IMPERIAL

Simplified Template Cross Section measurements of $H \rightarrow \gamma\gamma$ at CMS

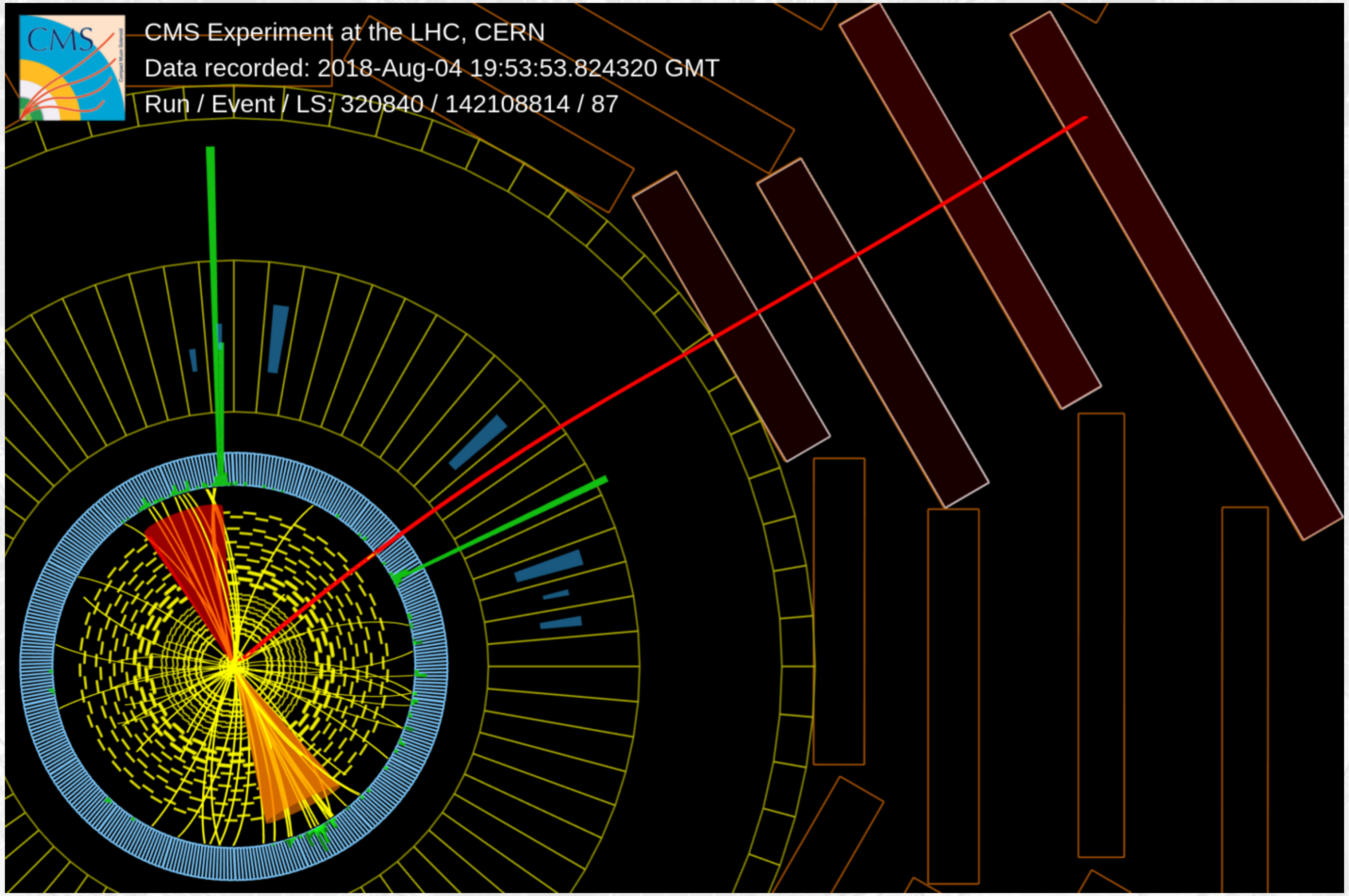
IOP Joint APP and HEPP Conference, Edinburgh

08th April 2026

Tom Runting

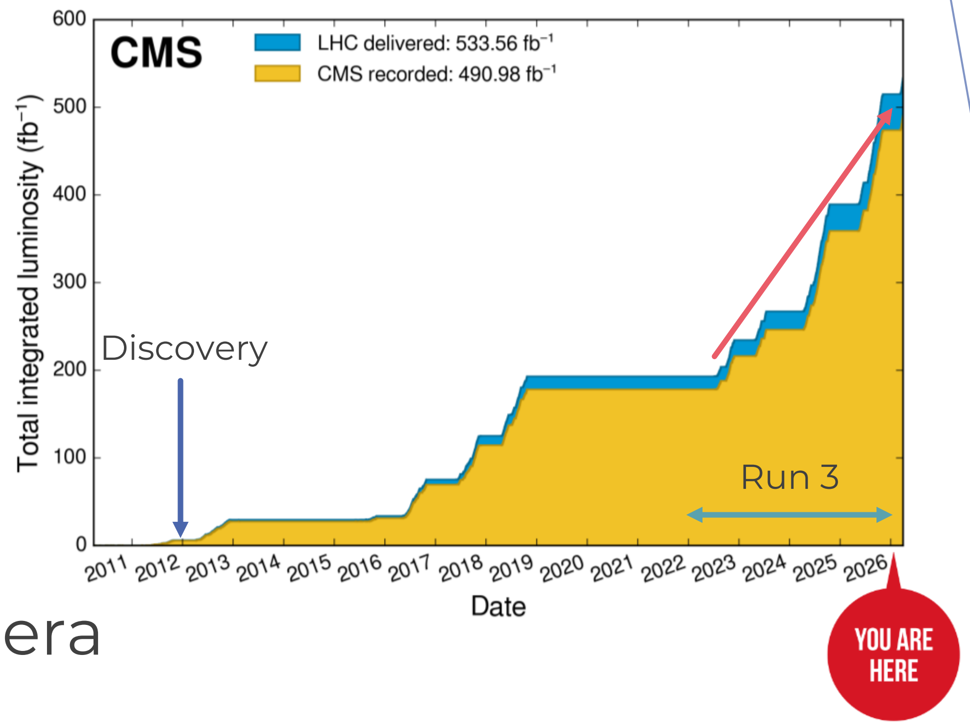
Supervisors: Jonathon Langford, Nicholas Wardle

The Higgs boson at CMS

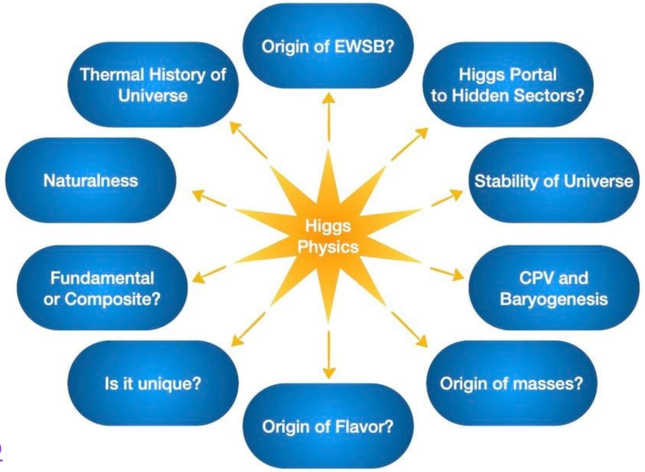


Higgs – from observation to precision

- Approaching 15 years since discovery, with ~25x discovery integrated luminosity already recorded in Run 3



- Accordingly, measurements have moved into the precision era

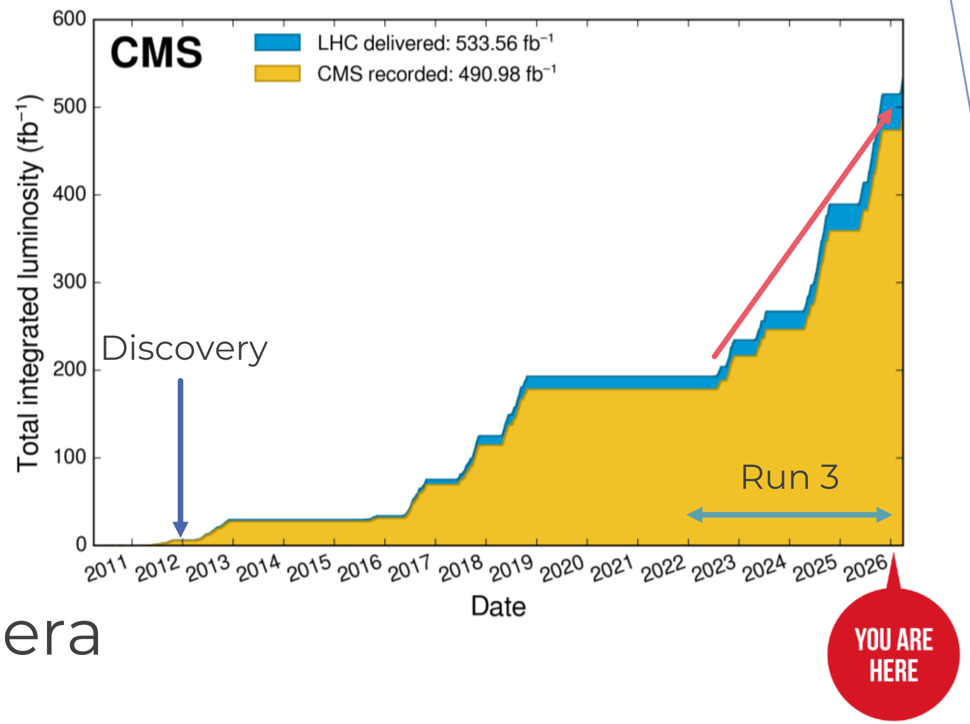


2103.06956

- Several outstanding questions about the Higgs sector remain and *all* require precise measurements

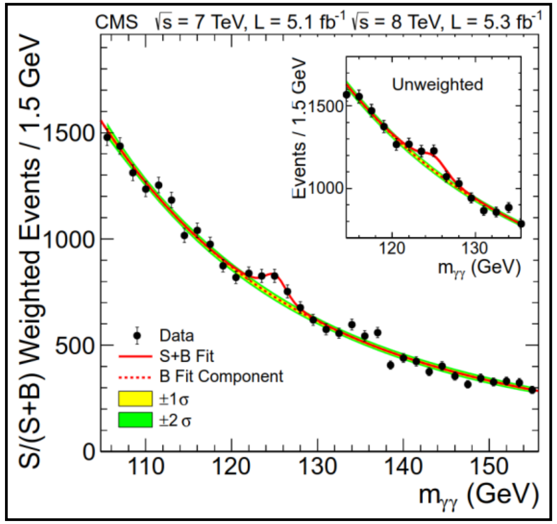
Higgs – from observation to precision

- Approaching 15 years since discovery, with ~25x discovery integrated luminosity already recorded in Run 3

