Likelihood-based unfolding with the CMS Higgs combination tool

Alessandro Tarabini on behalf of the CMS collaboration
(ETH Zürich, IPA)

Unfolding2024: France-Berkeley PHYSTAT Conference on Unfolding
11/06/2024

Likelihood-based unfolding with the CMS Hi~~ination tool~~

Combine

Alessandro Tarabini on behalf of the CMS collaboration
(ETH Zürich, IPA)

Unfolding2024: France-Berkeley PHYSTAT Conference on Unfolding
11/06/2024

CMS Combine

ETH zürich

IPA

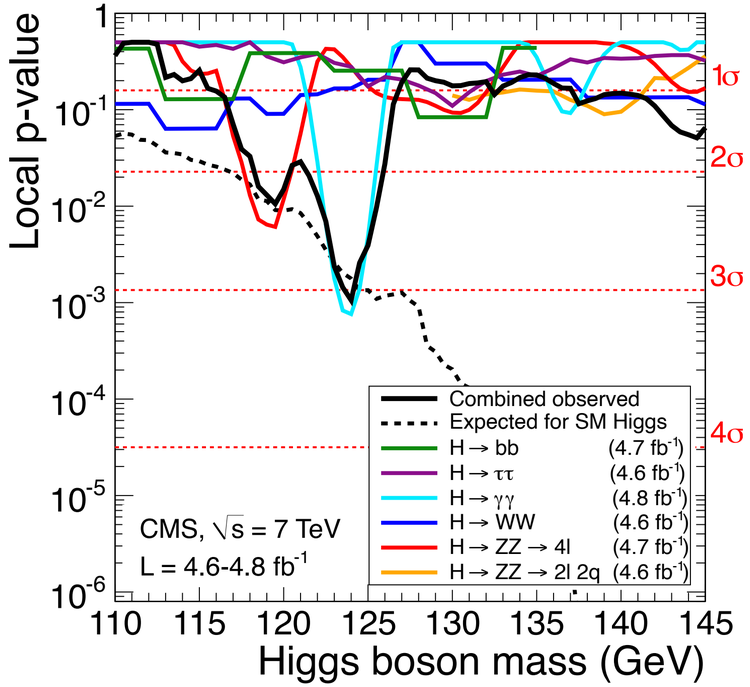
What is Combine?

Combine is the software package used for statistical analyses by the CMS collaboration

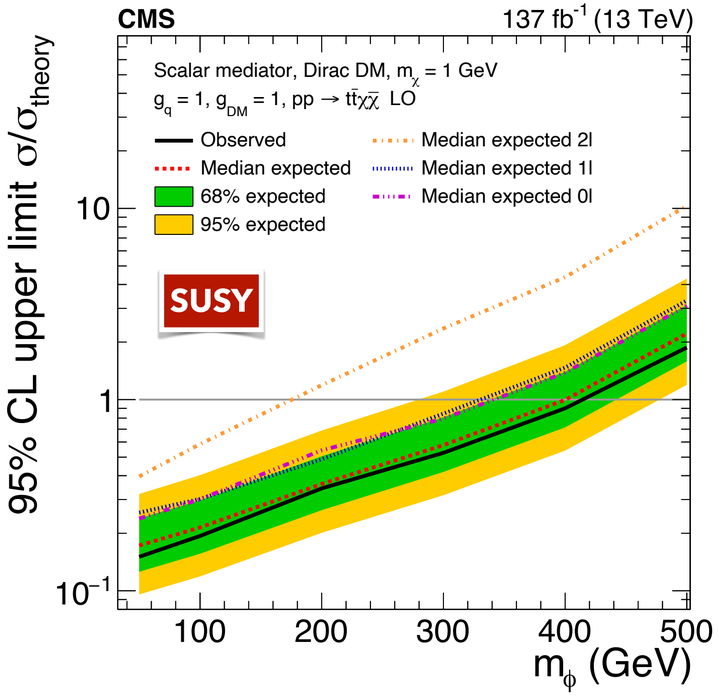
Built around the ROOT, RooFit, and RooStats packages (+ additional libraries for optimised algebraic calculations)

Originally designed for searches for the Higgs boson and their combination…

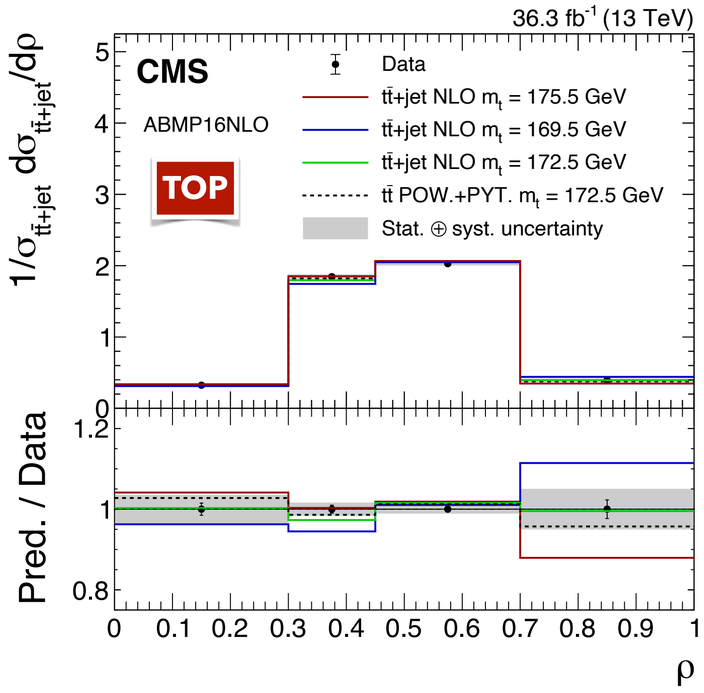
… it has been extended and used in numerous publications for the CMS Collaborations



10.1016/j.physletb.2012.02.064



10.1140/epjc/s10052-021-09721-5

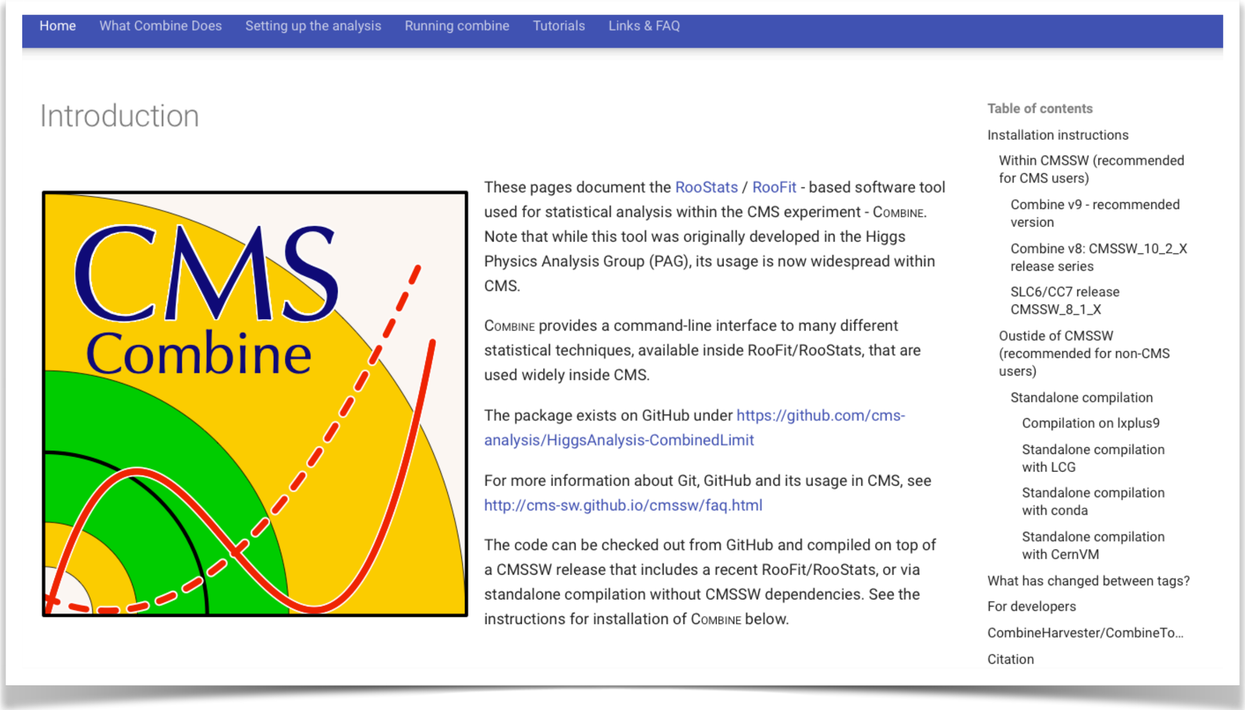


10.1007/JHEP07(2023)077

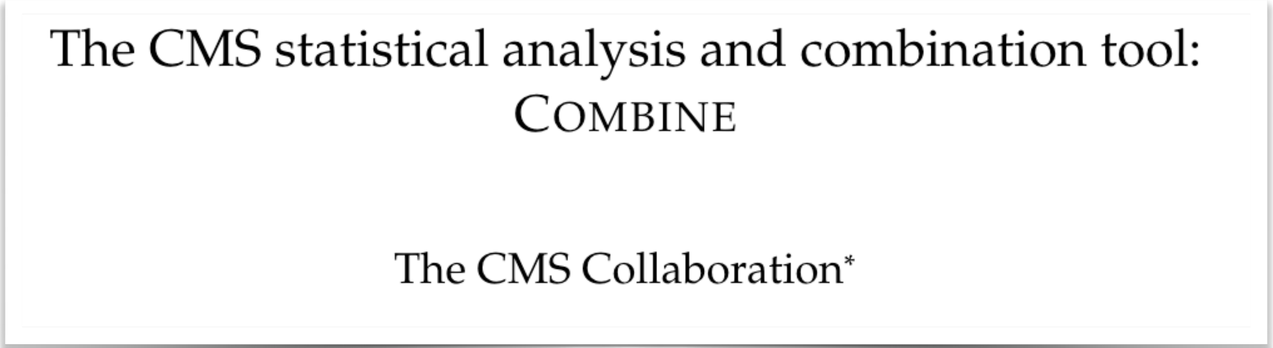
Combine is public!

