



Automating Leptonic Current and Phase Space

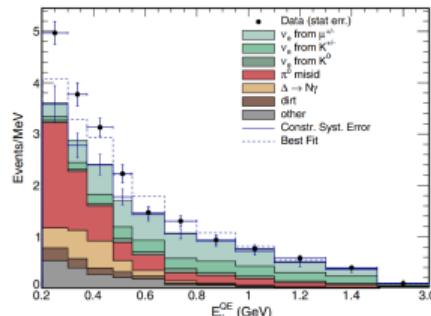
Joshua Isaacson

In Collaboration with Stefan Höche, Diego Lopez Gutierrez, and Noemi Rocco

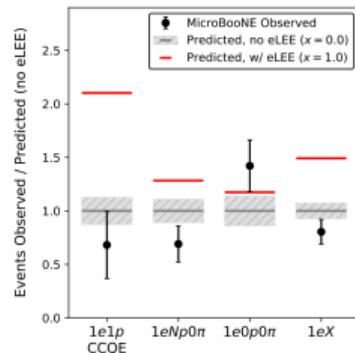
Based on arxiv:2110.15319

11 November 2021

Motivation: MiniBooNE and MicroBooNE

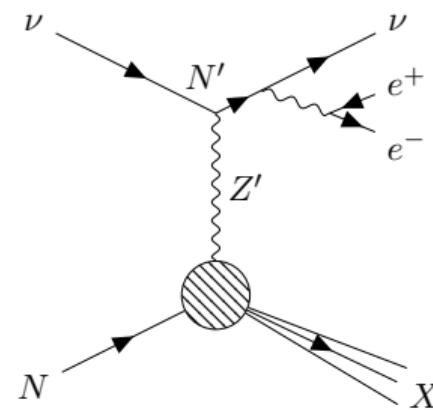


[PRL 121, 221801]

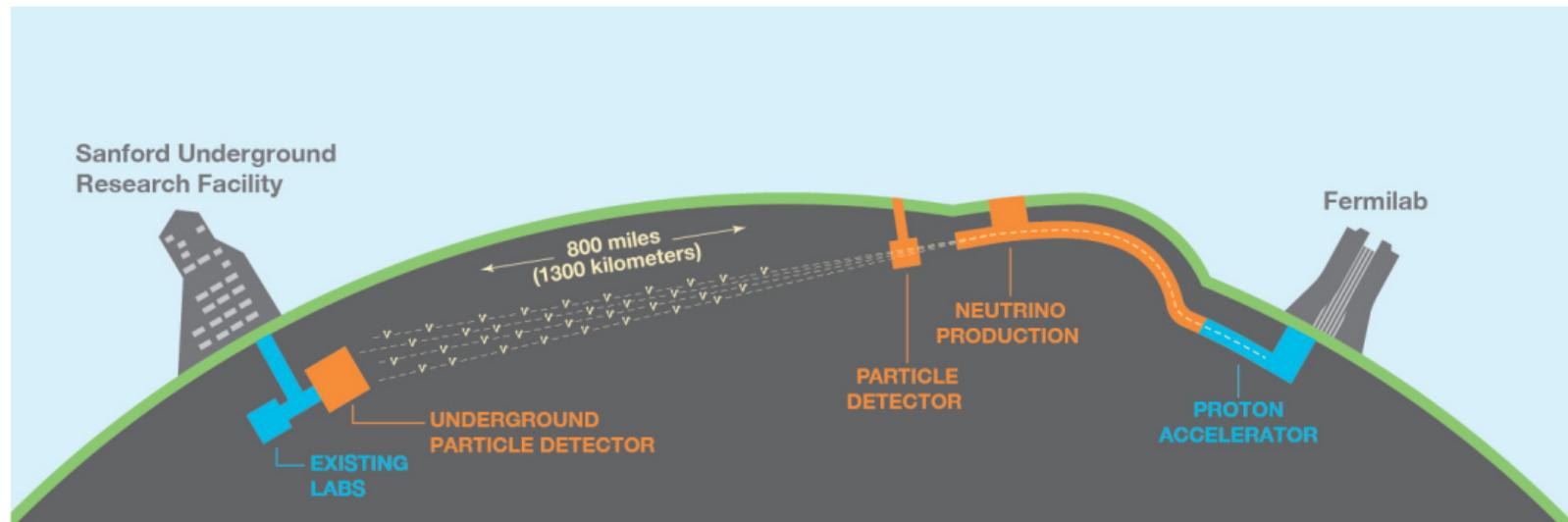


[arXiv:2110.14054]

- MiniBooNE sees excess of events
- MicroBooNE does not see excess of single electron events
- Excess can be from multiple lepton final states
- Event generators can not simulate these processes



Motivation: DUNE and HyperK



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

Motivation: DUNE and HyperK

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 GOUVÉA * 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. OLNESS⁶, 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

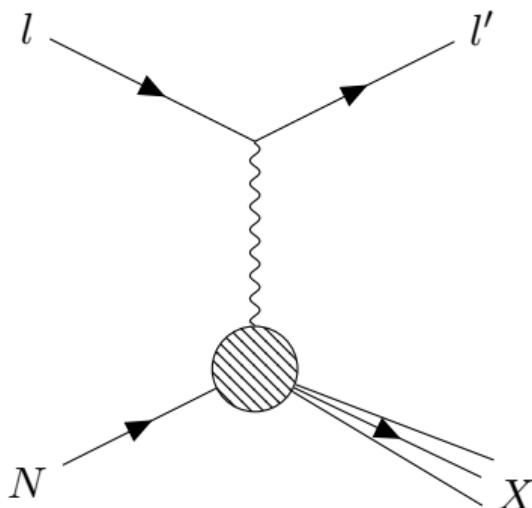
2.5.2 Factors to consider for interface design

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>]

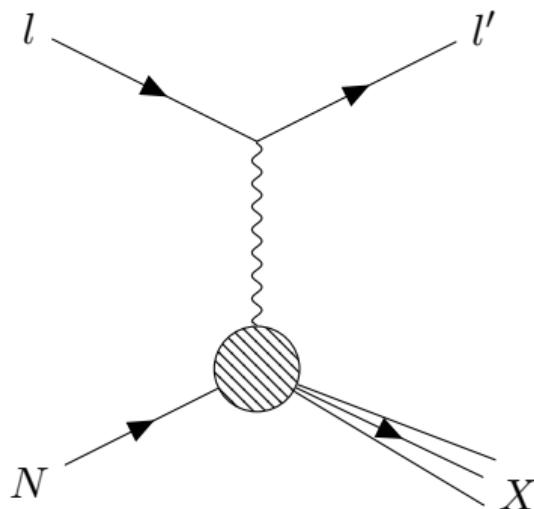
Leptonic and Hadronic Tensor



Observations:

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

Leptonic and Hadronic Tensor



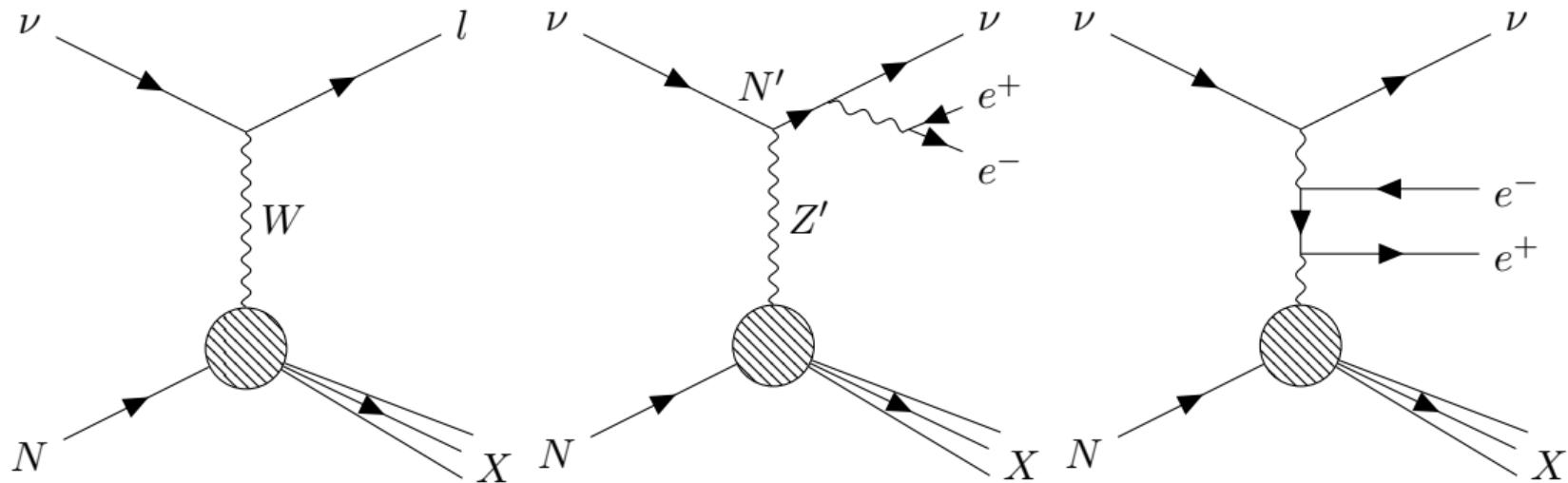
Observations:

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

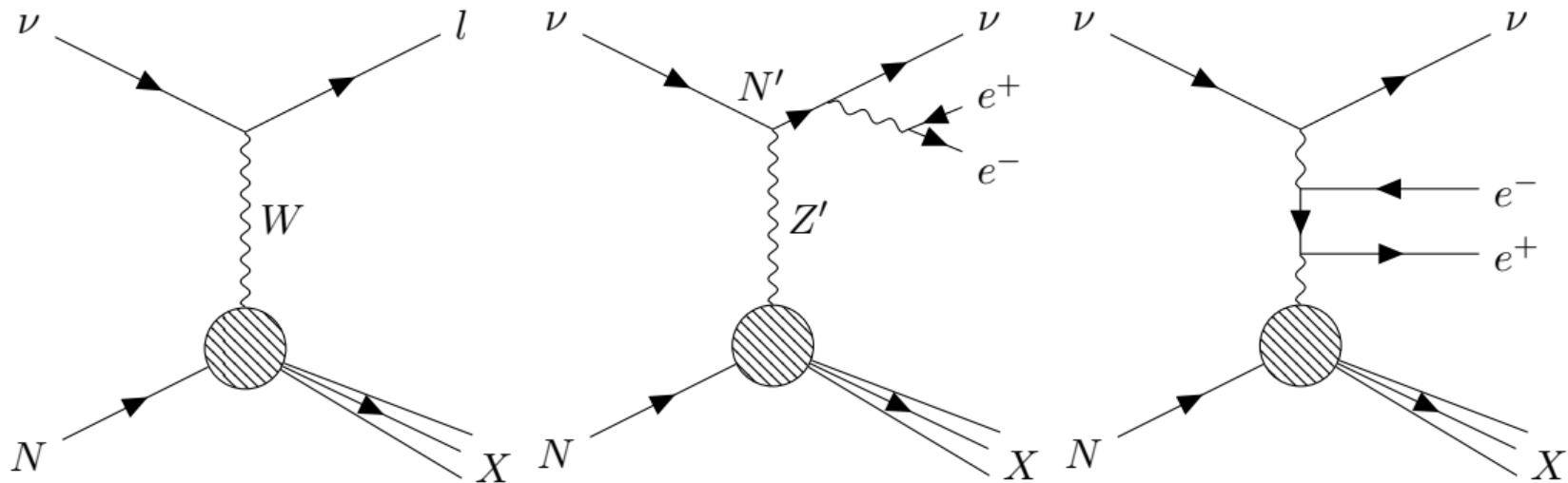
Question

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

Leptonic and Hadronic Tensor



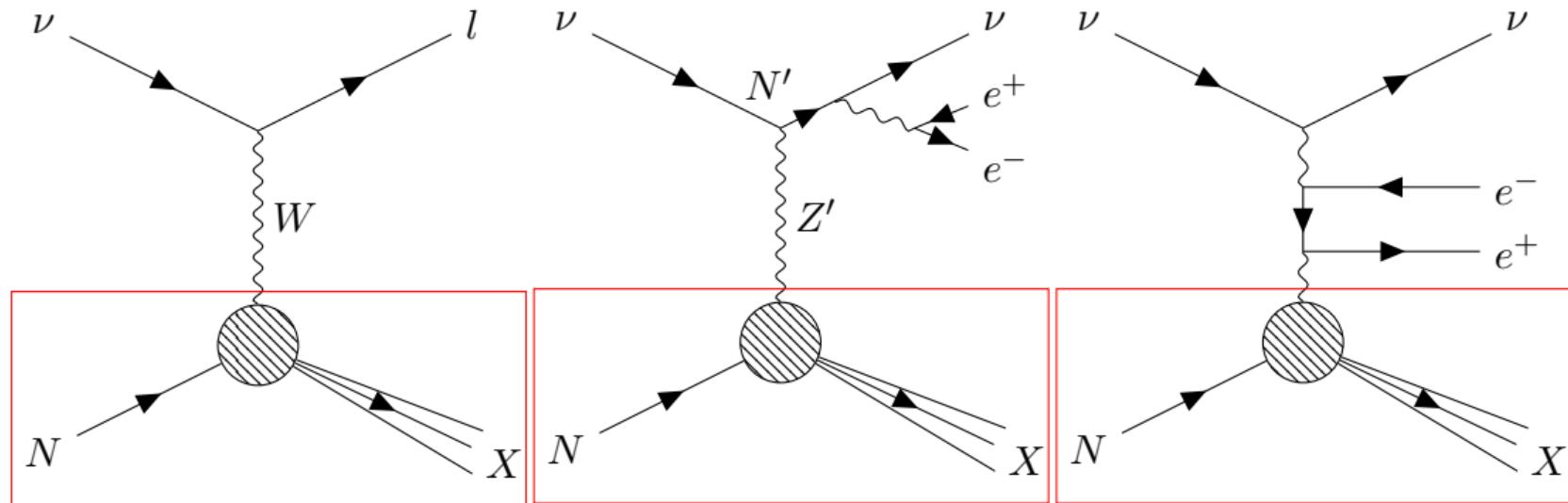
Leptonic and Hadronic Tensor



Question

What do all the diagrams above have in common?

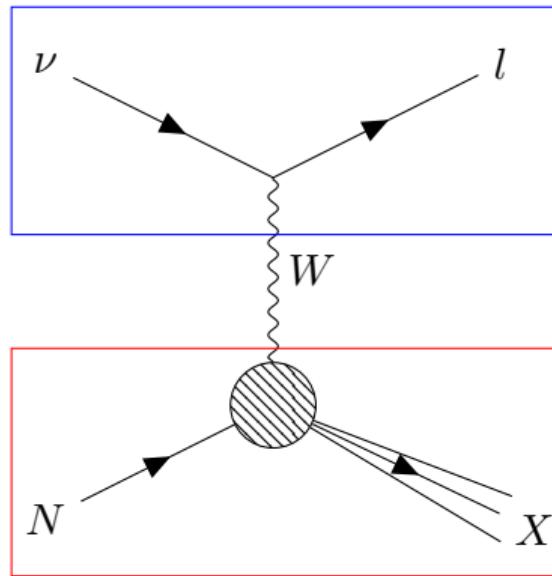
Leptonic and Hadronic Tensor



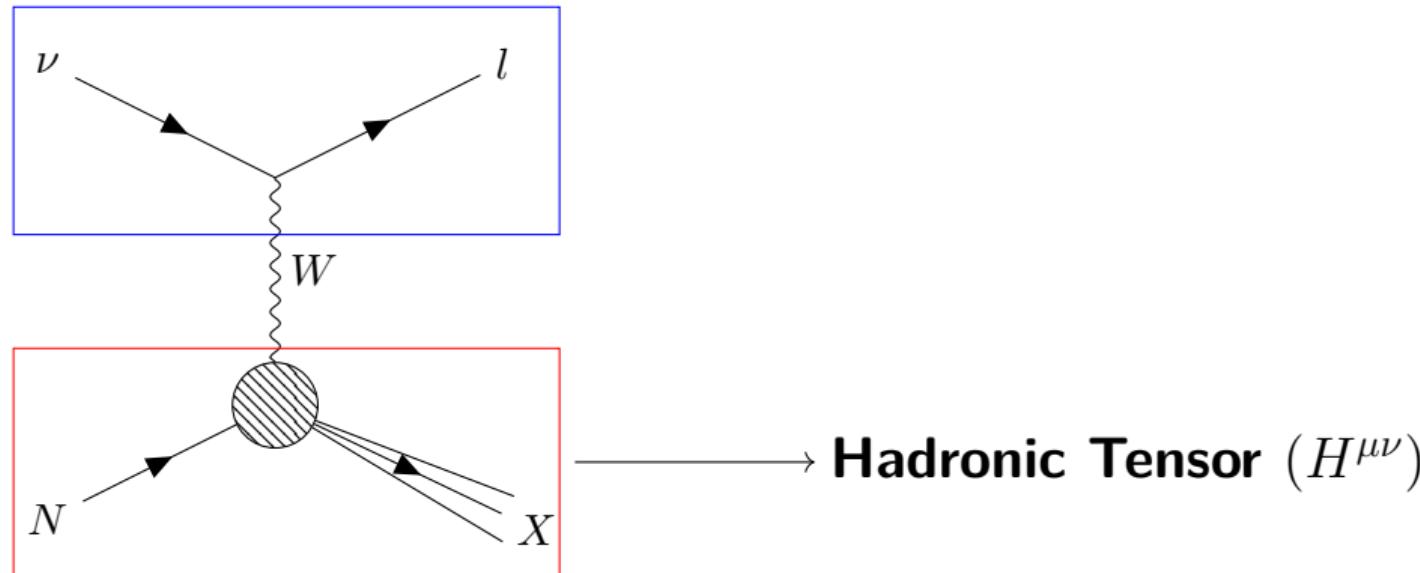
Question

What do all the diagrams above have in common?

