# Discovering CP-Sensitive Observables with Symbolic Regression

**Group 6 · TIFR | ML4HEP-V4 School Workshop**


Under the supervision of

**Dr. Akanksha Bhardwaj**

Jasmine Chhikara, Pradeep Singh Shekhawat, Chesta Batra, Chandramauli, Somnath Mandal, Ravindra Singh Dhami, Ria Kataria, Dhairyashil

# Goal of the Project

**Question** : Can ML algorithm discover the analytic expression  that is sensitive to CP violation in Higgs production?

- Evaluate the performance of the symbolic observable in VBF-like (Vector Boson Fusion) regions.
- Train a symbolic regression model to learn an event score for CP sensitive operator.
- Compare the performance of Symbolic Regression with Neural Network methods.

# SMEFT

- Evaluate the performance of the symbolic observable in VBF-like (Vector Boson Fusion) regions. Lack of experimental evidence of new physics indicate a mass gap between SM and BSM scales

SMEFT Lagrangian:

$$\mathcal{L} = \mathcal{L}_{\rm SM} + \sum_i \frac{c_i}{\Lambda^2} \widetilde{\mathcal{O}}_i \,,$$

Any amplitude square follows:

$$|\mathcal{M}|^2 = |\mathcal{M}_{\rm SM}|^2 + \frac{c_i}{\Lambda^2}\, 2\,\Re\left[\mathcal{M}_{\rm SM}\mathcal{M}^*_{d6,i}\right] + \frac{c_i c_j}{\Lambda^4}\, \mathcal{M}_{d6,i}\mathcal{M}^*_{d6,j}$$

- New physics effect can be seen as deformation of these distribution

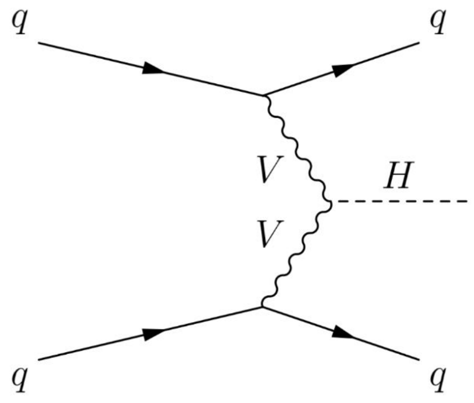
$$\begin{aligned} \mathcal{O}_{\Phi\widetilde{B}} &= \Phi^\dagger \Phi B^{\mu\nu} \widetilde{B}_{\mu\nu} \,, \\ \mathcal{O}_{\Phi\widetilde{W}} &= \Phi^\dagger \Phi W^{i\,\mu\nu} \widetilde{W}^i_{\mu\nu} \,, \\ \mathcal{O}_{\Phi\widetilde{W}B} &= \Phi^\dagger \sigma^i \widetilde{W}^{i\,\mu\nu} B_{\mu\nu} \,, \end{aligned}$$

$$\mathcal{O}_{\Phi\widetilde{W}} = \frac{c_{\Phi\widetilde{W}}}{\Lambda^2} \Phi^\dagger \Phi\, W^{i\,\mu\nu} \widetilde{W}^i_{\mu\nu}$$

# Observables

**What is there in the data file?**

- Each event has kinematic information for two jets and the Higgs boson.
- **Input variables:** $p_{T,j_1}, \eta_{j_1}, \phi_{j_1}, p_{T,j_2}, \eta_{j_2}, \phi_{j_2}, p_{T,h}, \eta_h, \phi_h$




- The target label is **sign_w = +1** or **−1**.
- The symbolic regression task is to find f(x) such that
  sign(f(x)) ≈ sign_w

# Symbolic Regression: in a nutshell

It discovers the analytical expression learning from the data. It is not about fitting the parameters of the given equation with data.

$$(x_i, y_i) \longrightarrow y_i \approx f(x_i)$$

Builds equation from chosen set of operators **+ , - , x , / , sin, cos, tan, ..**

The inspirations behind the algorithmic search:

- Mutation
  - Small random changes create new variants
- Reproduction
  - Good candidates are copied to the next generation
- Selection
  - Better equations are more likely to survive
- Evolution
  - Over many generations, solutions improve

Each candidate equation is judged by two criteria:

Loss = prediction error

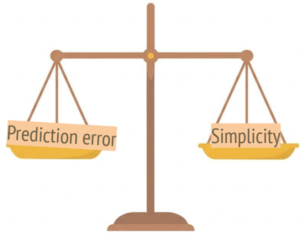
Complexity = Size of expression

Fitness = Error + λ complexity

Error could be MSE

Complexity depends on:

- Size of nodes
- Tree depth
- Number of operations



# What is PySR?

Python symbolic regression library built on python interface and julia backend.