While Combine is developed for CMS analysis, and with CMS users in mind, the code is public and can be compiled in *standalone* mode

Website



Paper



CMS will release **full statistical models** of analyses, the one for the Higgs Boson discovery is already available here

How Combine works? – The datacard

Datacard: configuration file in a plain text format that represents the primary input for Combine

Example of a simple datacard for a counting experiment

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin             ch1    ch1    ch1
process         ppX    WW     tt
process         0      1      2
rate            1.47   0.64   0.22
# ----------
lumi     lnN    1.11   1.11    1.11
xs       lnN    1.20    -       -
nWW      gmN 4   -    0.16     -
```

How Combine works? – The datacard

Datacard: configuration file in a plain text format that represents the primary input for Combine

Example of a simple datacard for a counting experiment

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin             ch1    ch1    ch1
process         ppX    WW     tt
process         0      1      2
rate            1.47   0.64   0.22
# ----------
lumi     lnN    1.11   1.11    1.11
xs       lnN    1.20    -       -
nWW      gmN 4   -     0.16     -
```

Number of bins/channel

Number of processes

Number of nuisance parameters

Unique channel label

Number of observed events in a channel

Process label

Process ID (<=0 for signal)

Expected number of events

Systematic uncertainties

Name Type Effect on process

How Combine works? – text2workspace

text2workspace.py: Convert info in the datacard into a binary ROOT file containing the statistical model in the form of a RooFit workspace

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin
process
process
rate
```

- The probability to observe n events is described by a Poisson distribution
- $\lambda(r, \vec{\nu})$ represents the total number of expected events

$$p(n, \vec{y}; r, \vec{\nu}) = \frac{\lambda(r, \vec{\nu})^n}{n!}$$

$$\rightarrow n = 0$$

$$\rightarrow \lambda(r, \vec{\nu}) =$$

How Combine works? – text2workspace

text2workspace.py: Convert info in the datacard into a binary ROOT file containing the statistical model in the form of a RooFit workspace

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin             ch1
process         ppX
process         0
rate            1.47
```

- The probability to observe n events is described by a Poisson distribution
- $\lambda(r, \vec{\nu})$ represents the total number of expected events
- One signal process (**ppX**)

$$p(n, \vec{y}; r, \vec{\nu}) = \frac{\lambda(r, \vec{\nu})^n}{n!}$$

$$\rightarrow n = 0$$

$$\rightarrow \lambda(r, \vec{\nu}) = \boxed{1.47}$$

How Combine works? – text2workspace

text2workspace.py: Convert info in the datacard into a binary ROOT file containing the statistical model in the form of a RooFit workspace

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin             ch1    ch1    ch1
process         ppX    WW     tt
process         0      1      2
rate            1.47   0.64   0.22
```

- The probability to observe n events is described by a Poisson distribution
- $\lambda(r, \vec{\nu})$ represents the total number of expected events
- One signal process (ppX)
- Two background processes (**WW** and **tt**)

$$p(n, \vec{y}; r, \vec{\nu}) = \frac{\lambda(r, \vec{\nu})^n}{n!}$$

$$\rightarrow n = 0$$

$$\rightarrow \lambda(r, \vec{\nu}) = 1.47 + 0.22 + 0.64$$

How Combine works? – text2workspace

text2workspace.py: Convert info in the datacard into a binary ROOT file containing the statistical model in the form of a RooFit workspace

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin             ch1    ch1    ch1
process         ppX    WW     tt
process         0      1      2
rate            1.47   0.64   0.22
# ----------
lumi     lnN    1.11   1.11   1.11
xs       lnN    1.20    -      -
nWW      gmN 4   -     0.16    -
```

- The probability to observe n events is described by a Poisson distribution
- $\lambda(r, \vec{\nu})$ represents the total number of expected events
- One signal process (ppX)
- Two background processes (WW and tt)
- Each systematic uncertainty results in an associated probability term

$$p(n, \vec{y}; r, \vec{\nu}) = \frac{\lambda(r, \vec{\nu})^n}{n!} e^{-\lambda(r,\vec{\nu})} \frac{1}{2\pi} \boxed{e^{-(\nu_{\text{lumi}} - y_{\text{lumi}})^2}} \boxed{e^{-(\nu_{\text{xs}} - y_{\text{xs}})^2}} \boxed{\frac{(\nu_{\text{nWW}})^{y_{\text{nWW}}}}{y_{\text{nWW}}!} e^{-(\nu_{\nu_{\text{nWW}}})}}$$

$$\rightarrow n = 0,\ y_{\text{lumi}} = y_{\text{xs}} = 0,\ y_{\text{nWW}} = 4$$

$$\rightarrow \lambda(r, \vec{\nu}) = 1.47\, \boxed{(1.11)^{\nu_{\text{lumi}}}} \boxed{(1.2)^{\nu_{\text{xs}}}} + 0.22\, \boxed{(1.11)^{\nu_{\text{lumi}}}} + 0.64\, \boxed{(1.11)^{\nu_{\text{lumi}}}} \boxed{\frac{\nu_{\text{nWW}}}{0.64}}$$

How Combine works? – text2workspace

text2workspace.py: Convert info in the datacard into a binary ROOT file containing the statistical model in the form of a RooFit workspace

```
imax 1
jmax 2
kmax 3
# A single channel - ch1 - in which 0 events are observed in data
bin             ch1
observation     0
# ----------
bin             ch1    ch1    ch1
process         ppX    WW     tt
process         0      1      2
rate            1.47   0.64   0.22
# ----------
lumi     lnN    1.11   1.11    1.11
xs       lnN    1.20    -      -
nWW      gmN 4   -     0.16    -
```

- **Default physics model**: the rate of every signal process is multiplied by a common factor r
- **Customised physics model**: scaling signals with different scaling parameters, parametrisation of signal processes for interpretations (κ-framework, EFT, AC, …), signal and background interference, …

$$p(n, \vec{y}; r, \vec{\nu}) = \frac{\lambda(r,\vec{\nu})^n}{n!} e^{-\lambda(r,\vec{\nu})} \frac{1}{2\pi} e^{-(\nu_{\text{lumi}} - y_{\text{lumi}})^2} \; e^{-(\nu_{\text{xs}} - y_{\text{xs}})^2} \; \frac{(\nu_{\text{nWW}})^{y_{\text{nWW}}}}{y_{\text{nWW}}!} e^{-(\nu_{\nu_{\text{nWW}}})}$$

$$\rightarrow n = 0,\; y_{\text{lumi}} = y_{\text{xs}} = 0,\; y_{\text{nWW}} = 4$$

$$\rightarrow \lambda(r,\vec{\nu}) = \boxed{r}\, 1.47\; (1.11)^{\nu_{\text{lumi}}}\; (1.2)^{\nu_{\text{xs}}} + 0.22\; (1.11)^{\nu_{\text{lumi}}} + 0.64\; (1.11)^{\nu_{\text{lumi}}} \frac{\nu_{\text{nWW}}}{0.64}$$

Let's make it more complicated

Default binned model

$$\Lambda(\vec{\alpha}) = \frac{L(\vec{\alpha}, \hat{\hat{\vec{\theta}}}(\vec{\alpha}))}{L(\hat{\vec{\alpha}}, \hat{\vec{\theta}})}$$

$$\mathcal{L}(\text{data} \mid \vec{\alpha}, \vec{\theta}, \vec{\rho}) = \prod_{c=1}^{N_C} \prod_{b=1}^{N_B^c} \text{Poisson}(n_{cb} \mid \lambda_{cb}(\vec{\alpha}, \vec{\theta}, \vec{\rho})) \prod_{i=1}^{N_E} p_e(\tilde{\theta}_e \mid \theta_e).$$

Exp. events per bin

NP constraints

$$\lambda_{cb} = \max\left(0, \sum_p M_{cp}(\vec{\alpha}) N_{cp}(\vec{\theta}_L, \vec{\theta}_S, \vec{\theta}_G, \rho) y_{cbp}(\vec{\theta}_S) + E_{cb}(\vec{\theta}_B)\right)$$

Physics model scaling — Process norm. — Process templates — MC stats

Gamma — Log-normal — Asymmetric log-normal

$$N = N_0(\theta_G) \prod_n \kappa_n^{\theta_{L,n}} \prod_a \kappa_a^A(\theta^a_{L(S)}, \kappa_a^+, \kappa_a^-)^{\theta^a_{L(S)}} \prod_r F_r(\vec{\rho}),$$

rateParams

$\theta_G / \tilde{\theta}_G$

$$y_b(\vec{\theta}_S) = \begin{cases} \max\left(0, y^0 \left(f_b^0 + \sum_s F(\theta_s, \delta_b^{s,+}, \delta_b^{s,-}), \epsilon_s\right)\right) & \text{(direct)}, \\ \max\left(0, y^0 \exp\left(\ln(f_b^0) + \sum_s F(\theta_s, \Delta_b^{s,+}, \Delta_b^{s,-}, \epsilon_s)\right)\right) & \text{(logarithmic)}, \end{cases}$$

