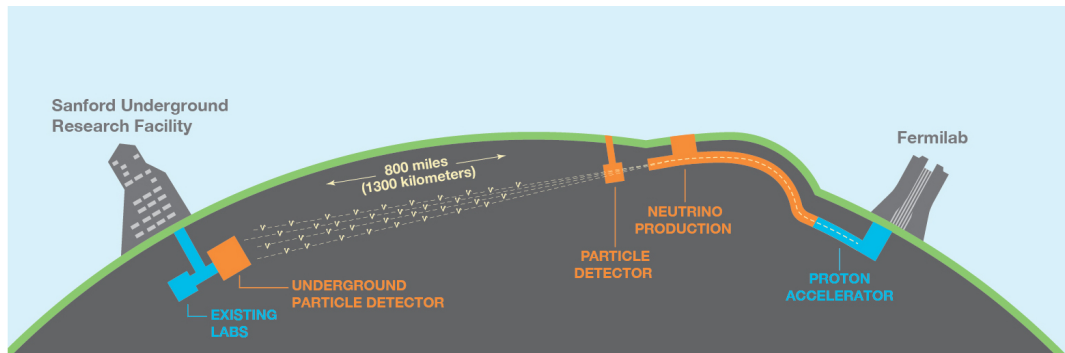




Automating the Leptonic Tensor

Joshua Isaacson
24 August 2020

Motivation: DUNE and T2HK



- Entering Precision Neutrino Physics with DUNE and T2HK
- Incorporating BSM into generators will be required

Motivation: DUNE and T2HK

WHITE PAPER ON NEW OPPORTUNITIES AT THE NEXT-GENERATION NEUTRINO EXPERIMENTS (PART 1: BSM NEUTRINO PHYSICS AND DARK MATTER)

C.A. ARGÜELLES¹, A.J. AURISANO², B. BATELL³, J. BERGER³, M. BISHAI⁴, T. BOSCHI⁵, N. BYRNES⁶,
A. CHATTERJEE⁶, A. CHODOS⁶, T. COAN⁷, Y. CUI⁸, A. DE GOUVÊA^{* 9}, P.B. DENTON⁴,
A. DE ROECK^{* 10}, W. FLANAGAN¹¹, D.V. FORERO¹², R.P. GANDRAJULA¹³, A. HATZIKOUTELIS¹⁴,
M. HOSTERT¹⁵, B. JONES⁶, B.J. KAYSER¹⁶, K.J. KELLY¹⁶, D. KIM¹⁷, J. KOPP^{10,18}, A. KUBIK¹⁹,
K. LANG²⁰, I. LEPETIC²¹, P.A.N. MACHADO¹⁶, C.A. MOURA²², F. OLNES⁶, J.C. PARK²³,
S. PASCOLI¹⁵, S. PRAKASH¹², L. ROGERS⁶, I. SAFA²⁴, A. SCHNEIDER²⁴, K. SCHOLBERG²⁵, S. SHIN^{26,27},
I.M. SHOEMAKER²⁸, G. SINEV²⁵, B. SMITHERS⁶, A. SOUSA^{* 2}, Y. SUI²⁹, V. TAKHISTOV³⁰,
J. THOMAS³¹, J. TODD², Y.-D. TSAI^{16,32}, Y.-T. TSAI³³, J. YU^{* 6}, AND C. ZHANG⁴

5.1 BSM Physics Tools and Prospect for Future Improvements

In order to accurately assess the capabilities of current and future neutrino experiments to discover BSM physics and to develop appropriate analysis strategies, **accurate simulation of both SM and BSM phenomenology is required**. This proves challenging, particularly in the energy regime most relevant to accelerator and atmospheric neutrino experiments, where strongly coupled QCD effects are present. Simulation of the beam-generated fluxes, non-standard-neutrino and dark matter interactions, and atmospheric neutrino backgrounds are all subject to large uncertainties. In this section, we provide an overview of the current status of tools for simulating these phenomena, discuss areas that are challenging, and assess the potential for future improvement.

