MIXED QCD-EW CORRECTIONS TO NEUTRAL CURRENT DRELL YAN

Milano-Pisa PRIN meeting

06/10/2021

Automation of the calculation

UNIVERSITÀ
DEGLI STUDI
DI MILANO

Simone Devoto

STRUCTURE OF A LOOP COMPUTATION

Process definition →

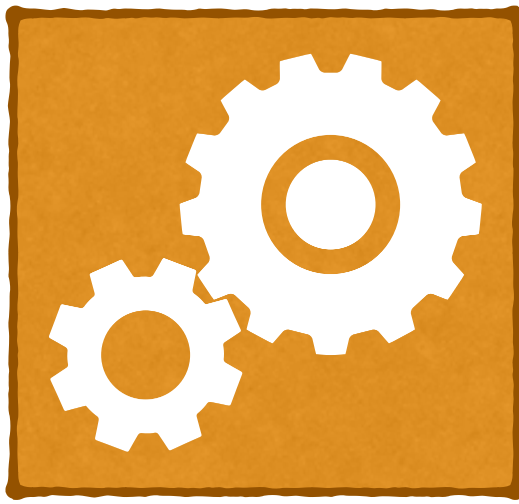
Feynman Amplitudes
↓
Computation of the interference terms
↓
Reduction to a set of Master Integrals
↓
Evaluation of the Master Integrals
↓
Subtraction of the UV poles (renormalisation)
↓
Subtraction of the IR poles
↓
Numerical evaluation in phase-space points
↓
Numerical grid

CAN WE AUTOMATISE THIS PROCEDURE?

- Internal cross check;
- Reproducible computation;
- Straightforward generalisation to different processes.

STRUCTURE OF A LOOP COMPUTATION

Process definition



Feynman Amplitudes

Computation of the interference terms

Reduction to a set of Master Integrals

Evaluation of the Master Integrals

Subtraction of the UV poles (renormalisation)

Subtraction of the IR poles

Numerical evaluation in phase-space points

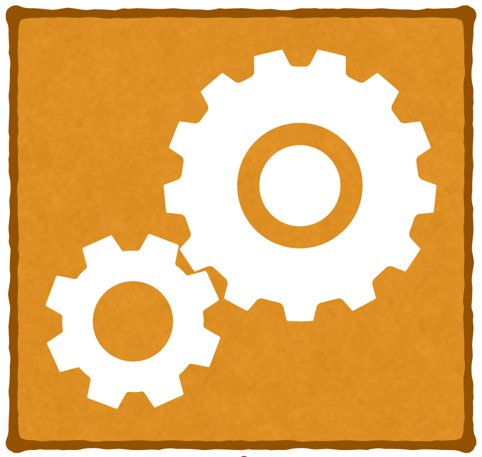
Numerical grid

CAN WE AUTOMATISE THIS PROCEDURE?

- Internal cross check;
- Reproducible computation;
- Straightforward generalisation to different processes.