Vertical morphing

$$f_b = y_b / \sum y_b$$

$$y^0 = \sum y_b^0, \delta^\pm = f_i^\pm - f_i^0, \text{ and } \Delta^\pm = \ln(f_i^\pm) - \ln(f_i^0).$$

$$\kappa^A(\theta, \kappa^+, \kappa^-) = \begin{cases} \kappa^+, & \text{for } \theta \geq 0.5; \\ \kappa^-, & \text{for } \theta \leq -0.5; \\ \exp\left(\frac{1}{2}\left((\ln\kappa^+ + \ln\kappa^+) + \frac{1}{4}(\ln\kappa^+ - \ln\kappa^-) I(\theta)\right)\right), & \text{otherwise}, \end{cases}$$

Interpolation between up and down variations (norm)

$$\kappa_s^\pm = \frac{\sum_b y_b^{s,\pm}}{\sum_b y_b^0}.$$

Shape uncert. norm. change factored out

$$F(\theta, \delta^+, \delta^-) = \begin{cases} \frac{1}{2}\theta'\left((\delta^+ - \delta^-) + \frac{1}{8}(\delta^+ + \delta^-)(3\bar{\theta}^5 - 10\bar{\theta}^3 + 15\bar{\theta})\right), & \text{for } \theta < q; \\ \theta'\delta^+, & \text{for } \theta > q; \\ -\theta'\delta^-, & \text{for } \theta < -q; \end{cases}$$

Interpolation between up and down variations (shape)

Barlow-Beeston

"lite"

$$E_{cb}(\theta) = \theta\left(\sum_p (e_{cpb} N_{cp} M_{cp}(\vec{\alpha}))^2\right)^{\frac{1}{2}}.$$

full

$$E_{cb}(\vec{\theta}) = \overset{\text{Poisson}}{\sum_i} \left(\frac{\theta_i}{\tilde{\theta}_i} - 1\right) y_{cib} N_{cp} M_{ci}(\vec{\alpha}) + \overset{\text{Gaussian}}{\sum_j} \theta_j e_{cjb} N_{cj} M_{cj}(\vec{\alpha}),$$

Credits: A. Gilbert

How Combine works? – Running the tool

Combine is run through a command line

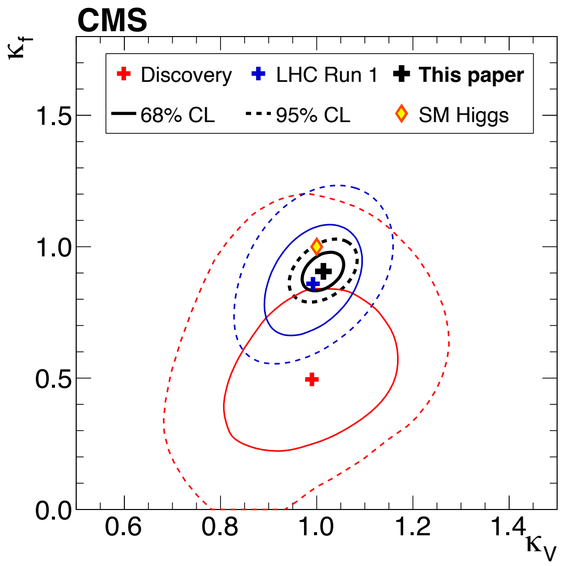
```
combine -n _mass4l_r_smH_0 -M MultiDimFit SM_125_all_13TeV_xs_mass4l_bin_v3.root -m 125.38 --freezeParameters MH --saveWorkspace --algo=grid --floatOtherPOIs=1 --points=200 --cminDefaultMinimizerStrategy 0 -P r_smH_mass4l_0 --setParameterRanges r_smH_mass4l_0=0.0,2 --redefineSignalPOI r_smH_mass4l_0
```

- All **commonly used statistical methods** are implemented
  - Limit setting (asymptotic and toy-based)
  - Significance / p-value calculation
  - Confidence interval

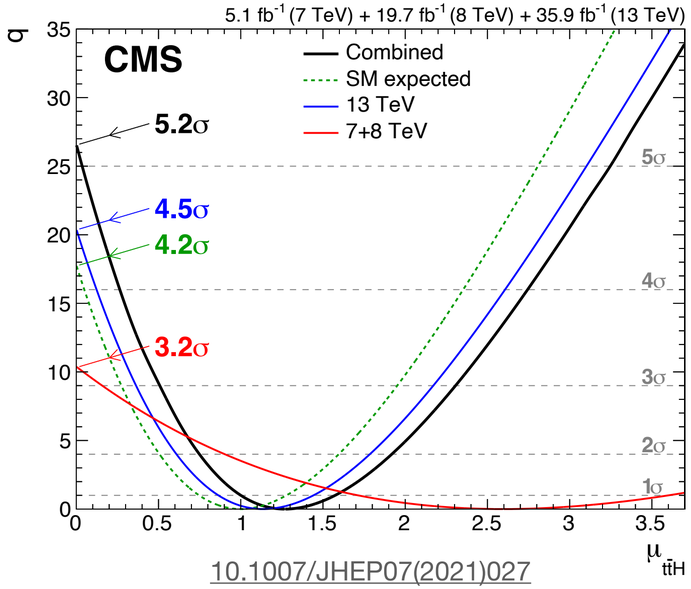
Typically, these methods make use of the **profile negative-log-likelihood function**, in which the nuisance parameters are profiled

$$\mathscr{L}(\vec{\mu},\vec{\nu}) = \prod_d p(\vec{x_d};\vec{\mu},\vec{\nu}) \prod_k p_k(y_k;\nu_k)$$

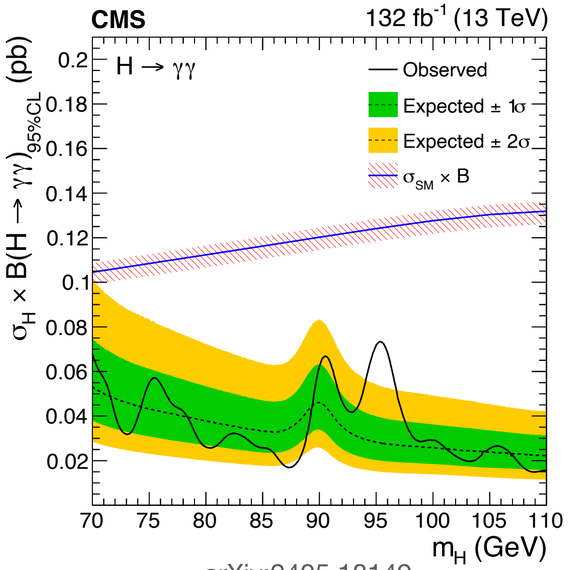
$$-\ln \mathscr{L}(\vec{\mu},\hat{\hat{\vec{\nu}}}(\vec{\mu}))$$