[<https://arxiv.org/abs/1907.08311>]

Motivation: Theory

Summary of Workshop on Common Neutrino Event Generator Tools

2.5.2 Factors to consider for interface design

While our goal should be to standardize the interface as much as possible, the workshop raised several issues that should be considered when evaluating possible approaches. These issues are summarized below:

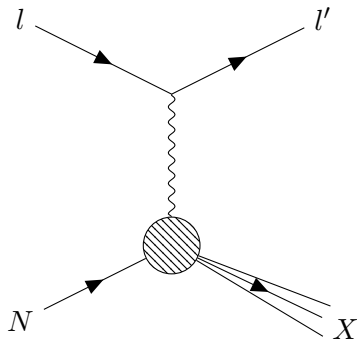
4. **Human factors** Our interface should be designed for ease of use, and should consider the skills and limitations of the theorists likely to use it. It was pointed out that many theorists are PhD students or postdoctoral researchers working on limited-term contracts. To fit in with this way of working, **it should be possible to develop, implement, and test a model against data on timescales of the order of a year.** We must also bear in mind that many theorists are not primarily programmers, and **that models may be developed using tools, such as Mathematica, that are not natively compatible with the languages used in generator software.** **If we restricted ourselves to an interface in a particular programming language, we could severely limit the accessibility of the interface to new models.**

[<https://arxiv.org/abs/2008.06566>]

Outline

- Introduction
- LHC Tools
- Current Status
- Possible Shortcomings
- Future Steps

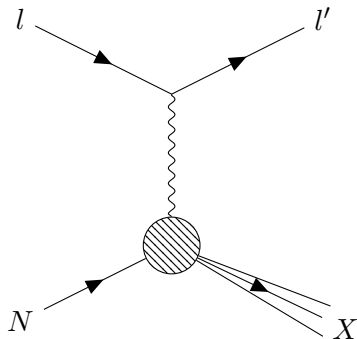
Leptonic and Hadronic Tensor



Observations:

- Nuclear physics calculations are hard
- Calculating arbitrary perturbative diagrams is a solved problem
- BSM effects of interest for DUNE and T2HK only weakly couple to quarks and gluons \Rightarrow no corrections to the nuclear physics

Leptonic and Hadronic Tensor



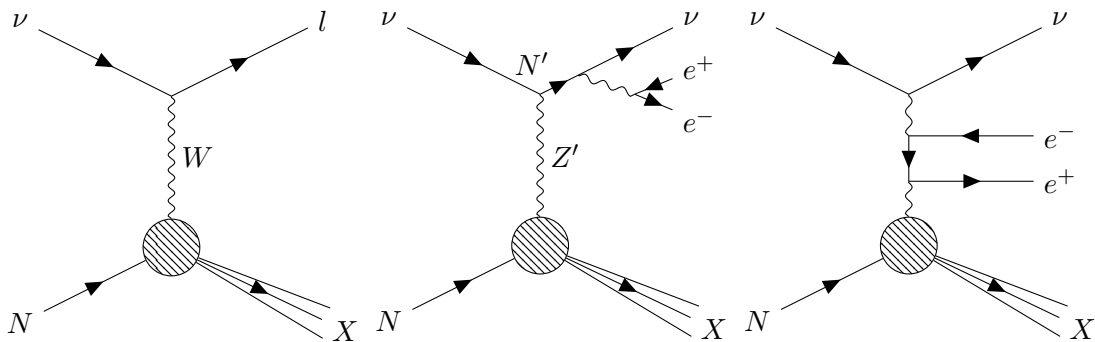
Observations:

- Nuclear physics calculations are hard
- Calculating arbitrary perturbative diagrams is a solved problem
- BSM effects of interest for DUNE and T2HK only weakly couple to quarks and gluons \Rightarrow no corrections to the nuclear physics

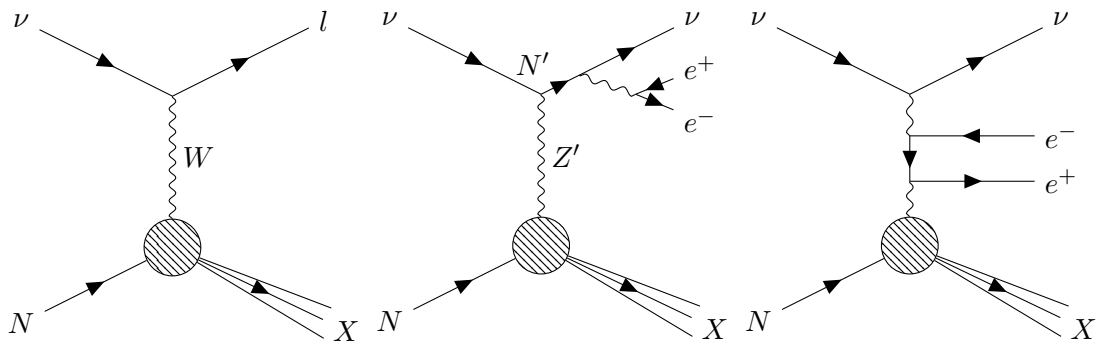
Question

Can we use these observations to automate Beyond the Standard Model Physics?

