

*u<sup>b</sup>*

[WWW.ALPHALOOP.CH](http://WWW.ALPHALOOP.CH)

# QUICKLY WHY DO WE NEED TOOLING?



# LOCAL UNITARITY: TL;DR

$$\sigma_{\gamma^* \rightarrow d\bar{d}} = \left( \text{Diagram 1} + \text{Diagram 2} \right) \times \left( \text{Diagram 3} + \text{Diagram 4} \right)^*$$

$$+ \left( \text{Diagram 5} + \text{Diagram 6} \right) \times \left( \text{Diagram 7} + \text{Diagram 8} \right)^*$$

$$\sigma_{\gamma^* \rightarrow d\bar{d}}^{(\text{LU})} = \text{Diagram 9} + \text{Diagram 10} + \text{Diagram 11} + \text{Diagram 12} + \text{Diagram 13} + \text{Diagram 14}$$

- LO
- NLO, Double-Triangle (DT)
- NLO, Self-Energy (SE)

$$\text{Diagram 15} \equiv \frac{p^2}{2p^0} \delta(p^2) \Theta(p^0)$$

# LOCAL UNITARITY: A CONCEPTUAL SHIFT

$$\sigma_{\gamma^* \rightarrow d\bar{d}}^{(\text{normal})} = \int \Pi^{(\text{phase-space})} \left| \text{---} \left( \text{---} + \text{---} + \text{---} + \text{---} \right) \right|^2$$

↓

$$\sigma_{\gamma^* \rightarrow d\bar{d}}^{(\text{LU})} = \text{LU} \left[ \text{---} \right] + \text{LU} \left[ \text{---} \right] + \text{LU} \left[ 2 \times \text{---} \right]$$

$$\sum_{c \in \{RRR, RRV, RVV, \dots\}} \int \Pi_c^{(\text{phase-space})} \left| \sum_{i_c=1}^{n_{\text{amplitudes}}(c)} \int \Pi_{i_c}^{(\text{loop})} \mathcal{A}_{i_c} \right|_{\text{truncated}}^2$$

IR-subtraction numerical  $d = 4$

analytic  $d = 4 - 2\epsilon$

$$\sum_{j=1}^{n_{\text{supergraphs}}} \int \Pi g_j^{(\text{LU})}$$

numerical  $d = 4$   
**NO** IR-subtraction

# KEY INGREDIENTS



User Interface



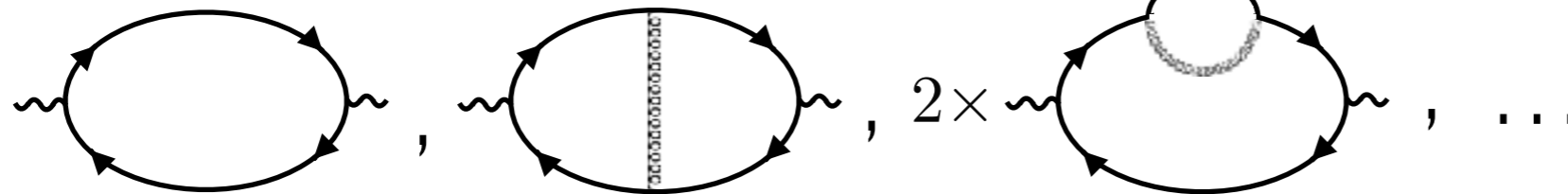
Low-level backend



Symbolica

“Open-source **free** fast Mathematica”

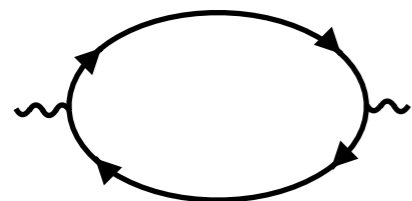
- Diagram generation



feyngen

<https://github.com/alpha100p/gammaloop>

- Fast and flexible numerator algebra



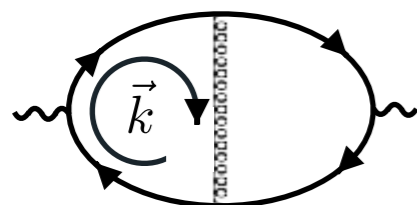
$$\epsilon(p)^\rho \text{Tr} [\gamma^\rho (\gamma^{\mu_1} (k + p)^{\mu_1} + m \mathbf{1}) \gamma^\sigma (\gamma^{\mu_2} k^{\mu_2} + m \mathbf{1})] \epsilon^*(p)^\sigma$$



