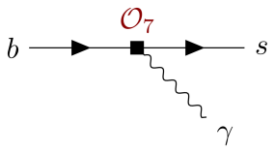
Hyperiso : A general BSM calculator for flavour observables

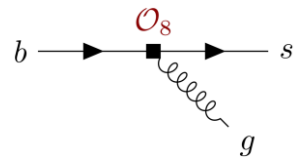
Niels Fardeau, Théo Reymermier
November 14 2024

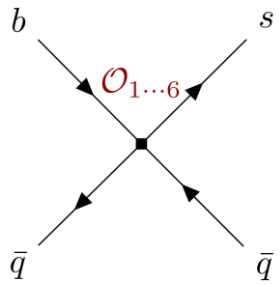
IRN Terascale

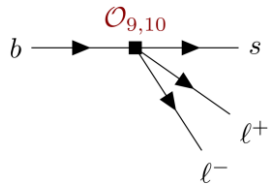
Calculations in flavor physics

$$L_{(B)SM} \to H_{\text{eff}}(b \to sX) = -\frac{4G_F}{\sqrt{2}} V_{tb} V_{ts}^* \sum_{i=1}^{10} C_i(\mu) O_i(\mu)$$

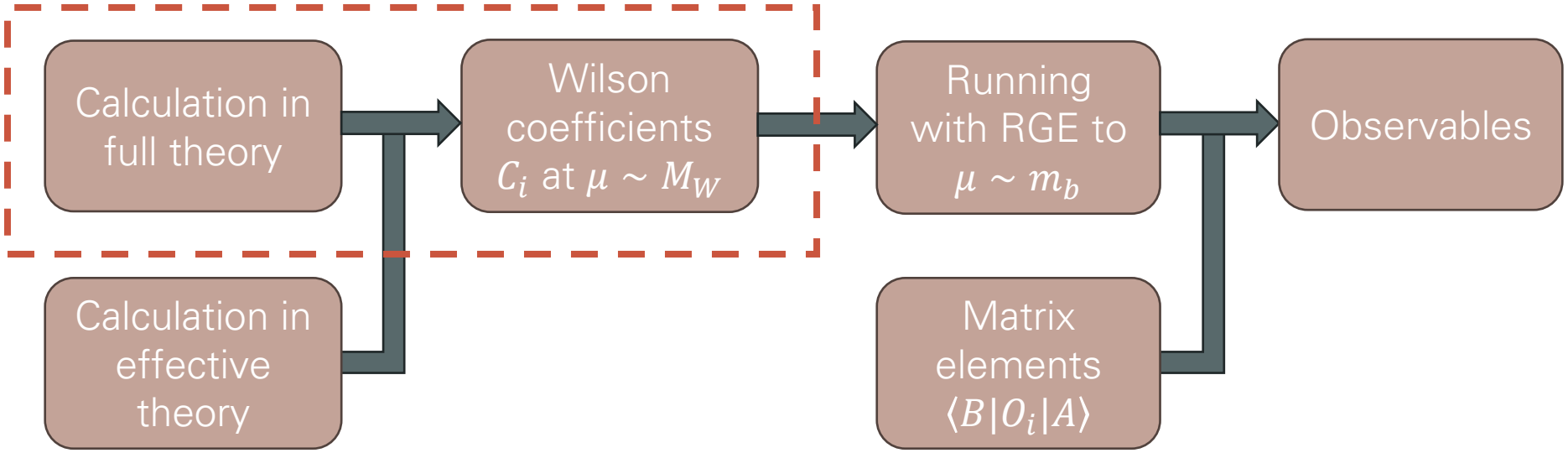








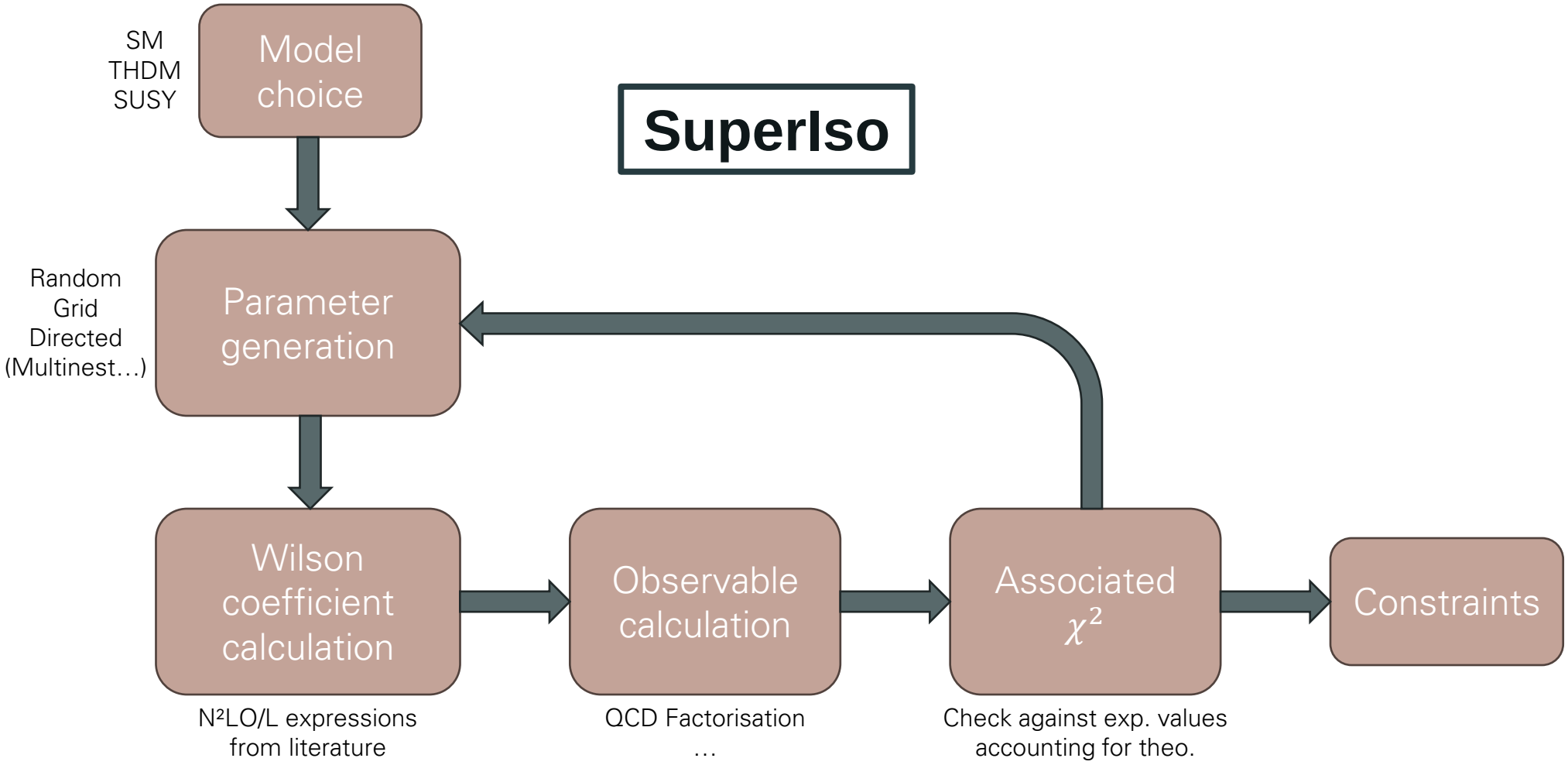
For each BSM model



Need for automated calculations !