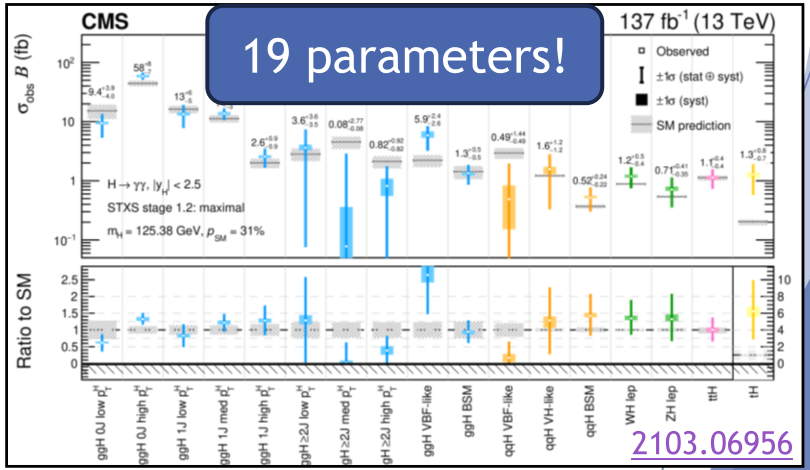


- Accordingly, measurements have moved into the precision era

1 parameter affecting total Higgs yield



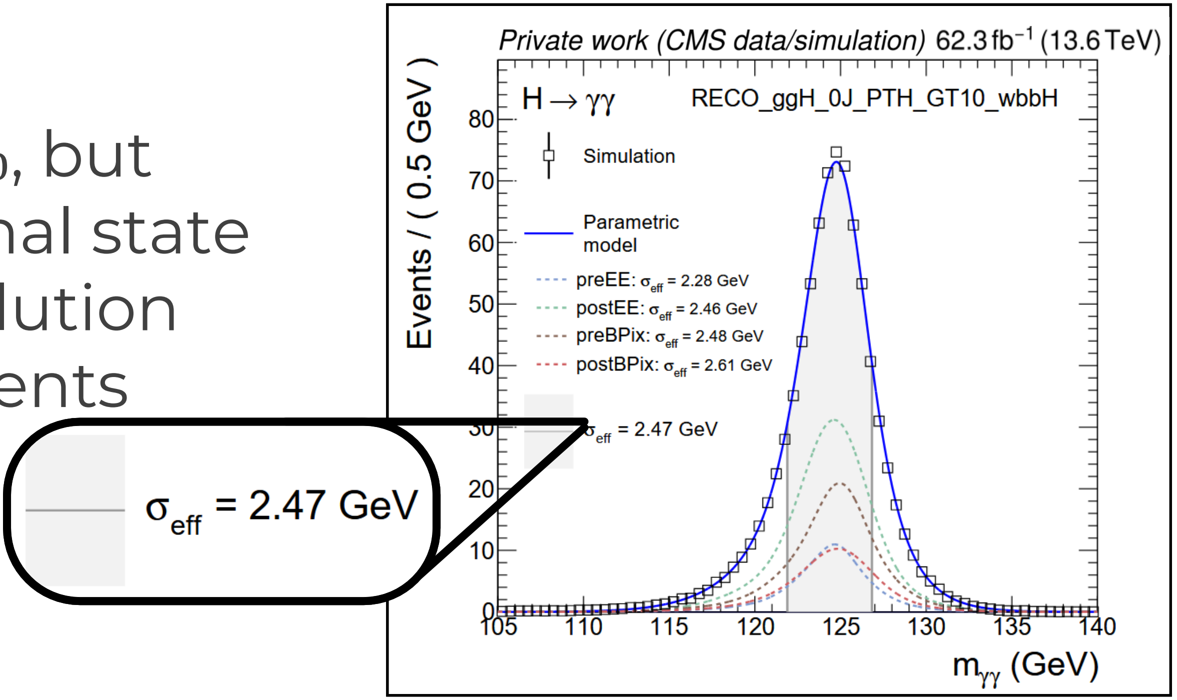
1303.4571

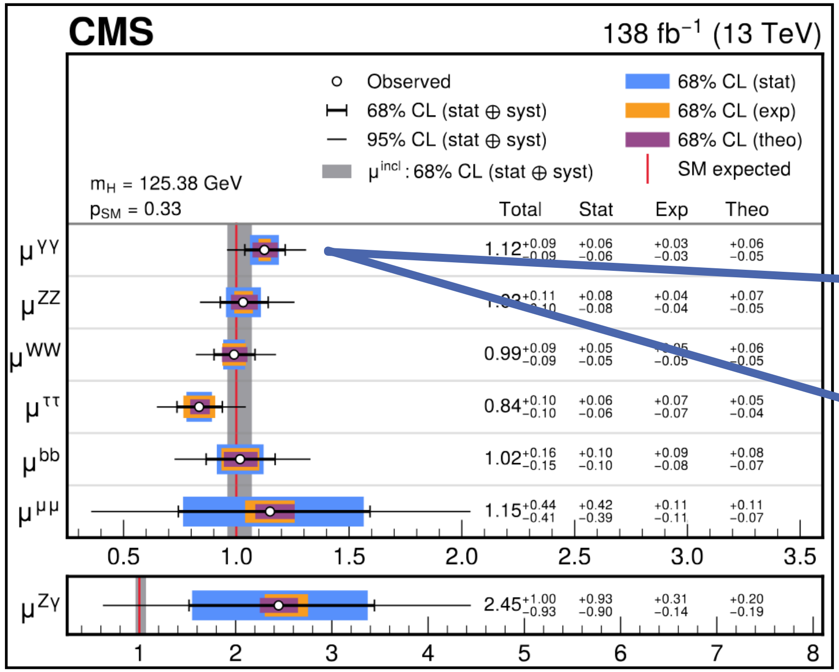


2103.06956

Higgs – why $H \to \gamma\gamma$?

- Small $\mathcal{B}(H \to \gamma\gamma) \approx 0.23\%$, but experimentally clean final state yielding very good resolution and precise measurements



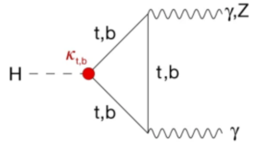


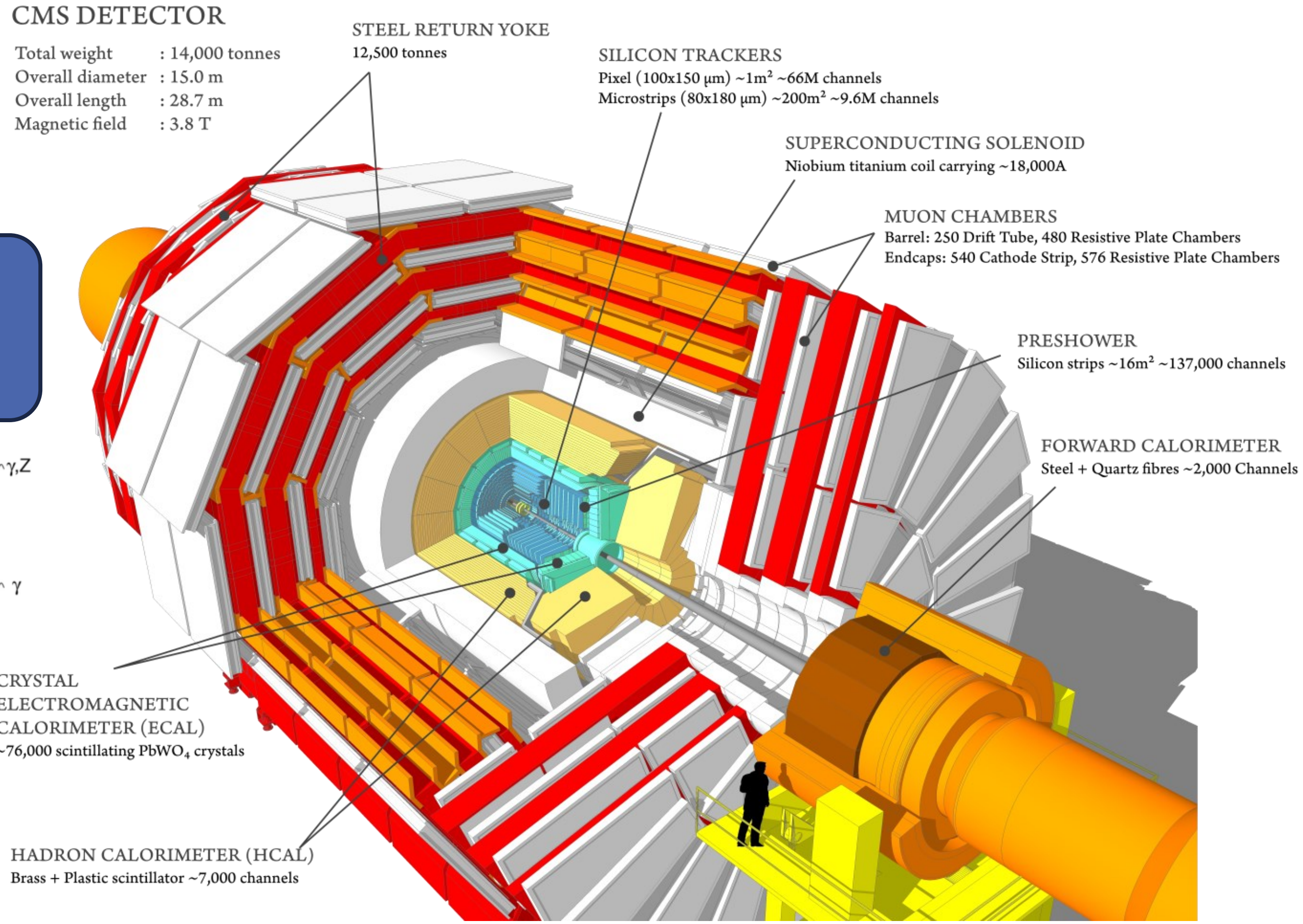
2602.18611

Higgs – how $H \to \gamma\gamma$?

- CMS is a general-purpose detector, and for precision measurements we need every part of it

ECAL: Photons deposit ~all their energy here

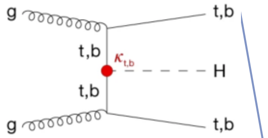




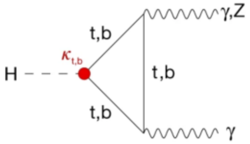
Higgs – how $H \rightarrow \gamma\gamma$?

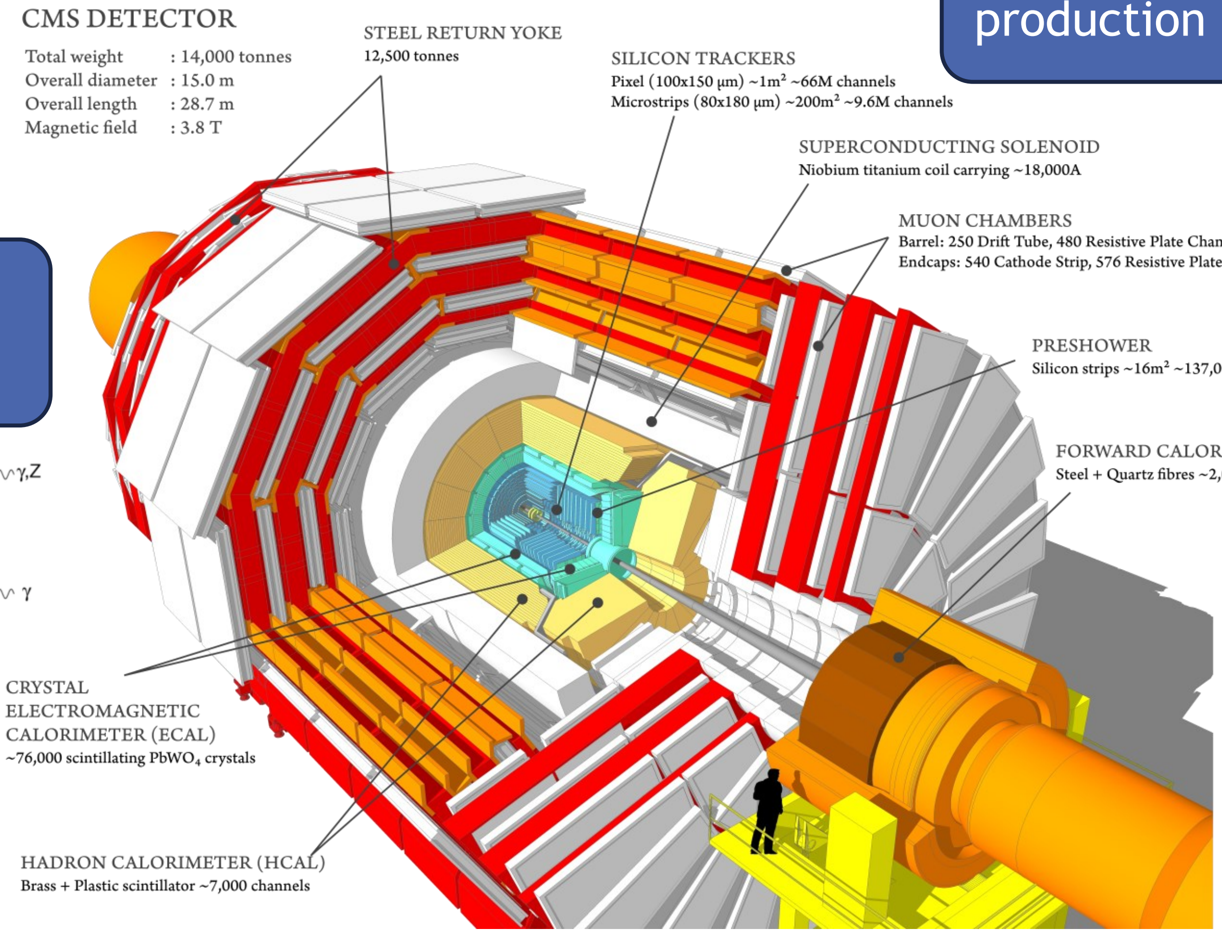
- CMS is a general-purpose detector, and for precision measurements we need every part of it

