Recent developments of GRACE system

Yoshimasa Kurihara (KEK)
J. Vermaseren (Nikhef)
T. Ueda (Seikei Univ.)

12/Mar./2019@ACAT2019 (Saas-Fee, Swissland)

Contents:

- Introduction
  - GRACE System
  - Calculations for the ILC
- Code optimization using FORM
- Physics Results
  - Higgs productions
- Summary

Introduction

GRACE system

- GRACE is an automatic system to calculate tree and ELWK-loop cross sections with spin-polarizations based on SM and MSSM.

$$e^+e^- \to t\bar{t}\gamma$$

$$e^+e^- \to e^+e^-\gamma$$

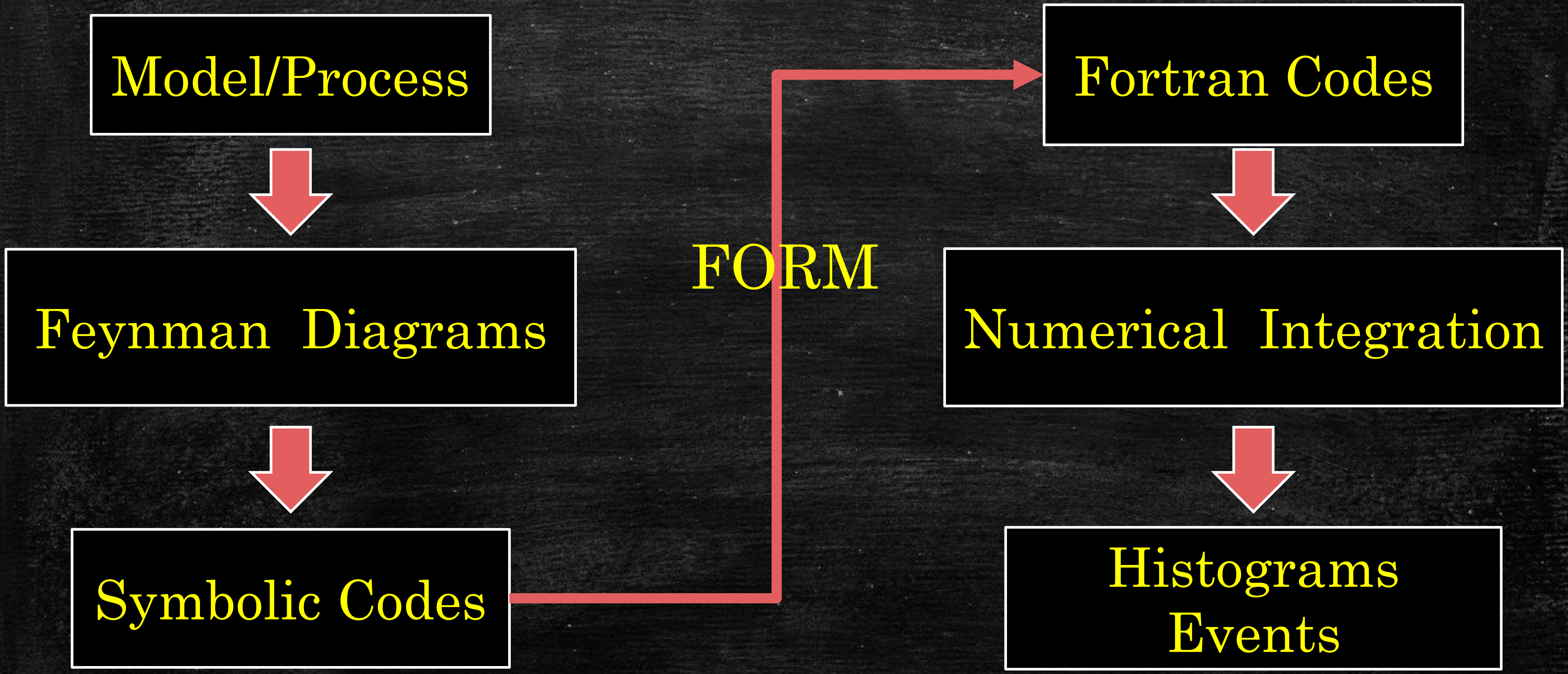
$$e^+e^- \to t\bar{t} \text{ w/ beam pol.}$$

Full-ELWK Eur. Phys. J. C **73**, 2400 (2013)

Full-ELWK Phys. Lett. **B740,** 192 (2014)

Full-ELWK Eur. Phys. J. C **78**, 422 (2018)

GRACE system



GRACE system

NLG fixing (Phys, Rept. **430**, 117 (2006))

$$\mathcal{L}_{GF} = -\frac{1}{\xi_W} \mid (\partial_\mu - ie\tilde{\alpha}A_\mu - igc_W\tilde{\beta}Z_\mu)W^{\mu+} + \xi_W\frac{g}{2}(v + \tilde{\delta}H + i\tilde{\kappa}\chi_3)\chi^+ \mid^2$$
$$-\frac{1}{2\xi_Z}(\partial.Z + \xi_Z\frac{g}{2c_W}(v + \tilde{\epsilon}H)\chi_3)^2 - \frac{1}{2\xi_A}(\partial.A)^2.$$

The results must be independent of non-linear gauge parameters ($\tilde{\alpha}, \tilde{\beta}, \tilde{\delta}, \tilde{\kappa}, \tilde{\epsilon}$).