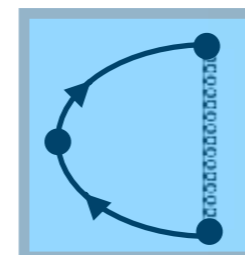
spenso

<https://github.com/alpha100p/spenso>

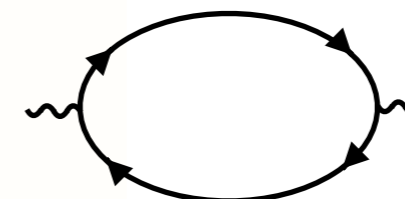
- Analytical vacuum integrals



$$\lim_{|\vec{k}| \rightarrow \infty}$$



$$d \rightarrow 4 - 2\epsilon$$



vakint

<https://github.com/alpha100p/vakint>

# FEYNGEN: EASY DIAGRAM GENERATION

<https://github.com/alpha00p/gammaploop>

First ingredient: Symbolica graph generation :

```
let mut graphs: HashMap<Graph<NodeColorWithoutVertexRule,... = SymbolicaGraph::generate(  
  external_edges: external_edges_for_generation.as_slice(),  
  vertex_signatures: vertex_signatures_for_generation.as_slice(),  
  max_vertices: None,  
  max_loops: Some(self.options.loop_count_range.1),  
  max_bridges: filters.get_max_bridge(),  
  allow_self_loops: !filter_self_loop,  
);
```

And the corresponding canonization with node and edge coloring :

```
let canonized_g: CanonicalForm<NodeColor, ...> = g.canonize();
```

Sprinkling physics on top:

```
Generating diagrams for process:  $\xi g \rightarrow h t \bar{t} / a z w+ g h a g h z g h w p g h m v e v m v t u c s \rightarrow g \theta g+ e- \mu- \tau- \text{QCD} \rightarrow 2 \text{ Q:D} \rightarrow 1 [ \{2\} ]$   
0s |  $\Delta=0s$  | Starting Feynman graph generation with Symbolica...  
0s |  $\Delta=0s$  | Number of graphs resulting from Symbolica generation: 45800  
0s |  $\Delta=0s$  | Number of graphs retained after removal of vetoed topologies: 9268  
0s |  $\Delta=0s$  | Number of graphs after vertex info assignment: 9268  
0s |  $\Delta=0s$  | Number of graphs after all complete graphs filters are applied: 3690  
0s |  $\Delta=0s$  | Number of graphs after closed fermion chains analysis: 3690  
0s |  $\Delta=0s$  | Number of graphs after symmetrization of external states: 3690  
0s |  $\Delta=0s$  | Number of graphs after canonization of vertices, edges and flows (graph labels assigned here): 3690  
10s |  $\Delta=9s$  | Number of graphs after numerator-aware grouping with strategy 'only detect zero numerators': 3634 (3013 isomorphically unique graphs, 56 color zeros, 0 lorentz zero, 0 grouped and 0 cancellation)  
( Sum of the symmetry factors from each graph generated = 565/3 )  
A total of 3634 graphs have been generated in 00:00:29.826.
```

Let's see it in action!

Process generation syntax and options detailed at :