Tracker: b-jets from rare production modes like $t\bar{t}H$



ECAL: Photons deposit ~all their energy here

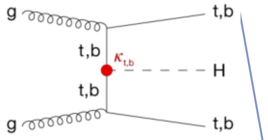




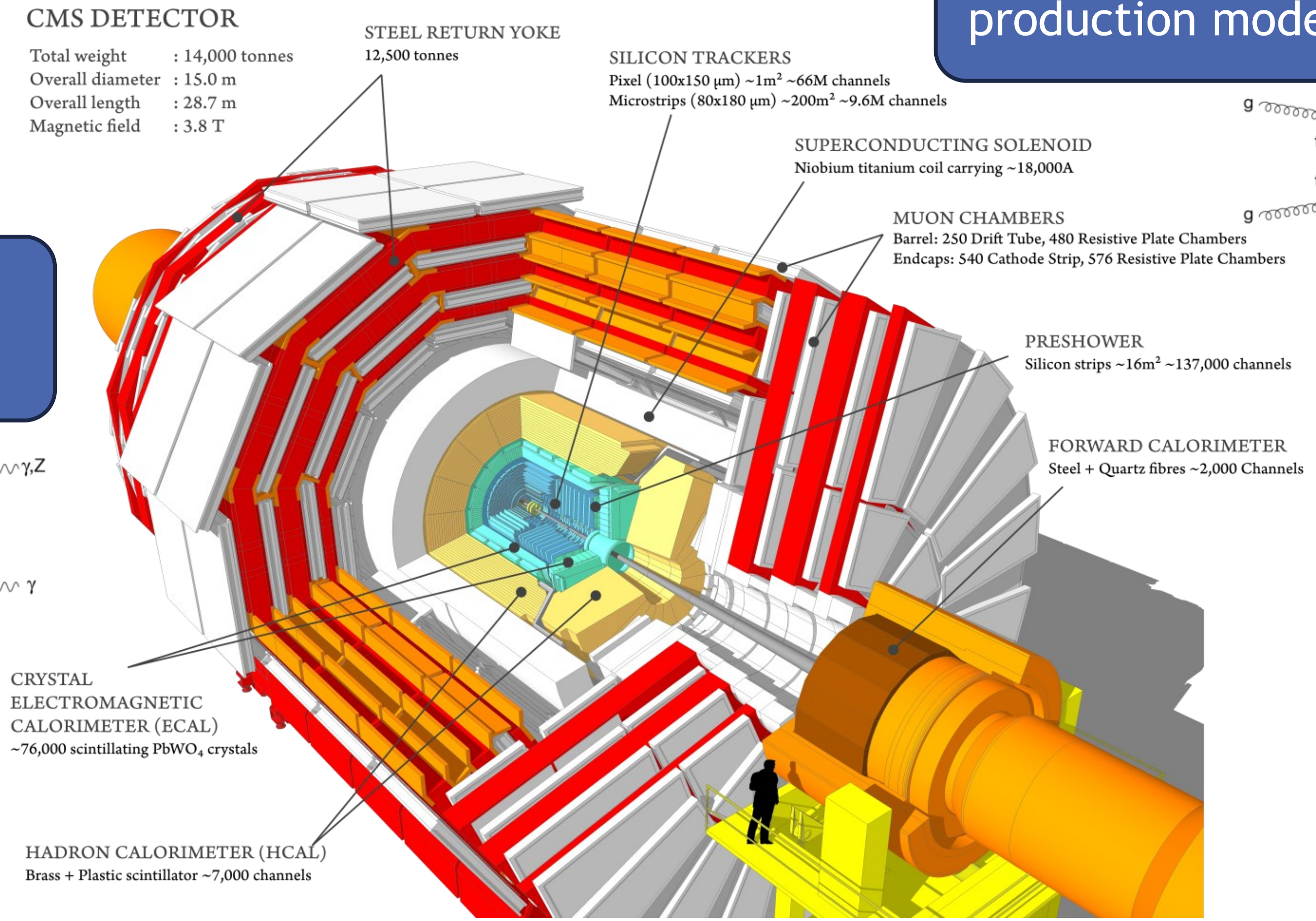
Higgs – how $H \to \gamma\gamma$?

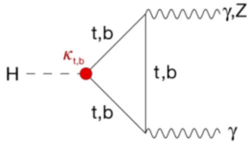
- CMS is a general-purpose detector, and for precision measurements we need every part of it

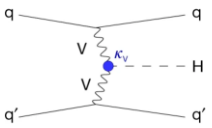
Tracker: b-jets from rare production modes like $t\bar{t}H$



ECAL: Photons deposit ~all their energy here





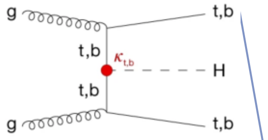


HCAL: Jets from e.g. VBF production

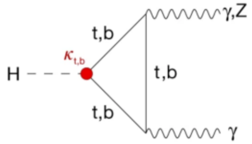
Higgs – how $H \to \gamma\gamma$?

- CMS is a general-purpose detector, and for precision measurements we need every part of it

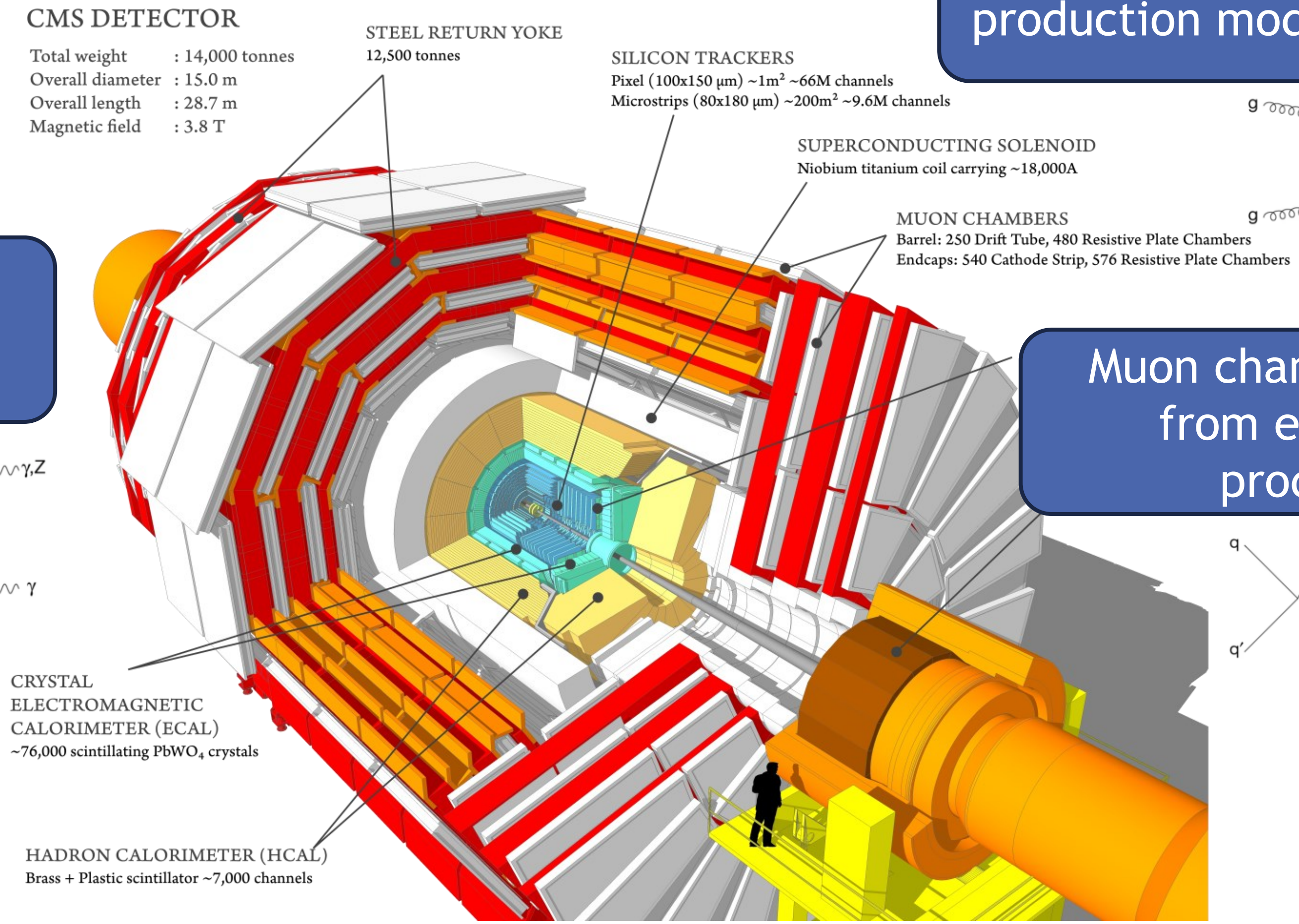
Tracker: b-jets from rare production modes like $t\bar{t}H$

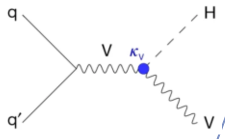


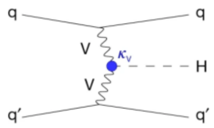
ECAL: Photons deposit ~all their energy here