[1] K. Aoki et al, Suppl. Prog. Theor. Phys. **73 (1982) 1**.

[2] G. Belanger, F. Boudjema, J. Fujimoto, T. Ishikawa, T. Kaneko, K. Kato, Y. Shimizu Phys, Rept. **430**, 117 (2006)

GRACE system

$$\sigma_{full} = \sigma_{Tree} + \sigma_{Loop}(C_{UV}, \tilde{\alpha}, \tilde{\beta}, \tilde{\delta}, \tilde{\epsilon}, \tilde{\kappa}, \lambda)$$
$$+ \sigma_{Tree}\delta_{soft}(\lambda, E_{\gamma} < k_c) + \sigma_{Hard}(k_c)$$

Soft photon: $E_{\gamma} < k_c$

Hard photon: $E_{\gamma} \geq k_c$

- C_{UV} (Ultra violet coefficient) independence
- Photon mass (λ) independence
- Gauge invariance
- k_c independence

Precision Control (top-pair) √s=500 GeV, kc= 1 MeV

λ	NLG	Cuv	pol (-,+)	Tree	(pb) loop	soft	T+L+S	δ
1E-17	(0,0,0,0,0)	0	(0,0)	5.0510E-01	-2.4358E+00	1.7120E+00	-2.1868E-01	0.0000E+00
			(L,R)	1.4026E+00	-6.8130E+00	4.7463E+00	-6.6402E-01	0.0000E+00
			(R,L)	6.1782E-01	-2.9304E+00	2.1016E+00	-2.1097E-01	0.0000E+00
1E-17	(0,0,0,0,0)	1000	(0,0)	5.0510E-01	-2.4358E+00	1.7120E+00	-2.1868E-01	1.0000E+00
			(L,R)	1.4026E+00	-6.8129E+00	4.7463E+00	-6.6393E-01	1.3554E-04
			(R,L)	6.1782E-01	-2.9304E+00	2.1016E+00	-2.1097E-01	1.0000E+00
1E-17	(10,20,30,40,50)	0	(0,0)	5.0510E-01	-2.4358E+00	1.7120E+00	-2.1868E-01	0.0000E+00
			(L,R)	1.4026E+00	-6.8129E+00	4.7463E+00	-6.6393E-01	1.3554E-04
			(R,L)	6.1782E-01	-2.9304E+00	2.1016E+00	-2.1097E-01	0.0000E+00
1E-19	(0,0,0,0,0)	0	(0,0)	5.0510E-01	-2.7340E+00	2.0099E+00	-2.1893E-01	-1.1066E-03
			(L,R)	1.4026E+00	-7.6397E+00	5.5727E+00	-6.6438E-01	-5.2860E-04
			(R,L)	6.1782E-01	-3.2962E+00	2.4670E+00	-2.1131E-01	-1.6069E-03

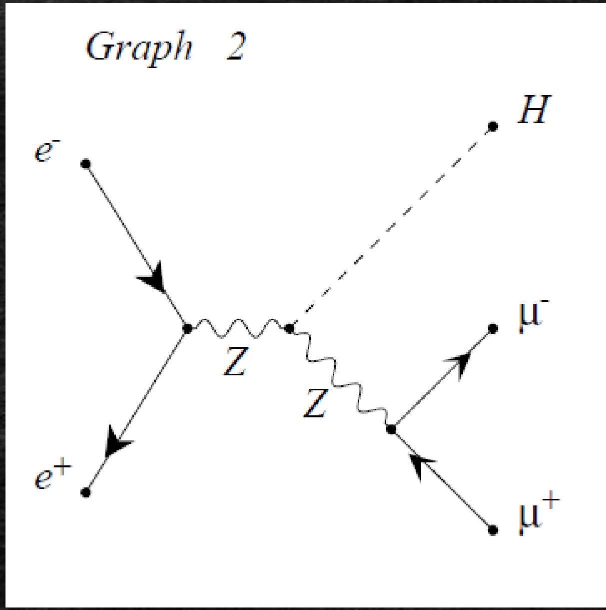
Precision Control (top-pair)