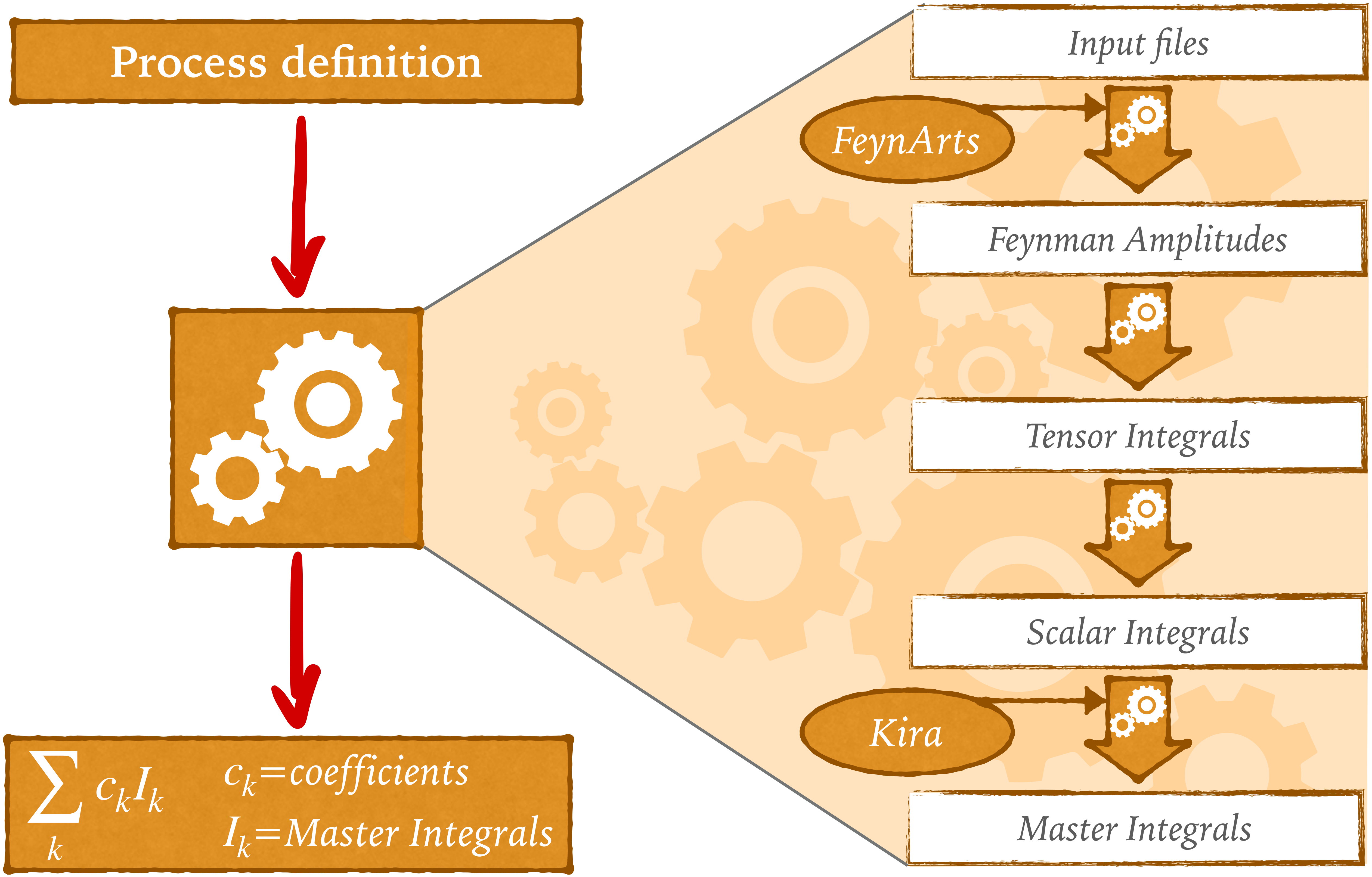
STRUCTURE OF A LOOP COMPUTATION

Process definition



$$\sum_k c_k I_k \qquad c_k = \text{coefficients} \qquad I_k = \text{Master Integrals}$$

STRUCTURE OF A LOOP COMPUTATION



GENERATION OF THE AMPLITUDES

FeynArts - *arXiv:hep-ph/0012260*

Mathematica package for the generation and visualisation of Feynman diagrams and amplitudes.

Input files → *FeynArts* → *Feynman Amplitudes*

- FeynArts work-flow:

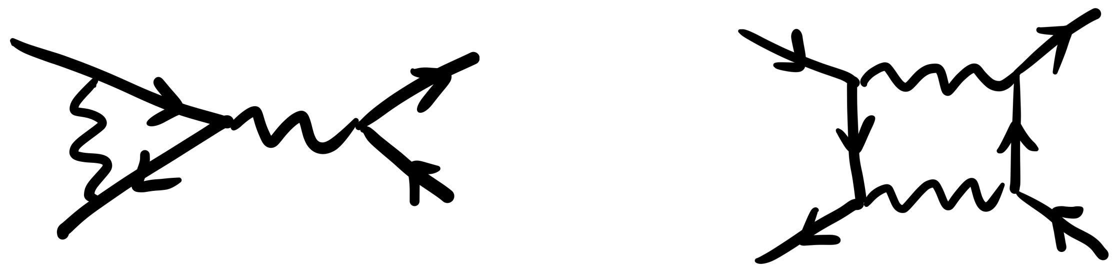
*Creation of the **topologies*** → *Insertion of **fields** into the topologies* → *Application of the Feynman rules to produce **Feynman amplitudes***

Output we are interested in!