Leptonic and Hadronic Tensor



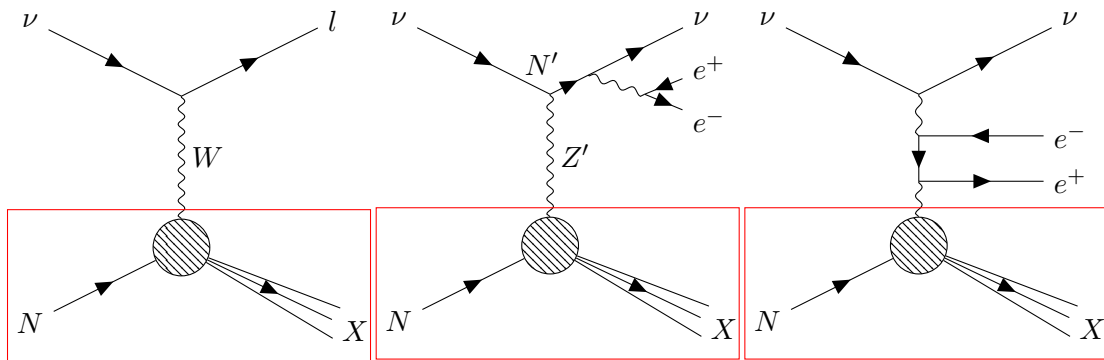
Leptonic and Hadronic Tensor



Question

What do all the diagrams above have in common?

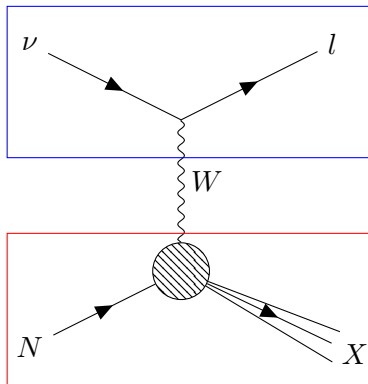
Leptonic and Hadronic Tensor



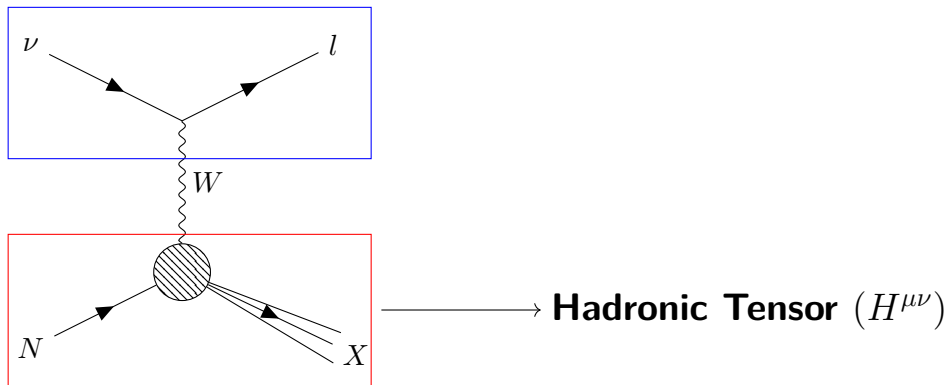
Question

What do all the diagrams above have in common?