## FEATURES

$$X.shape = (N_{samples}, N_{features})$$

## TARGETS

$$y.shape(N_{samples},)$$

Samples: No. of data points
Features: No. of input variables

## BINARY OPERATORS

Combine two expressions.

Builds complex expressions through combining subexpressions into larger ones

```
+ - * / ^
```

## UNARY OPERATORS

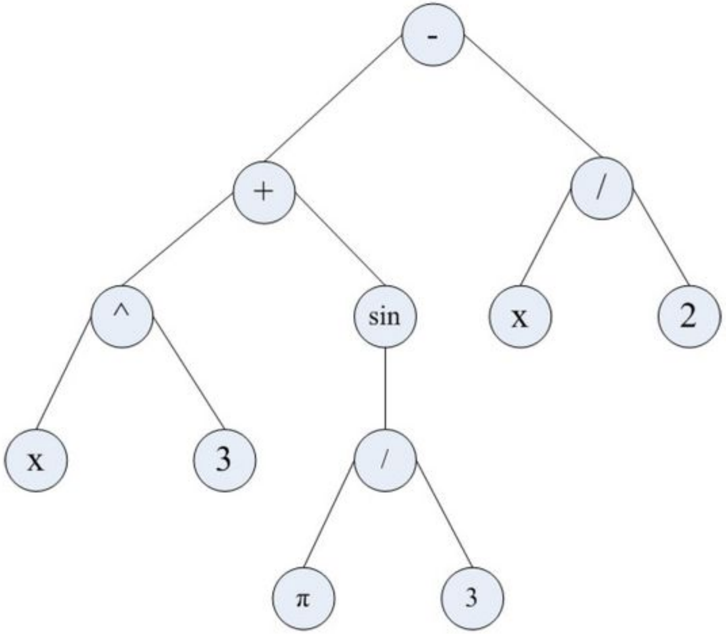
These act on single expressions.

Capture complex functional forms.

```
["sin", "cos", "tan",
 "log", "sqrt"]
```

# Expression Complexity

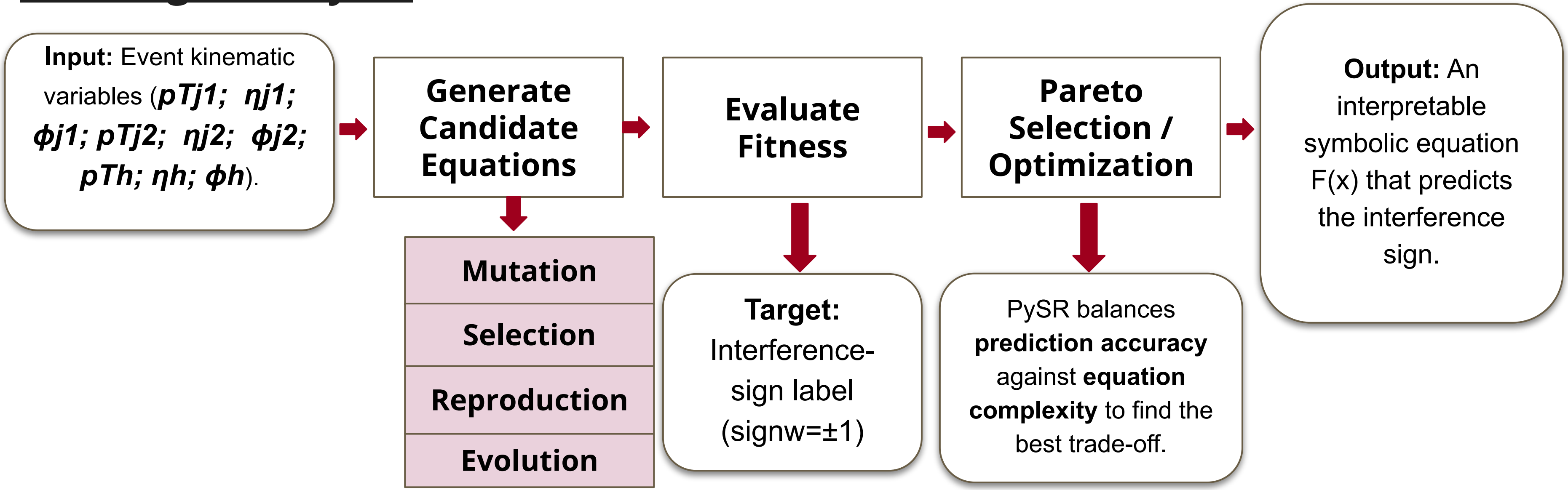
- A mathematical equation can be represented as an expression tree.
- The internal nodes are operations and the leaf nodes are variables or constants.
- SR algorithms often build and modify these expressions to find the best match.
- The complexity of an expression is roughly related to the size of this expression tree.