Muon chambers: Muons from e.g. $V(ll)H$ production



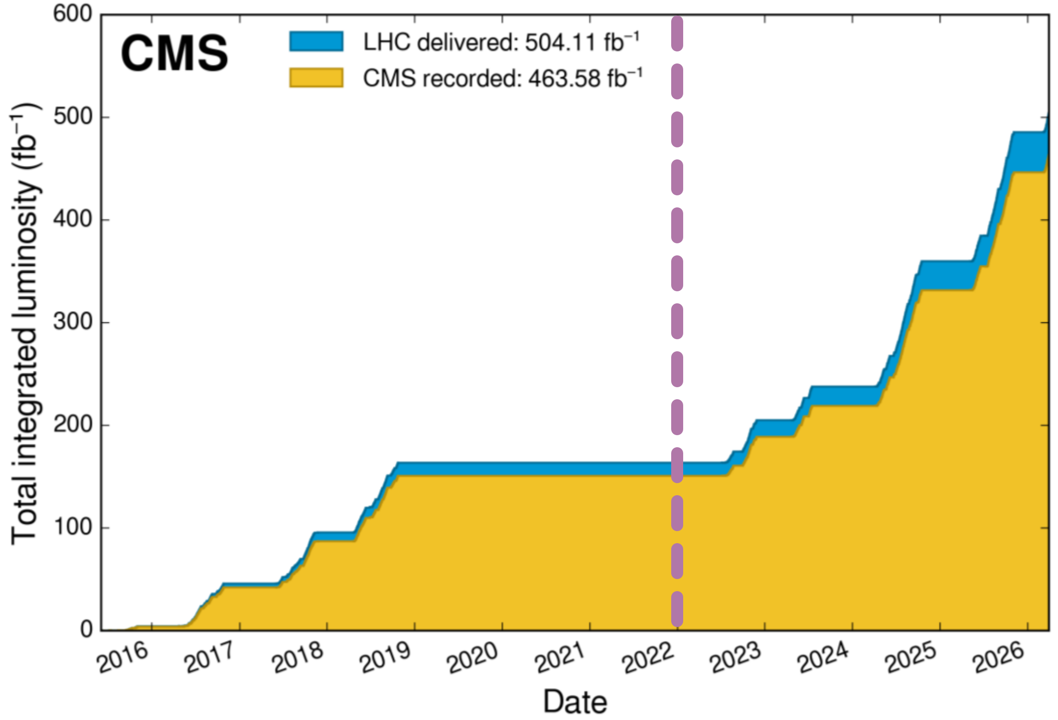




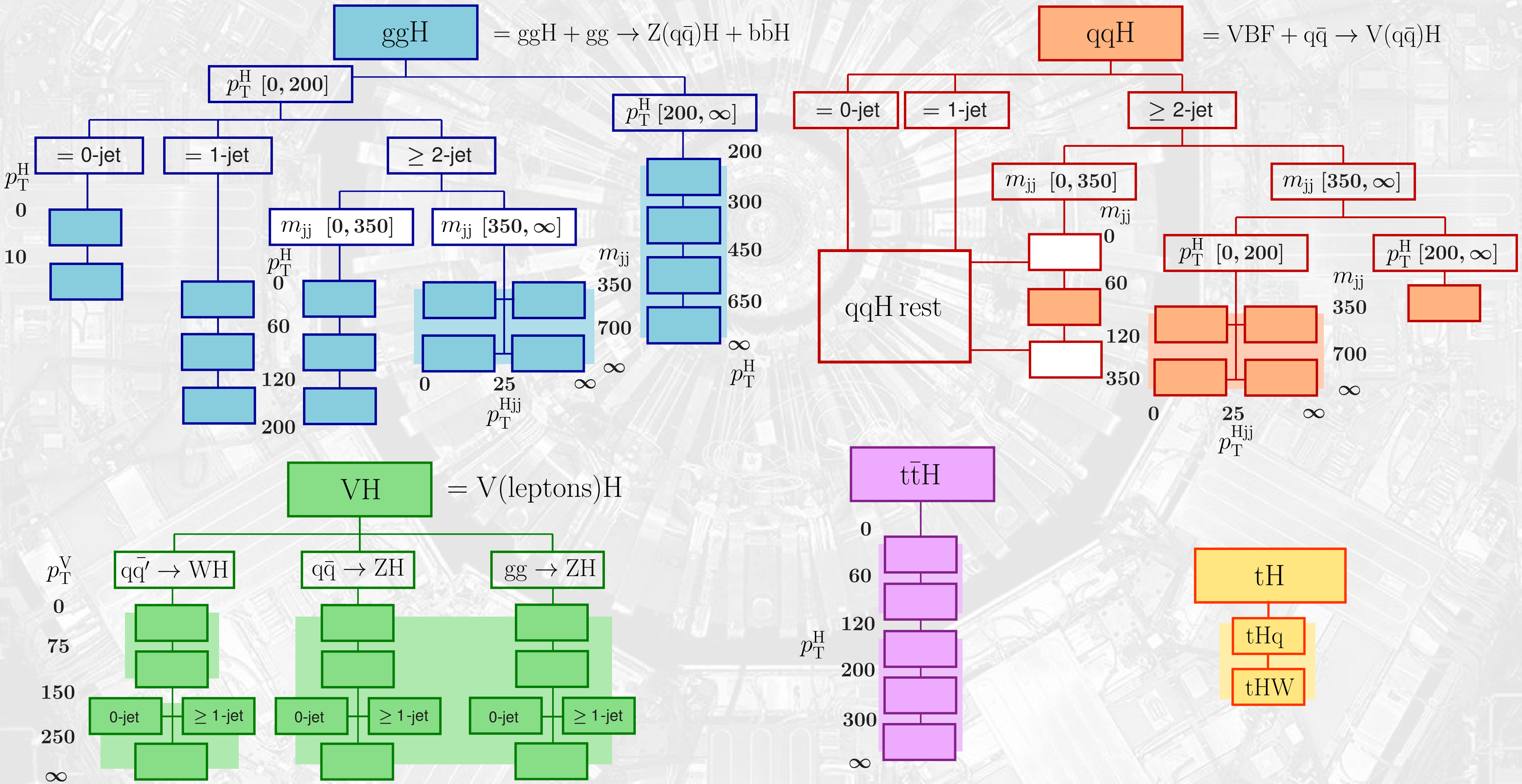
HCAL: Jets from e.g. VBF production

Higgs – how $H \to \gamma\gamma$?

- Since 2022, CMS has collected a dataset with $\mathcal{L} > 300\text{fb}^{-1}$
  - ~15 million events with a Higgs boson (~40k $H \to \gamma\gamma$) to analyse!
- An opportunity to try new analysis techniques, increase granularity and use this wealth of data to better understand the Higgs boson

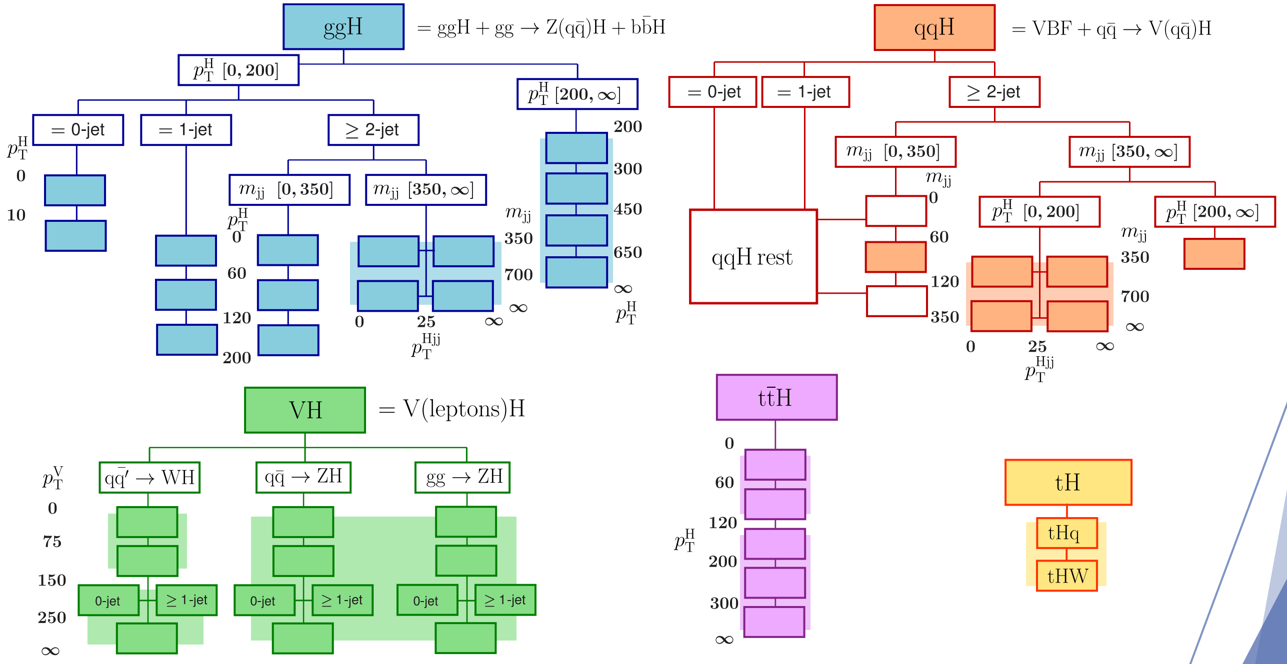


STXS $H \to \gamma\gamma$ in Run 3



Simplified Template Cross Sections

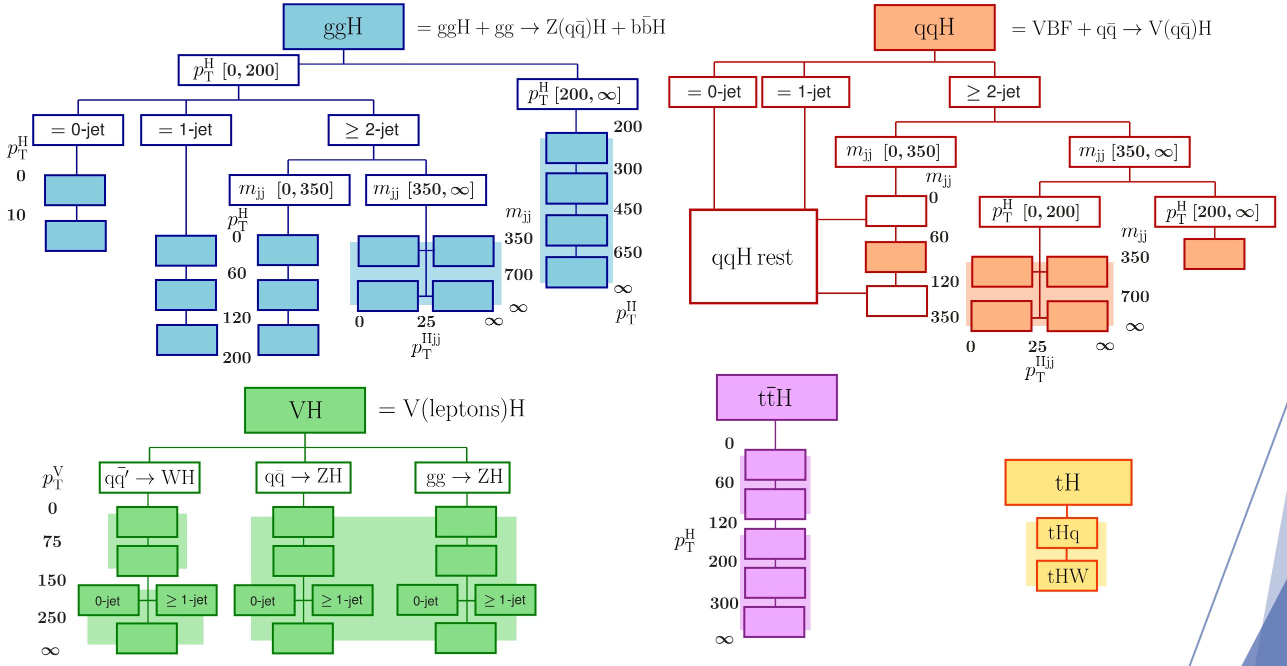
- So, we want to measure the Higgs with better precision and potentially discover new physics
- If the inclusive σ_H agrees with the SM, deviations could still occur in rarer regions of phase space (e.g., high p_T^H)
- Construct a few “particle-level” regions of phase space (split by production mode) and measure the cross-section in each



Simplified Template Cross Sections

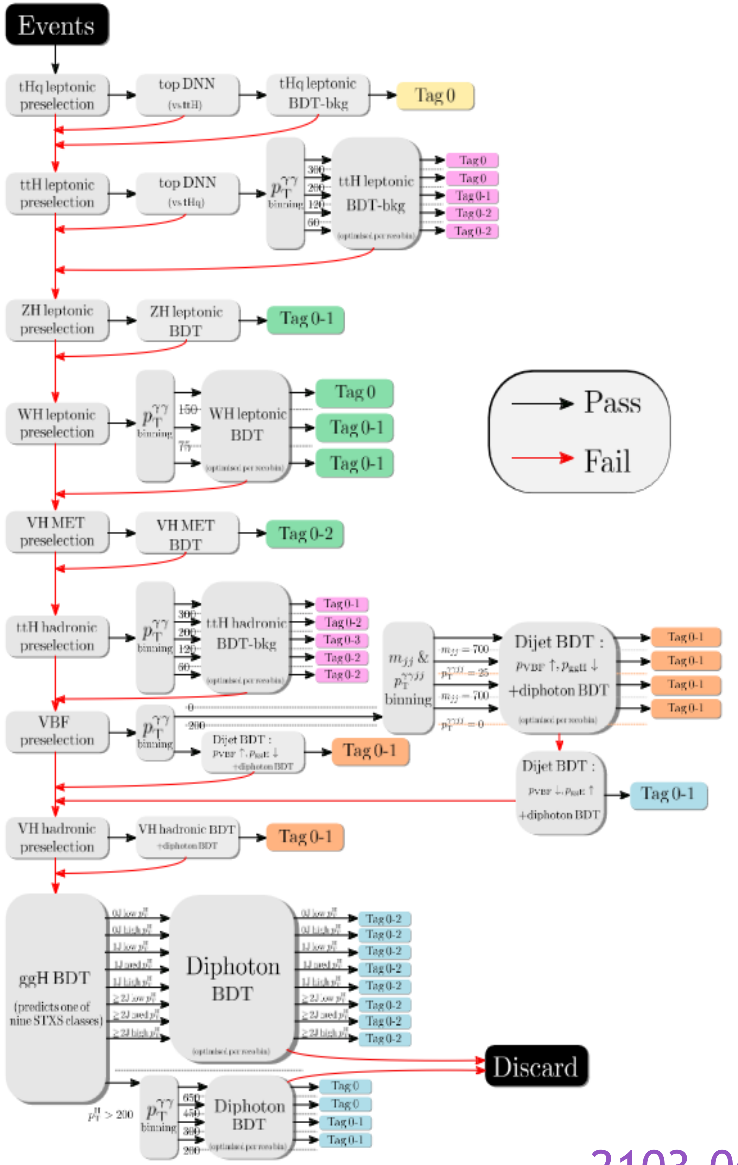
- Use simulation to relate the observed events to these underlying categories
- Complementary to differential measurements

- Unfolding to “particle-level” allows the results to remain long-term useful
- Fixed binning scheme useful for combinations!



