Matrix Element Regression with Deep Neural Networks - breaking the CPU barrier

Florian Bury

EOS PhD days 2020

ArXiv ePrint: 2008.10949

UCLouvain
Institut de recherche
en mathématique et physique

fnrs
LA LIBERTÉ DE CHERCHER

CMS

November 27, 2020

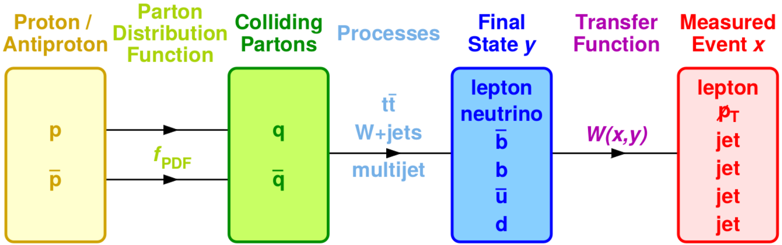
Matrix Element Method (MEM)

Matrix Element Method integral

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

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

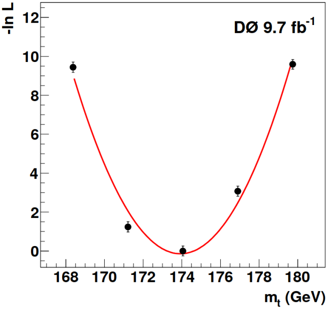
Matrix Element Method (MEM)

Matrix Element Method integral

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

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 in a nutshell

Classic MC integration

$$I = \int_\Omega f(x)dx \simeq \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Very slow convergence in high-dimension phase-space

Adaptive MC integration
Trick : introduce a sampling function g such that

$$I = \int_\Omega \frac{f(x)}{g(x)} g(x)dx \simeq \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{g(x_i)} \quad \text{where } x_i \sim g$$

Goal : variance will be reduced if $g \simeq f$
Most used algorithm in the market : Vegas

$$g(\vec{x}) = g_1(x_1) \times g_2(x_2) \times ... \times g_N(x_N) \quad \text{where } g_i \text{ are step functions}$$

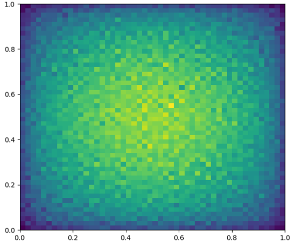
But the factorization approximation on which the algorithm is based on can impede the integration convergence

Numerical integration : a dummy example

Function to integrate : Disc $R = 0.5 \rightarrow$ Integral $= \frac{\pi}{4}$

Cartesian coordinates

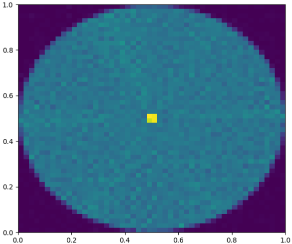
$$f(x,y) = \begin{cases} 1 & x^2 + y^2 < R^2 \\ 0 & x^2 + y^2 \geq R^2 \end{cases}$$



$I = 0.78736 \pm 0.00112$
on 10^5 points
Sharp edge at $x^2 + y^2 = R^2$

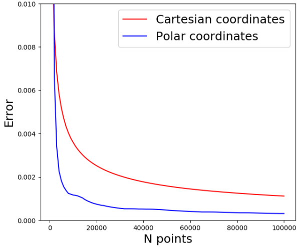
Polar coordinates

$$f(\rho,\phi) = \begin{cases} 1 & \rho < R \\ 0 & \rho \geq R \end{cases}$$



$I = 0.76968 \pm 0.00031$
on 10^5 points
Sharp edge at $\rho = R$

Variance



Conclusion : variance decreased if peak mapped onto a single variable of integration !

Matrix Element Method integral

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

Phase space parameterization

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

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

Peaks origins :

- 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 $|M_\alpha(q_1,q_2,y)|^2$: X
  Example : Breit-Wigner resonances

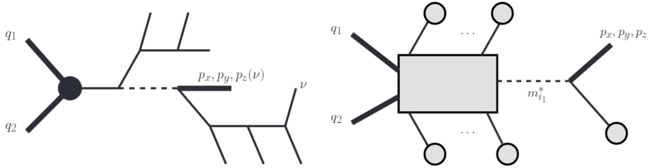
In addition, need to integrate out the δ of the momentum conservation