$$f(x) = x^3 + \sin(\frac{\pi}{3}) - \frac{x}{2}$$

# ML-Constructed CP-odd Observables

## Working with PySR-



# An example of search

Complexity ↑

Loss ↓

| Complexity | Loss | Score | Equation |
|---|---|---|---|
| 1 | 1.000e+00 | 0.000e+00 | y = 0.0058551 |
| 2 | 2.586e-01 | 1.352e+00 | y = sin(Δφ) |
| 4 | 2.037e-01 | 1.193e-01 | y = Δφ/\|Δφ\| |
| 5 | 1.966e-01 | 3.581e-02 | y = sin(Δφ/\|Δφ\|) |
| 6 | 1.934e-01 | 1.647e-02 | y = (\|Δφ\|/Δφ)*0.8981276 |
| 7 | 1.934e-01 | 5.960e-08 | y = sin(Δφ/(\|Δφ/2.0259852\|)) |
| 8 | 1.934e-01 | 5.960e-08 | y = $\sin(\sin(\Delta\phi/(\Delta\phi)) * 2.4077)$ |
| 9 | 1.933e-01 | 1.878e-04 | y = $\sin(\Delta\phi/((\Delta\phi/-2.0251)+0.0025726))$ |

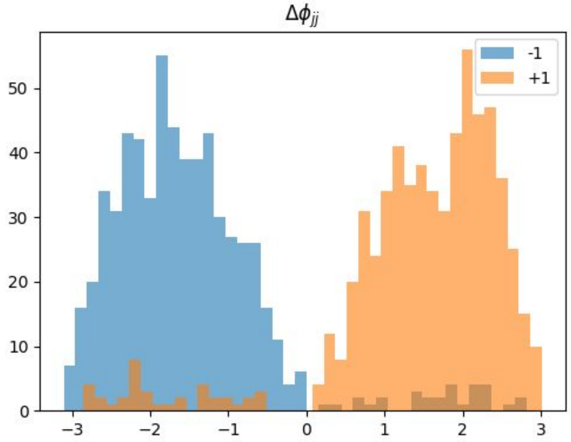
# Rediscovering the angular observable

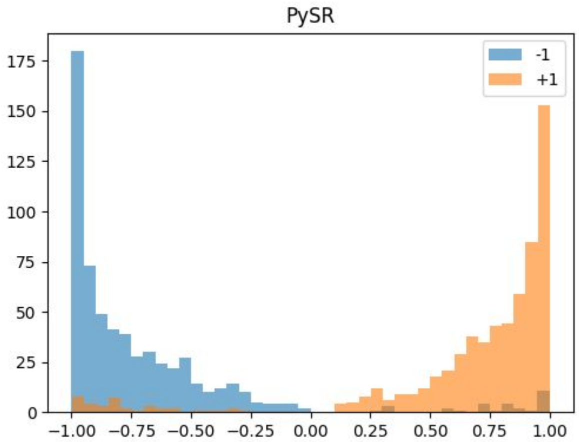
$$A_f = \frac{N(f > 0) - N(f < 0)}{N(f > 0) + N(f < 0)}$$

| observable | asymmetry |
|---|---|
| $\Delta\phi$ | 22.21 |
| Sin($\Delta\phi$) | 22.25 |
| sign($\Delta\phi$) | 22.24 |

| observable | accuracy | AUC |
|---|---|---|
| $\Delta\phi_{jj}$ | 0.950 | 0.951 |
| $\sin(\Delta\phi_{jj})$ | 0.950 | 0.949 |
| **PySR-discovered** | **0.950** | **0.949** |

Score by class: $\sin(\Delta\phi_{jj})$ cleanly separates sign_w at f = 0.