Analysis strategy

Run 2



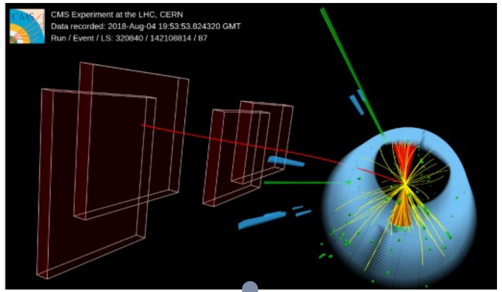
[2103.06956](2103.06956)

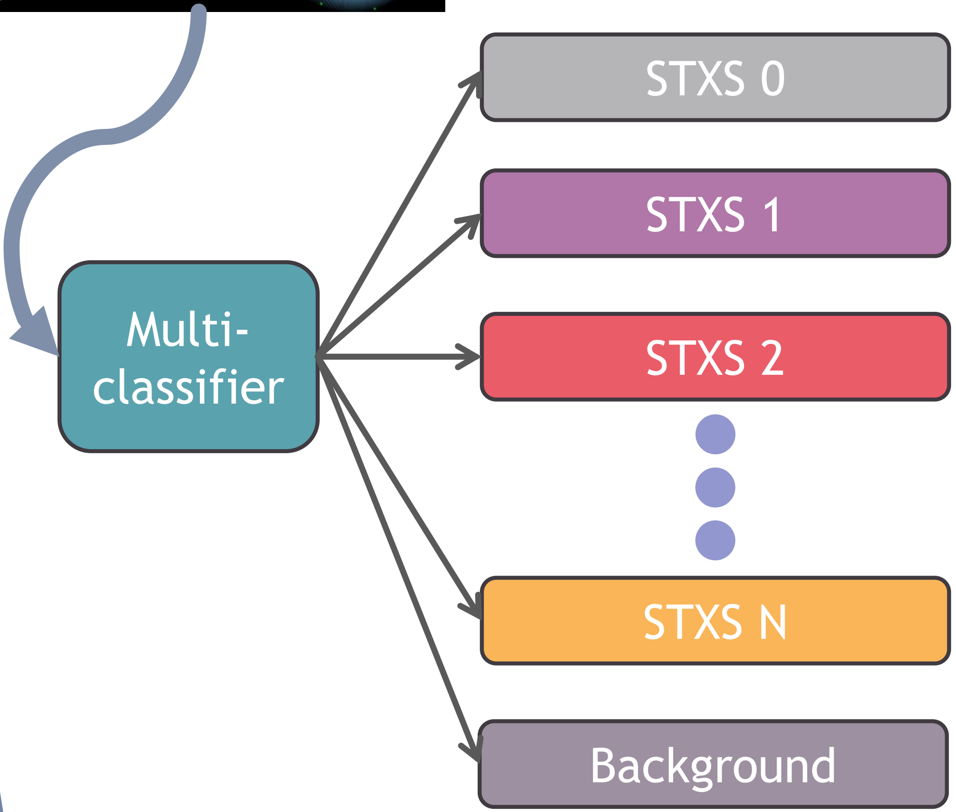
- Run 2 analysis used a complicated sequence of selection cuts + BDTs to categorise events into analysis categories (~STXS regions)
- Define “tag sequence” to prioritise rare production modes
  - Each BDT optimised sensitivity to just a few STXS regions
  - No global view of how categorising an event affects the other “out-of-scope” regions

Analysis strategy

Run 3

- Goal for Run 3 is to use classification techniques which maintain a *global view*

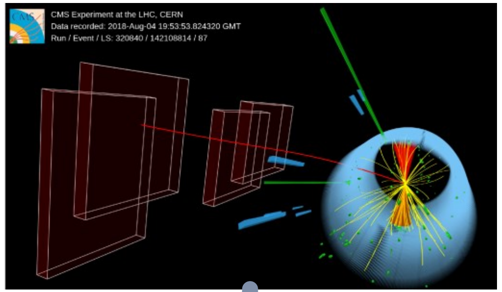


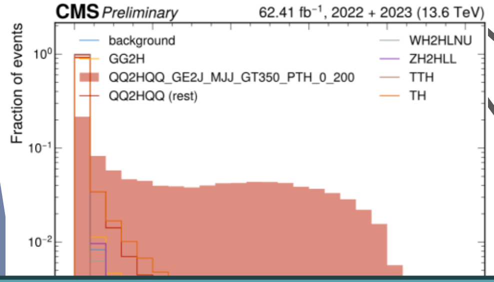


Analysis strategy

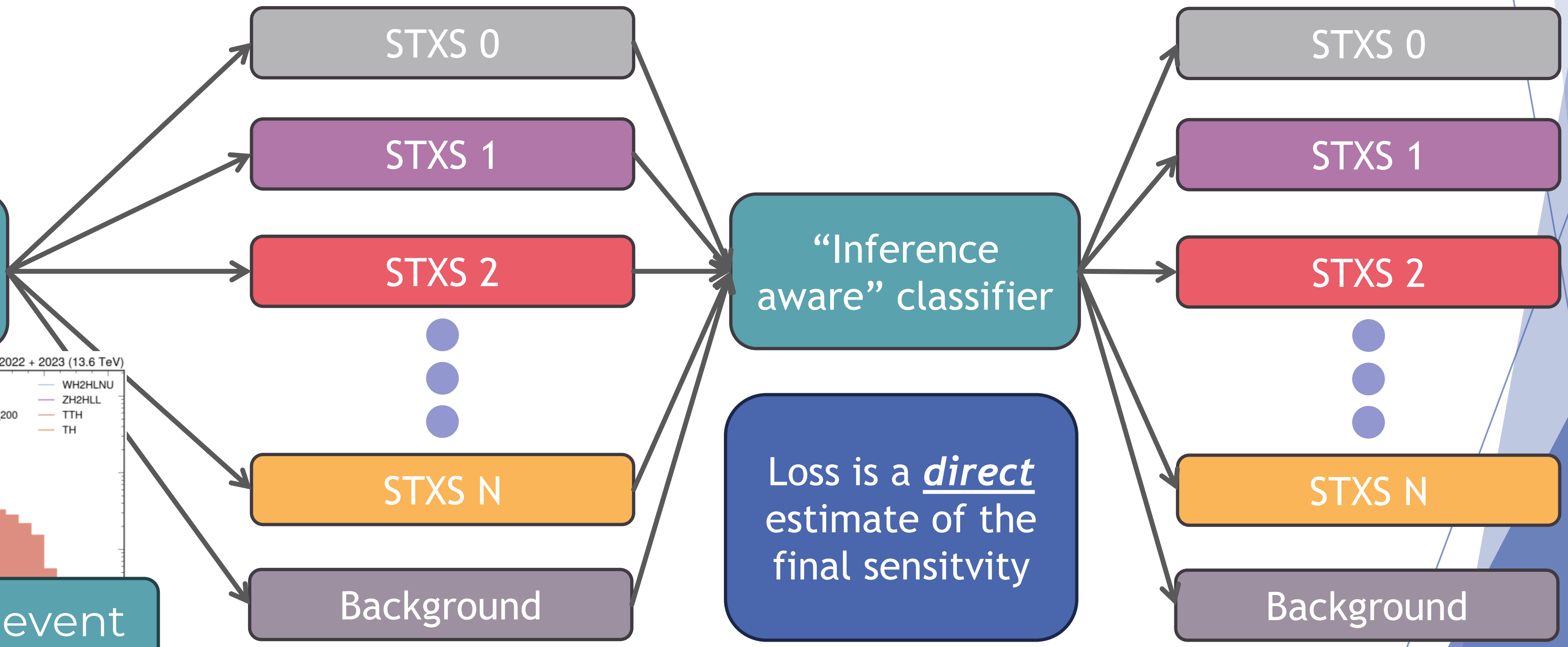
Run 3

- Goal for Run 3 is to use classification techniques which maintain a <u>*global view*</u>
- Current implementation has two steps