- Several models are already defined, but full **customisation** is possible;
- We use the Standard Model with **Background Field Gauge**, which restores some QED-like Ward identities in the full Electroweak Standard Model.

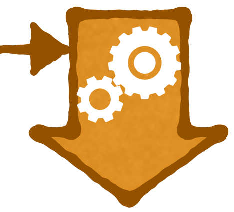
GENERATION OF THE AMPLITUDES

1Loop (EW/QCD)



Input files

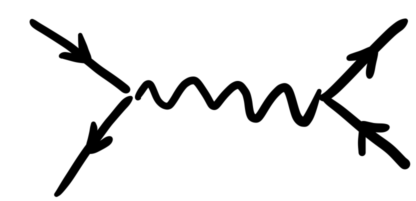
FeynArts



Feynman Amplitudes

FeynArts - *arXiv:hep-ph/0012260*

Born

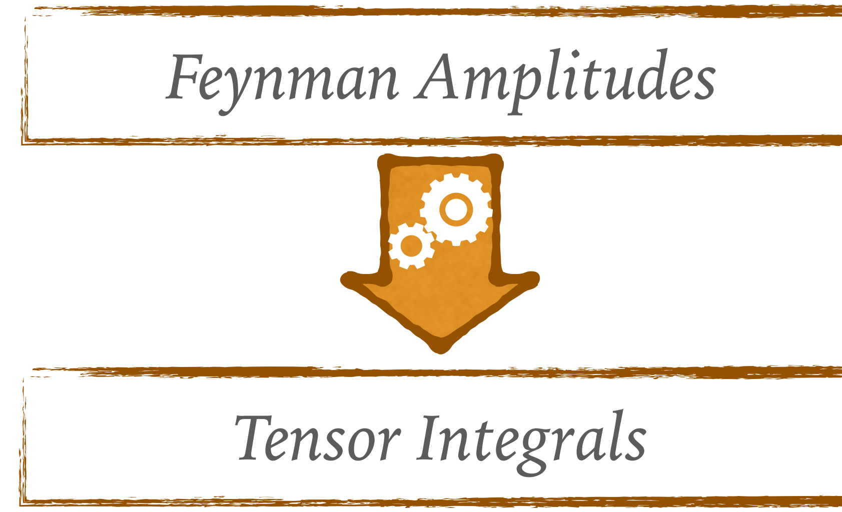


2Loop (QCDxEW)

COMPUTATION OF THE INTERFERENCE TERMS

Mathematica

In-house routines to automatically perform the Dirac and Lorentz Algebra in dimensional regularisation.



➤ The result can be written as a sum of **tensor integrals** in the form:

$$\int \prod_{i=1}^{L} dq_i \, \frac{q_1^{\mu_1}\ldots q_1^{\mu_j}\ldots q_L^{\mu_1}\ldots q_L^{\mu_l}}{P_1\ldots P_t}$$

where:

- $q_i \rightarrow$ loop momentum;
- $L \rightarrow$ number of independent loop momenta;
- $P_i = k_i^2 - m^2 \rightarrow$ inverse propagator, k_i being a linear combination of external and loop momenta.

<u>ISSUE</u>: Handling γ_5 in dimensional regularisation.

Object inherently 4D: how can we use it in arbitrary space-time dimension?

COMPUTATION OF THE INTERFERENCE TERMS

	ANTICOMMUTATION $\{\gamma_\mu, \gamma_5\} = 0$	CYCLICITY OF THE TRACE
't Hooft and Veltmann *Nucl. Phys. B 44 (1972) 189–213*	✗	✓
Kreimer et al. *Phys. Lett. B 237 (1990) 59–62*	✓	✗

Feynman Amplitudes

↓

Tensor Integrals

For neutral-current Drell Yan proven that at 2loops the two prescriptions yield:

- **different** scattering amplitudes;
- **same** finite corrections after subtraction.

(M. Heller, A. von Manteuffel, R. M. Schabinger and H. Spiesberger, arXiv:hep-ph/2012.05918)

Our procedure:

1. Use anticommutation relation, bring all γ_5 at the end of the Dirac trace;
2. Use $\gamma_5^2 = 1$, end up with zero or one γ_5 in each Dirac trace;
3. Replace the (single) leftover γ_5 with the relation: $\gamma_5 = \frac{i}{4!}\epsilon_{\mu\nu\rho\sigma}\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma$.