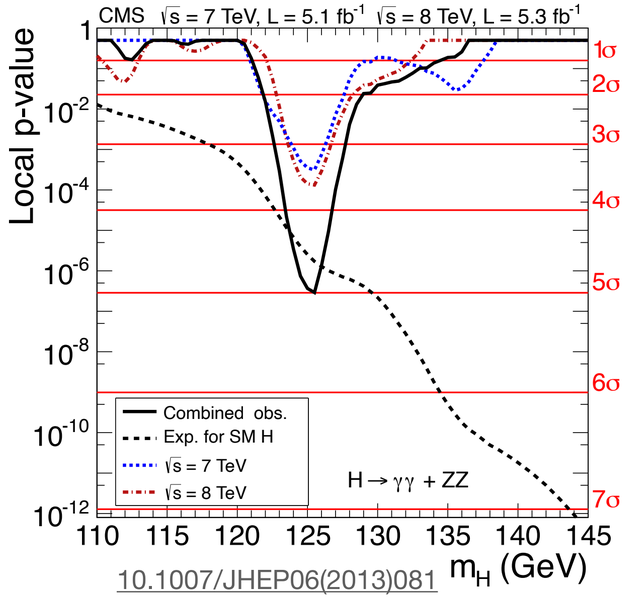
10.1038/s41586-022-04892-x



10.1007/JHEP07(2021)027



arXiv:2405.18149



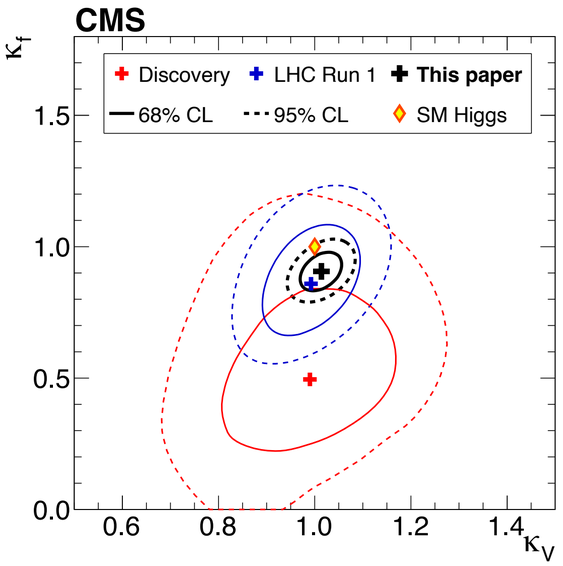
10.1007/JHEP06(2013)081

How Combine works? – Running the tool

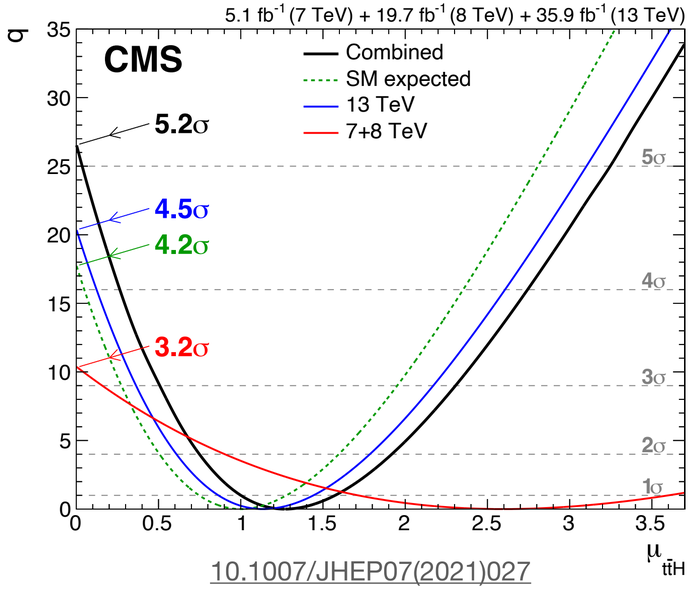
Combine is run through a command line

```
combine -n _mass4l_r_smH_0 -M MultiDimFit SM_125_all_13TeV_xs_mass4l_bin_v3.root -m 125.38 --freezeParameters MH --saveWorkspace --algo=grid --
floatOtherPOIs=1 --points=200 --cminDefaultMinimizerStrategy 0 -P r_smH_mass4l_0 --setParameterRanges r_smH_mass4l_0=0.0,2 --redefineSignalPOI r_smH_mass4l_0
```

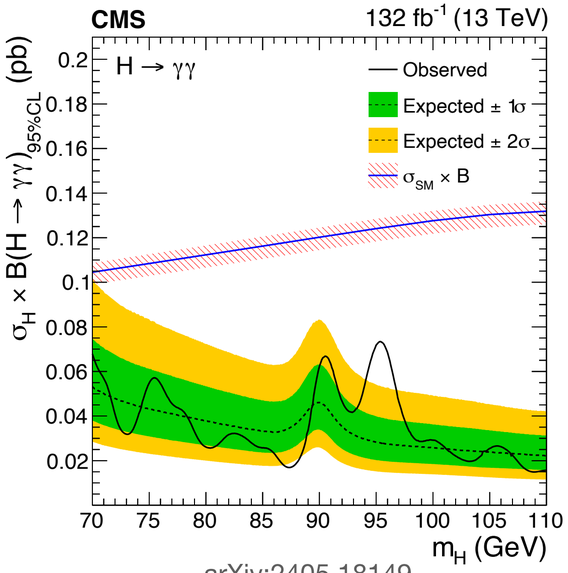
- All **commonly used statistical methods** are implemented
  - Limit setting (asymptotic and toy-based)
  - Significance / p-value calculation
  - Confidence interval
- All methods can run on **real data** or internally generated **toys/Asimov** datasets
- Also **diagnostics** and **model information**
  - Pre-/post-fit yields, shape, and uncertainties
  - Covariance matrices



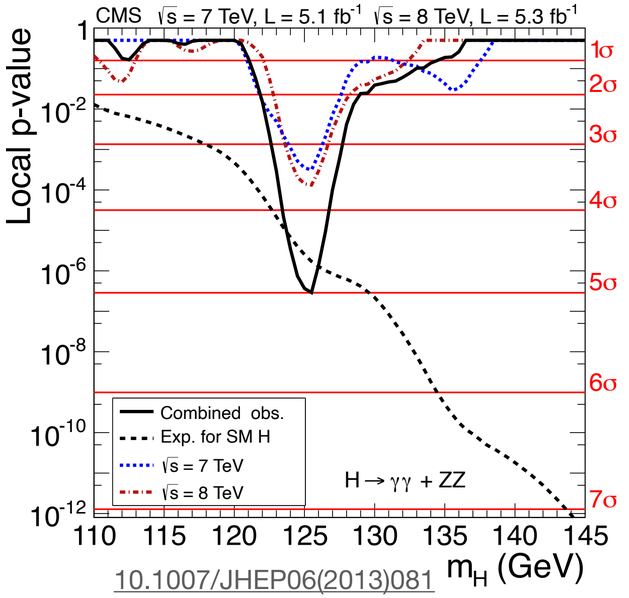
10.1038/s41586-022-04892-x



10.1007/JHEP07(2021)027



arXiv:2405.18149



10.1007/JHEP06(2013)081

Likelihood-based unfolding

Since Combine has access to the full likelihood, it can be used to perform likelihood-based unfolding

$$\vec{x}_{\text{reco}} = \mathbf{R} \cdot \vec{x}_{\text{true}} + \vec{b}$$

$$\mathscr{L}(\vec{\mu}, \vec{\nu}) = \prod_{i}^{\text{reco}} \text{Poiss}\left(x_{\text{reco},i} \middle| \sum_{j}^{\text{true}} \mu_j \cdot x_{\text{true},i}(\vec{\nu}) \cdot \text{R}_{ij}(\vec{\nu}) + b_i(\vec{\nu})\right) \prod_{k} p_k(\tilde{\nu}_k \,|\, \nu_k)$$

Advantages of likelihood-based unfolding

- Signal extraction + unfolding + inclusion of systematic uncertainties + measurement of physics quantities (+ regularisation) in one shot
- Background subtraction is accounted directly in the likelihood
- Systematic uncertainties are accounted for directly during the unfolding as nuisance parameters
- We can profile the nuisance parameters during the unfolding to make the most of the data available

Likelihood-based unfolding: counting experiment

$$\mathscr{L}(\vec{\mu},\vec{\nu}) = \prod_{i}^{\text{reco}} \text{Poiss}\left(x_{\text{reco},i} \middle| \sum_{j}^{\text{true}} \mu_j \cdot x_{\text{true},i}(\vec{\nu}) \cdot \text{R}_{ij}(\vec{\nu}) + b_i(\vec{\nu})\right) \prod_{k} p_k(\tilde{\nu}_k \,|\, \nu_k)$$

- Input parameters in the model are the signal, bkg, and data yields
- In each reco-level category, consider the contributions of all true-level bins
- Rarely used, lower sensitivity

Unfolding with two reco-level categories and two true-level categories

```
imax *
jmax *
kmax *
# ---------------------------------
bin            reco1 reco2
observation    1500  1900
# ---------------------------------
bin        reco1 reco1 reco1 reco2 reco2 reco2
process    true1 true2 bkg   true1 true2 bkg
process    0     -1    1     0     -1    1
rate       1000  70    200   40    1300  250
# ---------------------------------
# systematics
# ---------------------------------
```

Likelihood-based unfolding: counting experiment

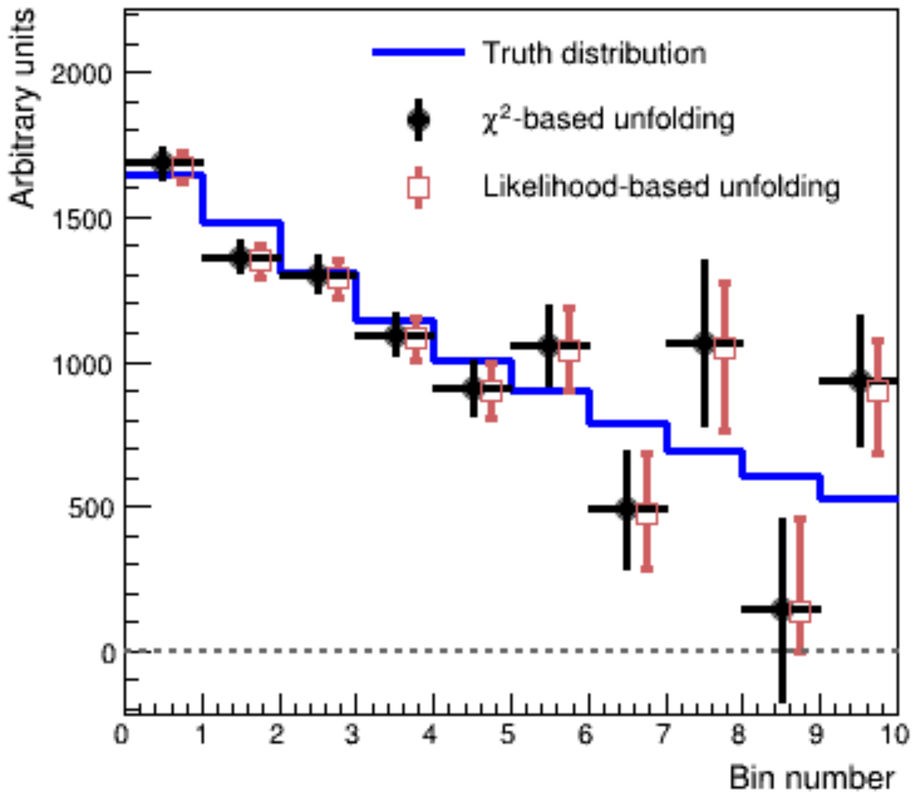
$$\mathscr{L}(\vec{\mu},\vec{\nu}) = \prod_{i}^{\text{reco}} \text{Poiss}\left(x_{\text{reco},i} \,\middle|\, \sum_{j}^{\text{true}} \mu_j \cdot x_{\text{true},i}(\vec{\nu}) \cdot \text{R}_{ij}(\vec{\nu}) + b_i(\vec{\nu})\right) \prod_{k} p_k(\tilde{\nu}_k \,|\, \nu_k)$$