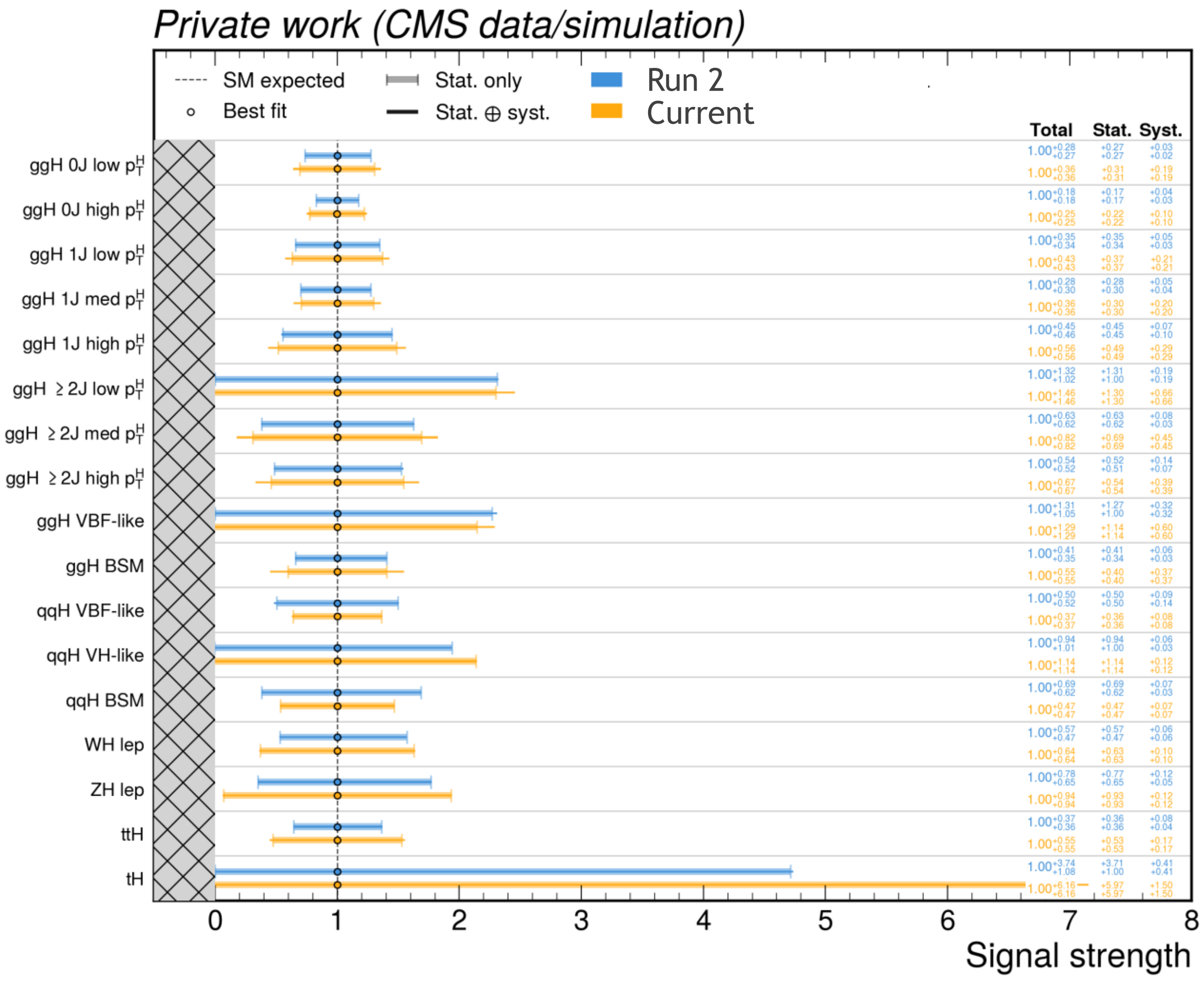
P(STXS) per-event



1806.04743

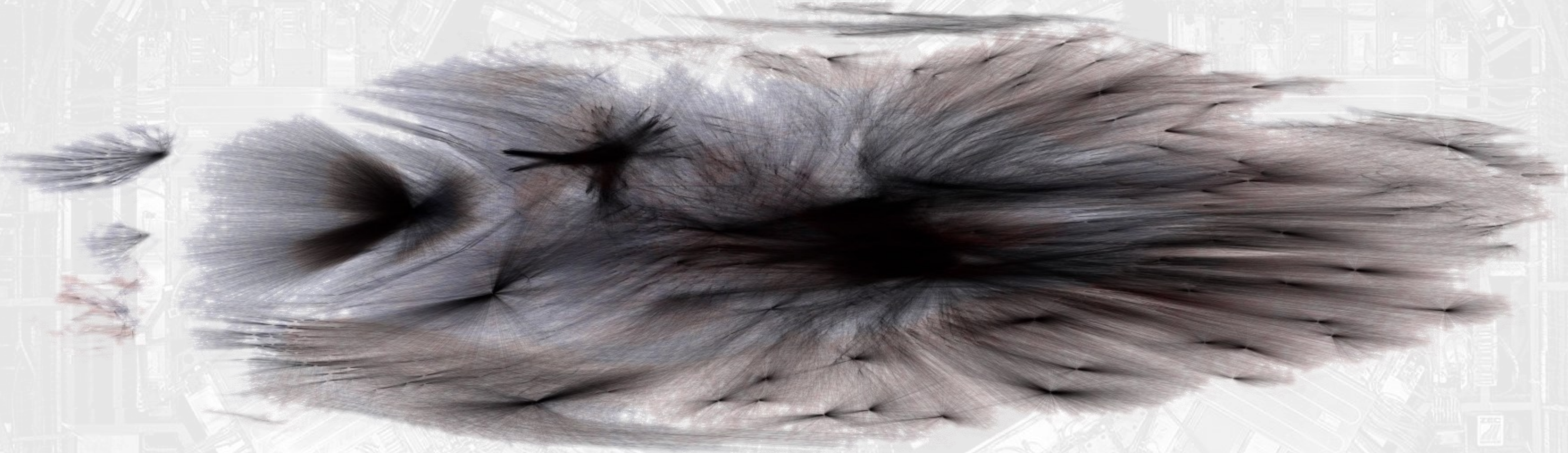
Current performance

- Approaching Run 2 sensitivity when scaled to the same luminosity



- Significant improvements expected soon from:
  - Splitting categories by resolution
  - Splitting categories like $Z(\mathrm{lep})H$ into $Z(l^+l^-)H$ and $Z(\nu\nu)H$
  - Including data from 2024 ($\sim 110\mathrm{fb}^{-1}$)

Statistical procedure



Statistical model

- Eventually, we fit to the $m_{\gamma\gamma}$ distribution

$$\mathcal{L} = \prod_{c}^{N_C} \prod_{b=1}^{N_B^c} \mathrm{Poiss}\left(n_{cb}; n_{cb}^{\exp}(\vec{\mu}, \vec{\nu})\right) \times \prod_{e} p_e(y_e; \nu_e)$$

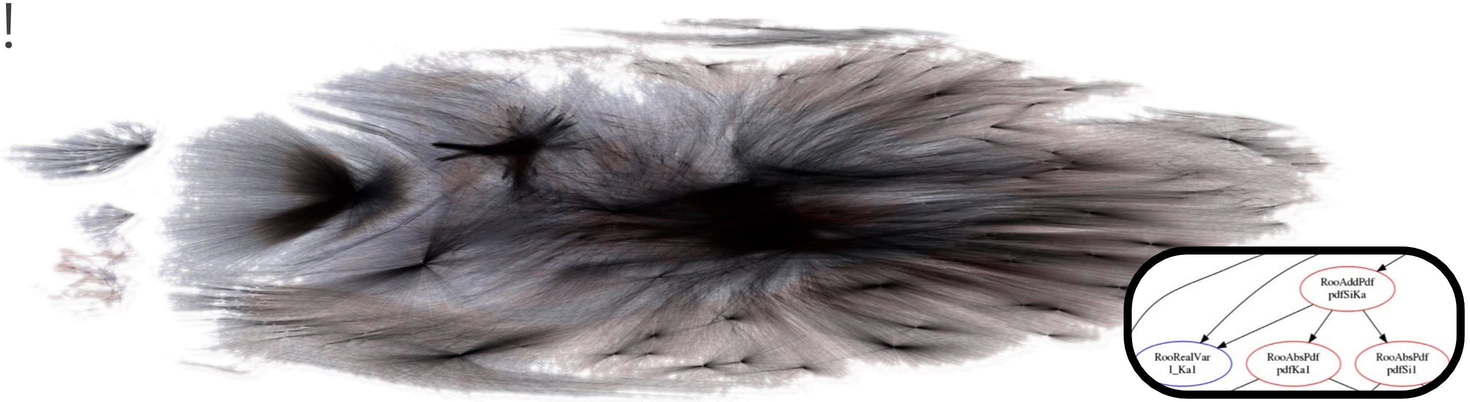
Signal and background modelling

$$n_{cb}^{\exp} = \sum_{p} \mu_p S_{cb}(\vec{\mu}, \vec{\nu}) + B_{cb}(\vec{\mu}, \vec{\nu})$$

Nuisance parameters

where we have c categories with bins b, processes p and nuisances e

- Can get large quickly – the STXS combination in Run 2 had $\mathcal{O}(10^4)$ parameters!



Statistical model

- Eventually, we fit to the $m_{\gamma\gamma}$ distribution

$$\mathcal{L} = \prod_{c}^{N_C} \prod_{b=1}^{N_B^C} \text{Poiss}\left(n_{cb}; n_{cb}^{\text{exp}}(\vec{\mu}, \vec{\nu})\right) \times \prod_{e} p_e(y_e; \nu_e)$$

Signal and background modelling

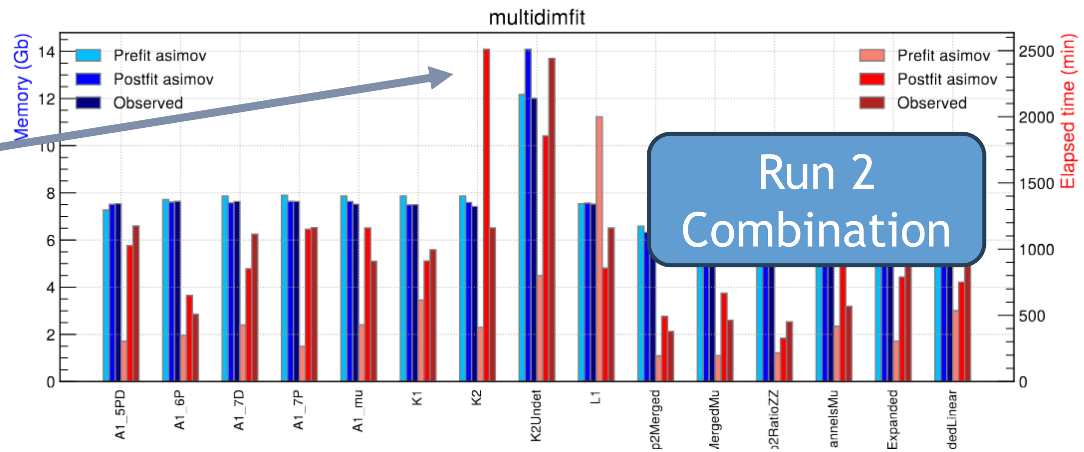
$$n_{cb}^{\text{exp}} = \sum_{p} \mu_p S_{cb}(\vec{\mu}, \vec{\nu}) + B_{cb}(\vec{\mu}, \vec{\nu})$$

Nuisance parameters

where we have c categories with bins b, processes p and nuisances e

- Can get large quickly – the STXS combination in Run 2 had $\mathcal{O}(10^4)$ parameters!

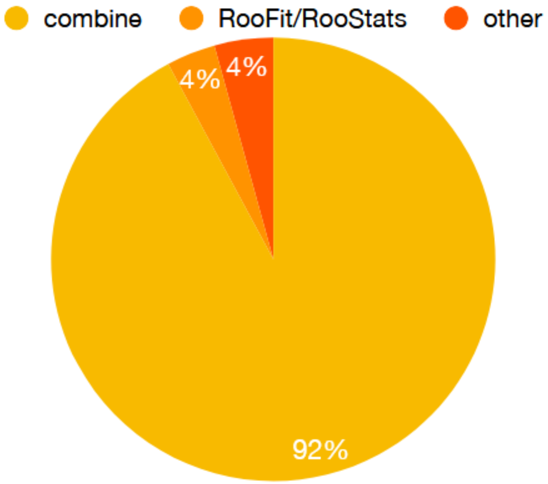
Some fits taking almost 2 days!



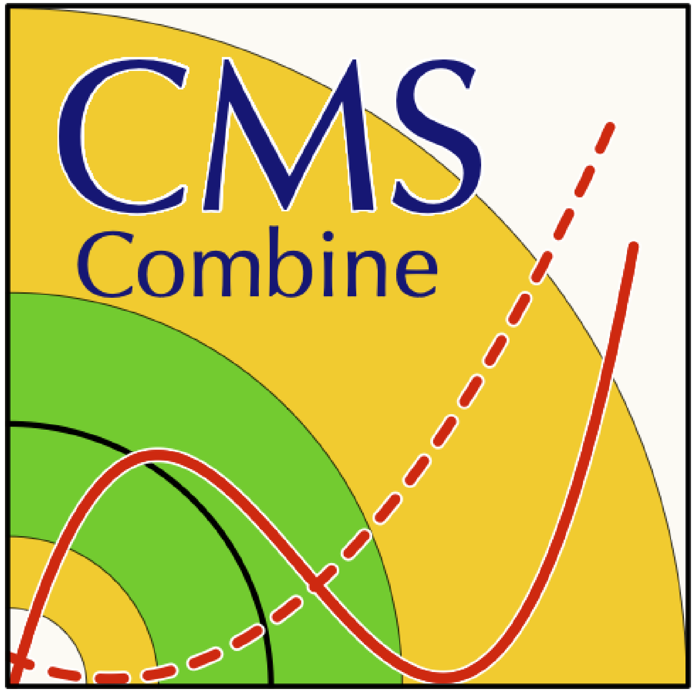
Run 2 Combination

Automatic Differentiation in Combine

- Combine is the most widely-used statistical analysis tool within CMS, providing a command-line interface to many statistical techniques
- Originally developed during Run 1 for the Higgs search, now used by the majority of analyses