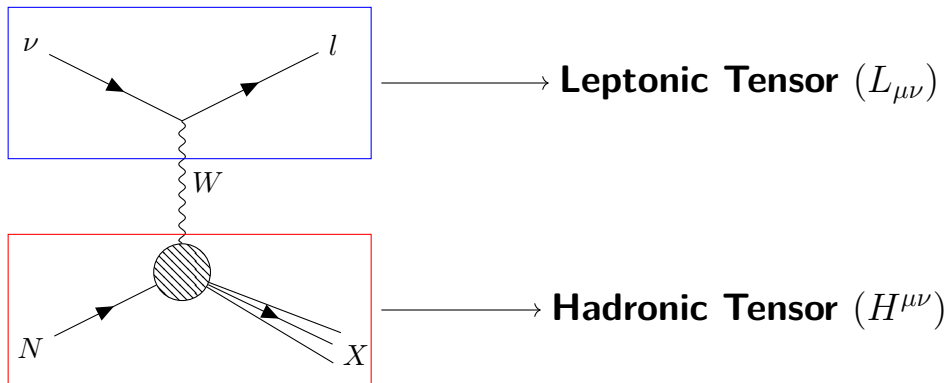
Leptonic and Hadronic Tensor



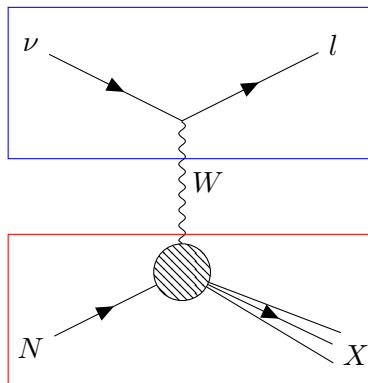
Leptonic and Hadronic Tensor



Leptonic and Hadronic Tensor



Leptonic and Hadronic Tensor

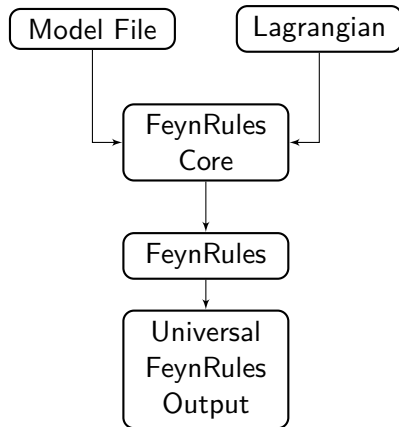


Notes:

- Leptonic tensor only contains perturbative physics.
- Can use LHC tools to calculate Leptonic tensor
- Hadronic tensor is difficult, but event generators have these calculations implemented already.

FeynRules

- *Mathematica* Program
- Takes model file and Lagrangian as input
- Calculates the Feynman rules
- Outputs in Universal FeynRules Output (UFO) format



[[arXiv:0806.4194](https://arxiv.org/abs/0806.4194), [arXiv:1310.1921](https://arxiv.org/abs/1310.1921)]

Universal FeynRules Output (UFO)

- Python output files
- Contains model-independent files and model-dependent files
- Contains all information to calculate any tree level matrix element
- Has parameter file to adjust model parameters to scan allowed regions

Example QED ($e^+e^-\gamma$ Vertex):

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \bar{\psi}(iD^\mu\gamma_\mu - m)\psi$$

$$V_{e^+e^-\gamma} = ie\gamma^\mu = \gamma \text{ ~~~~~ } \begin{array}{l} \nearrow \\ \searrow \end{array}$$

[arXiv:1108.2040]

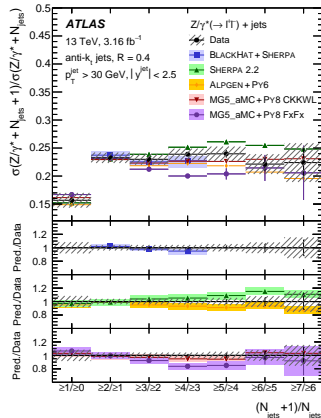
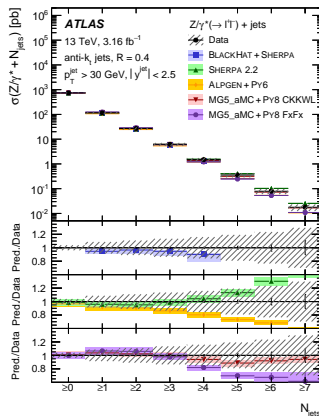
Universal FeynRules Output (UFO)