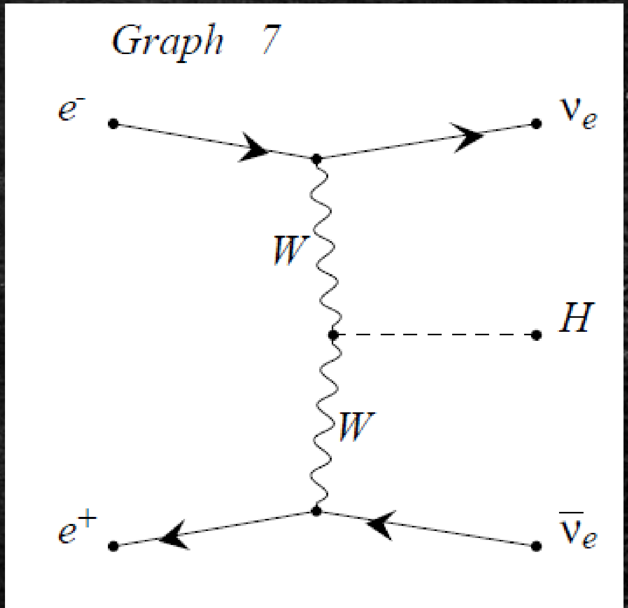
$\sqrt{s}=500$ GeV, $\lambda=10^{-17}$ GeV

Kc (GeV)	H (pb)	T+S	sum	δ
1e-3	7.4172E-01	2.2168E+00	2.9585E+00	2.9E-06
1e-4	8.9080E-01	2.0677E+00	2.9585E+00	0.0E+00
1e-5	1.0399E+00	1.9187E+00	2.9585E+00	-2.1E-06

GRACE for ILC physics

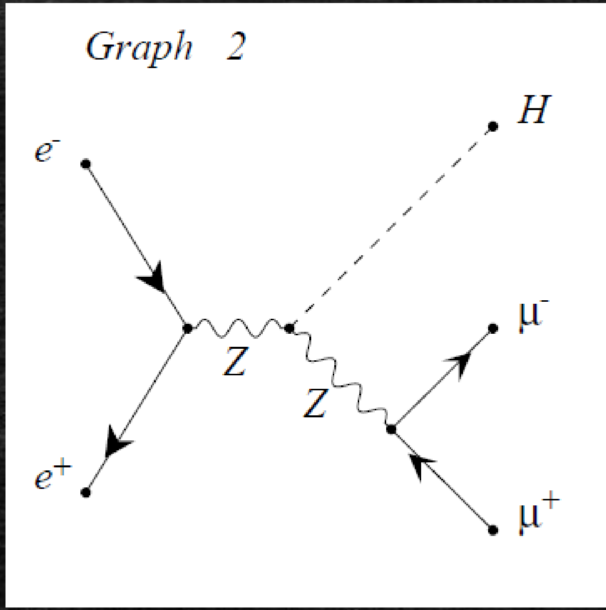
- Higgs physics @ $\sqrt{s}$=250 GeV
  - $e^+e^- \to H\mu^+\mu^-$
  - $e^+e^- \to H\overline{\nu}_e \nu_e$

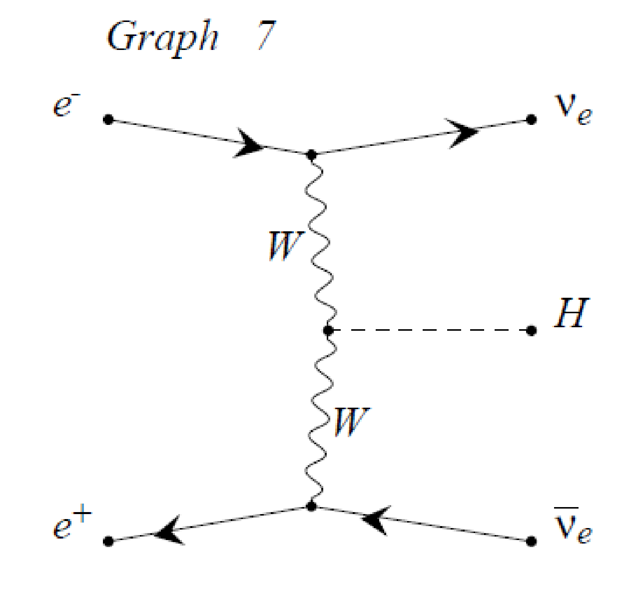




GRACE for ILC physics

- Higgs physics @ $\sqrt{s}$=250 GeV
  - $e^+e^- \to H\mu^+\mu^-$
  - $e^+e^- \to H\overline{\nu}_e \nu_e$
- Beam polarization
  - e^- (80%) , e^+(30%)





Projection operator: $P_\lambda = \frac{1}{2}(1 + \lambda\gamma_5 \not{p}/m)$

GRACE for ILC physics

- Beam polarization
  - e^- (80%) , e^+(30%)

Projection operator: $P_\lambda = \frac{1}{2}(1 + \lambda\gamma_5 \not{p}/m)$

⬇

Code optimization

Code optimization using FORM

Code optimization using FORM

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Algorisms:

- Horner's method (Monte Carlo tree search)
- Common subexpression elimination
- Greedy optimizations
- Partial factorization
- Recycling variables

Code optimization using FORM

A numerator of loop amplitudes are polynomials of:

- Coupling constants:
- Kinematical invariants:
- Feynman parameters:
- Gauge parameters:
- Polarization parameters:

⟹ $x_1, x_2, \ldots, x_n$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

$e^+e^- \rightarrow H\mu^+\mu^-$

2235 loop diagrams

```
*
*     Define couplings
*
      xcp1=cael(1)**4*chmu(1)*cwne(1)**2*chww
*
      ztd = cc1
      CALL snprpdn(pphase,ztd,vn7,ama**2,ama*aga)
      CALL snprpdn(pphase,ztd,vn15,ammu**2,0.0d0)
      CALL snprpdc(pphase,ztd,vn7,ama**2,ama*aga)
      CALL snprpdc(pphase,ztd,vn14,ammu**2,0.0d0)
      ztd = xcp1/ztd
*
      xre=-32*ammu*amel2*ammu2**2-96*ammu*amel2**2*ammu2-32*ammu*
     &    amel2**3+8*ammu*amh2*amel2*ammu2+16*ammu*amh2*amel2**2-16*
     &    lam2*amel*ammu*amel2*es234S2-16*lam2*amel*ammu*amel2*es134S2+
     &    32*lam2*amel*ammu*amel2*ammu2*e4S2+32*lam2*amel*ammu*amel2*
     &    ammu2*e3S2+32*lam2*amel*ammu*amel2**2*e4S2-16*lam2*amel*ammu*
     &    amh2*amel2*e3S2-16*lam1*amel*ammu*amel2*es234S1-16*lam1*amel*
     &    ammu*amel2*es134S1+32*lam1*amel*ammu*amel2*ammu2*e4S1+32*lam1
     &    *amel*ammu*amel2*ammu2*e3S1+32*lam1*amel*ammu*amel2**2*e4S1-
     &    16*lam1*amel*ammu*amh2*amel2*e3S1-8*lam1*lam2*ammu*ammu2*e2S2
     &    *es134S1+16*lam1*lam2*ammu*ammu2*e2S2*es124S1+8*lam1*lam2*
     &    ammu*ammu2*e2S2*es123S1-8*lam1*lam2*ammu*ammu2*e4S1*es123S2+8
     &    *lam1*lam2*ammu*ammu2*e3S1*es124S2+8*lam1*lam2*ammu*ammu2*
     &    e2S1*es123S2+8*lam1*lam2*ammu*ammu2*e1S1*es234S2+16*lam1*lam2
     &    *ammu*ammu2*e1S1*es124S2+16*lam1*lam2*ammu*ammu2*e1S1*es123S2
     &    -32*lam1*lam2*ammu*ammu2**2*e2S1*e1S2-32*lam1*lam2*ammu*
     &    ammu2**2*e1S1*e2S2
      xre=xre+16*lam1*lam2*ammu*amel2*e2S2*es124S1-16*lam1*lam2*ammu*
     &    amel2*e4S1*es123S2+16*lam1*lam2*ammu*amel2*e3S1*es124S2+16*
```

Code optimization using FORM

Horner's form:

$$a(x) = \sum_{i=0}^{n} a_i x^i = a_0 + x(a_1 + x(a_2 + x(\ldots + x \cdot a_n))).$$

Horner's form (multi-parameters):

$$a(x,y,z) = y - 3x + 5xz + 2x^2yz - 3x^2y^2z + 5x^2y^2z^2$$

18 mult. & 5 add.

$$a(x,y,z) = y + x(-3 + 5z + x(y(2z + y(z(-3 + 5z)))))$$

8 mult. & 5 add.

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

Horner's form: $$a(x) = \sum_{i=0}^{n} a_i x^i = a_0 + x(a_1 + x(a_2 + x(\ldots + x \cdot a_n))).$$

Horner's form (multi-parameters):

$$a(x,y,z) = y - 3x + 5xz + 2x^2yz - 3x^2y^2z + 5x^2y^2z^2$$

18 mult. & 5 add.

Not unique!

$$a(x,y,z) = y + x(-3 + 5z + x(y(2z + y(z(-3 + 5z)))))$$

8 mult. & 5 add.

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Monte Carlo tree search

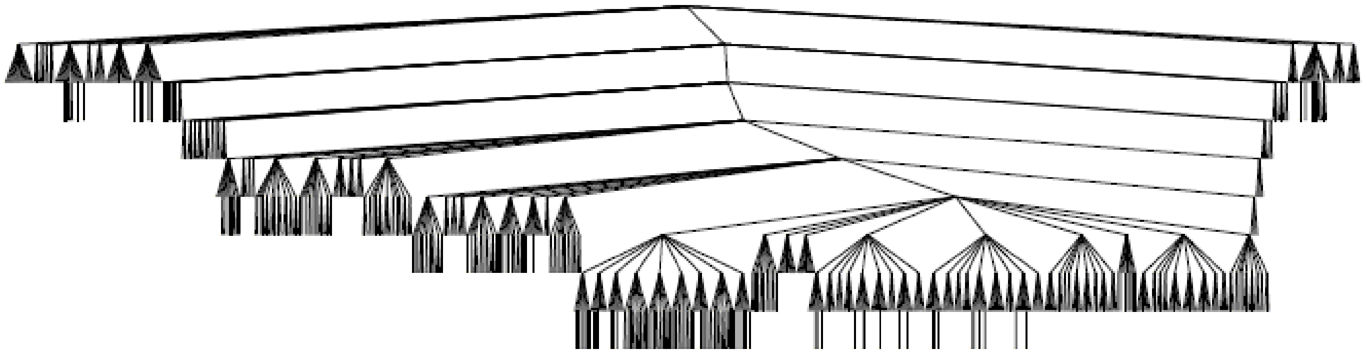


Figure 1: The traversed part of an MCTS tree in the search for a good Horner scheme.