REDUCTION TO MASTER INTEGRALS (I)

Mathematica

*In-house routines to automatically write the tensor integrals in terms of **Integral families** of **Scalar integrals**, better suited for the application of reduction algorithms*

Tensor Integrals

Scalar Integrals

Tensor Integral

$$\int \prod_{i=1}^{L} d^n q_i \, \frac{q_1^{\mu_1} \ldots q_1^{\mu_j} \ldots q_L^{\mu_1} \ldots q_L^{\mu_l}}{P_1 \ldots P_t}$$

→

Scalar Integral

$$\int \prod_{i=1}^{L} d^n q_i \, \frac{1}{P_1^{\alpha_1} \ldots P_t^{\alpha_t} P_{t+1}^{\alpha_{t+1}} \ldots P_N^{\alpha_N}}$$

- Write all the scalar products involving loop momenta in terms of **inverse propagators**:

$$\begin{cases} P_0 = q^2 - m_0^2 \\ P_1 = (q - p_1)^2 - m_1^2 \end{cases} \Rightarrow \begin{cases} q^2 = P_0 + m_0^2 \\ q \cdot p_1 = 1/2\,(P_0 + m_1 - P_1 - m_2) \end{cases}$$

- To this end, it is necessary to introduce **auxiliary propagators** $P_{t+1}^{\alpha_t+1} \ldots P_N^{\alpha_N}$

REDUCTION TO MASTER INTEGRALS (I)

Mathematica

*In-house routines to automatically write the tensor integrals in terms of **Integral families** of **Scalar integrals**, better suited for the application of reduction algorithms*

Tensor Integrals

Scalar Integrals

Tensor Integral

$$\int \prod_{i=1}^{L} d^n q_i \, \frac{q_1^{\mu_1} \ldots q_1^{\mu_j} \ldots q_L^{\mu_1} \ldots q_L^{\mu_1}}{P_1 \ldots P_t}$$

Scalar Integral

$$\int \prod_{i=1}^{L} d^n q_i \, \frac{1}{P_1^{\alpha_1} \ldots P_t^{\alpha_t} P_{t+1}^{\alpha_{t+1}} \ldots P_N^{\alpha_N}}$$

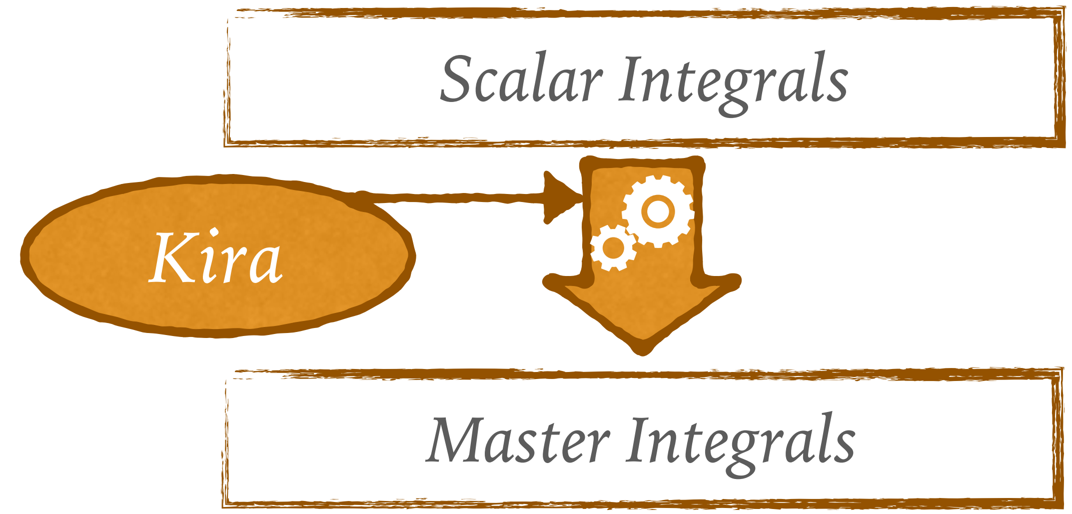
- A set of inverse propagators $\{P_1^{\alpha_1}, \ldots, P_N^{\alpha_N}\}$ defines an **integral family**;
- A scalar integral in an integral family is **uniquely identified** by the (positive or negative) powers of the exponents of the inverse propagators:

$$\int \prod_{i=1}^{L} dq_i \, \frac{1}{P_1^{\alpha_1} \ldots P_t^{\alpha_t} P_{t+1}^{\alpha_{t+1}} \ldots P_N^{\alpha_N}} = \mathrm{Family1}[\alpha_1, \ldots, \alpha_t, \alpha_{t+1}, \ldots, \alpha_N]$$