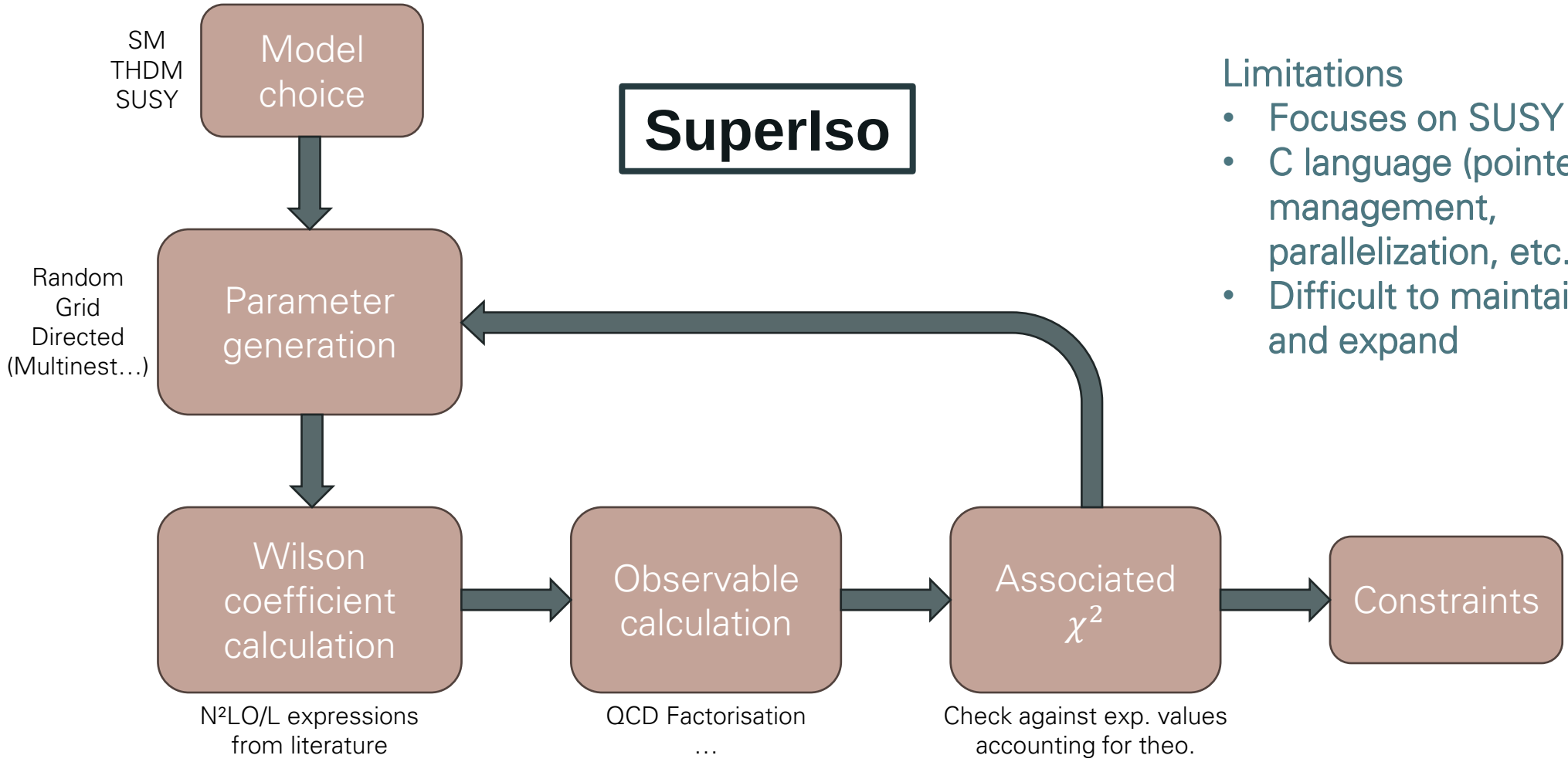
SuperIso

F. Mahmoudi, Comput. Phys. Commun. **178**, 745 (2008) [0710.2067]



SuperIso

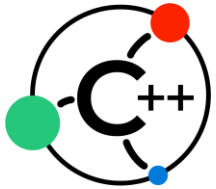
F. Mahmoudi, Comput. Phys. Commun. **178**, 745 (2008) [0710.2067]



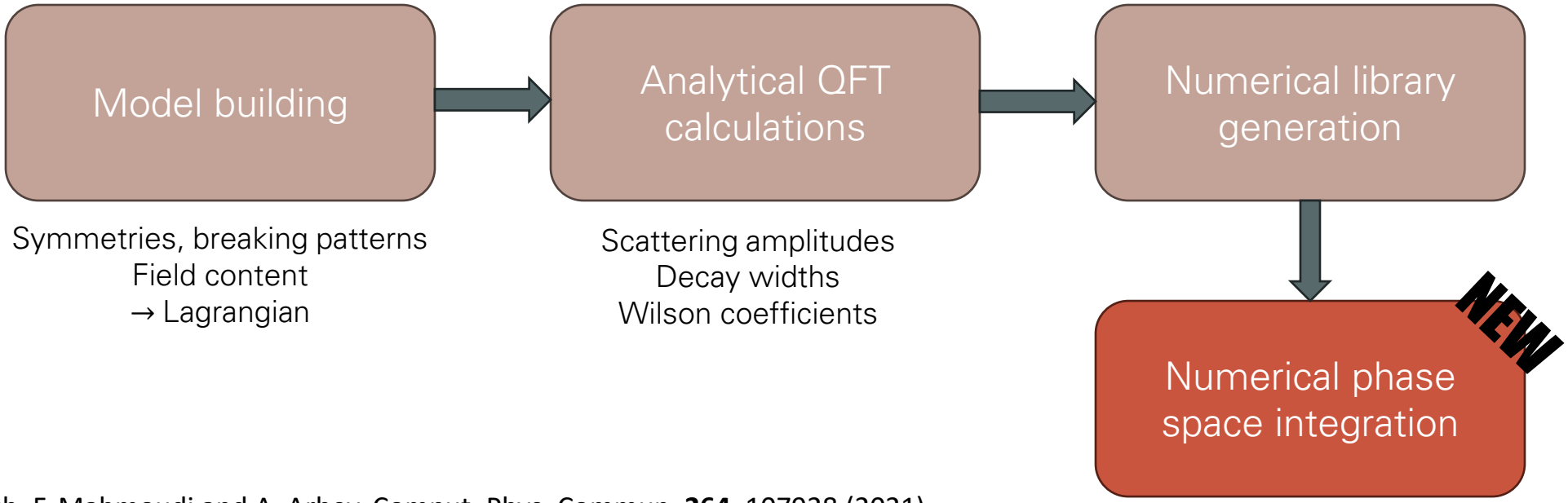
Limitations

- Focuses on SUSY
- C language (pointer management, parallelization, etc.)
- Difficult to maintain and expand

MARTY (1.5)