MoMEMta ▸ Link

A modular toolkit for the Matrix Element Method at the LHC

MoMEMta can perform the MEM integration almost out of the box

- Matrix element provided by MadGraph (with exporter ▸ Link)
- PDF from LHAPDF ▸ Link
- Transfer function : parameterized or in 2D histograms
- Integration with Cuba ▸ Link
- Modular : blocks that encompass the changes of variables ▸ Link
  - Delta integration
  - Change of variables
  - Associated Jacobians



Remaining obstacle

Gain from MoMEMta

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

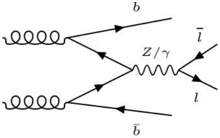
Idea

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

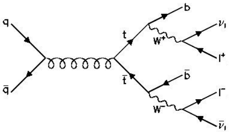
Is a function of $x = P_1, P_2, P_3, ...$ that can be learnt by a DNN

Case study : CMS 2HDM $l^+l^-b\bar{b}$ final-state analysis, Link

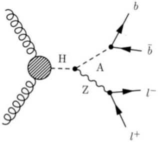
Drell-Yan process



$t\bar{t}$ (fully leptonic) process



$H \to ZA \to llbb$ process



Training specifications

Inputs :
Same as MoMEMta $\rightarrow$ 4-momentas of visible particles (here : 2 leptons + 2 jets)
In addition, several improvements :

- $(E, P_x, P_y, P_z) \rightarrow (P_T, \eta, \phi)$: not having to learn about the boost in Z direction yields better performances
- ϕ angle is relative (here : compared to leading lepton) : cylindrical symmetry
- Preprocessing : $x \rightarrow \frac{x-\mu}{\sigma}$

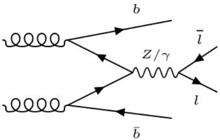
Target : $-log_{10}$(MEM weight)

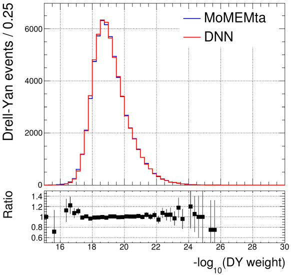
DNN architecture : results of an hyperparameter scan

- Fully-connected DNN
  - Drell-Yan process : 6 x 200 neurons
  - $t\bar{t}$ (fully leptonic) process : 8 x 500 neurons
  - $H \rightarrow ZA \rightarrow llbb$ process : 8 x 300 neurons
    Note : Parametric DNN in M_H and M_A
- Adam optimizer : LR = 0.001
- Activation functions : ReLU (hidden) + SeLU (output)
- Very small L2, dropout was not necessary
- Typical training time : < 8 hours on CPU

DNN regression results

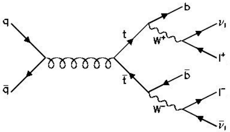
Drell-Yan process

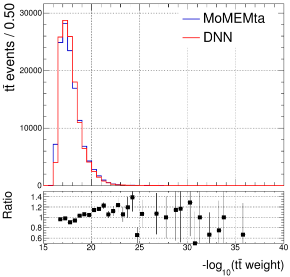




MoMEMta : 3.6 s / *event*
DNN : 110 μs / *event*

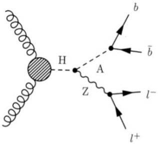
$t\bar{t}$ (fully leptonic) process

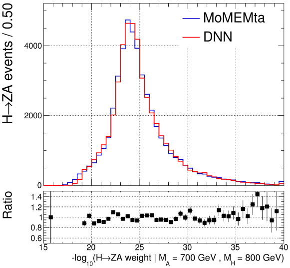




MoMEMta : 12 s / *event*
DNN : 150 μs / *event*

$H \rightarrow ZA \rightarrow llbb$ process





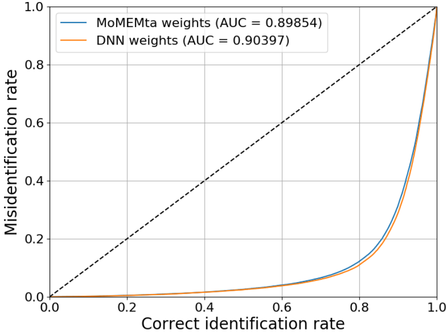
MoMEMta : 600 s / *event*
DNN : 120 μs / *event*

Conclusion : 4 to 6 orders of magnitude in time gain

Application : analytic discriminant

$$\mathcal{D}(x) = \frac{P(x|\alpha)}{P(x|\alpha) + P(x|\beta)} = \frac{W(x|\alpha)}{W(x|\alpha) + \gamma W(x|\beta)} \text{ where } \gamma = \frac{\sigma_\beta^{vis}}{\sigma_\alpha^{vis}}$$

