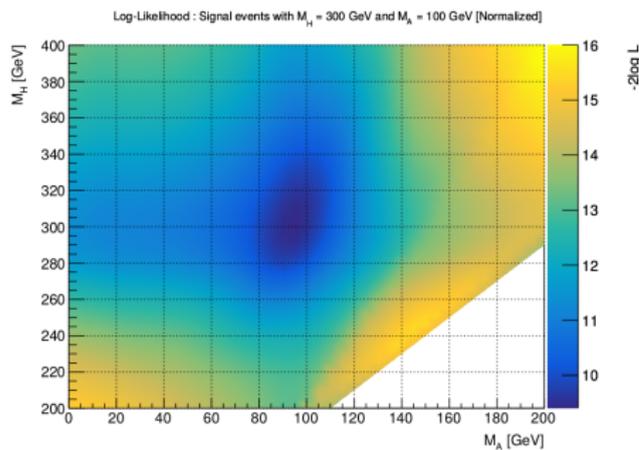


MoMEMta not directly applicable on LHC data analyses

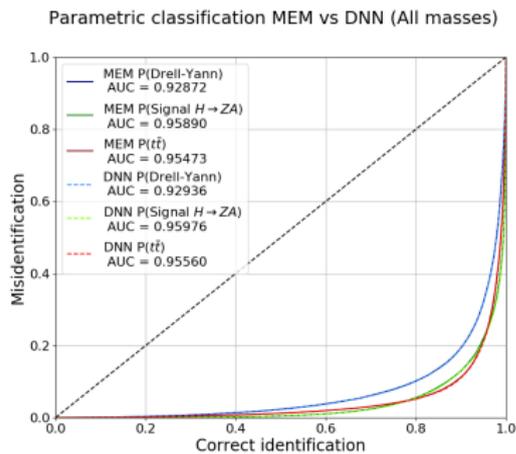
Example : classifier applied on Alessia's samples

New available applications

Likelihood with parameter scan



Classification



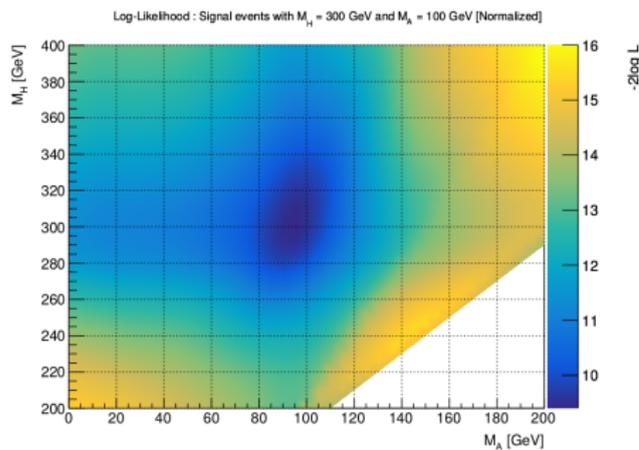
MoMEMta not directly applicable on LHC data analyses

Example : classifier applied on Alessia's samples

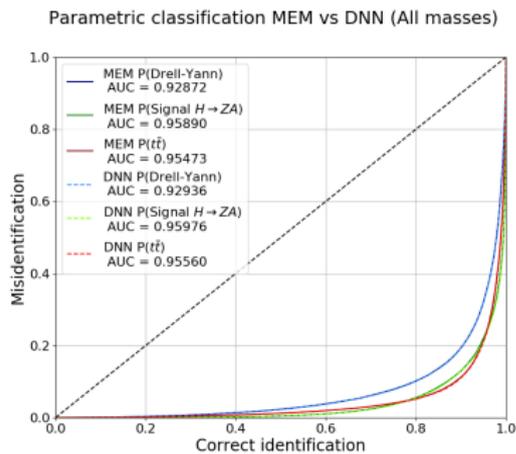
- Marginal gain in performances

New available applications

Likelihood with parameter scan



Classification



MoMEMta not directly applicable on LHC data analyses

Example : classifier applied on Alessia's samples

- Marginal gain in performances
- Computation time comparison
 - MoMEMta : ~ 3000 years (extrapolation)
 - DNN : ~ 10 hours

Take-home messages

MoMEMta [▶ Link](#)

- Modular and user-friendly (complexity hidden behind blocks)
- Customizable transfer functions
- Can use any LO model from MadGraph