MARTY



G. Uhlrich, F. Mahmoudi and A. Arbey, Comput. Phys. Commun. **264**, 107928 (2021)

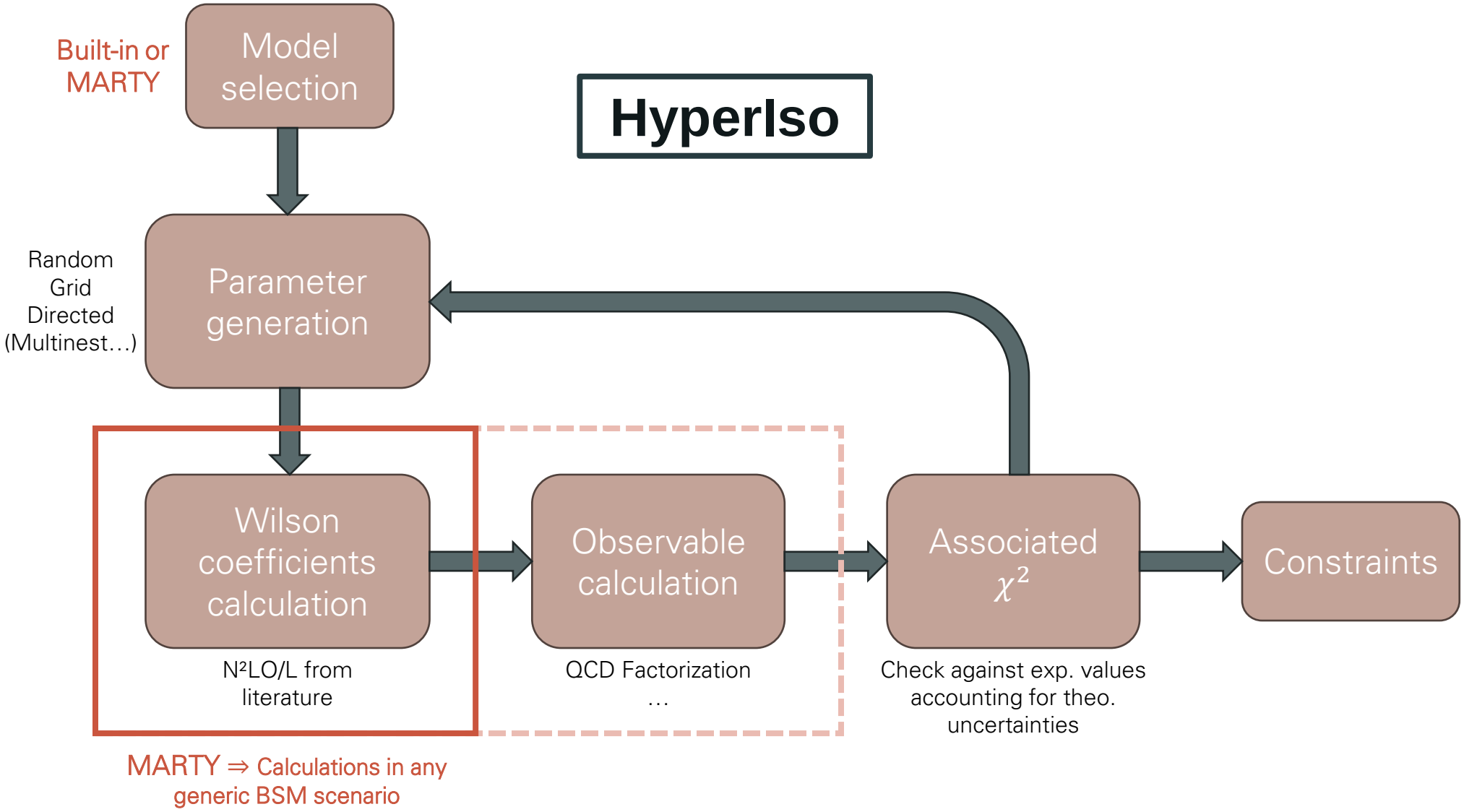
HyperIso

C++ redesign of SuperIso (C99). Workflow diagram unchanged.

Upgrades :

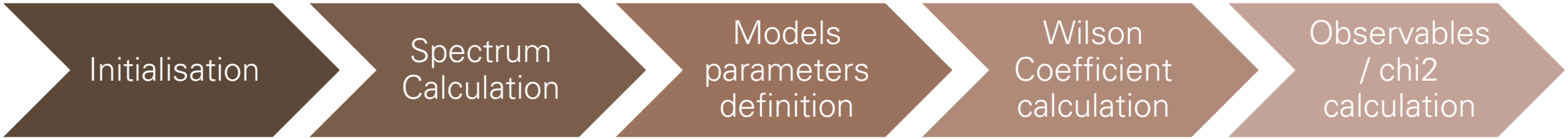
- Modern C++ features
- Clear software architecture
- Various optimizations
- Reproduces SuperIso's behavior for the calculation of Wilson Coefficients (in SM, THDM and SUSY) and observables (WIP)
- Greater flexibility to add new observables and in model choice
- Simplified python interface (+ GUI)