Example for photon-electron vertex

```
e__minus__ = Particle(pdg_code = 11, name = 'e-', antiname = 'e+',
                      spin = 2, color = 1, mass = Param.ZERO,
                      width = Param.ZERO, texname = 'e-',
                      antitexname = 'e+', charge = -1,
                      GhostNumber = 0, LeptonNumber = 1,
                      Y = 0)
V_77 = Vertex(name = 'V_77',
              particles = [ P.e__plus__, P.e__minus__, P.a ],
              color = [ '1' ], lorentz = [ L.FFV1 ],
              couplings = {(0,0):C.GC_3})
FFV1 = Lorentz(name = 'FFV1', spins = [ 2, 2, 3 ],
               structure = 'Gamma(3,2,1)')
GC_3 = Coupling(name = 'GC_3', value = '-(ee*complex(0,1))',
                order = {'QED':1})
```

Tree Level Matrix Element Generators

- ALPGEN [arXiv:hep-ph/0206293]
- AMEGIC [arXiv:hep-ph/0109036]
- COMIX [arXiv:0808.3674]
- CALCHEP [arXiv:1207.6082]
- HERWIG [arXiv:0803.0883]
- MADGRAPH [arXiv:1405.0301]
- WHIZARD [arXiv:0708.4233]
- etc.



[arXiv:1702.05725]

Recursive Matrix Element Generation

$$\mathcal{J}_\alpha(\rho) = P_\alpha(\rho) \sum_{\mathcal{V}_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\rho)} \mathcal{S}(\rho_1, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\rho_1, \rho_2) \mathcal{J}_{\alpha_1}(\rho_1) \mathcal{J}_{\alpha_2}(\rho_2)$$

$$A(\rho) = \mathcal{J}_{\alpha_n}(n) \frac{1}{P_{\bar{\alpha}_n}(\rho \setminus n)} \mathcal{J}_{\bar{\alpha}}(\rho \setminus n)$$

Brends-Giele Recursion

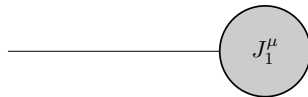
- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(n^a)$

[Nucl. Phys. B306(1988), 759]

Recursive Matrix Element Generation

$$\mathcal{J}_\alpha(\rho) = P_\alpha(\rho) \sum_{\nu_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\rho)} \mathcal{S}(\rho_1, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\rho_1, \rho_2) \mathcal{J}_{\alpha_1}(\rho_1) \mathcal{J}_{\alpha_2}(\rho_2)$$

$$A(\rho) = \mathcal{J}_{\alpha_n}(n) \frac{1}{P_{\bar{\alpha}_n}(\rho \setminus n)} \mathcal{J}_{\bar{\alpha}}(\rho \setminus n)$$



Brends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(n^a)$

[Nucl. Phys. B306(1988), 759]

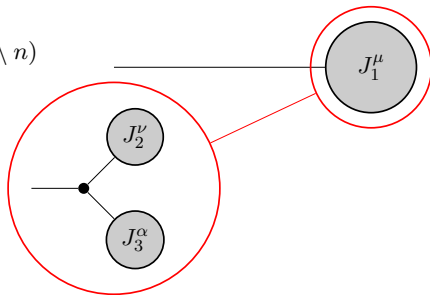
Recursive Matrix Element Generation

$$\mathcal{J}_\alpha(\rho) = P_\alpha(\rho) \sum_{\nu_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\rho)} \mathcal{S}(\rho_1, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\rho_1, \rho_2) \mathcal{J}_{\alpha_1}(\rho_1) \mathcal{J}_{\alpha_2}(\rho_2)$$

$$A(\rho) = \mathcal{J}_{\alpha_n}(n) \frac{1}{P_{\bar{\alpha}_n}(\rho \setminus n)} \mathcal{J}_{\bar{\alpha}}(\rho \setminus n)$$

Brends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(n^a)$



[Nucl. Phys. B306(1988), 759]

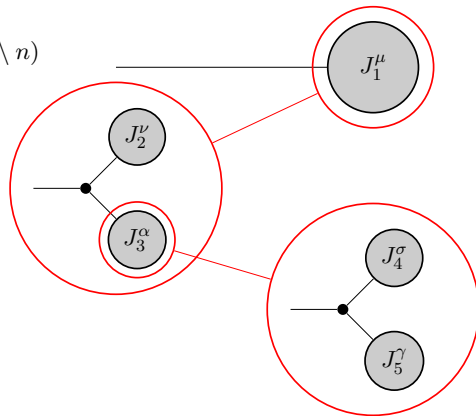
Recursive Matrix Element Generation

$$\mathcal{J}_\alpha(\rho) = P_\alpha(\rho) \sum_{\nu_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\rho)} \mathcal{S}(\rho_1, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\rho_1, \rho_2) \mathcal{J}_{\alpha_1}(\rho_1) \mathcal{J}_{\alpha_2}(\rho_2)$$

$$A(\rho) = \mathcal{J}_{\alpha_n}(n) \frac{1}{P_{\bar{\alpha}_n}(\rho \setminus n)} \mathcal{J}_{\bar{\alpha}}(\rho \setminus n)$$