<https://wiki.alpha00p.ch/en/gammaLoop/ProcessGeneration>

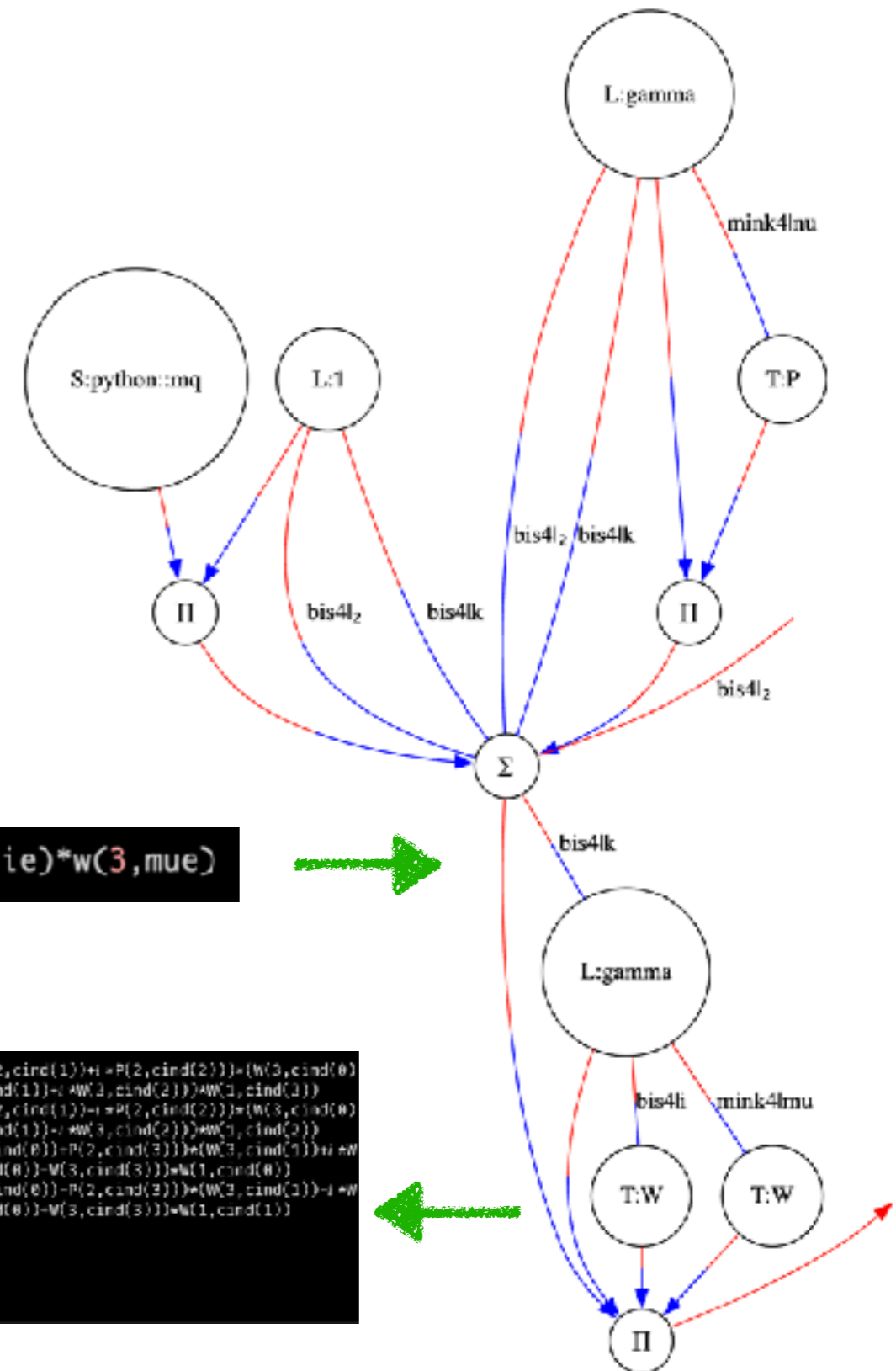
# SPENSO: FEYNMAN DIAGRAMS AS TENSOR NETWORKS

<https://github.com/alpha100p/spenso>

$$W_{1,i} \gamma_{ik}^\mu (p_2^\nu \gamma_{kj}^\nu + m_q \mathbf{1}_{kj}) W_3^\mu$$