Analysis specific example : discrimination between $\alpha = t\bar{t}$ and $\beta =$ Drell-Yan processes



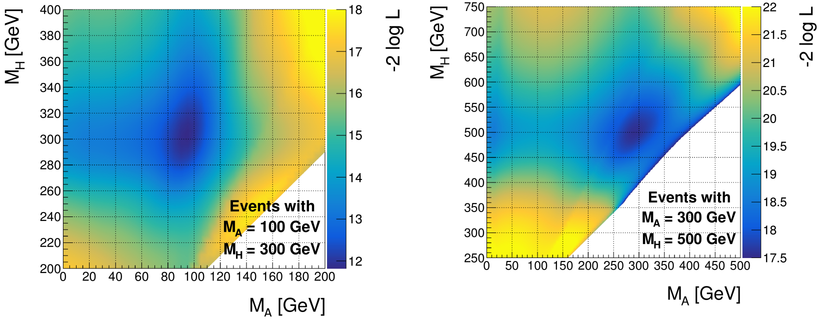
MEM weights from the DNN are as good as the ones from MoMEMta

Application : likelihood scan

$$-\log(L(x|\alpha)) = \frac{1}{n}\sum_{i=1}^{n} -\log(P(x_i|\alpha)) = \frac{1}{n}\sum_{i=1}^{n} -\log(W(x_i|\alpha)) + \log(\sigma_\alpha^{vis})$$

Analysis specific example : $H \to ZA \to llbb$ scan in M_A and M_H

Method : Use the parametric DNN to produce MEM weights as a function of (M_A, M_H)

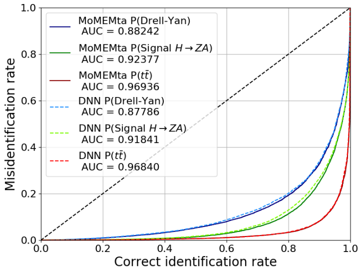


Calls to the MEM : $N_{events} \times (N_{params})^{dimension}$ $\rightarrow$ cannot be done with MoMEMta
Parametric DNN : only trained on few N_{params} $\rightarrow$ can be evaluated for any value

Application : multi-classification

Analysis specific example : use the MEM weights in a classifier (DNN)

Global classifier

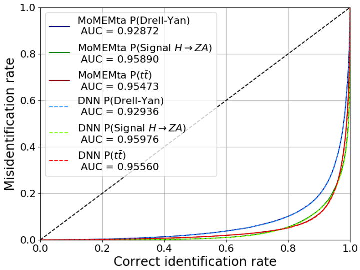


Inputs (25) :

- Drell-Yan weight
- $t\bar{t}$ weight
- $H \to ZA$ (x23 parameters) weights

Purpose : search for an excess on the whole mass plane (M_{bb},M_{llbb})

Parametric classifier



Inputs (5) :

- Drell-Yan weight
- $t\bar{t}$ weight
- $H \to ZA$ weight + M_A + M_H

Purpose : search for an excess at specific mass points ($\to$ look-elsewhere effect)

Same classifier performances when using MEM weights from MoMEMta or the DNNs

Additional studies

Systematic uncertainties
Case study : Jet Energy Scale (JES) $\rightarrow$ Apply a 10% increase in bjets P_T
Observations :

- DNN is able to reproduces the MEM weight of JES shifted events
- Analytic discriminant $\mathcal{D}$ shows no loss of performance

Guaranteed convergence
Case study : numerical integration may not converge in MoMEMta, not the case with the DNN
Observations :

- Not perfect agreement between recomputed weights and the DNN prediction
- Analytic discriminant $\mathcal{D}$ shows better performances with weights from the DNN

Real-life analysis
Case study : Combination of method in CMS $H \rightarrow ZA$ analysis arXiv and the classifiers
Observations :

- No gain from using the global classifier
- Marginal gain when combining the method from CMS and the parametric classifier
- Time estimation to produce MEM weights needed by the classifiers
  - MoMEMta : $\sim$ 3000 years
  - DNN : $\sim$ 10 hours

Matrix element regression

Main caveat of the method : transfer functions cannot be changed after learning

Idea : only learn $\sum_{a_1,a_2} f_{a_1}(q_1)f_{a_2}(q_2)|M_\alpha(q_1,q_2,y)|^2$ and integrate over the DNN output

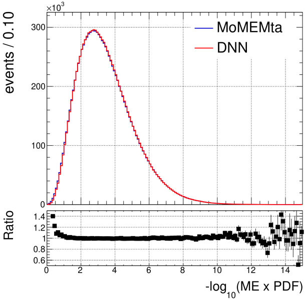
Case study : $t\bar{t}$ fully leptonic $\rightarrow$ 6 generator level particles