- Input parameters in the model are the signal, bkg, and data yields
- In each reco-level category, consider the contributions of all true-level bins
- Rarely used, lower sensitivity

Unfolding with two reco-level categories and two true-level categories

```
imax *
jmax *
kmax *
# ---------------------------------
bin             reco1  reco2
observation     1500   1900
# ---------------------------------
bin         reco1  reco1  reco1  reco2  reco2  reco2
process     true1  true2  bkg    true1  true2  bkg
process     0      -1     1      0      -1     1
rate        1000   70     200    40     1300   250
# ---------------------------------
# systematics
# ---------------------------------
```

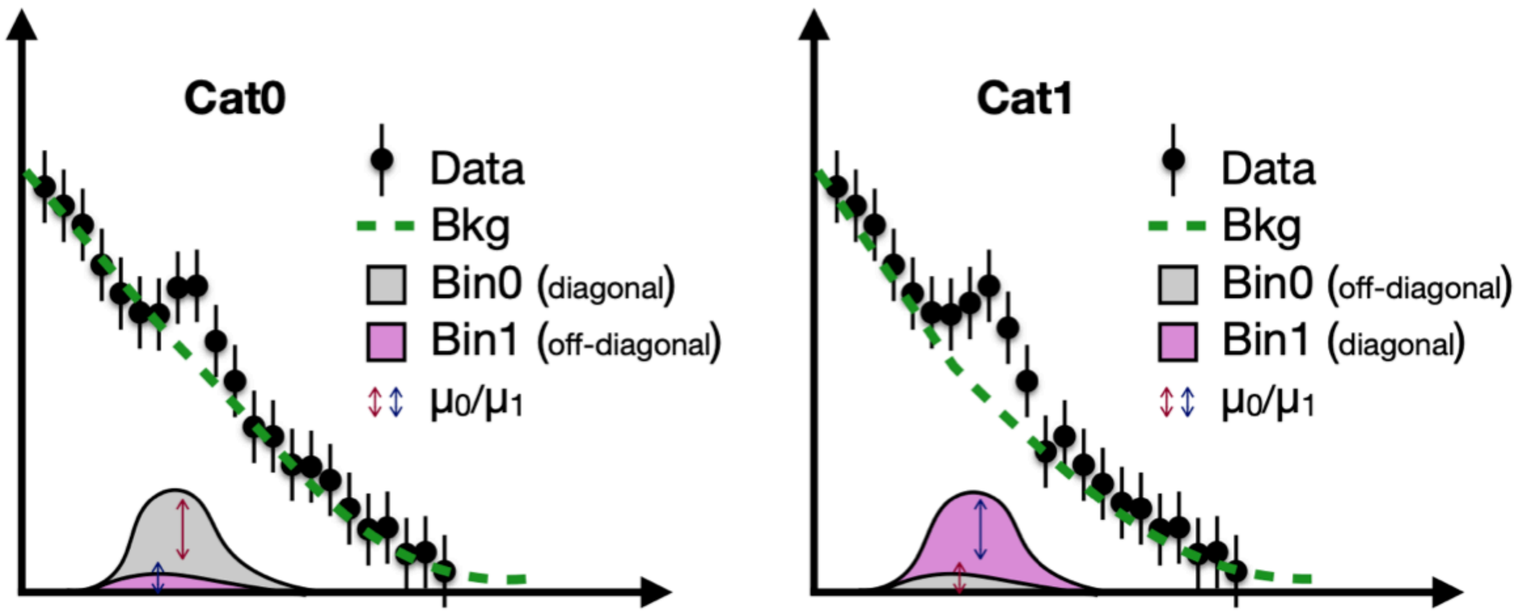
Comparison of likelihood-based unfolding and least-squares based unfolding as implemented in RooUnfold



Likelihood-based unfolding: shape-based

It is also possible to move to **shape-based (unbinned) analysis** to make the most of the data

$$\mathscr{L}(\vec{\mu},\vec{\nu}) = \prod_{i}^{\text{reco}} \text{Poiss}\left(x_{\text{reco},i} \middle| \sum_{j}^{\text{true}} \mu_j \cdot x_{\text{true},i}(\vec{\nu}) \cdot \text{R}_{ij}(\vec{\nu}) \cdot f_{\text{sig}}(\vec{\nu}) + b_i(\vec{\nu}) \cdot f_{\text{bkg}}(\vec{\nu})\right) \prod_{k} p_k(\tilde{\nu}_k \,|\, \nu_k)$$



fit simultaneously in cat0/cat1 to get the
Bin strength modifiers $\mu=(\mu_0,\mu_1)$

Likelihood-based unfolding: shape-based

It is also possible to move to **shape-based (unbinned) analysis** to make the most of the data

$$\mathscr{L}(\vec{\mu},\vec{\nu}) = \prod_{i}^{\text{reco}} \text{Poiss}\left(x_{\text{reco},i} \middle| \sum_{j}^{\text{true}} \mu_j \cdot x_{\text{true},i}(\vec{\nu}) \cdot \text{R}_{ij}(\vec{\nu}) \cdot f_{\text{sig}}(\vec{\nu}) + b_i(\vec{\nu}) \cdot f_{\text{bkg}}(\vec{\nu})\right) \prod_{k} p_k(\tilde{\nu}_k \,|\, \nu_k)$$

```
imax 1
jmax 1
kmax 2
# ---------------
shapes data_obs   bin1 parametric-analysis-datacard-input.root w:
  ↪ data_obs
shapes signal     bin1 parametric-analysis-datacard-input.root w:sig
shapes background bin1 parametric-analysis-datacard-input.root w:bkg
# ---------------
bin                bin1
observation        567
# ---------------
bin                bin1    bin1
process            signal background
process            0       1
rate               10      1
# ---------------
lumi         lnN   1.1     -
sigma        param 1.0     0.1
alpha        flatParam
bkg_norm     flatParam
```

Import any arbitrary binned/unbinned RooFit pdfs

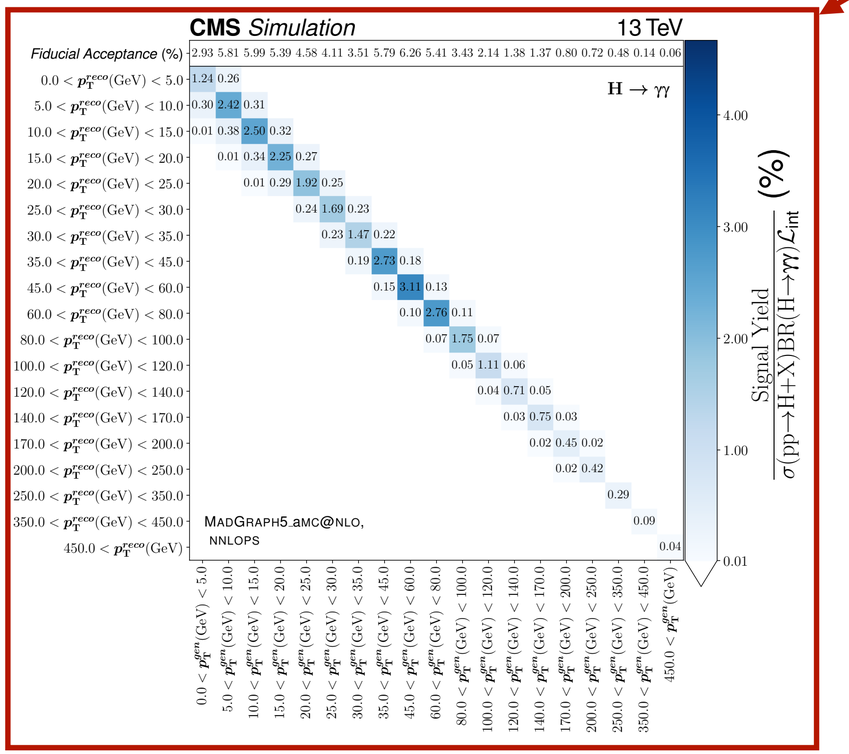
Links to the ROOT files with the corresponding RooAbsPdf/RooDataSet