HyperIso – MARTY

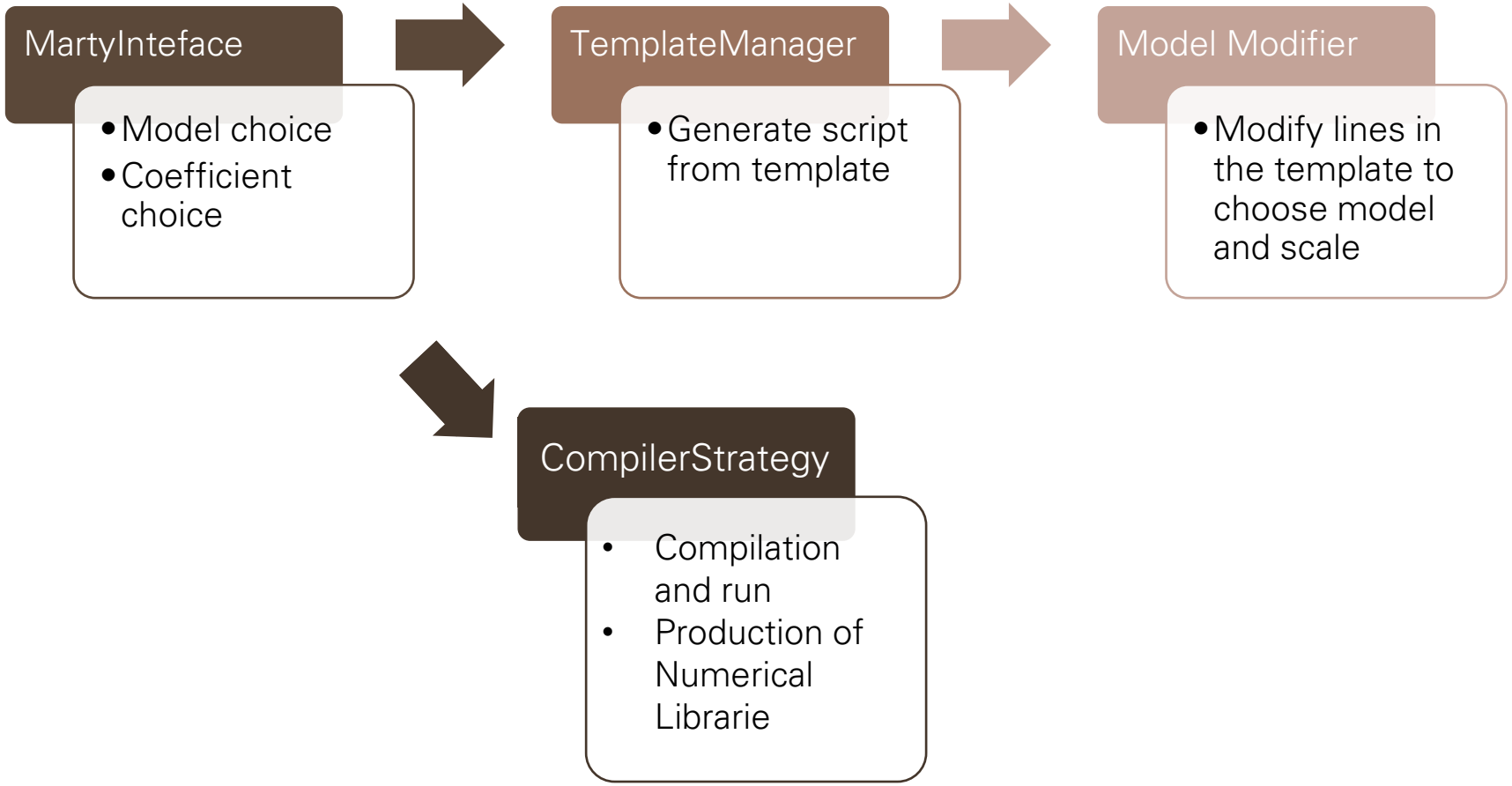


HyperIso – MARTY

Core	ExternalIntegration	DataBase	PhysicalModel	BusinessLogic
• Memory management • Parameter management	• SoftSusy • 2HDMC • MARTY	• LHA reader • Template management • JSON reader • CSV reader	• Wilson coefficients calculation • Wilson Manager • RGE running	• Observable Calculation • Statistical evaluation (χ^2)