REDUCTION TO MASTER INTEGRALS (II)

Kira - *arXiv:hep-ph/1705.05610*
- arXiv:hep-ph/2008.06494

C++ reduction program implementing Laporta algorithm



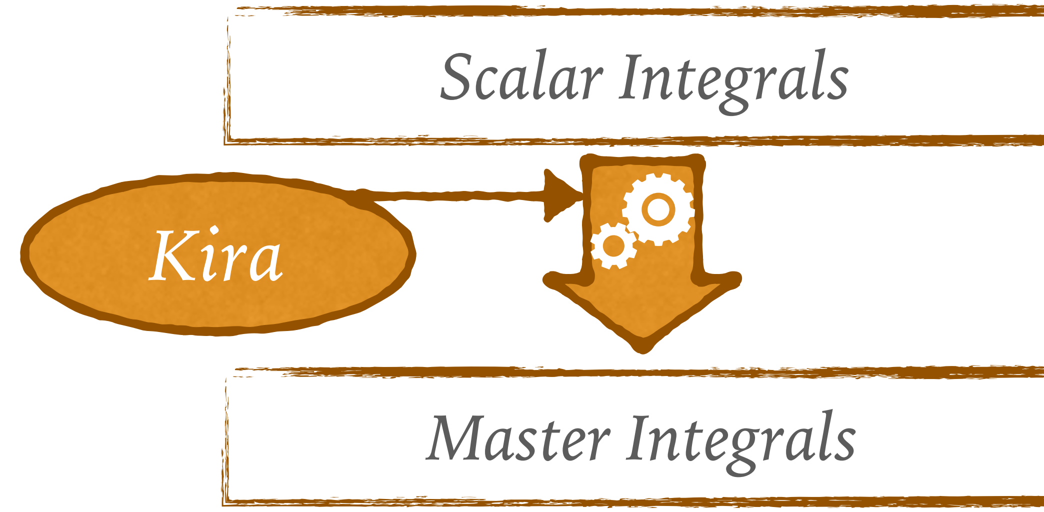
Scalar Integrals $\sum_{i=1}^{N} \hat{c}_i I_{S,i}$ → **Master Integrals** $\sum_{i=1}^{n} c_i I_{M,i}$ $\quad n \ll N$

- Expressions written as a sum of **scalar integrals** with the respective coefficient;
- All the scalar integrals are not independent: linear relations between them are provided by **integration by parts (IBP) identities**;
- We can reduce the large set of scalar integrals to a smaller set of **master integrals**;
- Kira applies **Laporta algorithm** to apply IBP identities to a set of integrals in order to find the linear relations between them.

REDUCTION TO MASTER INTEGRALS (II)

Kira - *arXiv:hep-ph/1705.05610*
- arXiv:hep-ph/2008.06494

C++ reduction program implementing Laporta algorithm



Integration by parts identities (IBP)

Gauss Theorem: $\int d^n q \frac{\partial}{\partial q^\mu} f^\mu(p_i^\mu, \cdots, q_i^\mu) = 0$

We choose, e.g.

$$f^\mu(p_i^\mu, \cdots, q_i^\mu) = \frac{p_1^\mu}{(q^2 - m_0^2)^{\alpha_0}((q+p_1)^2 - m_1^2)^{\alpha_1}}$$

$$\int d^n q \frac{\partial}{\partial q^\mu} f^\mu = -\int d^n q \left(\frac{2\alpha_0\, q \cdot p_1}{(q^2 - m_0^2)^{\alpha_0+1}((q+p_1)^2 - m_1^2)^{\alpha_1}} + \frac{2\alpha_1\, (q+p_1)\cdot p_1}{(q^2 - m_0^2)^{\alpha_0}((q+p_1)^2 - m_1^2)^{\alpha_1+1}} \right) = 0$$