A real-case scenario: H–>γγ

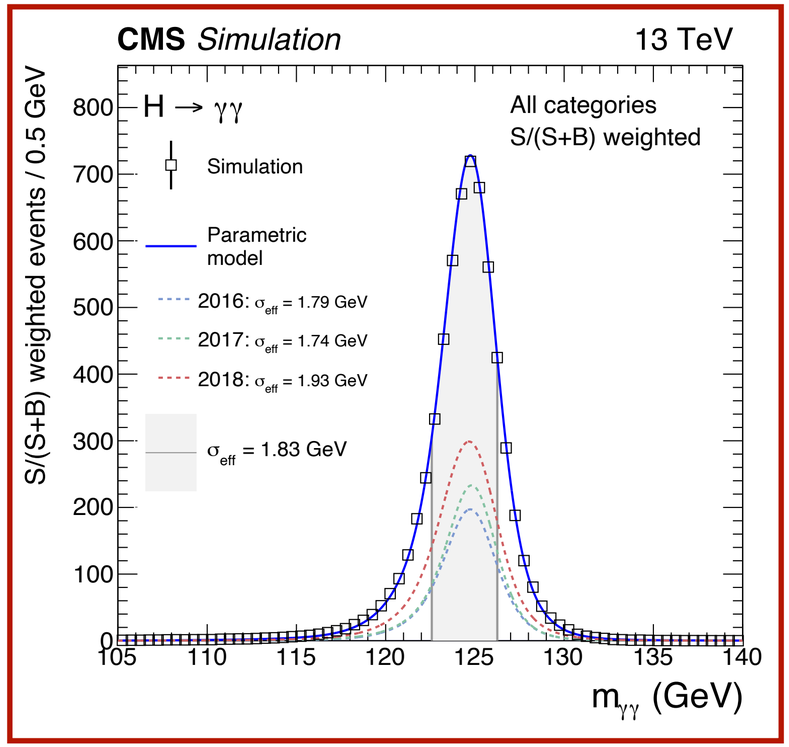
An example of **shape-based likelihood unfolding**: XS measurement in the H–>γγ channel

$$\mathscr{L}(\vec{\mu},\vec{\nu}) = \prod_{i}^{\text{reco}} \text{Poiss}\left(x_{\text{reco},i} \,\middle|\, \sum_{j}^{\text{true}} \mu_j \cdot \boxed{x_{\text{true},i}(\vec{\nu})} \cdot \boxed{\text{R}_{ij}(\vec{\nu})} \cdot \boxed{f_{\text{sig}}(\vec{\nu})} + b_i(\vec{\nu}) \cdot \boxed{f_{\text{bkg}}(\vec{\nu})}\right) \prod_{k} p_k(\tilde{\nu}_k \,|\, \nu_k)$$

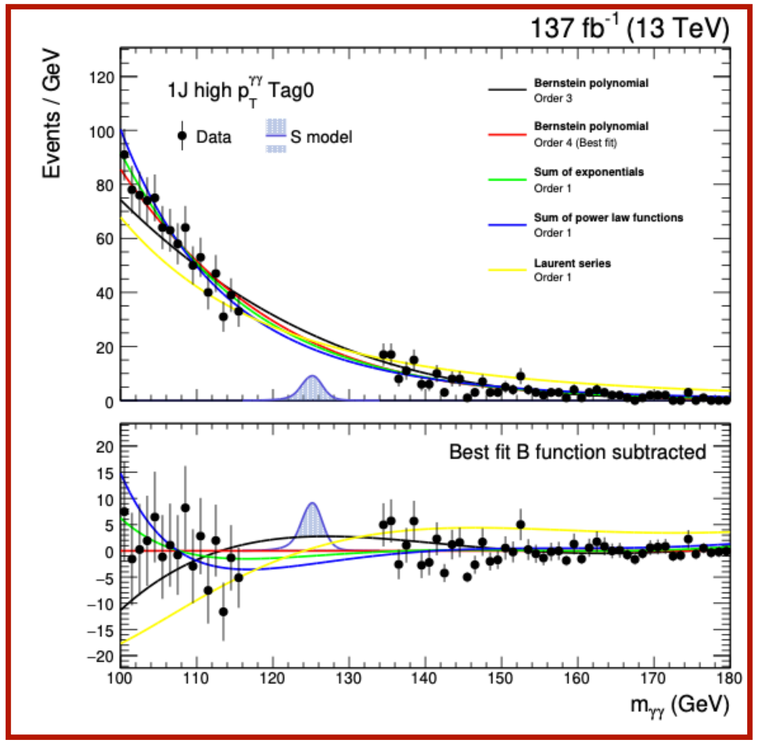
$$\boxed{\sigma \cdot \text{BR} \cdot \mathscr{A} \cdot \text{L}}$$



10.1007/JHEP07(2023)091



10.1007/JHEP07(2021)027

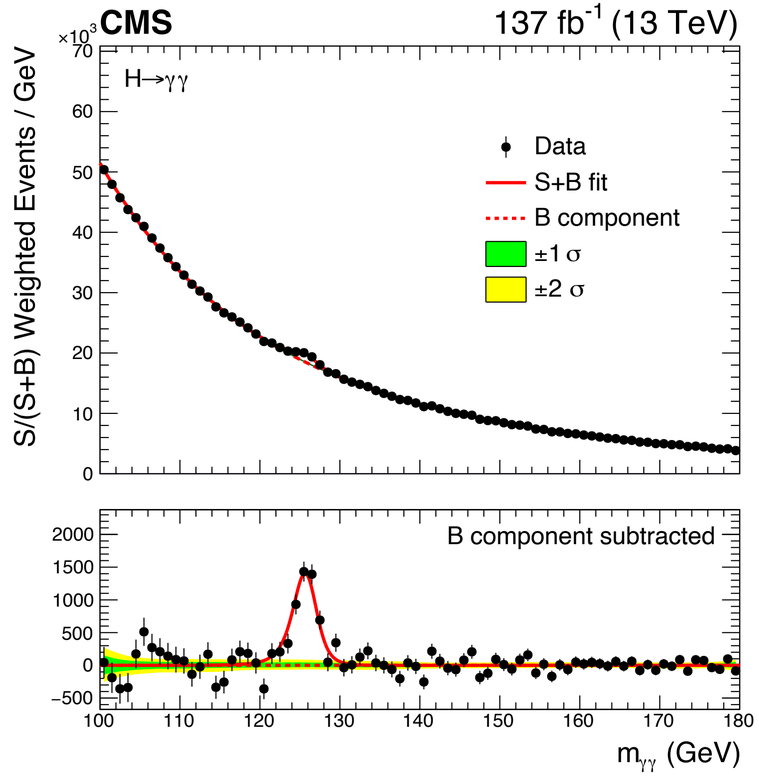


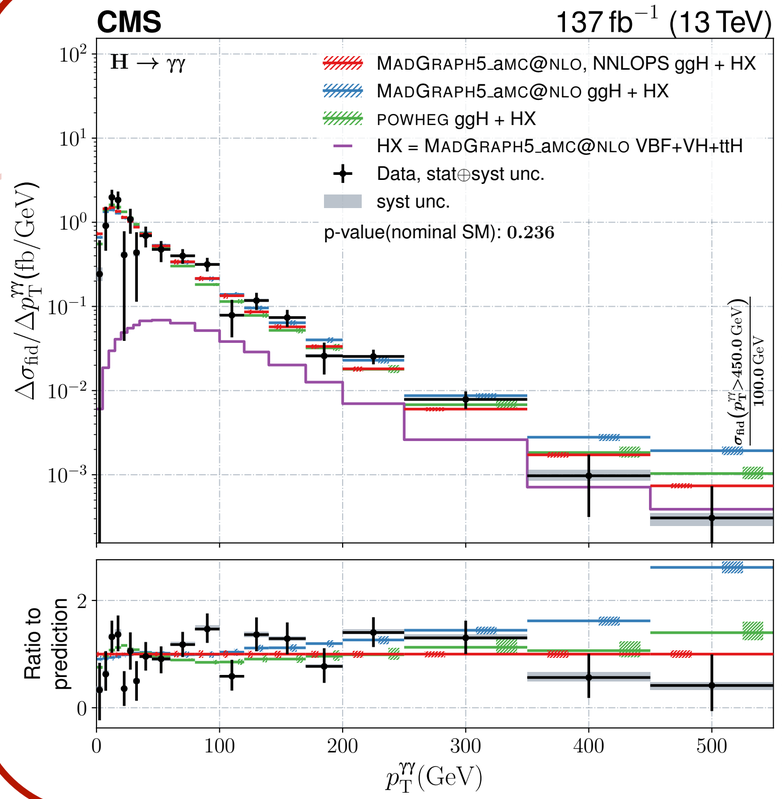
10.25560/93475

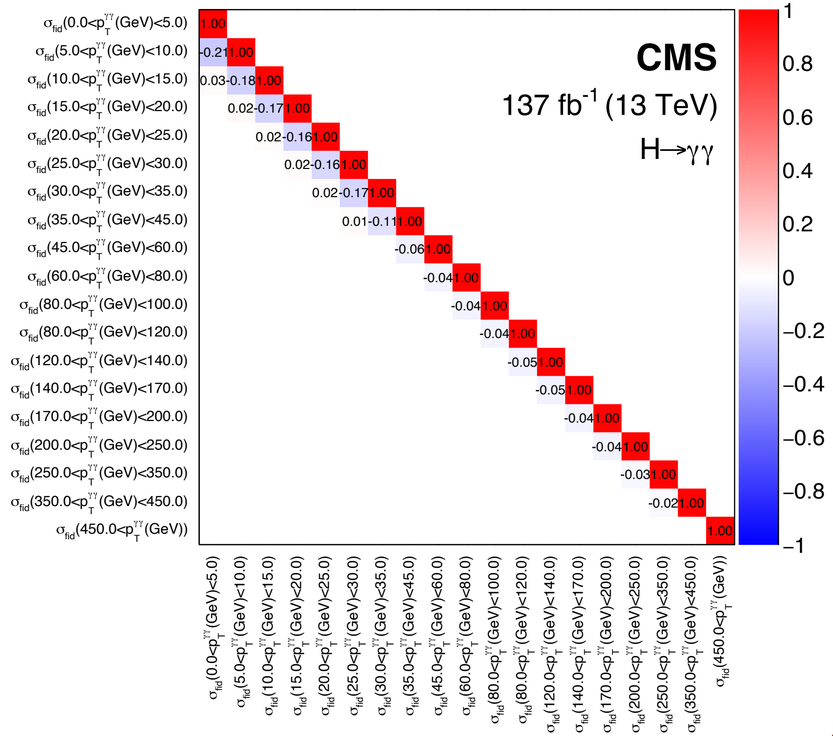
A real-case scenario: H–>γγ

10.1007/JHEP07(2023)091

An example of **shape-based likelihood unfolding**: XS measurement in the H–>γγ channel