Brends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(n^a)$



[Nucl. Phys. B306(1988), 759]

Recursive Matrix Element Generation

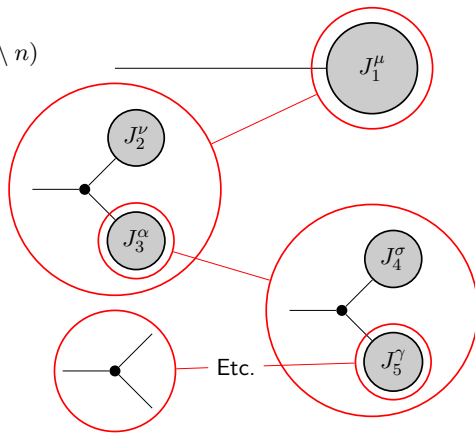
$$\mathcal{J}_\alpha(\rho) = P_\alpha(\rho) \sum_{\nu_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\rho)} \mathcal{S}(\rho_1, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\rho_1, \rho_2) \mathcal{J}_{\alpha_1}(\rho_1) \mathcal{J}_{\alpha_2}(\rho_2)$$

$$A(\rho) = \mathcal{J}_{\alpha_n}(n) \frac{1}{P_{\bar{\alpha}_n}(\rho \setminus n)} \mathcal{J}_{\bar{\alpha}}(\rho \setminus n)$$

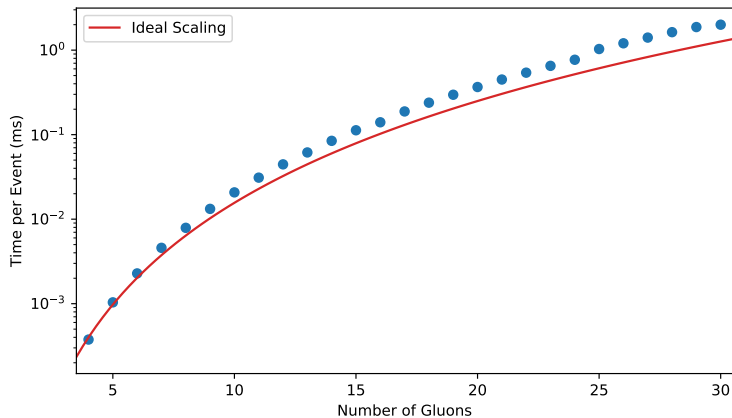
Brends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(n^a)$

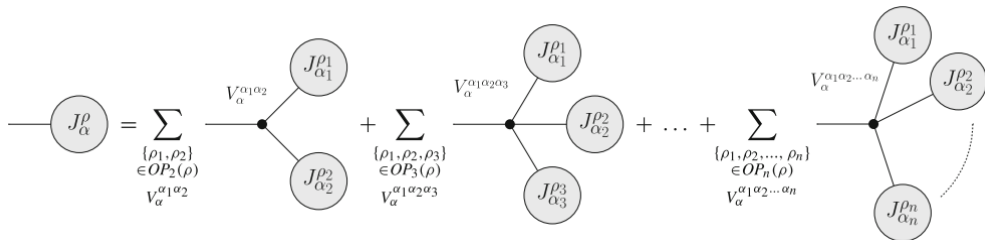
[Nucl. Phys. B306(1988), 759]



Brends-Giele Scaling



Recursion for BSM



$$\begin{aligned}
 \mathcal{J}_\alpha^\rho &= P_\alpha^\rho \sum_{m=2}^n \sum_{\substack{\{\rho_1, \dots, \rho_m\} \\ \in OP_m(\rho)}} \sum_{V_\alpha^{\alpha_1, \dots, \alpha_m}} \mathcal{S}(\rho_1, \dots, \rho_m) \\
 &\quad \times V_\alpha^{\alpha_1 \dots \alpha_m} \mathcal{J}_{\alpha_1}^{\rho_1} \dots \mathcal{J}_{\alpha_m}^{\rho_m}
 \end{aligned}$$