**What PySR found?**

- PySR converged to f ≈ sin($\Delta_\phi$) the known CP observable.
- Both separate the classes at the same level.
- sin($\Delta_\phi$) gives the cleanest left/right split about 0.
- PySR can reconstruct the angular information from raw ϕ.

# VBF-like region: $m_{jj}>400, |\Delta\eta_{jj}|>2.5, \eta_{j1}\eta_{j2}<0$

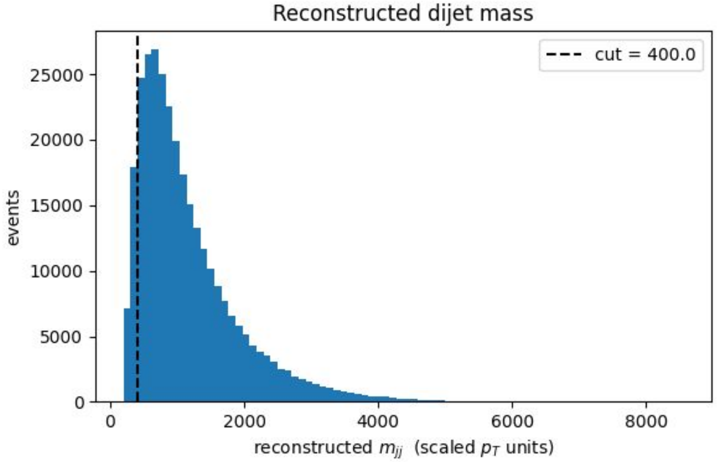
=== Inclusive (n = 307648) ===

| observable | accuracy | ROC AUC |
|---|---|---|
| dphi_jj | 0.9495 | 0.9506 |
| sin(dphi_jj) | 0.9495 | 0.9495 |
| PySR | 0.9495 | 0.9495 |

=== VBF-like (n = 284410) ===

| observable | accuracy | ROC AUC |
|---|---|---|
| dphi_jj | 0.9512 | 0.9522 |
| sin(dphi_jj) | 0.9512 | 0.9513 |
| PySR | 0.9512 | 0.9513 |

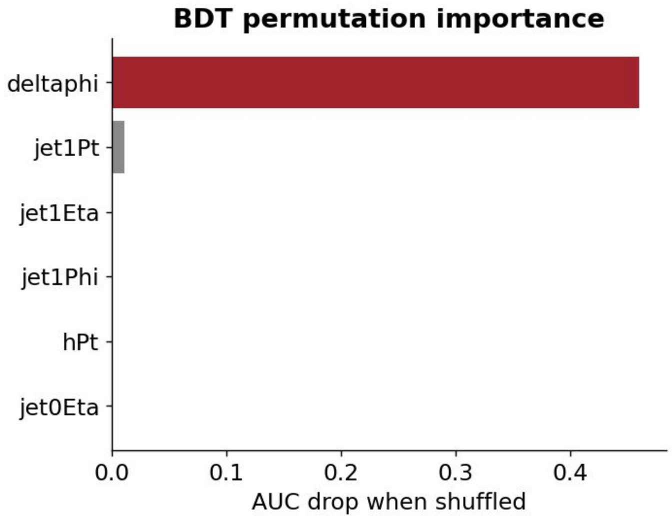
- Accuracy improves slightly: 0.9495 → 0.9512
- Formula stays $\sin(\Delta\phi_{jj})$ – no simpler, but confirmed.