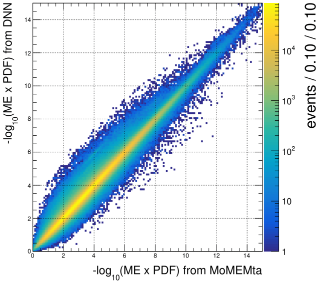
DNN inputs :

- Particles P_T + q_1 and q_2 E
- 2- and 3-objects invariant mass
- 2-objects 4-momentas products

DNN architecture :

- 10 x 200 neurons
- ReLU + ELU activations
- Generator (80M events, 20min/epoch)





Conclusion : $t_{DNN} \lessapprox t_{MadGraph}$ and the small fluctuations in the integrand made the effort not worth it in terms of computation time

Conclusion

Matrix Element Method

- Powerful tool to combine knowledge of the SM and the detector
- Numerical integration suffers from complexity and expensive computation time

MoMEMta Link

- Includes all the necessary components of the integration
- Hides the complexity behind change of variable blocks

$\Rightarrow$ MEM is now within reach of any physicist

With Deep Neural Networks : arXiv:2008.10949 (submitted to JHEP)

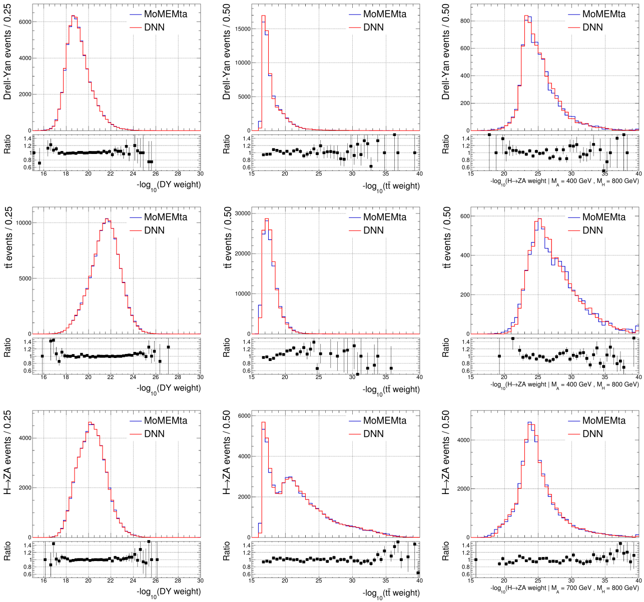
- Computation time gains : 4 to 6 orders of magnitude
- Allows parameters scans or up-down fluctuations
- Always converges
- Can be used on large datasets

$\Rightarrow$ MEM is now within reach of any physicist <u>for LHC-scale analyses</u>