Upper CL for tree:

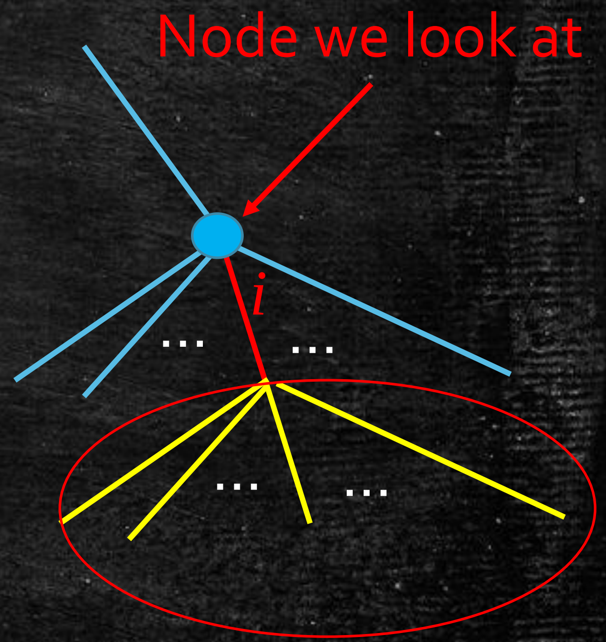
$$UCT_i = \langle x_i \rangle + 2C_p\sqrt{\frac{2\log n}{n_i}},$$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

Upper CL for tree: $$UCT_i = \langle x_i \rangle + 2C_p\sqrt{\frac{2\log n}{n_i}},$$

- $\langle x_i \rangle$ Average score of child i over the previous travels
- n_i Number of visits of child i
- n Number of visits for the node
- C_p Empirical constant



J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Common subexpression elimination

$$a(x,y,z) = y + x(-3+5z) + x(y(2z + y(z(-3+5z)))))$$

$$Z_1 = -3 + 5z$$
$$a = y + x(Z_1 + x(y(2z + y(zZ_1)))).$$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Common subexpression elimination

$$w^2y + w^2z + wx + wy + wz$$

$$\begin{aligned}
Z_1 &= w^2 \\
Z_2 &= y + z \\
Z_3 &= Z_1 * Z_2 \\
Z_4 &= x + y + z \\
Z_5 &= w * Z_4 \\
a &= Z_3 + Z_5
\end{aligned}$$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Common subexpression elimination