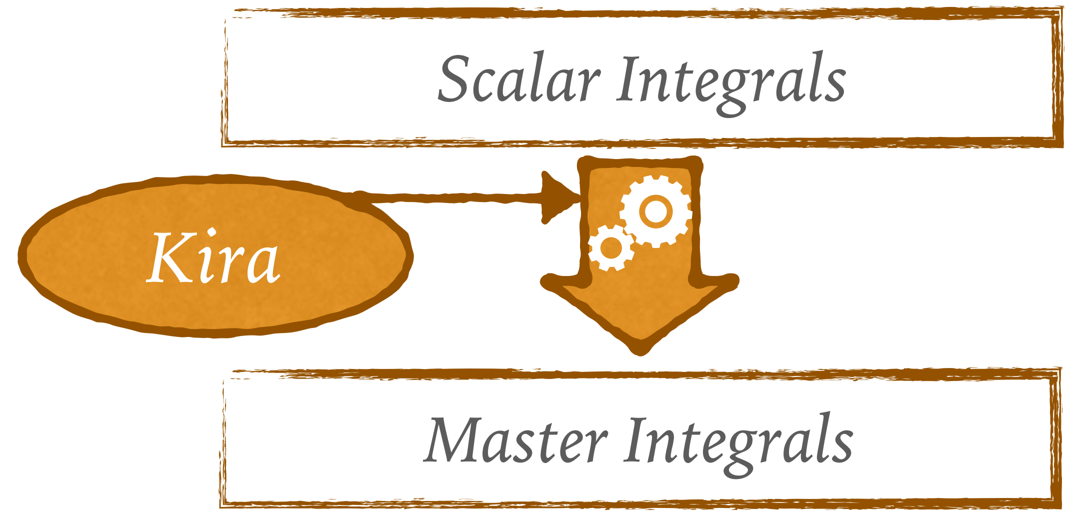
Relation between $F[\alpha_0 + 1, \alpha 1, 0, \ldots]$ and $F[\alpha_0, \alpha 1 + 1, 0, \ldots]$!

*By using different IBP relations it is possible to define **ladder operators** to rise and lower the indices in an integral family!*

REDUCTION TO MASTER INTEGRALS (II)