$$W_{1,i} \gamma_{ik}^\mu (p_2^\nu \gamma_{kj}^\nu + m_q \mathbf{1}_{kj}) W_3^\mu$$

```
x = g_muik.to_expression()*(p(2,nue)*gamma(nue,ke,je)+mq*id(ke,je))*w(1,ie)*w(3,mue)
```



```
[0]: ((-P(2,cind(1))-P(2,cind(2)))*(-W(3,cind(1))+W(3,cind(2)))-P(2,cind(0))-P(2,cind(3)))*(W(3,cind(0))-W(3,cind(3)))*W(1,cind(0))+((-P(2,cind(1))+P(2,cind(2)))-W(3,cind(0))+W(3,cind(3)))*(-W(3,cind(1))-W(3,cind(2)))*W(1,cind(1))+(-P(2,cind(1))-P(2,cind(2)))*W(3,cind(0))-W(3,cind(3)))*(-W(3,cind(1))+W(3,cind(2)))*W(1,cind(2)))+W(3,cind(0))+W(3,cind(3)))*W(1,cind(0))+mq*(W(3,cind(0))-W(3,cind(3)))*W(1,cind(1))+mq*(W(3,cind(1))-W(3,cind(2)))*W(1,cind(2))
[1]: ((-P(2,cind(1))-P(2,cind(2)))*(-W(3,cind(1))+W(3,cind(2)))-P(2,cind(0))-P(2,cind(3)))*(W(3,cind(0))-W(3,cind(3)))*W(1,cind(1))+((-P(2,cind(1))+P(2,cind(2)))-W(3,cind(0))-W(3,cind(3)))*(-W(3,cind(1))+W(3,cind(2)))*W(1,cind(2)))+W(3,cind(0))+W(3,cind(3)))*W(1,cind(0))+mq*(W(3,cind(0))-W(3,cind(3)))*W(1,cind(1))+mq*(W(3,cind(1))-W(3,cind(2)))*W(1,cind(2))
[2]: ((P(2,cind(0))+P(2,cind(3)))*(-W(3,cind(0))+W(3,cind(3)))-P(2,cind(1))-P(2,cind(2)))*(-W(3,cind(1))+W(3,cind(2)))*W(1,cind(0))+((-P(2,cind(0))-P(2,cind(3)))*(-W(3,cind(1))+W(3,cind(2)))*W(1,cind(1))+mq*(W(3,cind(0))-W(3,cind(3)))*W(1,cind(1))+mq*(W(3,cind(1))-W(3,cind(2)))*W(1,cind(2))
[3]: ((P(2,cind(0))-P(2,cind(3)))*(-W(3,cind(0))+W(3,cind(3)))-P(2,cind(1))+P(2,cind(2)))*(-W(3,cind(1))+W(3,cind(2)))*W(1,cind(0))+((-P(2,cind(0))-P(2,cind(3)))*(-W(3,cind(1))+W(3,cind(2)))*W(1,cind(1))+mq*(W(3,cind(0))-W(3,cind(3)))*W(1,cind(1))+mq*(W(3,cind(1))-W(3,cind(2)))*W(1,cind(2))
[0]: [-63.56167095136294+439.80791481890045i]
[1]: [33.399011649056+15+39.444256694814764i]
[2]: [117.80058178306877+88.05101385273053i]
[3]: [-157.20545431191258+117.68389663122473i]
```

# VAKINT: COMPUTATION OF UP TO 4-LOOP VACUUM INTEGRAL

<https://github.com/alpha100p/vakint>

FMFT [ [1707.01710](#) ] and MATAD [ [0009029](#) ]

$$\mathcal{N} = (k_1 \cdot k_2)^2 + (k_3 \cdot p_1)(k_3 \cdot p_2) + (p_1 \cdot p_2)(k_1 + k_2) \cdot k_2$$

```
integral = E("""
(
  k(1,11)*k(2,11)*k(1,22)*k(2,22)
+ p(1,11)*k(3,11)*k(3,22)*p(2,22)
+ p(1,11)*p(2,11)*(k(2,22)+k(1,22))*k(2,22)
)*topo(
  prop(1,edge(1,2),k(1),muvsq,1)
* prop(2,edge(2,3),k(2),muvsq,1)
* prop(3,edge(3,1),k(3),muvsq,1)
* prop(4,edge(1,4),k(3)-k(1),muvsq,1)
* prop(5,edge(2,4),k(1)-k(2),muvsq,1)
* prop(6,edge(3,4),k(2)-k(3),muvsq,1)
)""")
```

```
direct_evaluation = vakint.evaluate(integral)
```

```
num_eval, num_error = vakint.numerical_evaluation(
  evaluated_integral, params=masses, externals=external_momenta)
```

Numerical evaluation:

```
 $\epsilon^{-3}$  : (0+-11440.53140354612i)
 $\epsilon^{-2}$  : (0+57169.95521898031i)
 $\epsilon^{-1}$  : (0+-178748.9838377694i)
 $\epsilon^0$  : (0+321554.1122184795i)
```