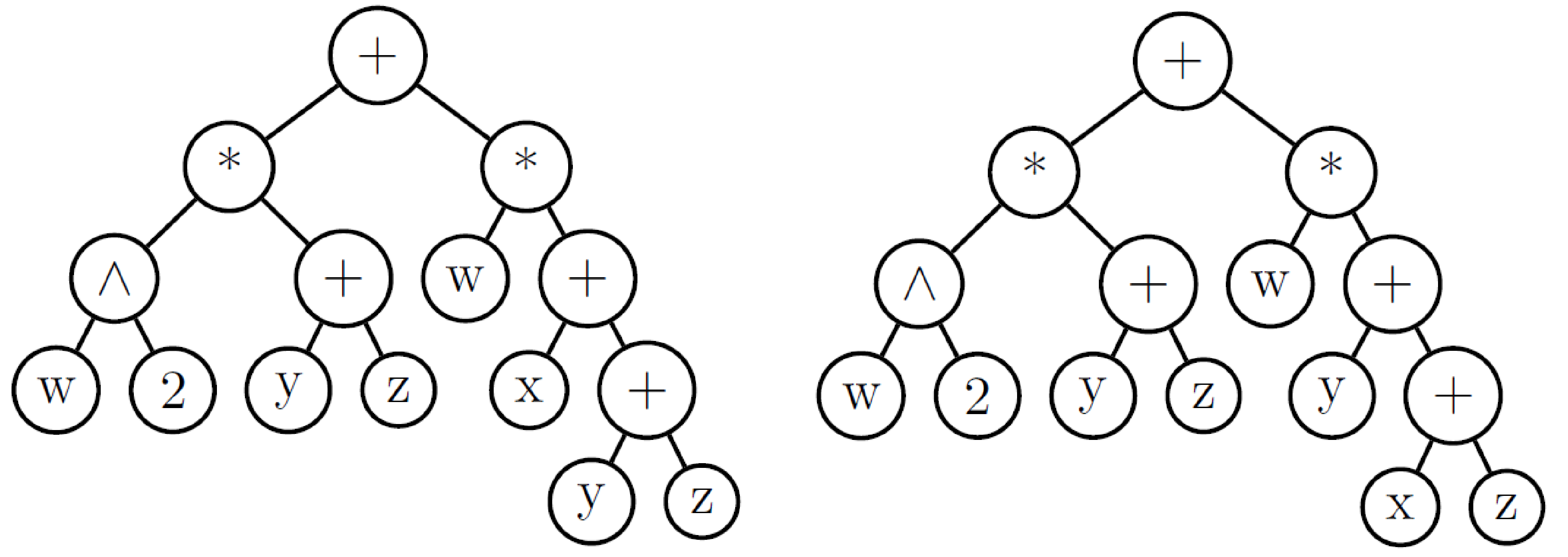
$$w^2y + w^2z + wx + wy + wz$$

$$\begin{aligned} Z_1 &= w^2 \\ Z_2 &= y + z \\ Z_3 &= Z_1 * Z_2 \\ Z_4 &= x + y + z \\ Z_5 &= w * Z_4 \\ a &= Z_3 + Z_5 \end{aligned} \quad \longrightarrow \quad \begin{aligned} Z_1 &= w^2 \\ Z_2 &= y + z \\ Z_3 &= Z_1 * Z_2 \\ Z_4 &= x + Z_2 \\ Z_5 &= w * Z_4 \\ a &= Z_3 + Z_5 \end{aligned}$$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

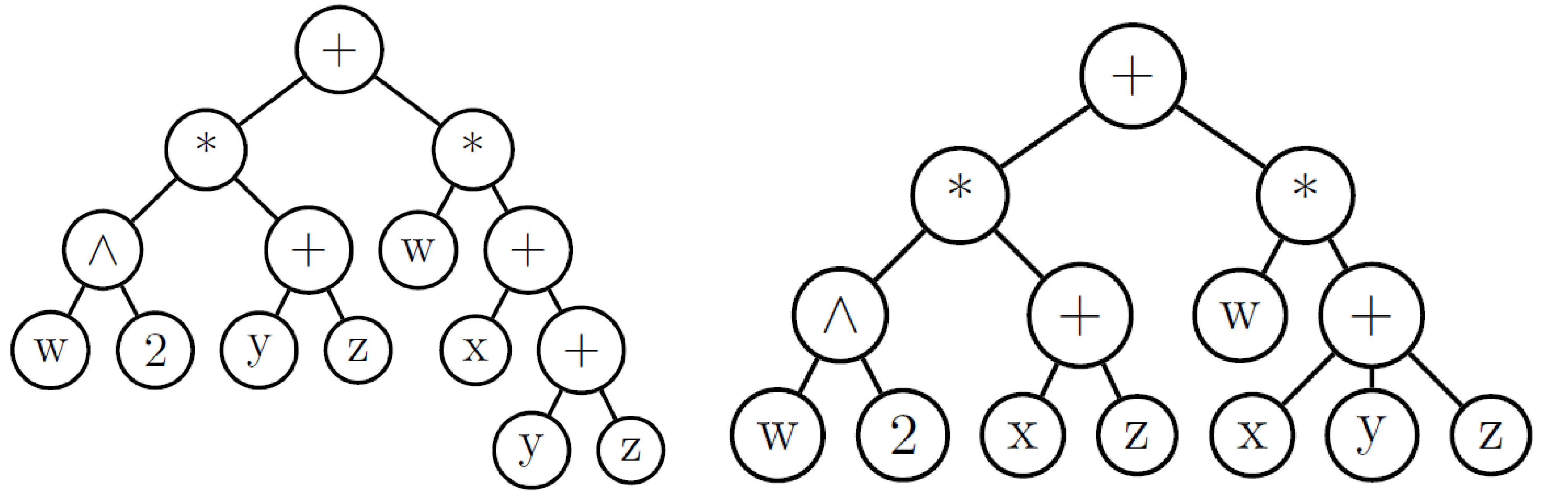
- Greedy optimizations



J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Greedy optimizations



J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Partial factorization

$$
\begin{aligned}
Z_1 &= w * y \\
Z_2 &= w * z \\
Z_3 &= x * y \\
Z_4 &= x * z \\
a &= Z_1 + Z_2 + Z_3 + Z_4
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
Z_5 &= y + z \\
Z_6 &= w * Z_5 \\
Z_7 &= y + z \\
Z_8 &= x * Z_7 \\
a &= Z_6 + Z_8
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
Z_5 &= y + z \\
Z_6 &= w * Z_5 \\
Z_8 &= x * Z_5 \\
a &= Z_6 + Z_8
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
Z_5 &= y + z \\
Z_9 &= w + x \\
a &= Z_5 * Z_9
\end{aligned}
$$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Partial factorization