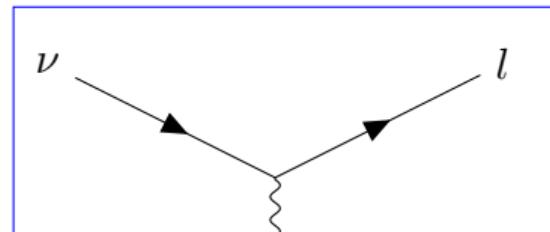
Leptonic and Hadronic Tensor



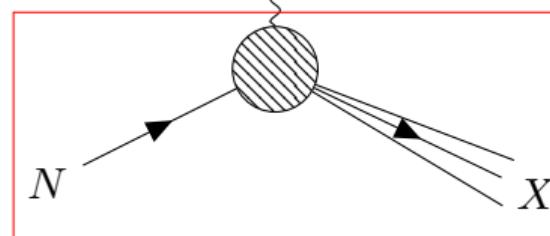
Leptonic and Hadronic Tensor



Leptonic and Hadronic Tensor

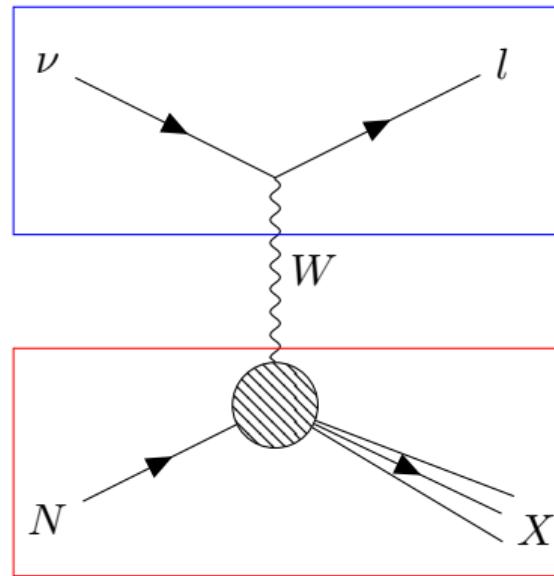


Leptonic Tensor ($L_{\mu\nu}$)



Hadronic Tensor ($H^{\mu\nu}$)

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.

Using Currents

Using tensors:

$$\frac{d\sigma}{d\Omega} = \sum_{i,j} L_{\mu\nu}^{(ij)} W^{(ij)\mu\nu} = L_{\mu\nu}^{(\gamma\gamma)} W^{(\gamma\gamma)\mu\nu} + L_{\mu\nu}^{(\gamma Z)} W^{(\gamma Z)\mu\nu} + L_{\mu\nu}^{(Z\gamma)} W^{(Z\gamma)\mu\nu} + L_{\mu\nu}^{(ZZ)} W^{(ZZ)\mu\nu} + \dots$$

Using Currents:

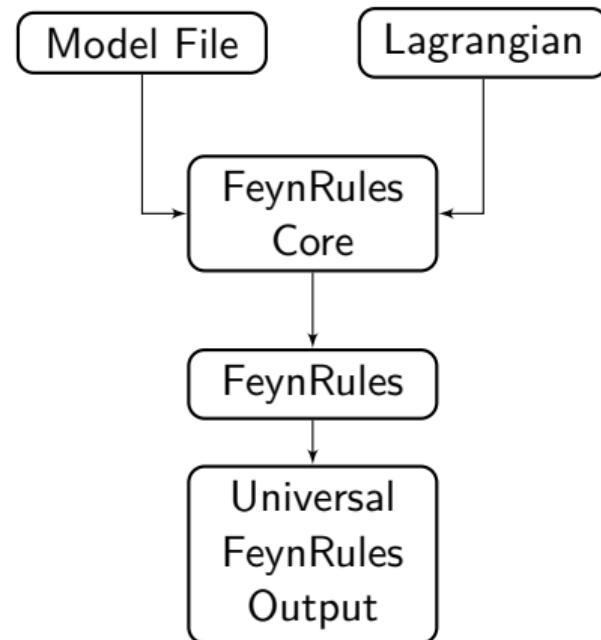
$$\frac{d\sigma}{d\Omega} = \left| \sum_i L_\mu^{(i)} W^{(i)\mu} \right|$$

Interferences handled automatically using currents

Interface to tensors provided for nuclear calculations that **must** be expressed using tensors.

FeynRules

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



[arXiv:0806.4194, arXiv:1310.1921]

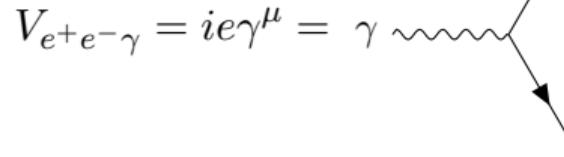
Universal FeynRules Output (UFO)

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

- 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

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

$$V_{e^+e^-\gamma}$$



[arXiv:1108.2040]

Handling Form Factors

Nuclear one-body current operators:

$$\mathcal{J}^\mu = (\mathcal{J}_V^\mu + \mathcal{J}_A^\mu)$$

$$\mathcal{J}_V^\mu = \gamma^\mu \mathcal{F}_1^a + i\sigma^{\mu\nu} q_\nu \frac{\mathcal{F}_2^a}{2M}$$

$$\mathcal{J}_A^\mu = -\gamma^\mu \gamma_5 \mathcal{F}_A^a - q^\mu \gamma_5 \frac{\mathcal{F}_P^a}{M}$$

Standard Model Form Factors:

$$\mathcal{F}_i^{\gamma(p,n)} = F_i^{p,n}, \quad \mathcal{F}_A^\gamma = 0$$

$$\mathcal{F}_i^{W(p,n)} = F_i^p - F_i^n, \quad \mathcal{F}_A^W = F_A$$

$$\mathcal{F}_i^{Z(p)} = \left(\frac{1}{2} - 2 \sin^2 \theta_W \right) F_i^p - \frac{1}{2} F_i^n,$$

$$\mathcal{F}_i^{Z(n)} = \left(\frac{1}{2} - 2 \sin^2 \theta_W \right) F_i^n - \frac{1}{2} F_i^p$$

$$\mathcal{F}_A^{Z(p)} = \frac{1}{2} F_A, \quad \mathcal{F}_A^{Z(n)} = -\frac{1}{2} F_A$$

Handling Form Factors

Nuclear one-body current operators:

$$\mathcal{J}^\mu = (\mathcal{J}_V^\mu + \mathcal{J}_A^\mu)$$

$$\mathcal{J}_V^\mu = \gamma^\mu \mathcal{F}_1^a + i\sigma^{\mu\nu} q_\nu \frac{\mathcal{F}_2^a}{2M}$$

$$\mathcal{J}_A^\mu = -\gamma^\mu \gamma_5 \mathcal{F}_A^a - q^\mu \gamma_5 \frac{\mathcal{F}_P^a}{M}$$

Standard Model Form Factors:

$$\mathcal{F}_i^{\gamma(p,n)} = F_i^{p,n}, \quad \mathcal{F}_A^\gamma = 0$$

$$\mathcal{F}_i^{W(p,n)} = F_i^p - F_i^n, \quad \mathcal{F}_A^W = F_A$$

$$\mathcal{F}_i^{Z(p)} = \left(\frac{1}{2} - 2 \sin^2 \theta_W \right) F_i^p - \frac{1}{2} F_i^n,$$