From Statistics Committee Questionnaires 2021-2022



2404.06614

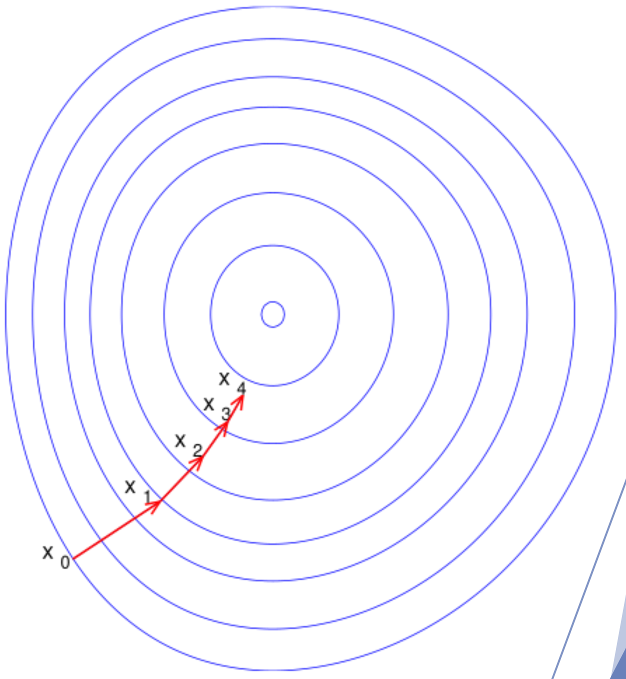
Automatic Differentiation in Combine

- Many choices for which specific optimizer to use within Combine, but almost exclusively the defaults are used: MINUIT with the MIGRAD optimizer
- At its core, MIGRAD is a gradient-descent algorithm, where parameters $\vec{x}$ are iteratively updated according to

$$\vec{x}_{n+1} = \vec{x}_n - \alpha \nabla f(\vec{x}_n)$$

Step-size

Gradient of target function

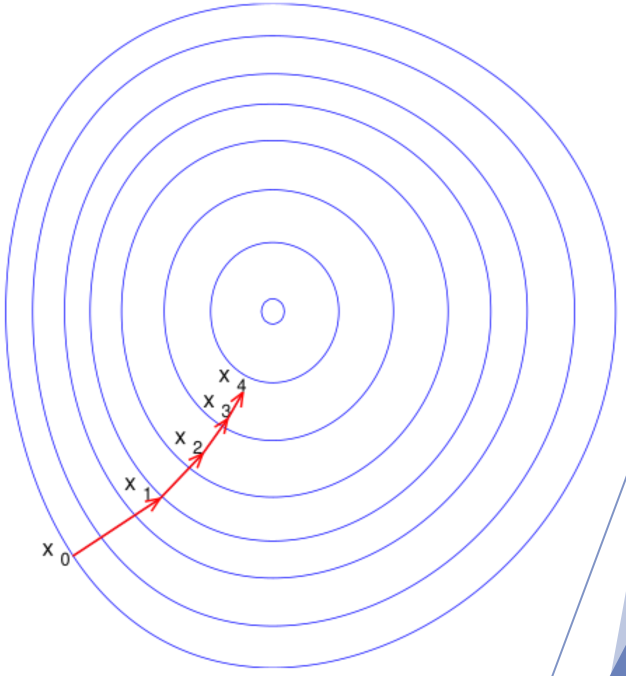


2404.06614

Automatic Differentiation in Combine

- Many choices for which specific optimizer to use within Combine, but almost exclusively the defaults are used: MINUIT with the MIGRAD optimizer
- At its core, MIGRAD is a gradient-descent algorithm, where parameters $\vec{x}$ are iteratively updated according to

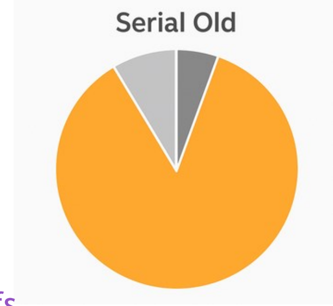
$$\vec{x}_{n+1} = \vec{x}_n - \alpha \nabla f(\vec{x}_n)$$



2404.06614

- At present, Combine differentiates numerically, where *for each* x_n^i in $\vec{x}_n$, we need to calculate

$$\frac{\partial f}{\partial x_n^i} \approx \frac{f(\vec{x}_n + h\vec{e}_i) - f(\vec{x}_n)}{h}$$



Z. Wolffs

Automatic Differentiation in Combine

- For the past ~4 years, the RooFit team have worked to provide an *automatic differentiation* backend
- Implemented using Cla∂, a source-code transformation plugin for the Clang compiler

```
double absFunc(double x) {
 if (x < 0) return -x;
 else return x;
}
```

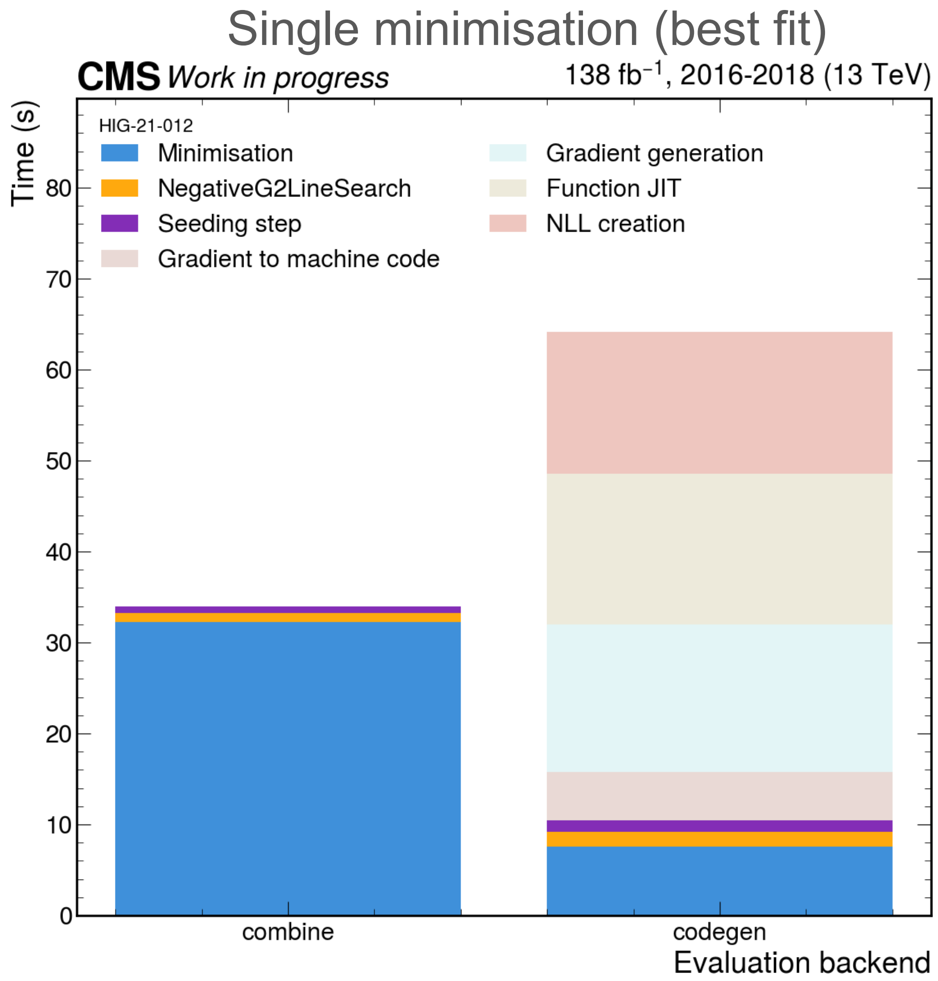
`clad::differentiate(absFunc)` →

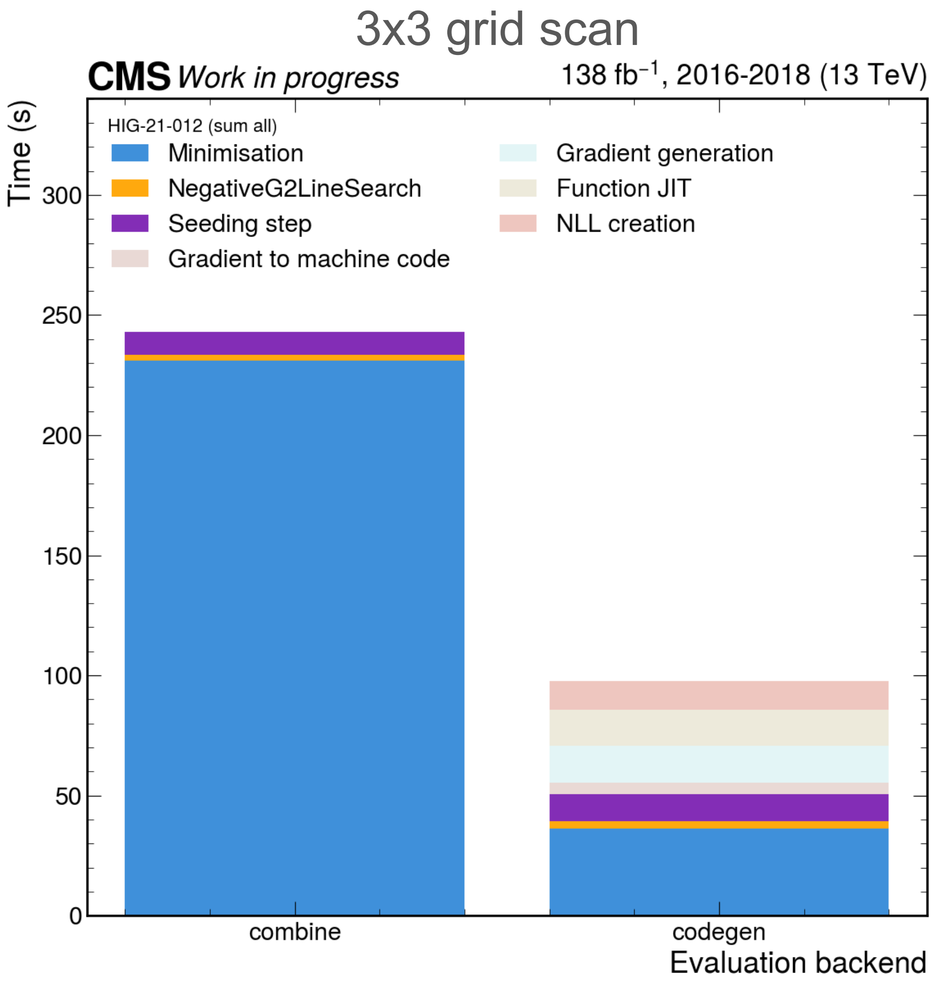
```
double absFunc_darg0(double x) {
   double _d_x = 1;
   if (x < 0) return -_d_x;
   else return _d_x;
}
```

- Source-code transformation allows for compiler optimisation of the gradient code, in principle faster than operator overloading (c.f. PyTorch, TensorFlow, JAX)

Automatic Differentiation in Combine

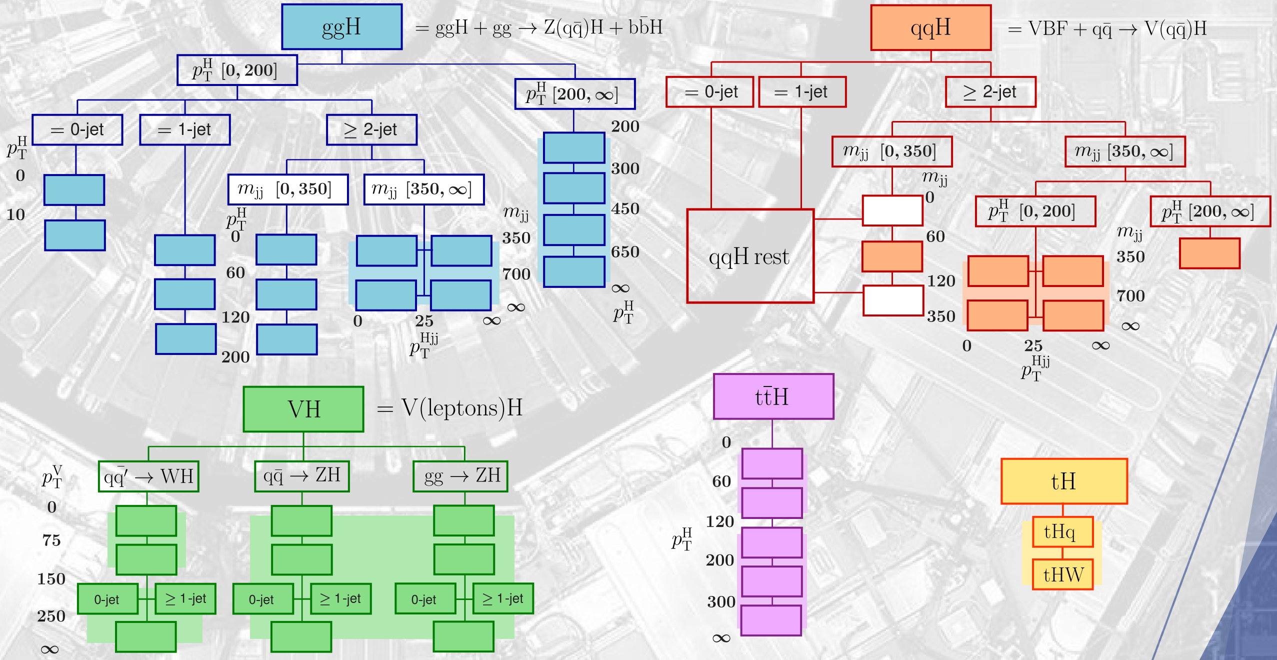
- Preliminary results look promising, with ~5x faster fits!