⇒ MEM is now within reach of any physicist

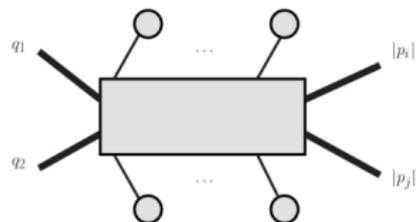
with Deep Neural Networks (paper not out yet)

- Computation time gains : 4 to 6 orders of magnitude
- Allows parameters scans or up-down fluctuations
- Always converges
- Can be used on large datasets

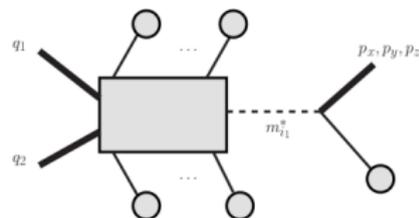
⇒ MEM is now within reach of any physicist for LHC-scale analyses

Thank you

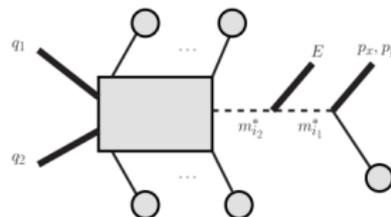
Back-up slides



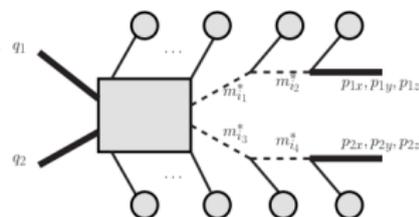
(a) MB A



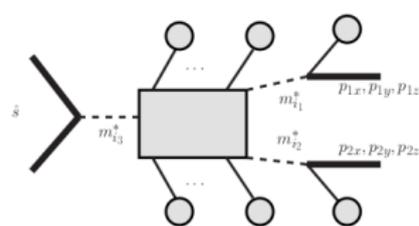
(b) MB B



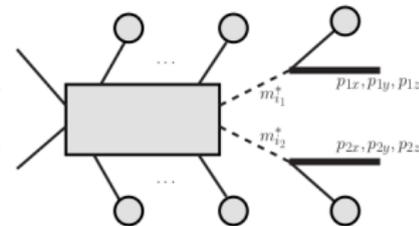
(c) MB C



(d) MB D

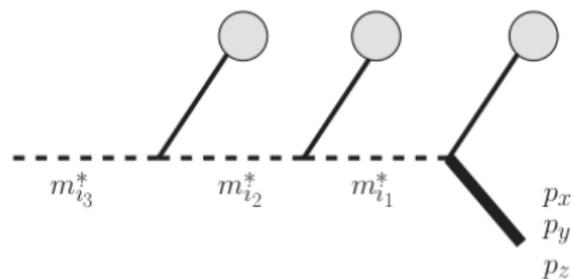


(e) MB E

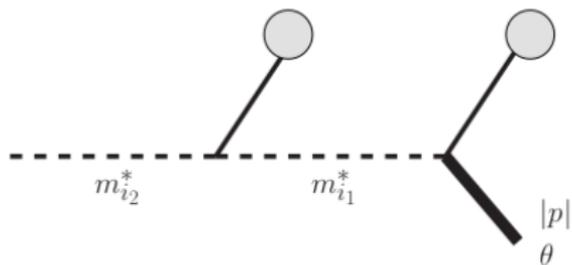


(f) MB F

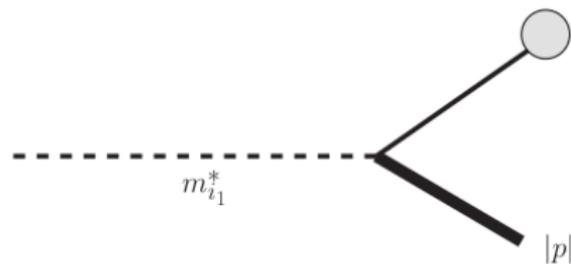
Secondary blocks [▶ Link](#)



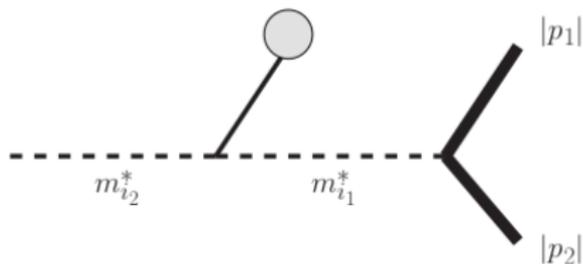
(a) SB A



(b) SB B



(c) SB C/D



(d) SB E