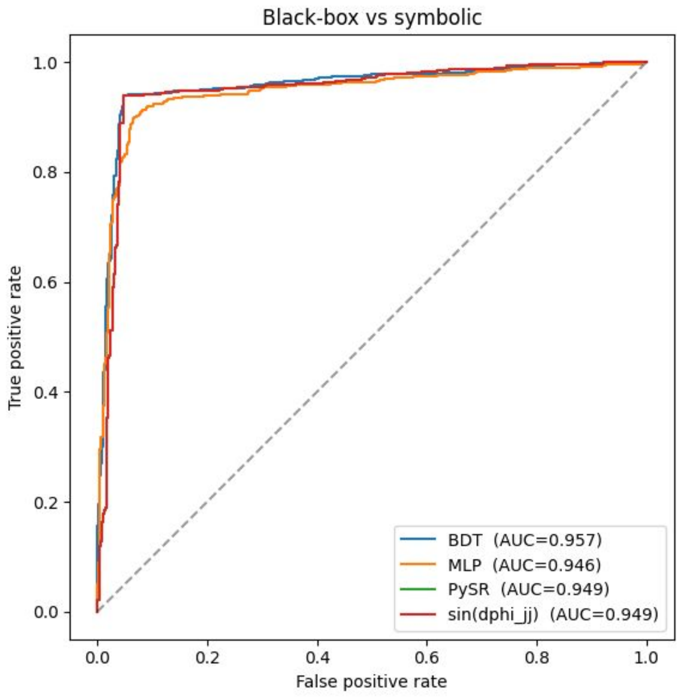



Reconstructed $m_{jj}$ with cut.

# Black-box vs symbolic Regression

- At 5k events the black boxes barely beat – or lose to – a one-line formula.
- BDT importance: $\Delta\phi_{jj}$ dominates (rediscovered), but only via a post-hoc plot.
- The BDT slowly climbs toward the optimal observable (~0.965) as data grows.
- PySR reaches the leading term instantly, data-efficiently, and in readable form.

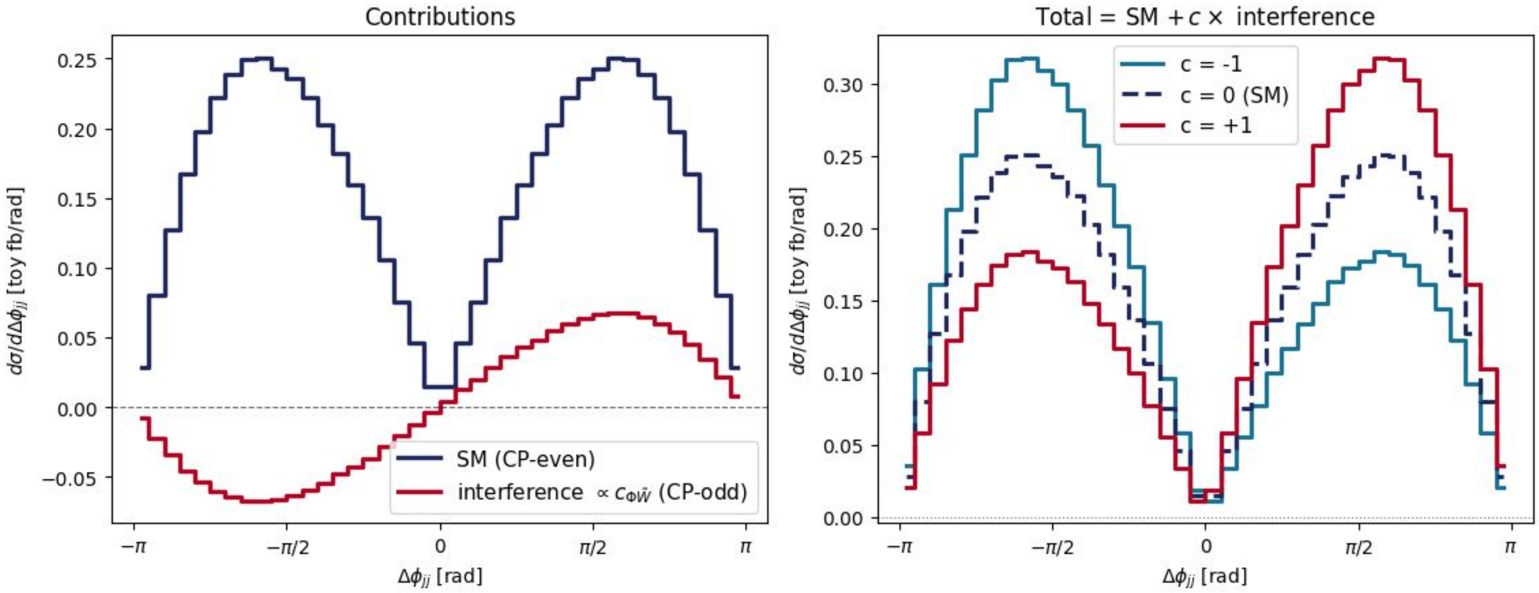



| model | AUC | readable? |
|---|---|---|
| BDT | 0.955 | no |
| MLP (64×64) | 0.946 | no |
| **PySR = sin(Δϕjj)** | **0.949** | **yes ✓** |

# Differential cross section dσ/dΔϕjj

- SM is CP-even: symmetric in $\Delta\phi_{jj}$ → no asymmetry.
- Interference is CP-odd: an antisymmetric sine – the piece sign_w tracks.