Summary

- Run 3 $H \rightarrow \gamma\gamma$ STXS measurement progressing well
- Improved, global categorisation strategy with clear avenues to improve
- Computational advances underway to enable the most precise measurements in the future



Thank you

Backup

Automatic Differentiation example

- Consider some general $f(\vec{x}) = \sum_{i=0}^{N} g_i(x_i)$
- Cost to evaluate the function is $T_f = \sum_{i=1}^{N} c_i$, where c_i is the cost to evaluate $g_i(x_i)$

Automatic Differentiation example

- Consider some general $f(\vec{x}) = \sum_{i=0}^{N} g_i(x_i)$
- Cost to *evaluate* the function is $T_f = \sum_{i=1}^{N} c_i$, where c_i is the cost to evaluate $g_i(x_i)$
- The gradient is $\nabla f(\vec{x}) = \left(g_1'(x_1), \ldots, g_n'(x_n)\right)$

Automatic Differentiation example

- The gradient is $\nabla f(\vec{x}) = \left(g_1'(x_1), \dots, g_n'(x_n)\right)$
- Numerically:
$$\nabla f(\vec{x}) \approx \left(\frac{f(\vec{x} + he_1) - f(\vec{x})}{h}, \dots, \frac{f(\vec{x} + he_n) - f(\vec{x})}{h}\right)$$
- Already evaluated $f(\vec{x})$ and the cost of $f(\vec{x} + he_i)$ is the same as that of $f(\vec{x})$
- Total cost for the *gradient* is N lots of T_f: $\mathcal{O}_{\text{num}}(N \cdot T_f)$
- For function + gradient, cost is
$$\mathcal{O}_{\text{num}}^{\text{grad}}\left(N \cdot T_f\right) + \mathcal{O}_{\text{num}}^{f}\left(T_f\right) = \mathcal{O}_{\text{num}}^{\text{tot}}(N \cdot T_f)$$

Automatic Differentiation example

- The gradient is $\nabla f(\vec{x}) = \left(g_1'(x_1), \dots, g_n'(x_n)\right)$
- With AD: When we evaluate $f(\vec{x})$, we store the intermediate values

$$x_i \to_{\text{store}} v_i \coloneqq g_i(x_i) \to_{\text{store}} f = \sum_i v_i$$

so we are left with a "tape": $\{x_1, \dots, x_N, v_1, \dots v_N, f\}$.

- When we come to compute the gradients, we go back through the tape, applying the chain rule

$$\bar{v}_i = \frac{\partial f}{\partial v_i} = 1$$

$\mathcal{O}(1)$ per node

$$\bar{x}_i = \bar{v}_i \cdot g_i'(x_i) = 1 \cdot g_i'(x_i)$$

Automatic Differentiation example

- The gradient is $\nabla f(\vec{x}) = \left(g_1'(x_1), \dots, g_n'(x_n)\right)$
- With AD: When we evaluate $f(\vec{x})$, we store the intermediate values

$$x_i \rightarrow_{\text{store}} v_i := g_i(x_i) \rightarrow_{\text{store}} f = \sum_i v_i$$

so we are left with a "tape": $\{x_1, \dots, x_N, v_1, \dots v_N, f\}$.

- When we come to compute the gradients, we go back through the tape, applying the chain rule

$$\bar{v}_i = \frac{\partial f}{\partial v_i} = 1$$

$\mathcal{O}(1)$ per node

$$\bar{x}_i = \bar{v}_i \cdot g_i'(x_i) = 1 \cdot g_i'(x_i)$$

$\mathcal{O}(1)$ per node, since we have $g_i'(x_i)$ from our AD tool

Automatic Differentiation example

- The gradient is $\nabla f(\vec{x}) = \left(g_1'(x_1), \dots, g_n'(x_n)\right)$
- With AD: When we evaluate $f(\vec{x})$, we store the intermediate values

$$x_i \to_{\text{store}} v_i \coloneqq g_i(x_i) \to_{\text{store}} f = \sum_i v_i$$

so we are left with a "tape": $\{x_1, \dots, x_N, v_1, \dots v_N, f\}$.

- When we come to compute the gradients, we go back through the tape, applying the chain rule

$$\bar{v}_i = \frac{\partial f}{\partial v_i} = 1$$

$\mathcal{O}(1)$ per node

$$\bar{x}_i = \bar{v}_i \cdot g_i'(x_i) = 1 \cdot g_i'(x_i)$$

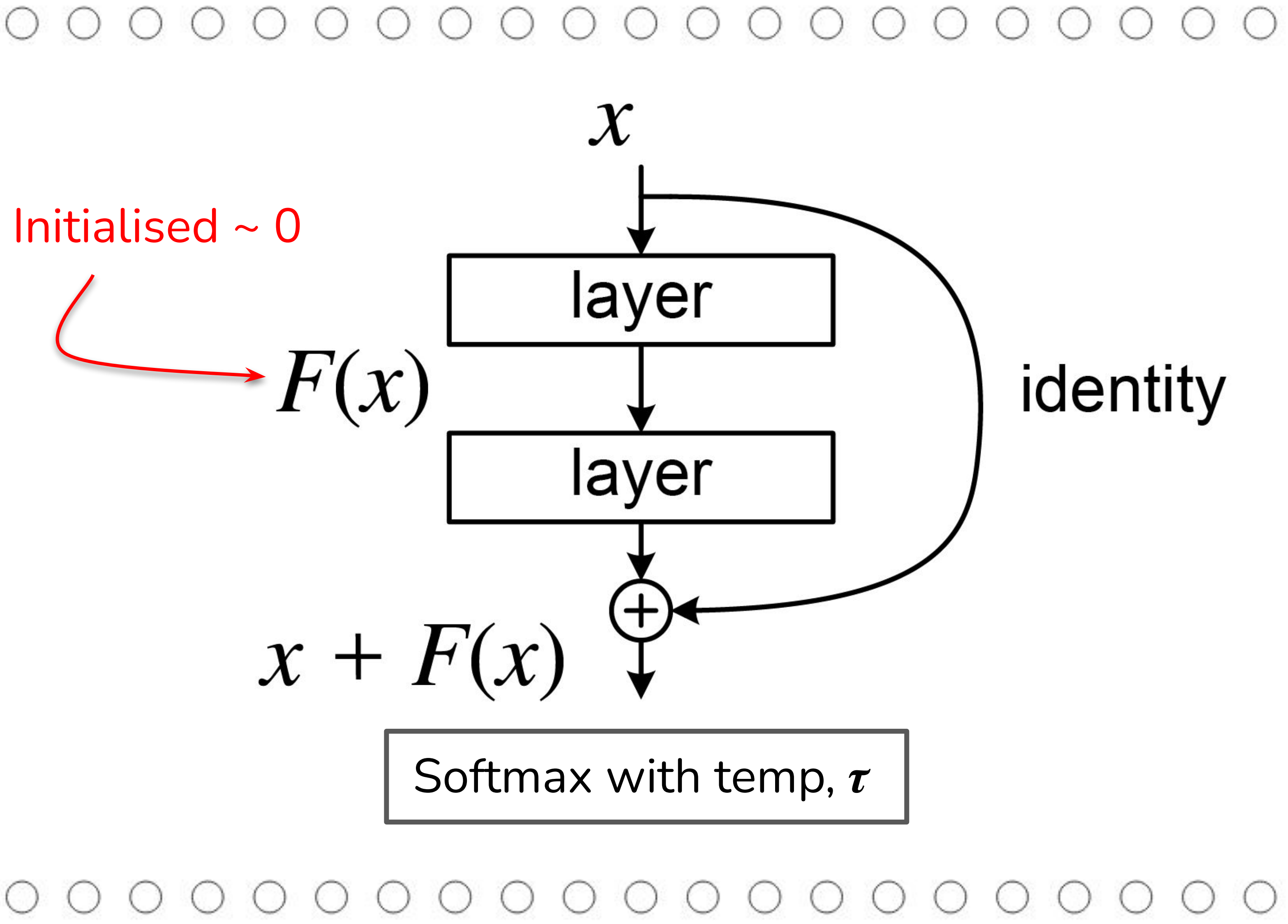
$\mathcal{O}(1)$ per node, since we have $g_i'(x_i)$ from our AD tool

Total cost:
$\mathcal{O}_{\text{AD}}^{\text{grad}}(N \cdot 1) + \mathcal{O}_{\text{AD}}^{f}(T_f) = \mathcal{O}_{\text{AD}}^{\text{tot}}(T_f)$

Current approach

21 XGBoost probabilities

Category weights applied for these studies



21 analysis categories

Inference-aware loss

Likelihood = Product of Poisson terms over categories, k

$$L(\vec{\mu}) = \prod_k e^{-\lambda_k(\vec{\mu})} \lambda_k(\vec{\mu})^{n_k} / n_k!$$

Signal + background yields from MC in [120, 130] GeV, where index i labels STXS region to be measured

$$\lambda_k(\vec{\mu}) = \sum_i \mu_i s_{ik} + b_k \qquad n_k = \lambda_k|_{\vec{\mu}=1} = \sum_i s_{ik} + b_k$$

Asimov

$$\mathrm{NLL}(\vec{\mu}) \equiv -\ln L(\vec{\mu}) = \sum_k \lambda_k(\vec{\mu}) - n_k \ln \lambda_k(\vec{\mu}) + \ln n_k!$$

Analytic expression for Hessian matrix, H

$$\frac{\partial \mathrm{NLL}(\vec{\mu})}{\partial \mu_i} = \sum_k s_{ik} - \frac{n_k s_{ik}}{\lambda_k(\vec{\mu})} \qquad \frac{\partial^2 \mathrm{NLL}(\vec{\mu})}{\partial \mu_i \partial \mu_j} = \sum_k \frac{n_k s_{ik} s_{jk}}{\lambda_k(\vec{\mu})}$$

1st derivative — 2nd derivative

$$\mathcal{H}_{ij} = \left.\frac{\partial^2 \mathrm{NLL}(\vec{\mu})}{\partial \mu_i \partial \mu_j}\right|_{\vec{\mu}=1} = \sum_k \frac{s_{ik} s_{jk}}{n_k}$$

$$\mathrm{Cov} = \mathcal{H}^{-1}$$

Loss functional of covariance matrix

$$\mathcal{L}[\mathrm{Cov}] = \mathrm{Tr}(\mathrm{Cov}) = \sum_i \mathrm{Var}(\mu_i)$$

Loss vs epoch

- Vast improvement on category weights approach used in July