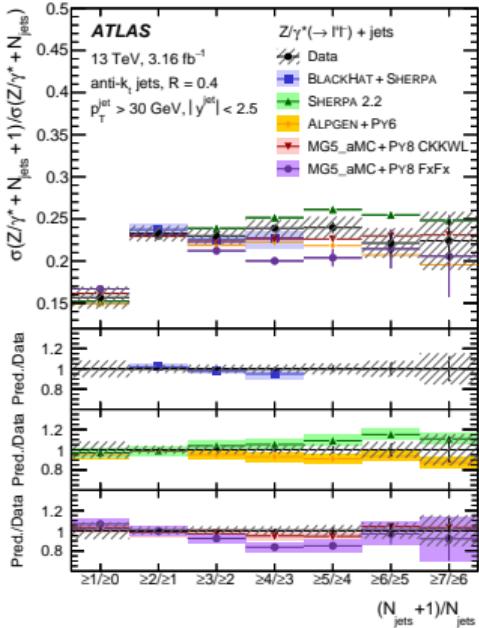
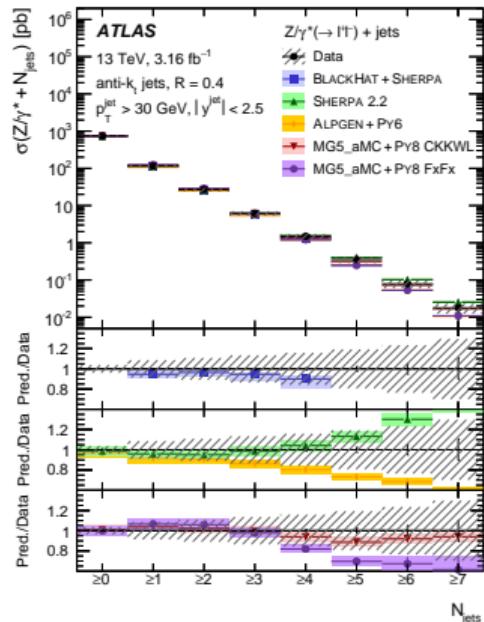
$$\mathcal{F}_i^{Z(n)} = \left(\frac{1}{2} - 2 \sin^2 \theta_W \right) F_i^n - \frac{1}{2} F_i^p$$

$$\mathcal{F}_A^{Z(p)} = \frac{1}{2} F_A, \quad \mathcal{F}_A^{Z(n)} = -\frac{1}{2} F_A$$

Straight-forward to extend to BSM if CVC is valid

Tree Level Matrix Element Generators

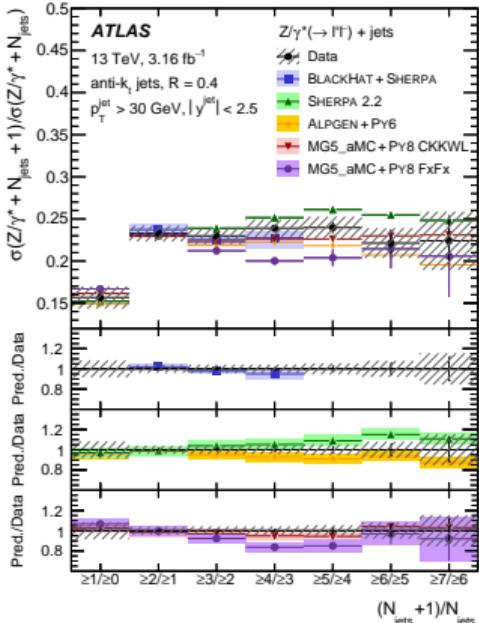
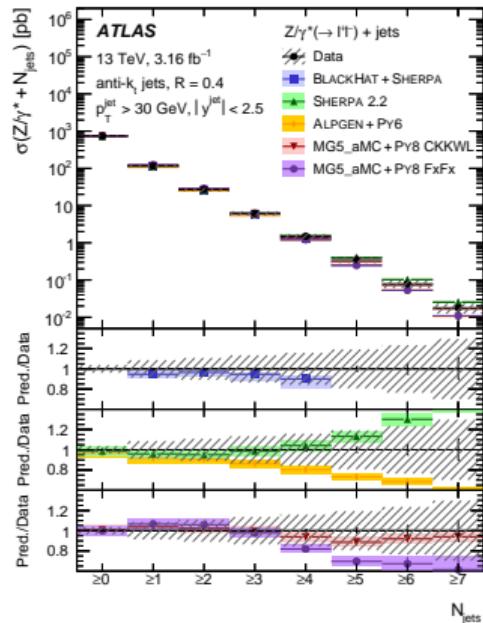
- ALPGEN [[arXiv:hep-ph/0206293](#)]
- AMEGIC [[arXiv:hep-ph/0109036](#)]
- COMIX [[arXiv:0808.3674](#)]
- CALCHEP [[arXiv:1207.6082](#)]
- HERWIG [[arXiv:0803.0883](#)]
- MADGRAPH [[arXiv:1405.0301](#)]
- WHIZARD [[arXiv:0708.4233](#)]
- etc.



[[arXiv:1702.05725](#)]

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

$$\mathcal{J}_\alpha(\pi) = P_\alpha(\pi) \sum_{V_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\pi)} \mathcal{S}(\pi, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\pi, \pi) \mathcal{J}_{\alpha_1}(\pi_1) \mathcal{J}_{\alpha_2}(\pi_2)$$

$$L_\mu^{(i)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m)$$

$$L_{\mu\nu}^{(i,j)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m) \mathcal{J}_\nu^{(j)\dagger}(1, \dots, m)$$

Berends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(3^n)$

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

Recursive Matrix Element Generation

$$\mathcal{J}_\alpha(\pi) = P_\alpha(\pi) \sum_{V_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\pi)} \mathcal{S}(\pi, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\pi, \pi) \mathcal{J}_{\alpha_1}(\pi_1) \mathcal{J}_{\alpha_2}(\pi_2)$$

$$L_\mu^{(i)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m)$$

$$L_{\mu\nu}^{(i,j)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m) \mathcal{J}_\nu^{(j)\dagger}(1, \dots, m)$$



Berends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(3^n)$

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

Recursive Matrix Element Generation

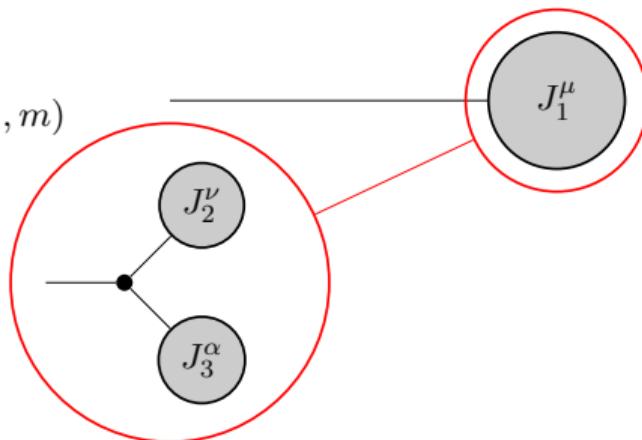
$$\mathcal{J}_\alpha(\pi) = P_\alpha(\pi) \sum_{V_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\pi)} \mathcal{S}(\pi, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\pi, \pi) \mathcal{J}_{\alpha_1}(\pi_1) \mathcal{J}_{\alpha_2}(\pi_2)$$

$$L_\mu^{(i)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m)$$

$$L_{\mu\nu}^{(i,j)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m) \mathcal{J}_\nu^{(j)\dagger}(1, \dots, m)$$

Berends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(3^n)$



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

Recursive Matrix Element Generation

$$\mathcal{J}_\alpha(\pi) = P_\alpha(\pi) \sum_{V_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\pi)} \mathcal{S}(\pi, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\pi, \pi) \mathcal{J}_{\alpha_1}(\pi_1) \mathcal{J}_{\alpha_2}(\pi_2)$$