[arXiv:1412.6478]

Handling Lorentz Structures

```

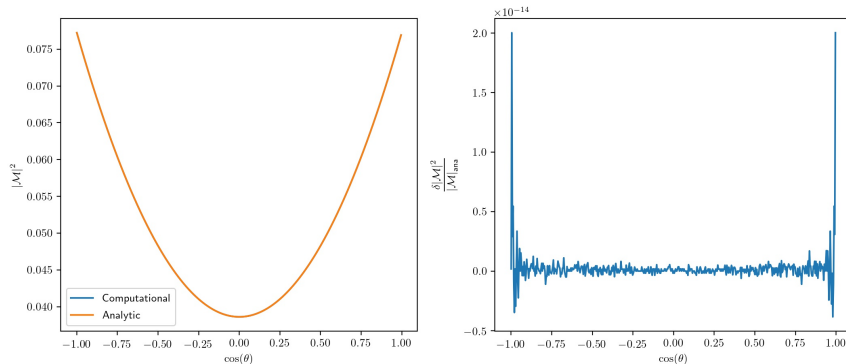
'Gamma(3,2,1)':
Tensor ( [[[ [ 0.+0.j  0.+0.j  1.+0.j  0.+0.j ]
             [ 0.+0.j  0.+0.j  0.+0.j  1.+0.j ]
             [ 1.+0.j  0.+0.j  0.+0.j  0.+0.j ]
             [ 0.+0.j  1.+0.j  0.+0.j  0.+0.j ] ]
          [[ [ 0.+0.j  0.+0.j  0.+0.j  1.+0.j ]
             [ 0.+0.j  0.+0.j  1.+0.j  0.+0.j ]
             [ 0.+0.j -1.+0.j  0.+0.j  0.+0.j ]
             [-1.+0.j  0.+0.j  0.+0.j  0.+0.j ] ]
          [[ [ 0.+0.j  0.+0.j  0.+0.j -0.-1.j ]
             [ 0.+0.j  0.+0.j  0.+1.j  0.+0.j ]
             [ 0.+0.j  0.+1.j  0.+0.j  0.+0.j ]
             [-0.-1.j  0.+0.j  0.+0.j  0.+0.j ] ]
          [[ [ 0.+0.j  0.+0.j  1.+0.j  0.+0.j ]
             [ 0.+0.j  0.+0.j  0.+0.j -1.+0.j ]
             [-1.+0.j  0.+0.j  0.+0.j  0.+0.j ]
             [ 0.+0.j  1.+0.j  0.+0.j  0.+0.j ] ] ] ],
          (Index(3, True), Index(2, False), Index(1, False)))

```

- Parse UFO strings using LARK (<https://github.com/lark-parser/lark>)
- Develop python structures to perform calculations once at code start-up
- Separate spin and lorentz indices to sum and contract over correctly

Basic Validation

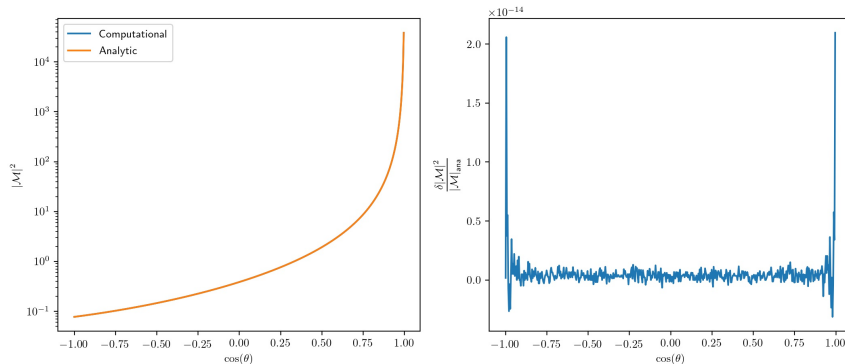
$e^+e^- \rightarrow \mu^+\mu^-$
Comparison of amplitudes for $\phi = 0.34$