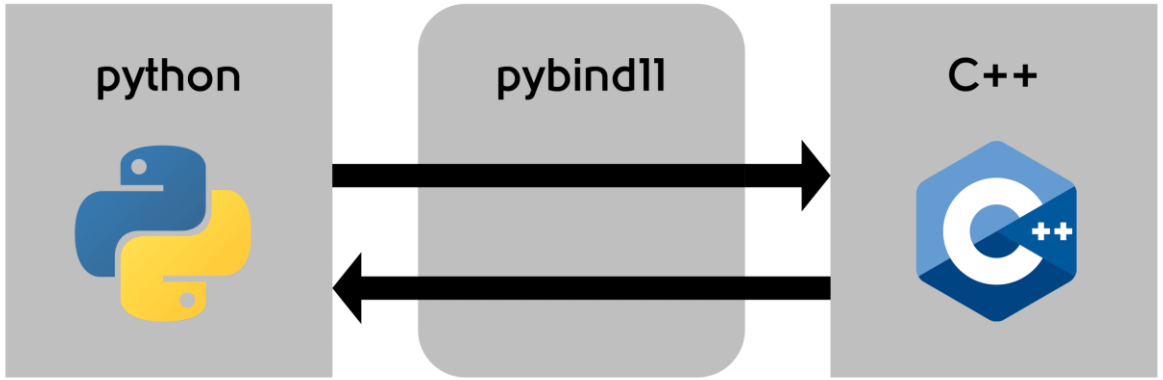


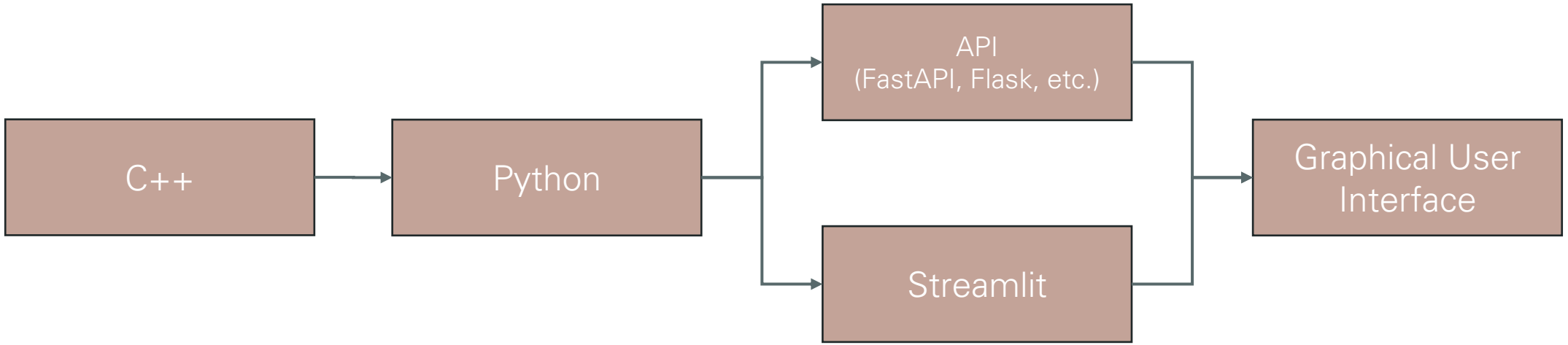
MARTY Interface



Python binding

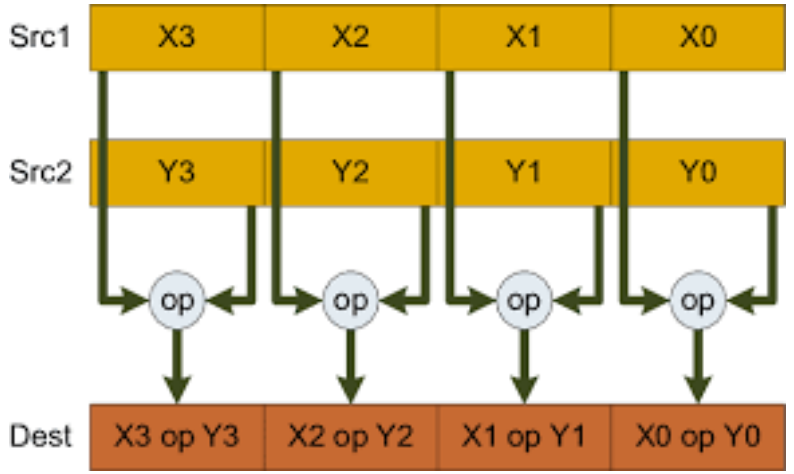
- Parameters management
- Wilson calculation
- Observable calculation