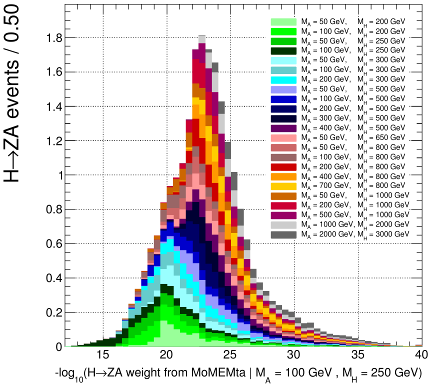
Thank you for your attention

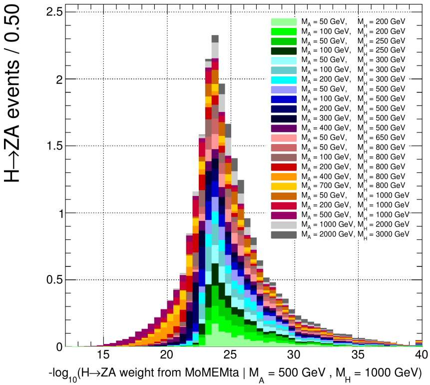
Backup

DNN regression plots

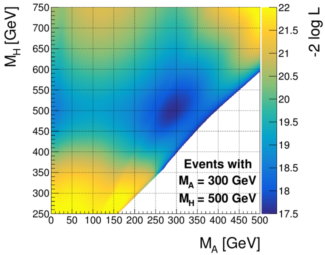


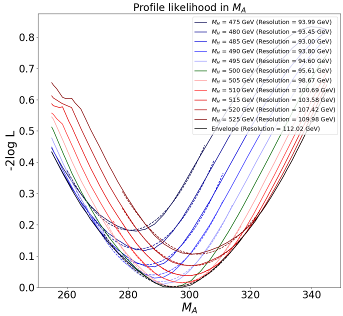
Signal MEM weights

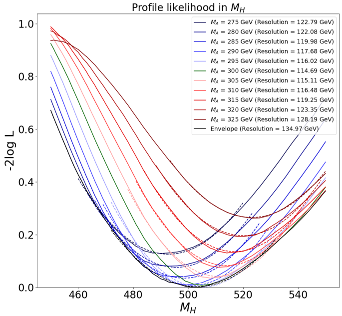




Likelihood scan profiles







Systematic uncertainties : JES shifted events

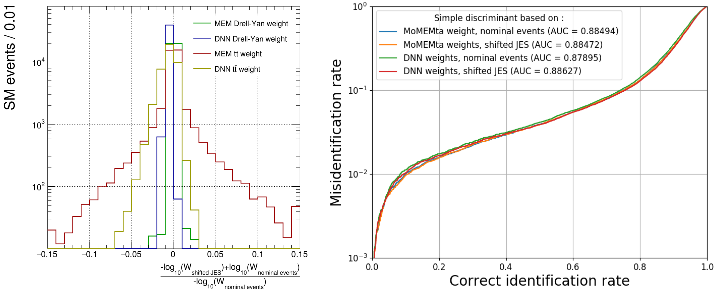
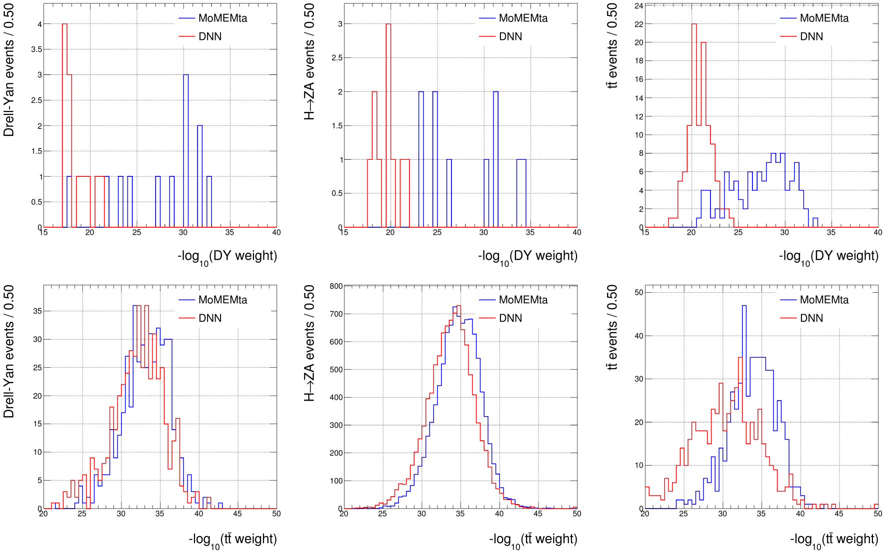


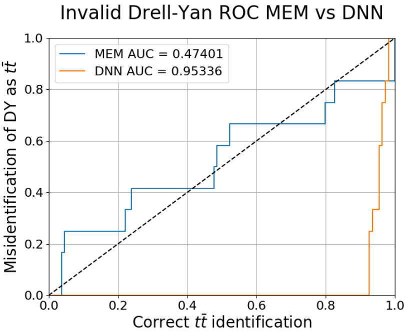
Table: Regression bias and resolution in the event information when replacing the integration with MoMEMta by the DNN ansatz for the two SM weights with nominal and shifted JES events.

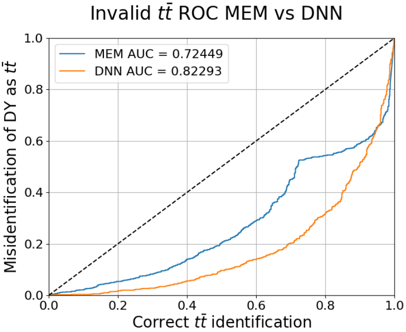
	Regression bias	Regression resolution
Nominal Drell-Yan	-0.1243	0.1383
Shifted JES Drell-Yan	0.0049	0.1351
Nominal $t\bar{t}$	-0.2758	0.4439
Shifted JES $t\bar{t}$	-0.1659	0.4137

Invalid weights (failed numerical convergence)



Analytic discriminant on invalid weights





Real-life analysis : CMS $H \rightarrow ZA \rightarrow l^+l^-b\bar{b}$