$$|\mathcal{M}|^2 = 8e^4 \left(\frac{t^2 + u^2}{s^2} \right)$$

Basic Validation

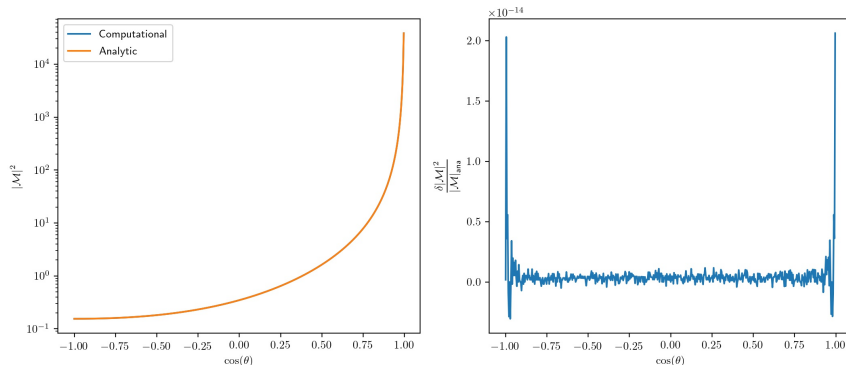
$e^- \mu^- \rightarrow e^- \mu^-$
Comparison of amplitudes for $\phi = 2.67$



$$|\mathcal{M}|^2 = 8e^4 \left(\frac{s^2 + u^2}{t^2} \right)$$

Basic Validation

$e^+e^- \rightarrow e^+e^-$
Comparison of amplitudes for $\phi = 0.00$



$$|\mathcal{M}|^2 = 8e^4 \left(\frac{u^2 + s^2}{t^2} + \frac{2u^2}{st} + \frac{u^2 + t^2}{s^2} \right)$$

Phase Space Generation

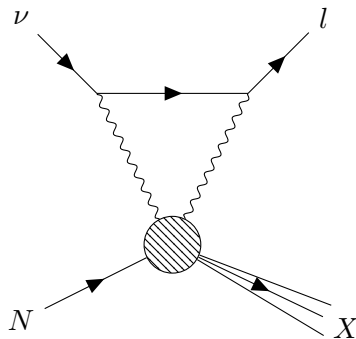
$$d\Phi_n(a, b; 1, \dots, n) = \delta^{(4)}\left(p_a + p_b - \sum_{i=1}^n p_i\right) \left[\prod_{i=1}^n \frac{d^4 p_i}{(2\pi)^3} \delta(p_i^2 - m_i^2) \Theta(p_{i0}) \right]$$

Algorithms for phase space generation

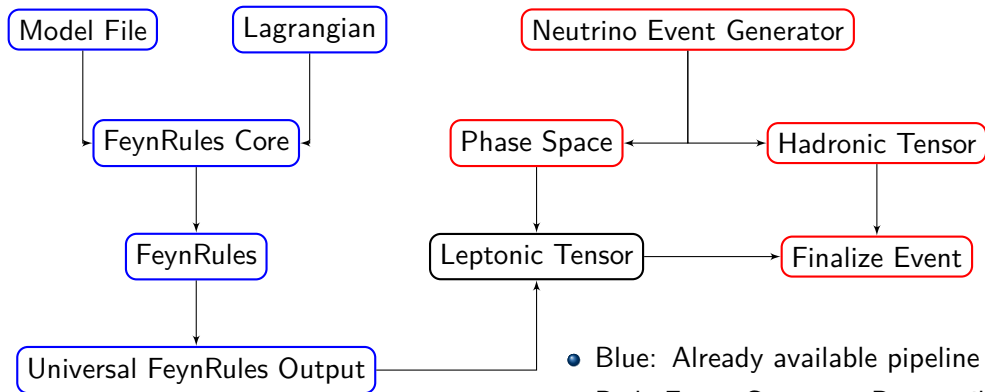
- RAMBO [[Comput. Phys. Commun. 40\(1986\) 359](#)]
- Multi-channel techniques [[hep-ph/9405257](#)]
- Recursive Phase Space [[arXiv:0808.3674](#)]

Leptonic and Hadronic Tensor: Possible issues

- Which part is responsible for phase space generation
- Handling of g_L and g_R (g_V and g_A) for nuclear side
- Can't handle multiple boson exchanges
- More issues will probably arise



Future / Proposed Interface



- Blue: Already available pipeline
- Red: Event Generator Responsibility

Conclusions

- BSM important for the next generation neutrino experiments
- Requires automating theory calculations
- Take advantage of LHC tools
- First proof of principle for test processes
- Discussion of next steps
- Proposal for separation of responsibilities