- χ^2 calculation





Optimizations

- Sparse matrix optimization
- Eigen
- SIMD optimization (AVX, MMX, etc.)
- Parallelization



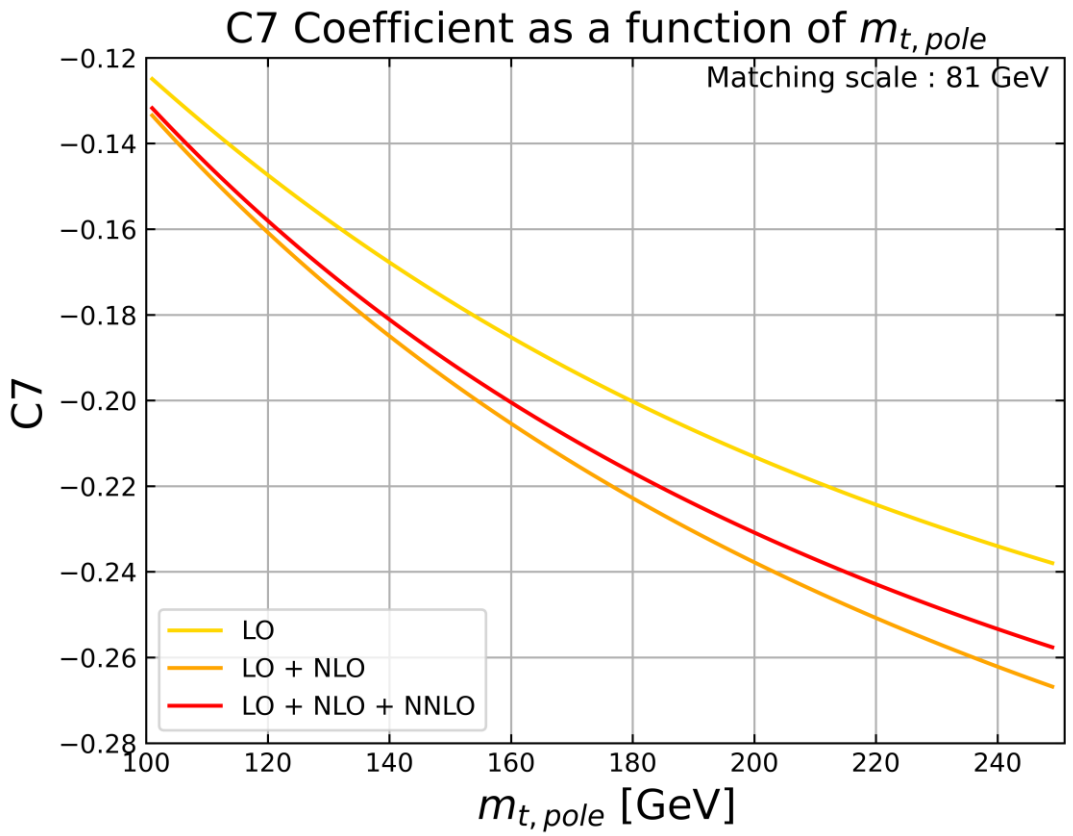
$$\vec{C}^{(0)\text{eff}}(\mu_b) = U^{(0)}\vec{C}^{(0)\text{eff}}(\mu_W)\,,$$