# Summary

| observable | region | AUC | interpretable |
|---|---|---|---|
| $\Delta\phi_{jj}$ | inclusive | 0.951 | yes |
| $\sin(\Delta\phi_{jj})$ = PySR | inclusive | 0.949 | yes |
| $\sin(\Delta\phi_{jj})$ | VBF-like | 0.951 | yes |
| BDT / MLP (black box) | inclusive | 0.955 / 0.946 | no |

- PySR rediscovers sin(Δϕjj), the CP-odd optimal observable.
- It is provably CP-odd and physically meaningful.
- VBF cuts give a small, consistent gain.
- Black boxes only match it with far more data – and stay opaque.

# References:


- H. Bahl, E. Fuchs, M. Menen and T. Plehn, SciPost Phys. 20, no.2, 040 (2026) doi:10.21468/SciPostPhys.20.2.040 [arXiv:2507.05858 [hep-ph]].
- A. Bhardwaj, C. Englert, R. Hankache and A. D. Pilkington, Phys. Lett. B 832, 137246 (2022) doi:10.1016/j.physletb.2022.137246 [arXiv:2112.05052 [hep-ph]].
- PySR official documentation: https://ai.damtp.cam.ac.uk/pysr/v1.5.9/


# Acknowledgement:

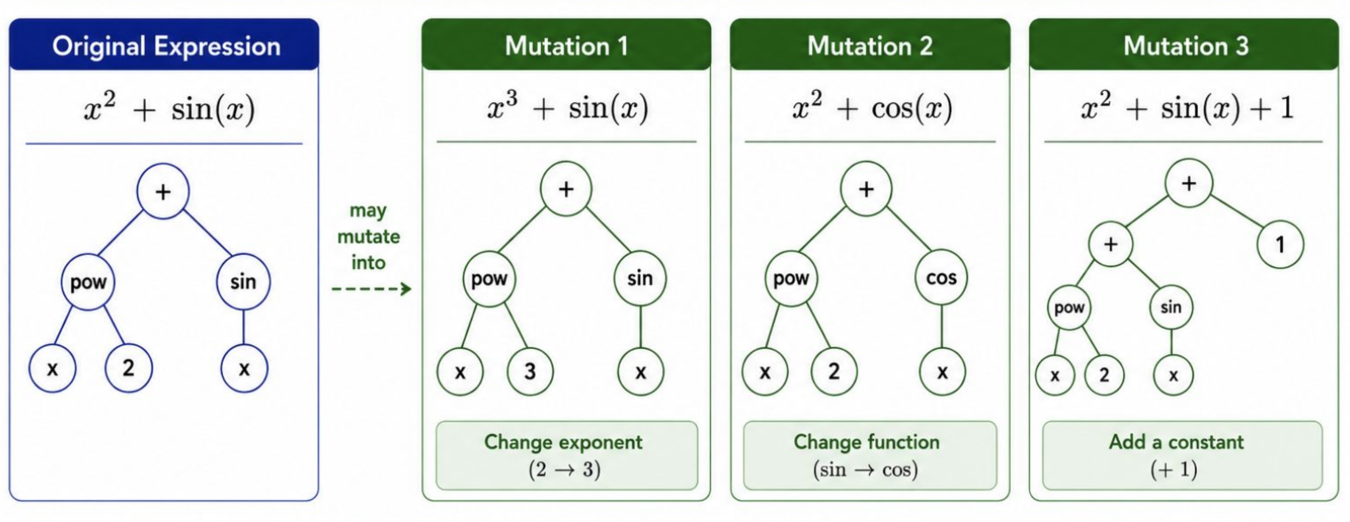

- We would like to express our gratitude to the **ML4HEP Organizers** for giving us the opportunity to explore ML and its applications in High Energy Physics.

- We sincerely thank **Dr. Akanksha Bhardwaj** for designing this workshop project and providing valuable guidance.

Thank You

# BACKUP

# An example of mutation in SR



# Selecting Equations

## Best equation is not always:

the one with the lowest loss.