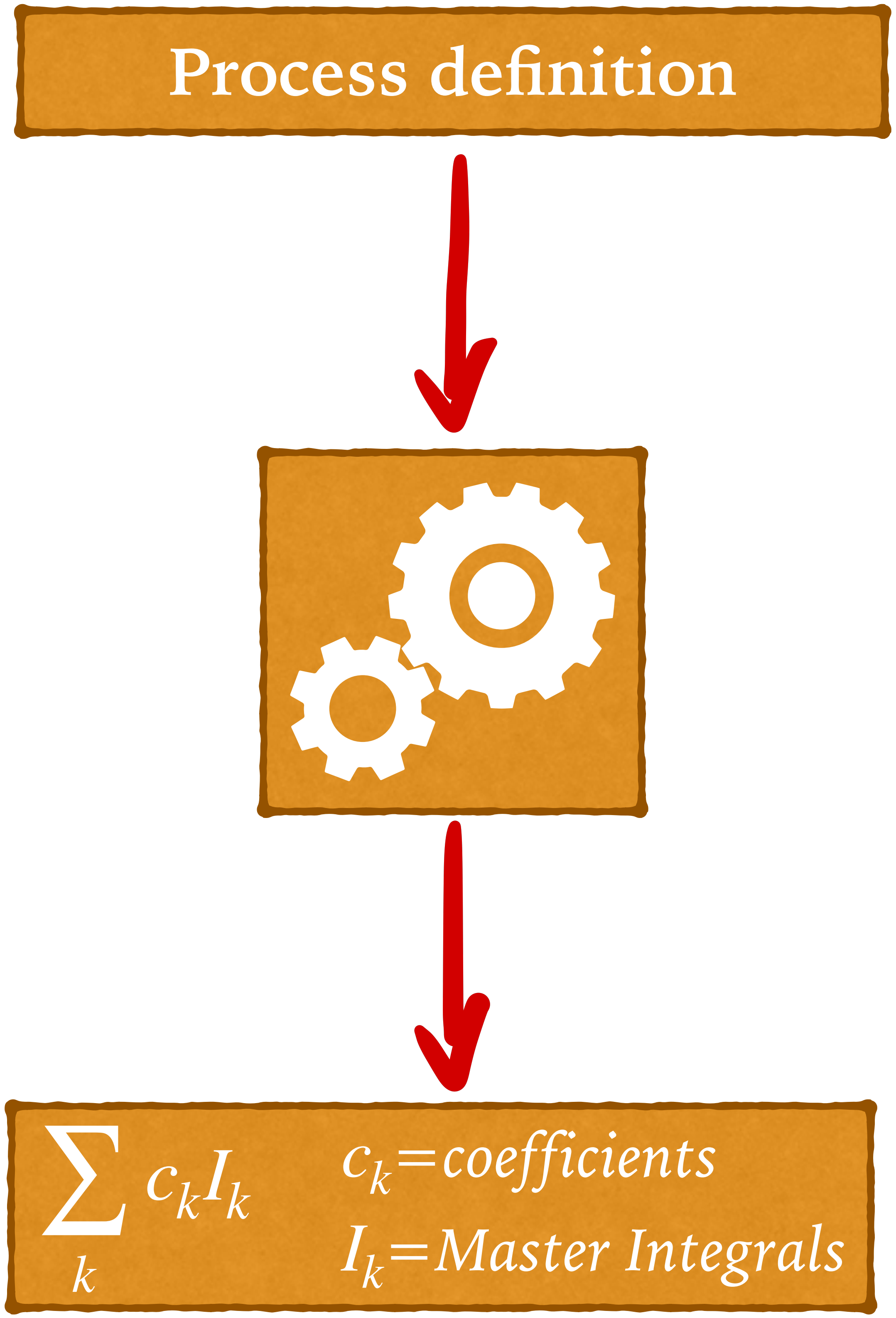
Kira - *arXiv:hep-ph/1705.05610*
- arXiv:hep-ph/2008.06494

C++ reduction program implementing Laporta algorithm



- The **ladder operators** would in principle generate an **infinite system of equations** for an infinite number of scalar integrals!
- **Kira** implements **Laporta algorithm**:
  - Isolates a set of scalar integrals (**seed integrals**) defined by maximum value of the sum of positive and negative propagator powers;
  - Orders the integrals by **complexity** (preferring lower propagator powers);
  - Generates a **finite system** of IBP equations;
  - Removes the **linearly dependent** relations;
  - Extracts a **set of master integrals** (the user can provide a preference);
  - Uses a **Gauss-type forward elimination algorithm** to solve the system of IBP equations and provide an expression of all the seed integrals in terms of master integrals.

FINAL RESULTS



- Kira produces as an output a **list of substitutions** for the scalar integrals provided as an input;
- Kira output can be naturally exported as a list of substitutions in Mathematica format;
- In-house Mathematica routine generates the final result:
  - **Imports** the substitutions to Master Integrals from Kira, **applies** them to the expressions in terms of scalar Integrals;
  - Performs a final **simplification** of the coefficients of the Master Integrals.

SUMMARY AND OUTLOOK

Process definition

$$\sum_k c_k I_k \qquad c_k = \text{coefficients} \qquad I_k = \text{Master Integrals}$$

No overlap at any stage of the computation, no FeynArts, no Kira!

*Neutral current-DY results **cross-checked** with **independent** computation from Narayan!*

*Can we **automatise extra steps** of the computation? (Renormalisation, subtraction…)*

***Process independent**, needs only definition from the user of the process and of the integral families!*

***Easy generalisation** to charged current, **additional challenges** from the evaluation of the new masters!*

Additional mass scale in the loops (WZ diagrams). Extra integral family? Mass expansion?

SUMMARY AND OUTLOOK

Process definition

$$\sum_k c_k I_k$$

c_k=*coefficients*

I_k=*Master Integrals*

No overlap at any stage of the computation, no FeynArts, no Kira!