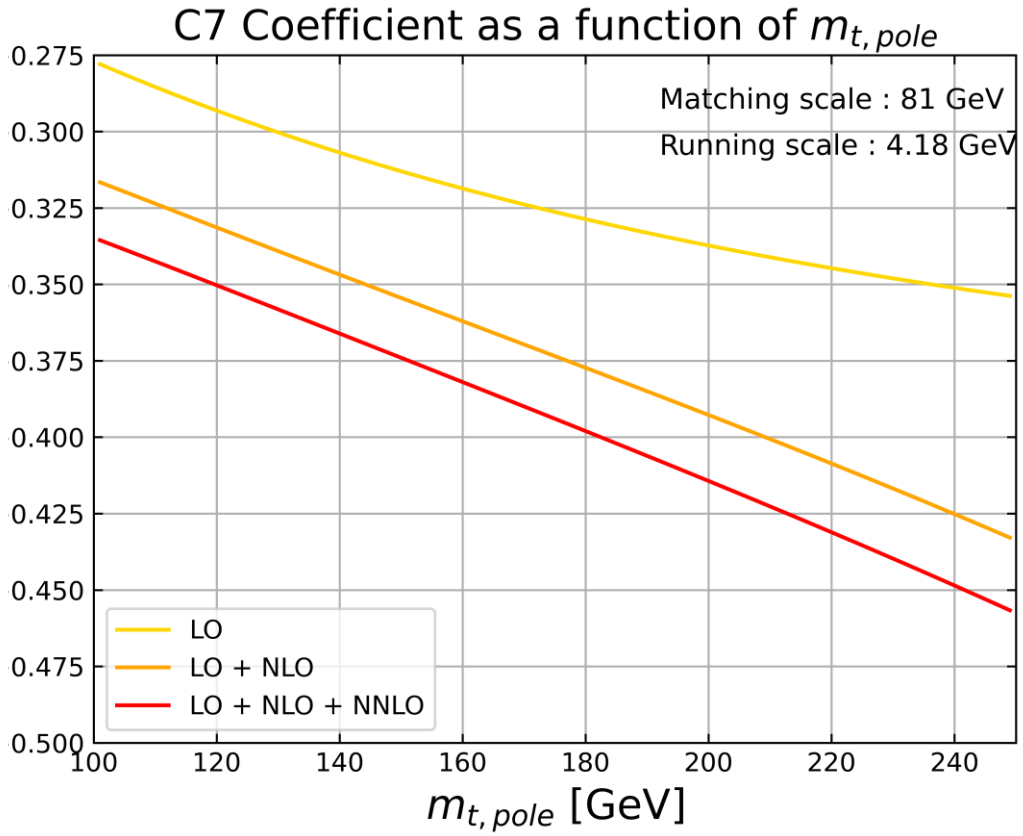
$$\vec{C}^{(1)\text{eff}}(\mu_b) = \eta\left[U^{(0)}\vec{C}^{(1)\text{eff}}(\mu_W) + U^{(1)}\vec{C}^{(0)\text{eff}}(\mu_W)\right]\,,$$

$$\vec{C}^{(2)\text{eff}}(\mu_b) = \eta^2\left[U^{(0)}\vec{C}^{(2)\text{eff}}(\mu_W) + U^{(1)}\vec{C}^{(1)\text{eff}}(\mu_W) + U^{(2)}\vec{C}^{(0)\text{eff}}(\mu_W)\right]$$