## Need to consider multiple factors:

**Interpretability**
Simple equations are easier to understand, analyze and communicate.

**Physical meaning**
Terms should correspond to known physical quantities or mechanisms.

**Dimensional consistency**
The equation must be dimensionally consistent.

**Robustness**
The equation should generalize well to different datasets and noise levels.

**Physics knowledge matters**
Domain knowledge helps validate, refine and trust the discovered equation.

Use PySR as a **discovery tool**, and physics as the **guide**.

## How to choose the right equation?

1. Inspect top equations returned by PySR.
2. Check simplicity (number of terms / complexity).
3. Verify dimensional consistency.
4. Look for physical interpretability.
5. Test robustness on validation data.
6. Prefer equations aligned with known physics.

### Example

| Rank | Equation | Loss (MSE) | Comment |
|---|---|---|---|
| 1 | $x^2 + \sin(x) + 1.3$ | 1.2e−4 | Low loss, but harder to interpret |
| 2 | $x^2 + \sin(x)$ | 1.8e−4 | Simple and interpretable |
| 3 | $x^2 + 0.5x$ | 2.5e−4 | Too simple, may miss physics |
| 4 | $\frac{(x^2 + \sin(x))}{(1 + x^2)}$ | 1.5e−4 | Good fit and physically plausible |

Choose the equation that balances **accuracy**, **simplicity**, and **physical insight**.

# ML-Constructed CP-odd Observables

## Working with BDT-

- Each new tree learns mistakes made by previous trees.
- BDT serves as a black-box benchmark.

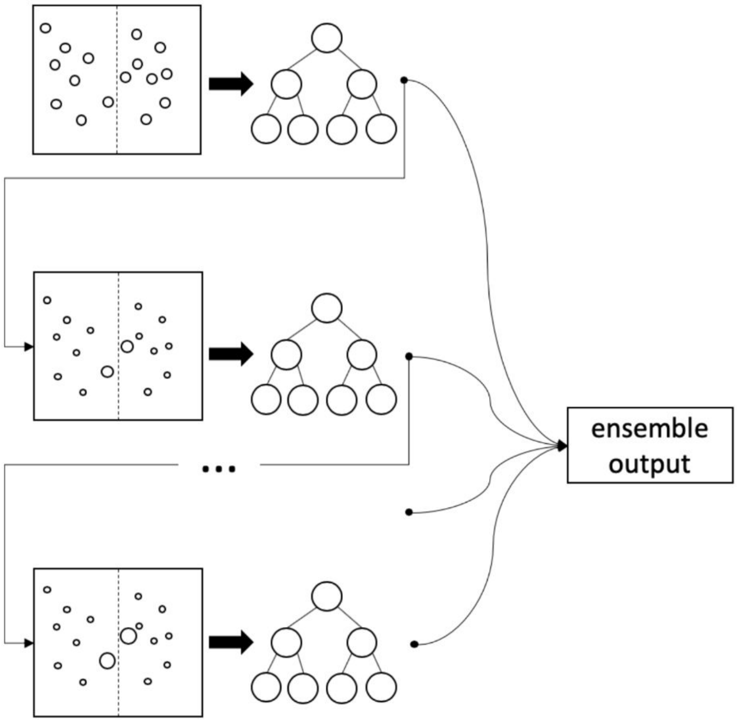



Fig : illustration of the boosted decision trees.

## Working with NN-

- Neural Networks can learn CP-sensitive information from all kinematic variables (e.g., Δηjj, mjj, pT), whereas PySR in our study primarily identified Δϕjj as the dominant contributor.

Note-VBF-like regions using cuts such as $m_{jj} > 400$ GeV, $|\Delta\eta_{jj}| > 2.5$, $\eta_{j1}\eta_{j2} < 0$.

# Search for "ideal" functional form to explain the data

| | pick | score | equation |
|---|---|---|---|
| 0 | | 0.000000e+00 | 0.0058551254 |
| 1 | ✅>>> | 1.352285e+00 | sin(Δφ) |
| 2 | | 1.193117e-01 | Δφ/\|Δφ\| |
| 3 | | 3.580818e-02 | sin(Δφ/\|Δφ\|) |
| 4 | | 1.646847e-02 | (\|Δφ\|/Δφ)*0.8981276 |
| 5 | | 1.034359e-07 | sin(Δφ/(\|Δφ/2.0259852\|)) |