Regularisation

The simplest way to introduce regularisation in the likelihood-based approach is to apply a **penalty term** in the likelihood function (so-called *Tikhonov regularisation*)

$$\mathscr{L}(\vec{\mu},\vec{\nu}) = \prod_{i}^{\text{reco}} \text{Poiss}\left(x_{\text{reco},i} \middle| \sum_{j}^{\text{true}} \mu_j \cdot x_{\text{true},i}(\vec{\nu}) \cdot \text{R}_{ij}(\vec{\nu}) \cdot f_{\text{sig}}(\vec{\nu}) + b_i(\vec{\nu}) \cdot f_{\text{bkg}}(\vec{\nu})\right) \prod_{k} p_k(\tilde{\nu}_k \,|\, \nu_k) \cdot \mathscr{K}(\vec{\mu})$$

In `Combine`, it is possible to implement both the **TUnfold** and **SVD** variant of the Tikhonov regularisation

Regularisation: H–>WW

Measurement of the inclusive and differential Higgs boson production cross sections in the leptonic WW decay mode at $\sqrt{s} = 13\,\text{TeV}$

$$\mathscr{K}(\vec{\mu}) = \prod_{i=2}^{N-1} \exp\left(\frac{-[(\mu_{i+1} - \mu_i) - (\mu_i - \mu_{i-1})]^2}{2\delta^2} \right)$$

- The regularisation term penalises the curvature (high-frequency fluctuations)
- Regularisation strength δ: optimised in order to minimise the mean of the global correlation coefficient on an Asimov dataset

The regularisation term can be added to the likelihood with **a simple line in the systematics section of the datacard**

<name> constr <formula> <args> <delta> $\longrightarrow$ $\exp\left[-\frac{1}{2}\left(\frac{< \text{formula} >}{< \text{delta} >}\right)\right]$

Regularisation: H–>WW

Measurement of the inclusive and differential Higgs boson production cross sections in the leptonic WW decay mode at $\sqrt{s} = 13\,\text{TeV}$

$$\mathscr{K}(\vec{\mu}) = \prod_{i=2}^{N-1} \exp\left(\frac{-[(\mu_{i+1} - \mu_i) - (\mu_i - \mu_{i-1})]^2}{2\delta^2} \right)$$

- The regularisation term penalises the curvature (high-frequency fluctuations)
- Regularisation strength δ: optimised in order to minimise the mean of the global correlation coefficient on an Asimov dataset

The regularisation term can be added to the likelihood with **a simple line in the systematics section of the datacard**

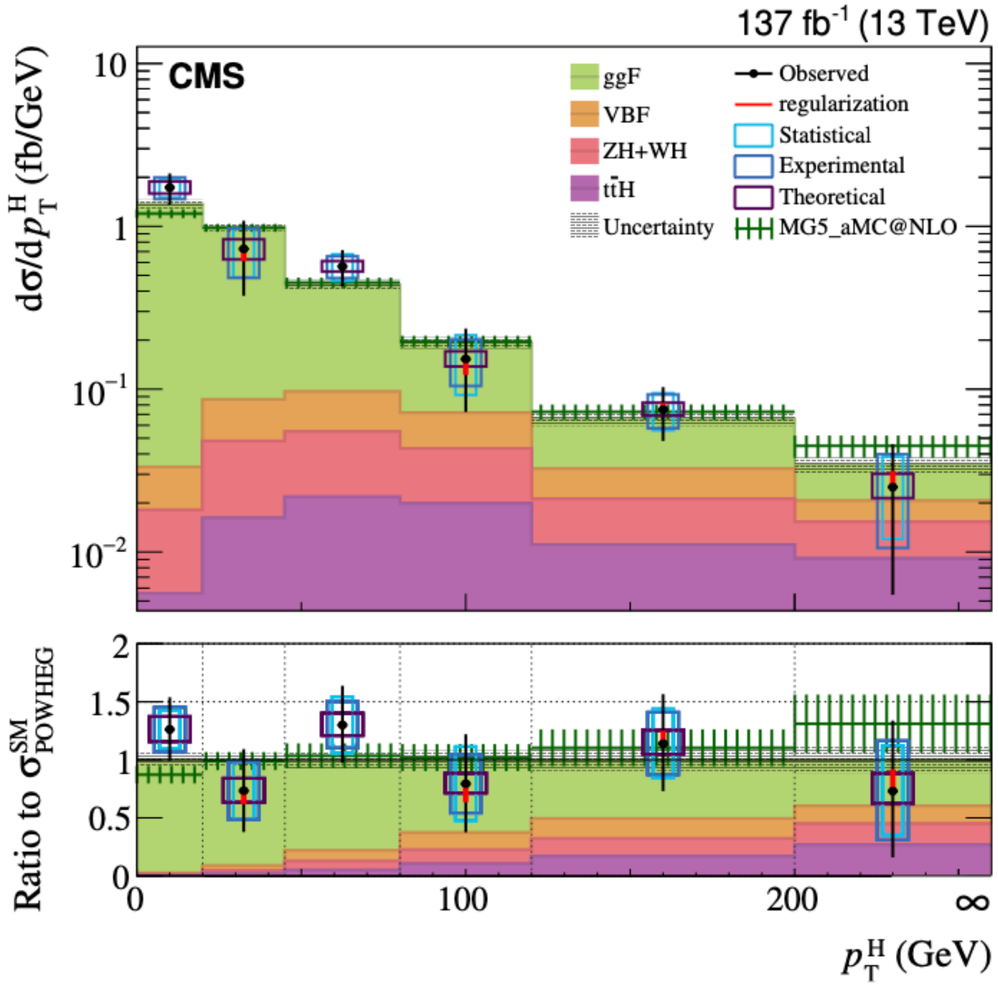
```
constr0  constr  @0-2*@1+@2  r_0,r_1,r_2  2.50
constr1  constr  @0-2*@1+@2  r_1,r_2,r_3  2.50
constr2  constr  @0-2*@1+@2  r_2,r_3,r_4  2.50
constr3  constr  @0-2*@1+@2  r_3,r_4,r_5  2.50
```

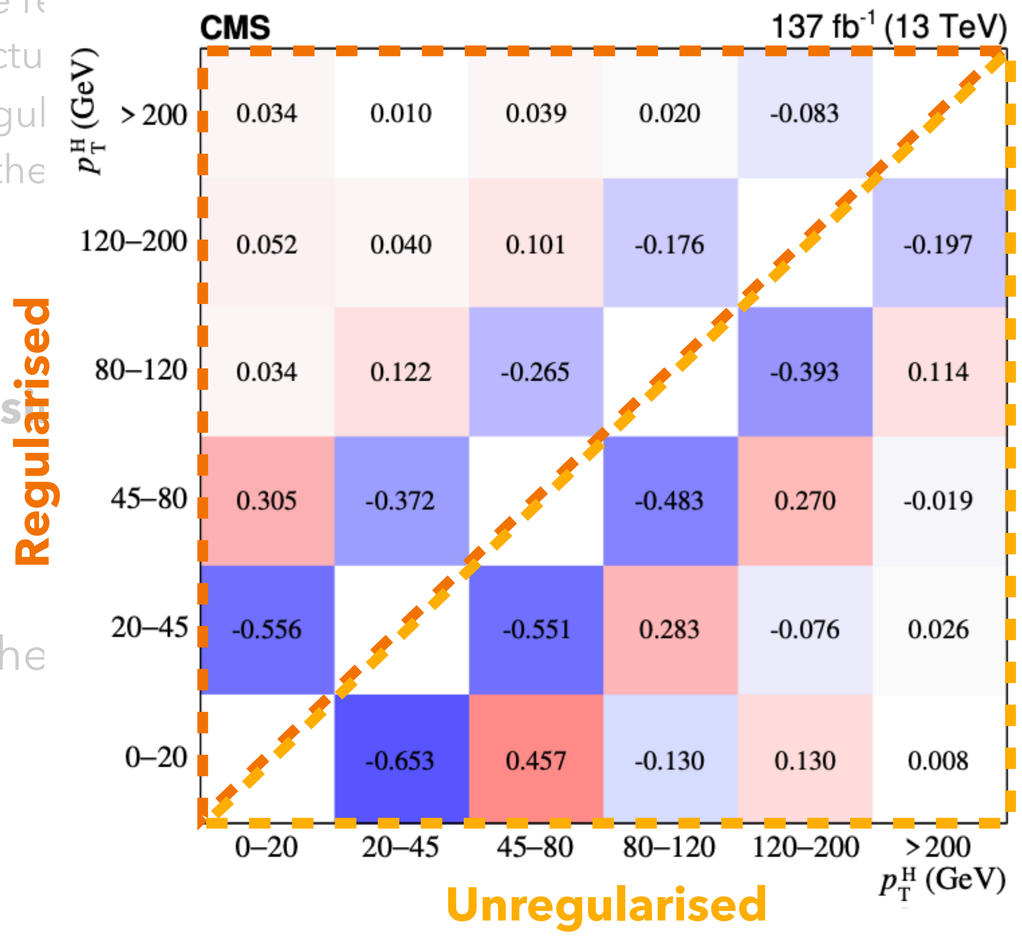
The analysis measures the XS in 6 p_T^H bins –> 4 penalty terms to add to the likelihood

Regularisation strength optimised to be 2.50

Regularisation: H->WW

Measurement of the inclusive and differential Higgs boson production cross sections in the leptonic WW decay mode at $\sqrt{s} = 13$ TeV