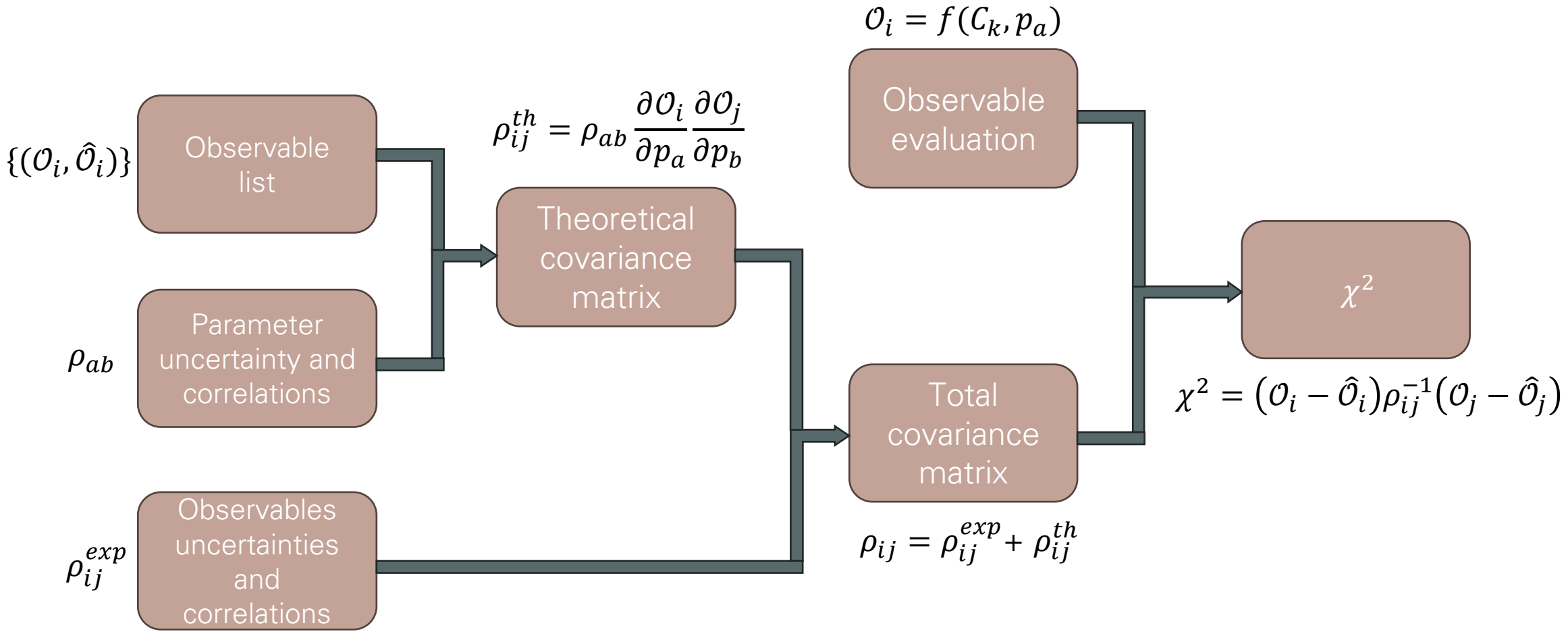
$$U_{kl}^{(n)} = \sum_{j=0}^{n}\sum_{i=1}^{9} m_{kli}^{(nj)}\eta^{a_i-j}\,.$$

Example plots

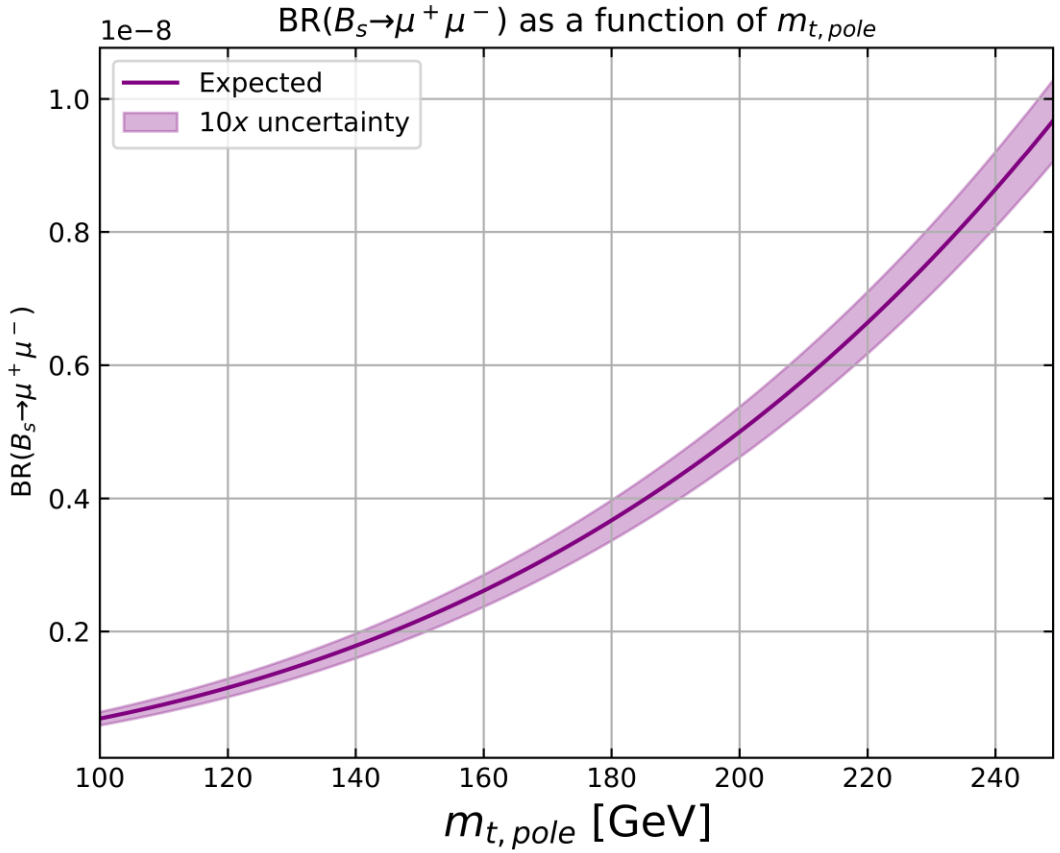




Uncertainty estimation and model validation



Example plots : Observable (Preliminary)



Future improvements

- Extension of statistical calculations to generic nuisance distributions / likelihoods
- Implement more flavour observables
- Optimization of MARTY interface
- Implement more EFTs in MARTY interface (currently only WET)
- Make interface more user-friendly

Thanks