$$
\begin{aligned}
Z_1 &= w * y \\
Z_2 &= w * z \\
Z_3 &= x * y \\
Z_4 &= x * z \\
a &= Z_1 + Z_2 + Z_3 + Z_4
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
Z_5 &= y + z \\
Z_6 &= w * Z_5 \\
Z_7 &= y + z \\
Z_8 &= x * Z_7 \\
a &= Z_6 + Z_8
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
Z_5 &= y + z \\
Z_6 &= w * Z_5 \\
Z_8 &= x * Z_5 \\
a &= Z_6 + Z_8
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
Z_5 &= y + z \\
Z_9 &= w + x \\
a &= Z_5 * Z_9
\end{aligned}
$$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Partial factorization

$$
\begin{array}{llll}
Z_1 = w * y & Z_5 = y + z & Z_5 = y + z & \\
Z_2 = w * z & Z_6 = w * Z_5 & Z_6 = w * Z_5 & Z_5 = y + z \\
Z_3 = x * y \quad \longrightarrow & Z_7 = y + z \quad \longrightarrow & Z_8 = x * Z_5 \quad \longrightarrow & Z_9 = w + x \\
Z_4 = x * z & Z_8 = x * Z_7 & a = Z_6 + Z_8 & a = Z_5 * Z_9 \\
a = Z_1 + Z_2 + Z_3 + Z_4 & a = Z_6 + Z_8 & &
\end{array}
$$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

- Recycling variables

$$
\begin{aligned}
Z_1 &= w^2 \\
Z_2 &= y + z \\
Z_3 &= Z_1 * Z_2 \\
Z_4 &= x + Z_2 \\
Z_5 &= w * Z_4 \\
a &= Z_3 + Z_5
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
Z_1 &= w^2 \\
Z_2 &= y + z \\
Z_1 &= Z_1 * Z_2 \\
Z_2 &= x + Z_2 \\
Z_2 &= w * Z_2 \\
a &= Z_1 + Z_2
\end{aligned}
$$

J. Kuipers, T. Ueda, J. Vermaseren, arXiv:1310.7007

Code optimization using FORM

$e^+e^- \to H\mu^+\mu^-$

2235 loop diagrams

```
      xy(12) = 2*e2e1a2+2*e3e2a2+2*e3e1a2-2*xb(1)*xb(3)+4*xb(3)*ammu2+
     & xb(3)*xb(69)-4*xb(56)*e2e1
      xy(13) = 2*xb(1)*xb(66)-xb(3)*xb(72)-2*xb(46)*e3e1-2*xb(53)*e3e2+
     & 4*xb(54)*e3e1-2*xb(62)*e2e1
      xy(14) = 3*xb(3)+xb(77)
      xy(15) = 2*xb(7)-xb(50)+2*xb(75)

*
      ztd = cc1
      CALL snprpdn(pphase,ztd,vn7,ama**2,ama*aga)
      CALL snprpdn(pphase,ztd,vn15,ammu**2,0.0d0)
      CALL snprpdc(pphase,ztd,vn7,ama**2,ama*aga)
      CALL snprpdc(pphase,ztd,vn14,ammu**2,0.0d0)
      ztd = xcp1/ztd
*
      ztd = ztd*(-ammu*cc2)
*
      xre=+cc2*(xv(1))
      g0e0(0,0)=g0e0(0,0)+ztd*xre
      xim=+cc4*(-xz(14))
      g0e0(0,0)=g0e0(0,0)+ztd*zimg*xim
      xre=+cc24*(xz(2)*xz(10)*xy(11))
      xre=xre+cc16*(xz(1)*xz(10)*xz(13)-xz(7)*xz(9))
      xre=xre+cc8*(xb(3)*xb(51)*xz(1)*xz(11)-xb(5)*xz(13)+xb(41)*xz(1)*
     &    xz(12)+xb(51)*xz(2)*xz(11)+xz(2)*xz(12)*xy(6)-xz(5)*xy(8)+xz(
     &    9)*xy(7))
      xre=xre+cc4*(xz(1)*xz(9)*xy(2)-xz(1)*xz(10)*xy(13)-xz(7)*xy(12)-
     &    xz(8)*xy(15))
      xre=xre+cc2*(xz(1)*xz(13)*xy(1)+xz(2)*xy(3)+xz(4)*xy(4))
      xre=xre-xz(1)*xy(5)-xy(9)
      g0e0(0,1)=g0e0(0,1)+ztd*xre
```

Code optimization using FORM

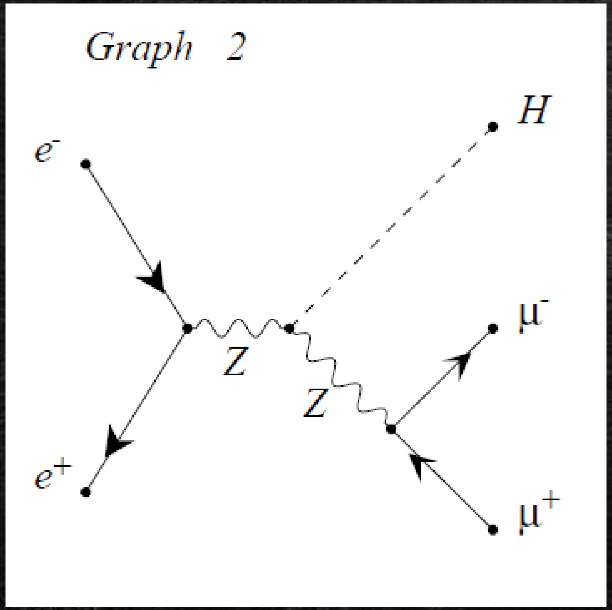
$e^+e^- \to H\mu^+\mu^-$ 2235 loop diagrams