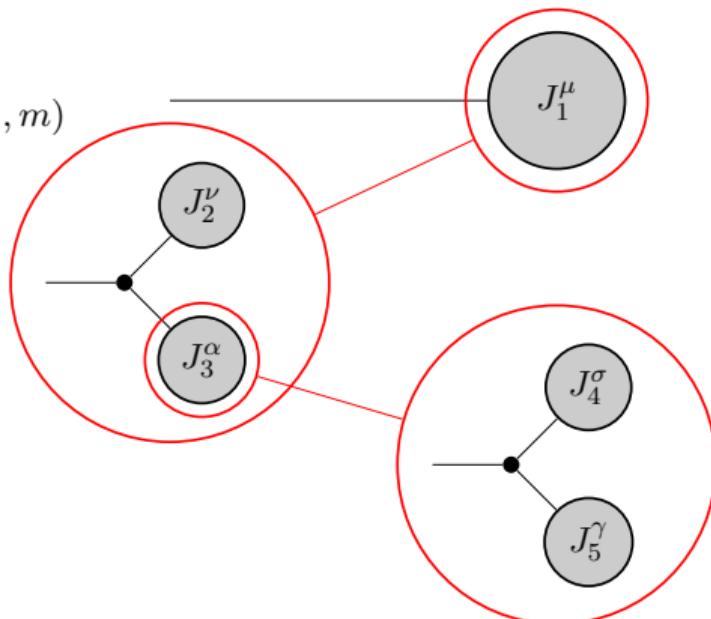
$$L_\mu^{(i)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m)$$

$$L_{\mu\nu}^{(i,j)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m) \mathcal{J}_\nu^{(j)\dagger}(1, \dots, m)$$

Berends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(3^n)$

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



Recursive Matrix Element Generation

$$\mathcal{J}_\alpha(\pi) = P_\alpha(\pi) \sum_{V_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\pi)} \mathcal{S}(\pi, \pi_2) V_\alpha^{\alpha_1, \alpha_2}(\pi, \pi) \mathcal{J}_{\alpha_1}(\pi_1) \mathcal{J}_{\alpha_2}(\pi_2)$$

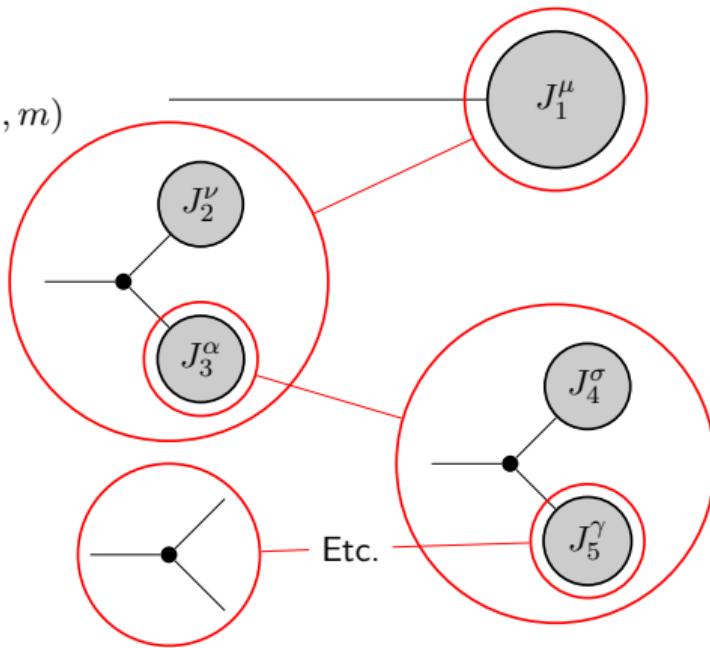
$$L_\mu^{(i)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m)$$

$$L_{\mu\nu}^{(i,j)}(1, \dots, m) = \mathcal{J}_\mu^{(i)}(1, \dots, m) \mathcal{J}_\nu^{(j)\dagger}(1, \dots, m)$$

Berends-Giele Recursion

- Reuse parts of calculation
- Most efficient for high multiplicity
- Reduces computation from $\mathcal{O}(n!)$ to $\mathcal{O}(3^n)$

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



Phase Space Generation

$$d\Phi_n(a, b; 1, \dots, n) = \delta^{(4)} \left(p_a + p_b - \sum_{i=1}^n p_i \right) \left[\prod_{i=1}^n \frac{d^4 p_i}{(2\pi)^3} \delta(p_i^2 - m_i^2) \Theta(p_{i_0}) \right]$$

The above phase space definition does not contain the handling of initial states.

Algorithms for n -body phase space generation

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

Phase Space Generation

$$d\Phi_n(a, b; 1, \dots, n) = \delta^{(4)} \left(p_a + p_b - \sum_{i=1}^n p_i \right) \left[\prod_{i=1}^n \frac{d^4 p_i}{(2\pi)^3} \delta(p_i^2 - m_i^2) \Theta(p_{i_0}) \right]$$

The above phase space definition does not contain the handling of initial states.

Algorithms for n -body phase space generation

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

2 → 2 Phase Space Example

Consider $l + {}^{12}C \rightarrow l' + N + X$ in the quasielastic regime.

$$d\sigma \propto d\Phi_2(a, b; 1, 2) \quad d^4 p_a \quad d^3 p_b$$

2 → 2 Phase Space Example

Consider $l + {}^{12}C \rightarrow l' + N + X$ in the quasielastic regime.

$$d\sigma \propto d\Phi_2(a, b; 1, 2) \quad d^4 p_a \quad d^3 p_b$$

- Phase space: $d\Phi_2(a, b; 1, 2) = \frac{\lambda(s_{ab}, s_1, s_2)}{16\pi^2 2s_{ab}} d\cos\theta_1 d\phi_1$

2 → 2 Phase Space Example

Consider $l + {}^{12}C \rightarrow l' + N + X$ in the quasielastic regime.

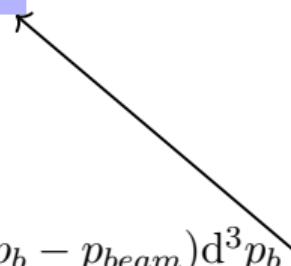
$$d\sigma \propto d\Phi_2(a, b; 1, 2) \quad d^4 p_a \quad d^3 p_b$$

- Phase space: $d\Phi_2(a, b; 1, 2) = \frac{\lambda(s_{ab}, s_1, s_2)}{16\pi^2 2s_{ab}} d\cos\theta_1 d\phi_1$
- Initial nucleon: $d^4 p_a = |\vec{p}_a|^2 dp_a dE_r d\cos\theta_a d\phi_a$

2 → 2 Phase Space Example

Consider $l + {}^{12}C \rightarrow l' + N + X$ in the quasielastic regime.

$$d\sigma \propto d\Phi_2(a, b; 1, 2) \quad d^4p_a \quad d^3p_b$$



- Phase space: $d\Phi_2(a, b; 1, 2) = \frac{\lambda(s_{ab}, s_1, s_2)}{16\pi^2 2s_{ab}} d\cos\theta_1 d\phi_1$
- Initial nucleon: $d^4p_a = |\vec{p}_a|^2 dp_a dE_r d\cos\theta_a d\phi_a$
- Initial lepton (Here only monochromatic): $d^3p_b = \delta^3(p_b - p_{beam}) d^3p_b$

2 → 2 Phase Space Example

Consider $l + {}^{12}C \rightarrow l' + N + X$ in the quasielastic regime.