TUnfold vs Combine

	TUnfold	Combine
Method	Least-square minimisation	Maximum likelihood
Speed	Linear algebra –> very fast	Numerical minimisation with Minuit and complex fit with nuisance parameters –> much slower
Number of unfolded bins	Up to very large numbers	Complexity of the fit increases with the number of bins
Regularisation	Possible	Possible
Background	Simple subtraction	Can do simultaneous binwise signal + background fit
Systematic uncertainties	Vary externally and repeat unfolding	Simultaneous fit of nuisance parameters and profiling them
Ideal application	High statistics, low background, precision analyses, e.g., inclusive jets, ttbar production	Anything, except cases with very large numbers of unfolded events

Credits: O. Behnke, P. Gras, G. Kasieczka

Conclusions

After a decade of development, the `Combine` package has become the **main tool used for statistical analysis of data by the CMS Collaboration**

The statistical model is constructed from a text file provided by the user and a configurable physics model that encodes the parameters of interest and the nuisance parameters that model systematic uncertainties

The `Combine` package can perform a variety of statistical procedures, including the possibility of performing **likelihood-based unfolding**

Backup

Nuisance parameters

Table 1: Available uncertainty types for counting experiments. The second and third columns indicate the entries for the datacard required to specify the type, and the relative effect on the yield of each process in each channel. The fourth and fifth columns indicate the resulting multiplicative factor by which COMBINE scales the normalization of the relevant process in the specified channel, and the term $p(y;\nu)$ that is included in Eq. (1). Finally, the last column indicates the default values of ν and y. Where relevant, the value of $\kappa - 1$ can be interpreted as the relative uncertainty in the process normalization in a given channel.

Uncertainty type	Directive	Inputs	Multiplicative factor, $f(\nu)$	$p(y;\nu)$	Default values
Log-normal	`lnN`	`kappa`	κ^{ν}	$\mathcal{N}(y;\nu,1)$	$\nu = y = 0$
Asymmetric log-normal	`lnN`	`kappaDown`, `kappaUp`	$\left(\kappa^{\mathrm{Down}}\right)^{-\nu}$ if $\nu < -0.5$, $\left(\kappa^{\mathrm{Up}}\right)^{\nu}$ if $\nu > 0.5$, $e^{\nu K\left(\kappa^{\mathrm{Down}},\kappa^{\mathrm{Up}},\nu\right)}$ otherwise.*	$\mathcal{N}(y;\nu,1)$	$\nu = y = 0$
Log-uniform	`lnU`	`kappa`	κ^{ν}	$\mathcal{U}\left(y, 1/\kappa, \kappa\right)$	$\nu = y = \frac{1}{2}\left(\kappa + 1/\kappa\right)$
Gamma	`gmN`	`N`, `alpha`†	ν/N	$\mathcal{P}(y;\nu)$	$\nu = N+1, y = N$‡

*$K\left(\kappa^{\mathrm{Down}},\kappa^{\mathrm{Up}},\nu\right) = \frac{1}{8}\left[4\ln\left(\kappa^{\mathrm{Up}}/\kappa^{\mathrm{Down}}\right) + \ln\left(\kappa^{\mathrm{Up}}\kappa^{\mathrm{Down}}\right)\left(48\nu^5 - 40\nu^3 + 15\nu\right)\right]$ ensures that the multiplicative factor and its first and second derivatives are continuous for all values of ν, and reduces to a log-normal for $\kappa^{\mathrm{Down}} = 1/\kappa^{\mathrm{Up}}$.

†The rate value for the affected process must be equal to $N\alpha$.

‡The default value for the nuisance parameter is set to the mean of a gamma distribution with parameters $\kappa = N+1$, $\lambda = 1$, as defined in Ref. [20].

Global correlation coefficients

V represents the covariance matrix

$$\rho_i = \sqrt{1 - \frac{1}{(\mathbf{V_{xx}}^{-1})_{ii}(\mathbf{V_{xx}})_{ii}}}.$$

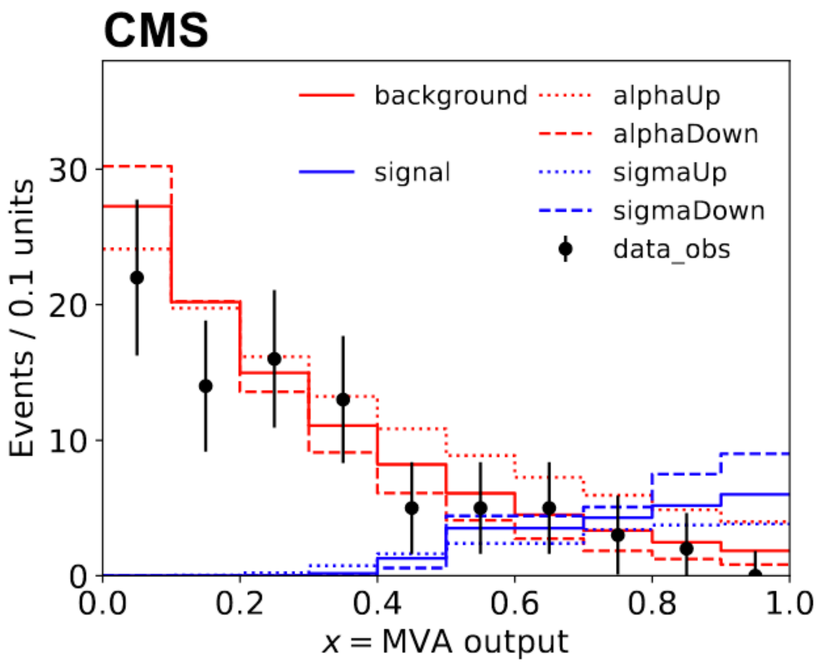
Datacard for a template-based analysis

```
imax 1  Number of bins/channel
jmax 1  Number of processes
kmax 4  Number of nuisance parameters
# ---------------
shapes * * template-analysis-datacard-input.root $PROCESS
  ↪ $PROCESS_$SYSTEMATIC
# ---------------
bin          ch1   Unique channel label
observation  85    Number of observed events in a channel
# -------------------------------
bin              ch1        ch1
process          signal     background   Process label
process          0          1            Process ID (<=0 for signal)
rate             24         100          Expected number of events
# --------------------------------
lumi     lnN     1.1        1.0
bgnorm   lnN     -          1.3
alpha    shape   -          1
sigma    shape   0.5        -
```

Links to the histograms saved in a root file

Systematic uncertainties

Name Type Effect on process



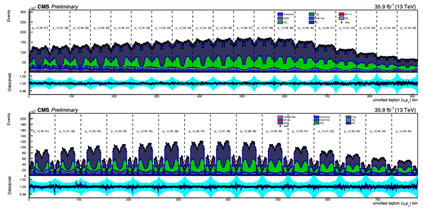
Root file containing the nominal histograms and two additional histograms per process related to the ±1σ shift

Other tools

- Several other tools exist which offer various advantages over the ROOT/RooFit base of combine:

combine-tensorflow

- Originally for W helicity analysis
- Combine binned model implemented
- Auto-grad+hessian, parallelisation
- + Alternative minimizers

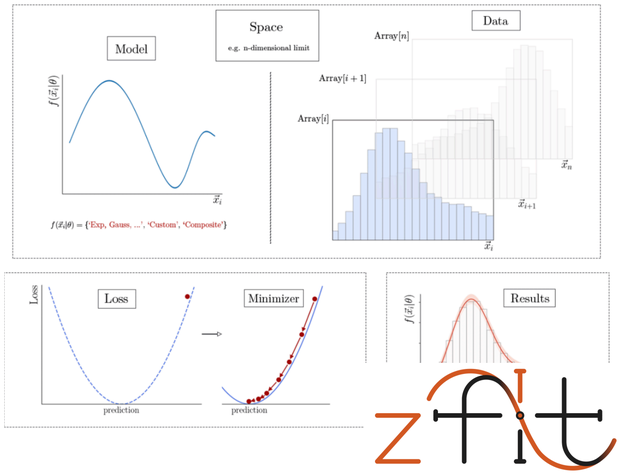


	Likelihood	Likelihood+Gradient	Hessian
Combine, TR1950X 1 Thread	10ms	830ms	-
TF, TR1950X 1 Thread	70ms	430ms	165s
TF, TR1950X 32 Thread	20ms	71ms	32s
TF, 2x Xeon Silver 4110 32 Thread	17ms	54ms	24s
TF, GTX1080	7ms	13ms	10s
TF, V100	4ms	7ms	8s

Performance > 100x wrt. current combine

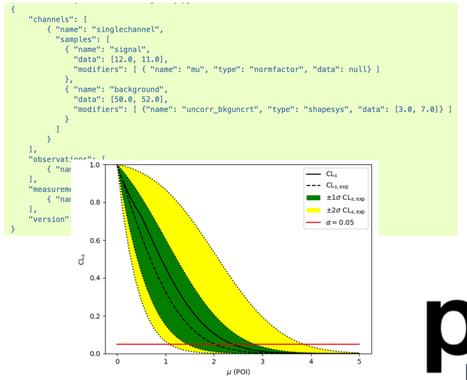
zfit

- Also TF based, supports parametric models



pyhf

- Pure-python implementation of HistFactory (used in ATLAS)
- Support for PyTorch/TF/JAX backends
- Becoming popular for sharing public models



+ several others

- Can lack some of the features needed to cover all our needs
- RooFit development currently quite active (vectorized evaluation, GPU dispatch and auto-grad on the horizon)
- Should continually survey the alternatives, and consider where future efforts from our side are best directed