Can we ***automatise extra steps*** *of the computation? (Renormalisation, subtraction…)*

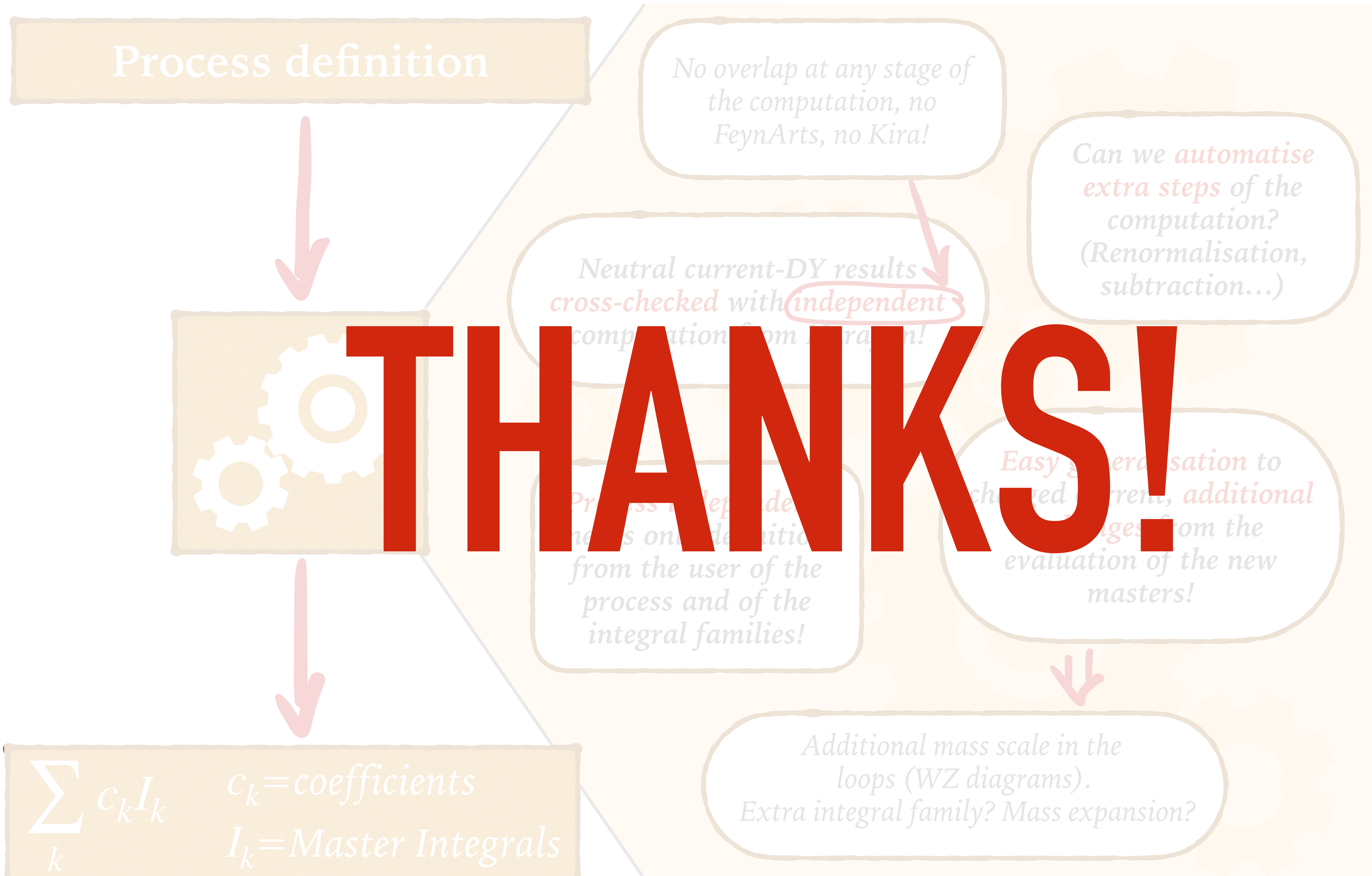
Neutral current-DY results ***cross-checked*** *with* ***independent*** *computation from Narayan!*

Process independent*, needs only definition from the user of the process and of the integral families!*

Easy generalisation *to charged current,* ***additional challenges*** *from the evaluation of the new masters!*

Additional mass scale in the loops (WZ diagrams). Extra integral family? Mass expansion?

SUMMARY AND OUTLOOK



THANKS!