$$d\sigma \propto d\Phi_2(a, b; 1, 2) \quad d^4 p_a \quad d^3 p_b$$

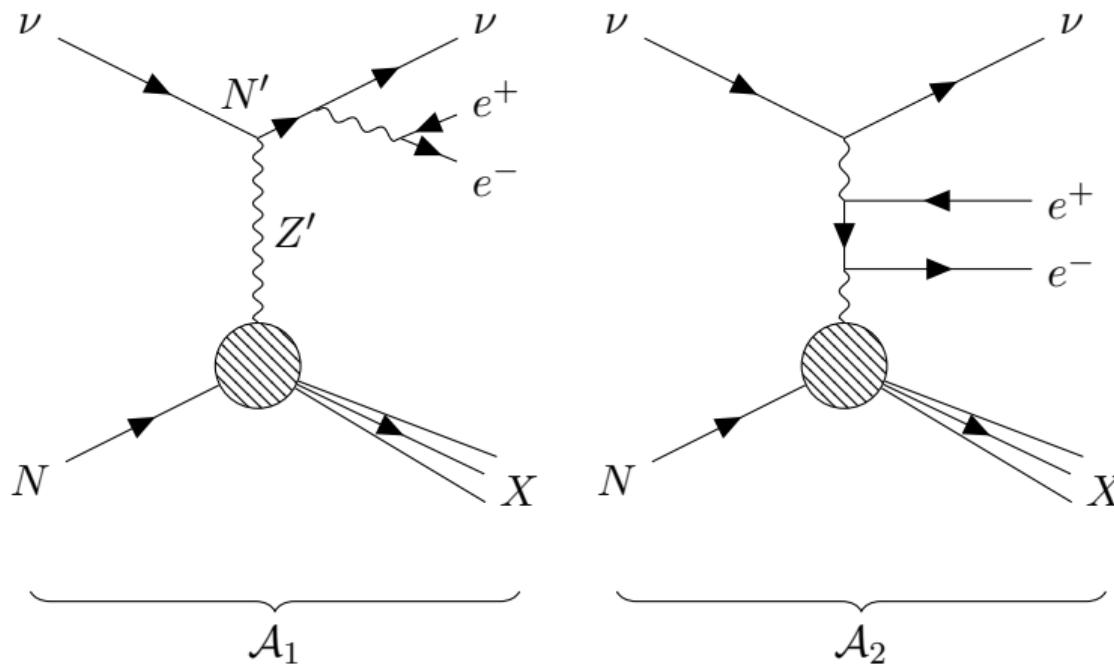
- Phase space: $d\Phi_2(a, b; 1, 2) = \frac{\lambda(s_{ab}, s_1, s_2)}{16\pi^2 2s_{ab}} d\cos\theta_1 d\phi_1$
- Initial nucleon: $d^4 p_a = |\vec{p}_a|^2 dp_a dE_r d\cos\theta_a d\phi_a$
- Initial lepton (Here only monochromatic): $d^3 p_b = \delta^3(p_b - p_{beam}) d^3 p_b$

Quasielastic Delta Function: $\delta(E_b - E_1 - E_r + m - E_2)$

Phase Space Delta Function: $\delta(E_a + E_b - E_1 - E_2)$

Define initial nucleon energy as $E_a = m - E_r$. Allows use of phase space tools developed at LHC.

Multi-channel Integration



- Both diagrams contribute to cross section
- They have different pole structures
- Need method to sample these structures efficiently
(i.e. $|\mathcal{A}_1 + \mathcal{A}_2|^2$)

Multi-channel Integration and VEGAS

Multi-channel Integration

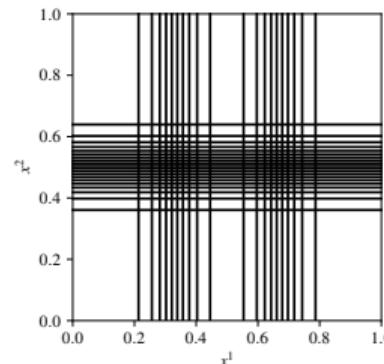
- Generate PS efficiently for $|\mathcal{A}_1|^2$ or $|\mathcal{A}_2|^2$
- Do not know how to efficiently sample $2Re(\mathcal{A}_1\mathcal{A}_2^\dagger)$
- Define channels: C_1 and C_2
- Generate events according to distributions g_i for channel i

$$\int d\vec{x} f(\vec{x}) = \sum_i \alpha_i \int d\vec{x} g_i(\vec{x}) \frac{f(\vec{x})}{g_i(\vec{x})}$$

- Optimize α_i to minimize variance

VEGAS

- Adaptive importance sampling
- Use this to get interference terms more accurately



VEGAS grid for $\int_0^1 d^4x \left(e^{-100(\vec{x}-\vec{r}_1)^2} + e^{-100(\vec{x}-\vec{r}_2)^2} \right)$

[J.Comput.Phys. 27 (1978) 291, 2009.05112]

Recursive Phase Space Decomposition

Phase space can be decomposed as:

$$d\Phi_n(a, b; 1, \dots, n) = d\Phi_{n-m+1}(a, b; m+1, \dots, n) \frac{ds_\pi}{2\pi} d\Phi_m(\pi; 1, \dots, m)$$

Iterate until only 1 → 2 phase spaces remain.

Basic building blocks:

$$S_\pi^{\rho, \pi \setminus \rho} = \frac{\lambda(s_\pi, s_\rho, s_{\pi \setminus \rho})}{16\pi^2 2 s_\pi} d\cos\theta_\rho d\phi_\rho$$

$$T_{\alpha, b}^{\pi, \overline{\alpha b \pi}} = \frac{\lambda(s_{\alpha b}, s_\pi, s_{\overline{\alpha b \pi}})}{16\pi^2 2 s_{\alpha b}} d\cos\theta_\pi d\phi_\pi$$

Momentum conservation: $(2\pi)^4 d^4 p_{\overline{\alpha b}} \delta^{(4)}(p_\alpha + p_b - p_{\overline{\alpha b}})$

Results

Processes Considered:

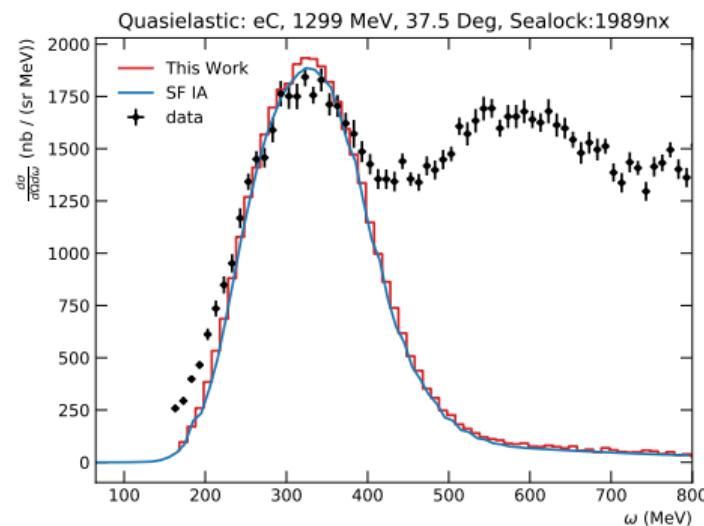
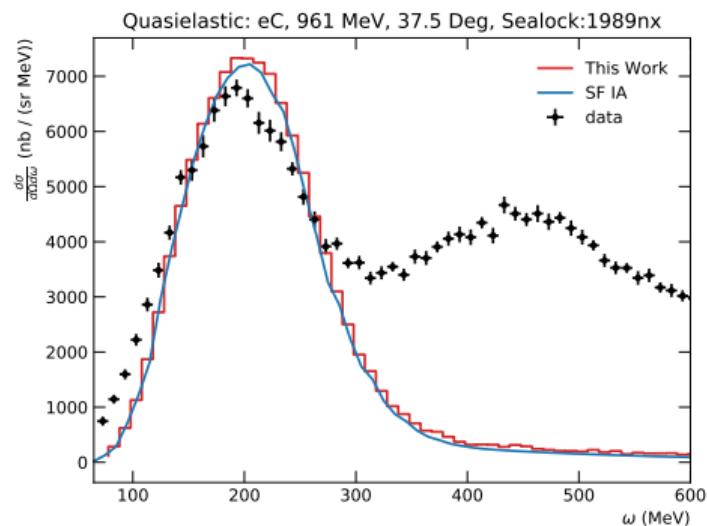
- Electron-Carbon Scattering
- Neutrino-Carbon Scattering
- Neutrino Tridents

NOTE: All processes are fully differential

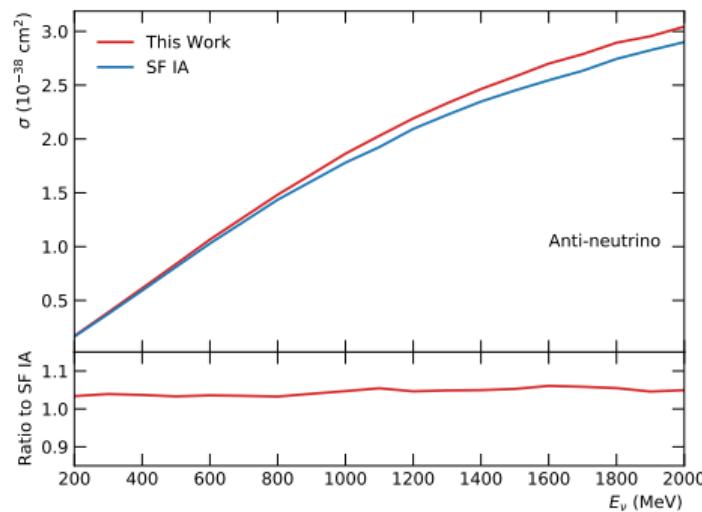
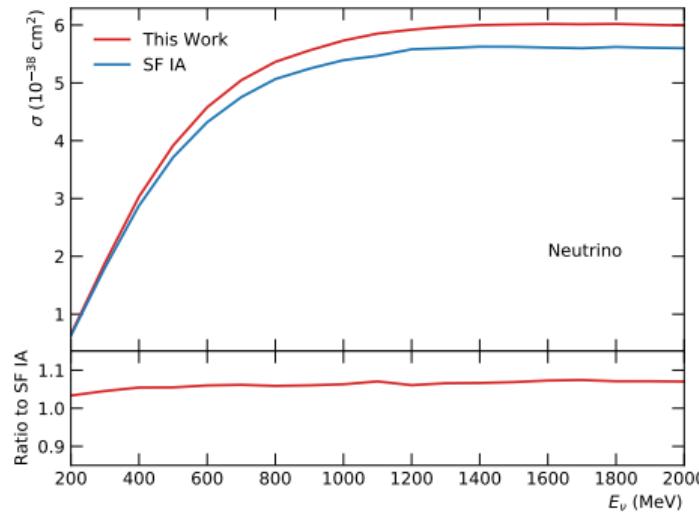
Parameters:

- Only quasielastic scattering is included and no FSI
- EM Form Factors: Kelly parameterization [[PRC 70, 068202 \(2004\)](#)]
- Axial Form Factor:
 - Dipole
 - $M_A = 1.0 \text{ GeV}$
 - $g_A = 1.2694$
- $\alpha = 1/137$
- $G_F = 1.16637 \times 10^{-5}$
- $M_Z = 91.1876 \text{ GeV}$

Electron Scattering

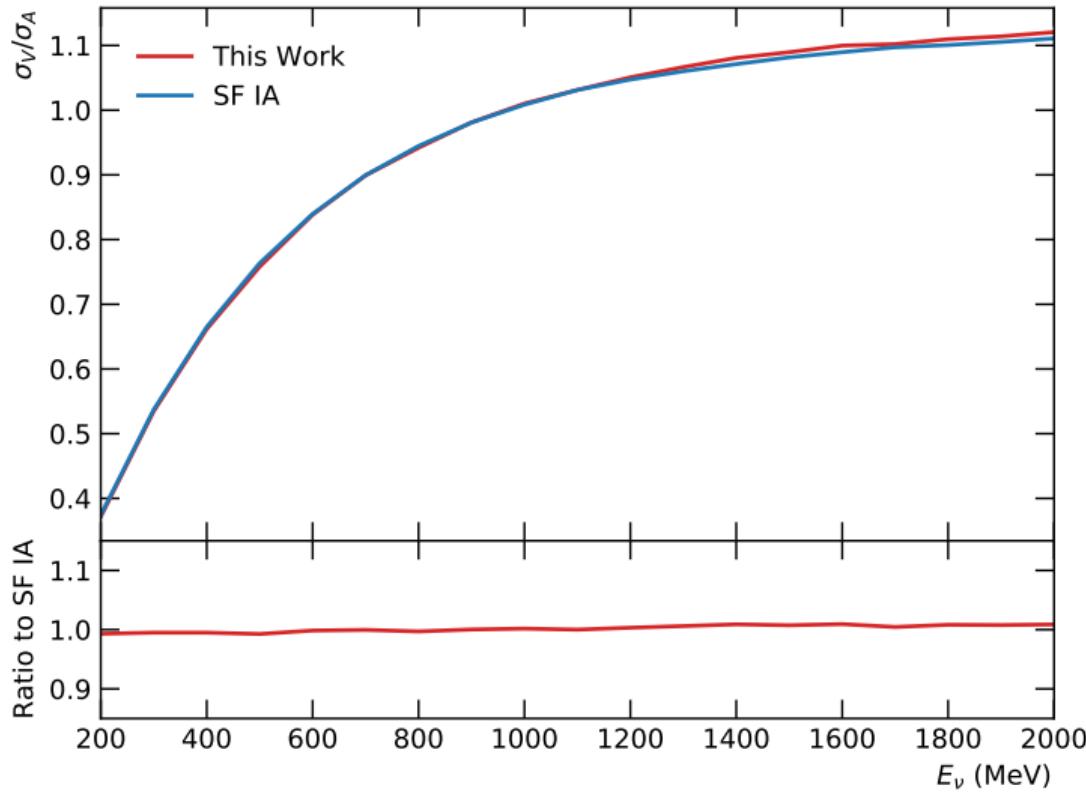


Neutrino Total Cross Section

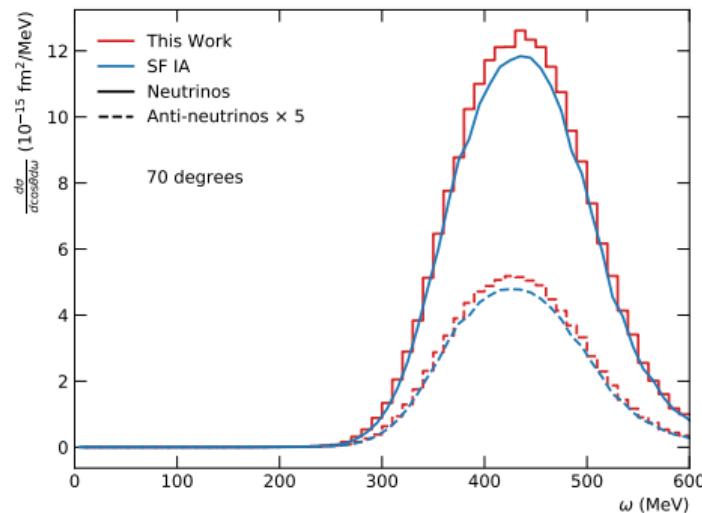
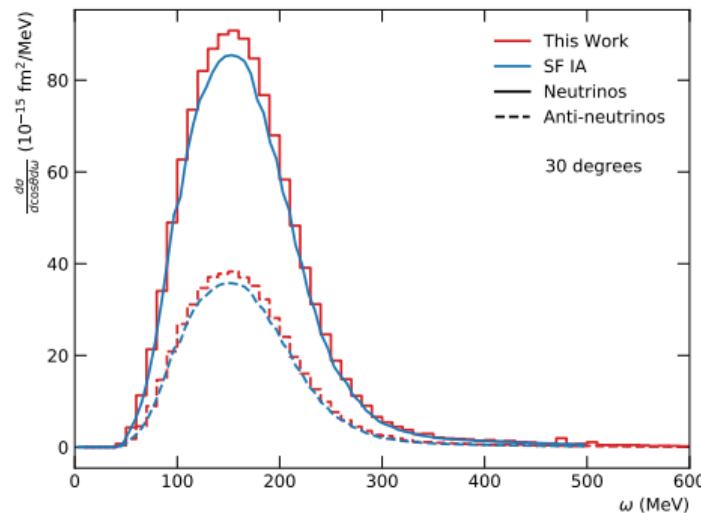


Neutrino Total Cross Section

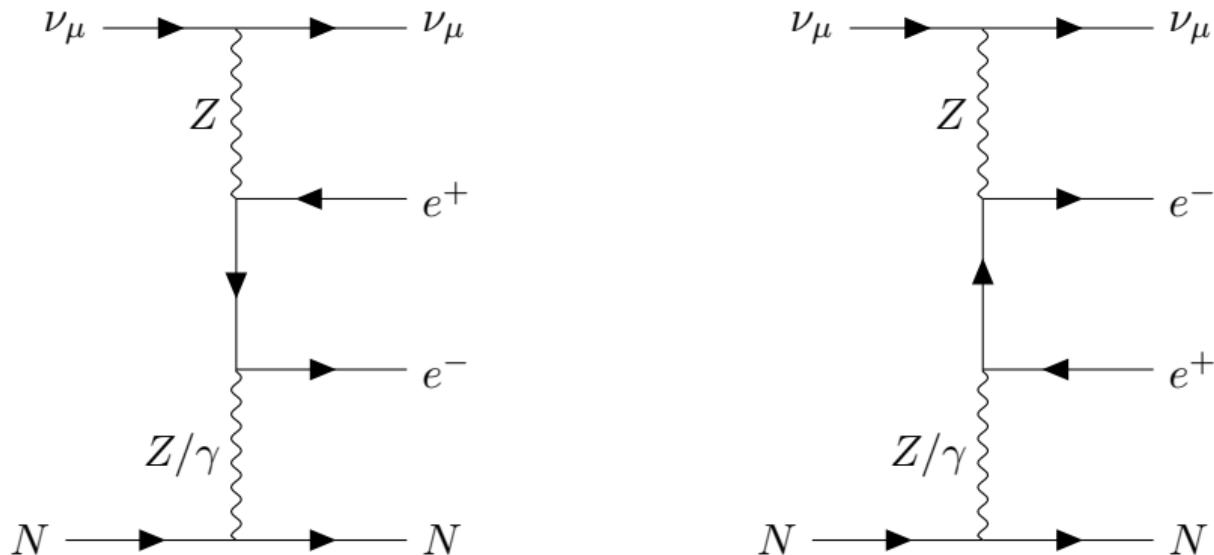
- Difference due to how couplings are handled
- Ratio of $\frac{\sigma_V}{\sigma_A}$ consistent



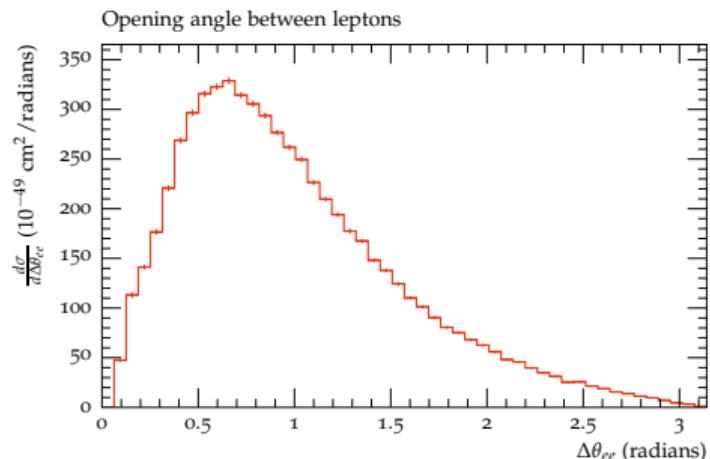
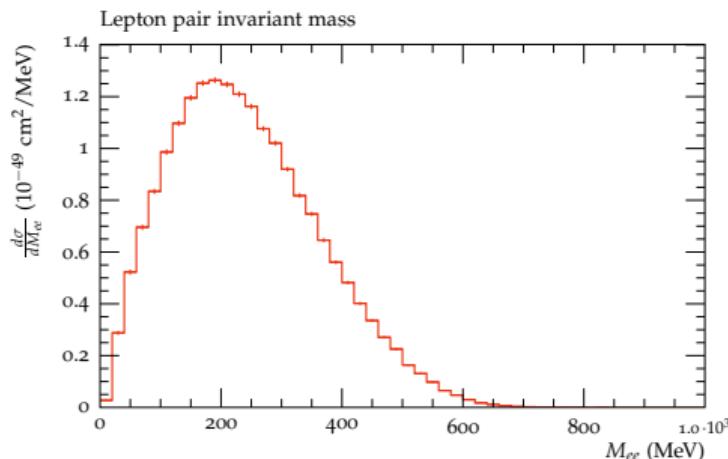
Neutrino Differential Cross Section



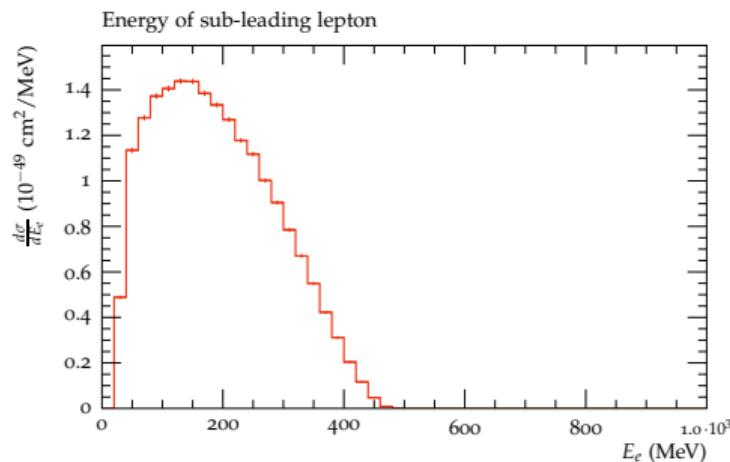
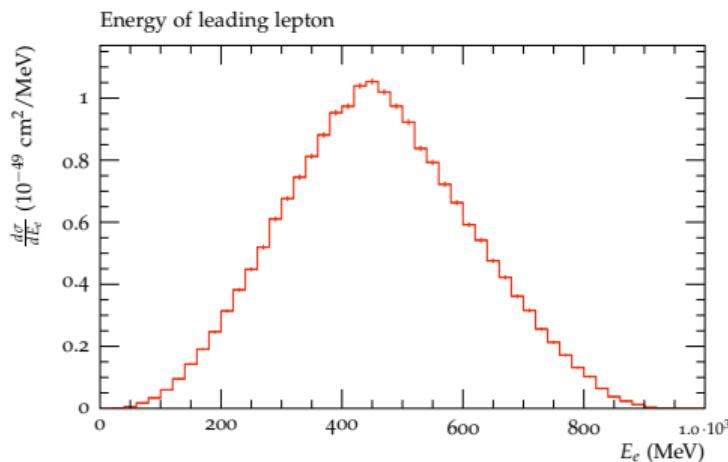
Neutrino Tridents



Neutrino Tridents



Neutrino Tridents



Conclusions

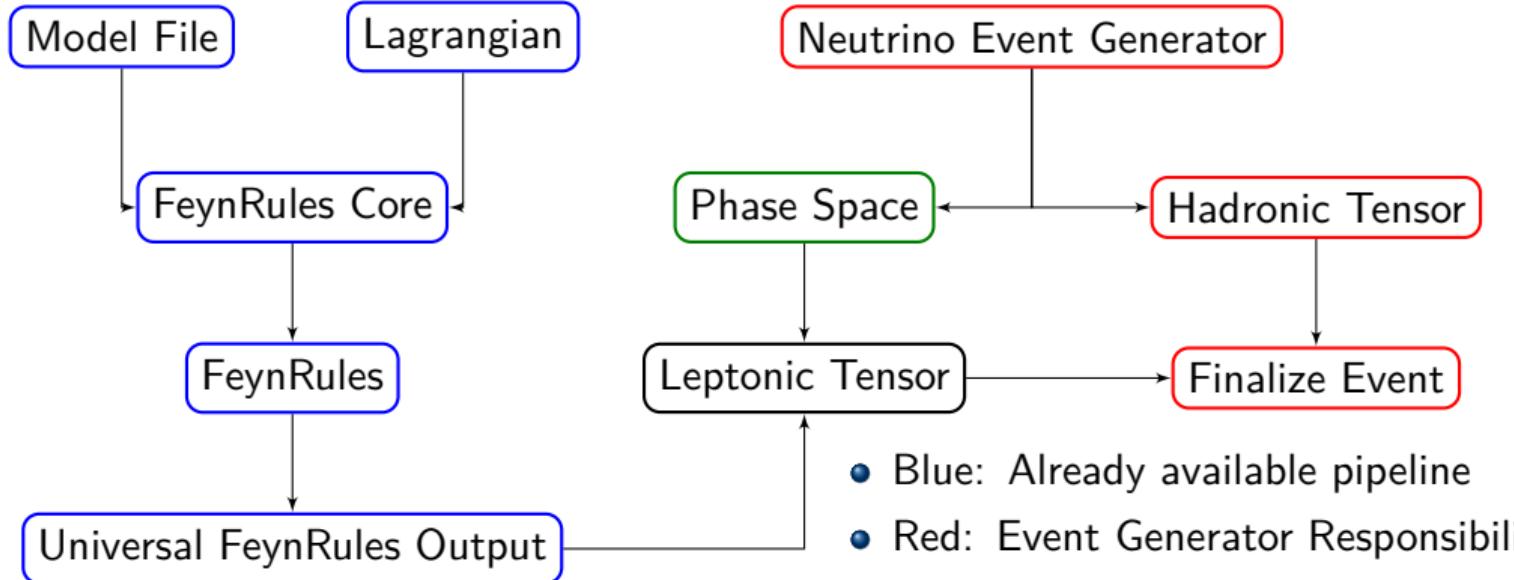
- BSM important for the current and next generation neutrino experiments
- Robust BSM program requires automating theory calculations
- Need new phase space and integration tools
- Develop method for arbitrary form factors
- Proof of principle for SM processes

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})
```

Proposed Interface



- Blue: Already available pipeline
- Red: Event Generator Responsibility
- Green: New code: n -body final state event generator: initial state momenta