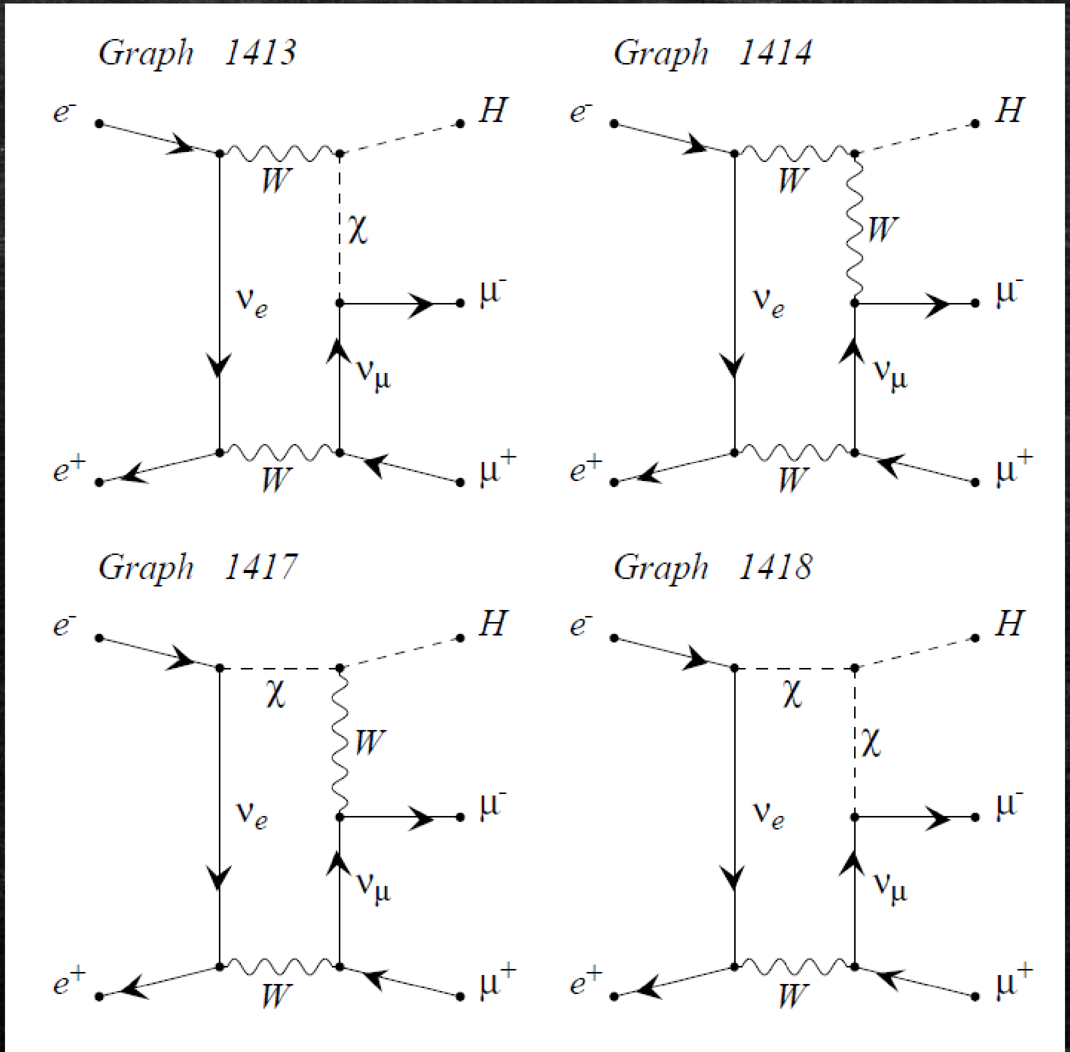
			Pol 0	Pol 1
Opt	0	src	1.0GB	5.9GB
		obj	4.9GB	31.5GB
	1	src		6.3GB
		obj		3.3GB

Intel FORTRAN compiler v15 Optimization 2

Physics Results

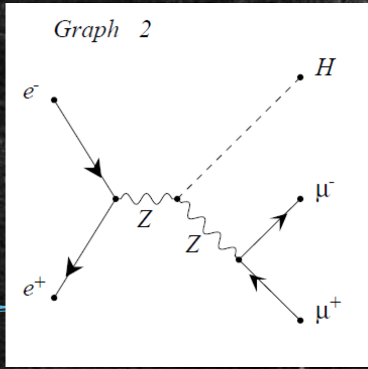
Physics results: $e^+e^- \to H\mu^+\mu^-$



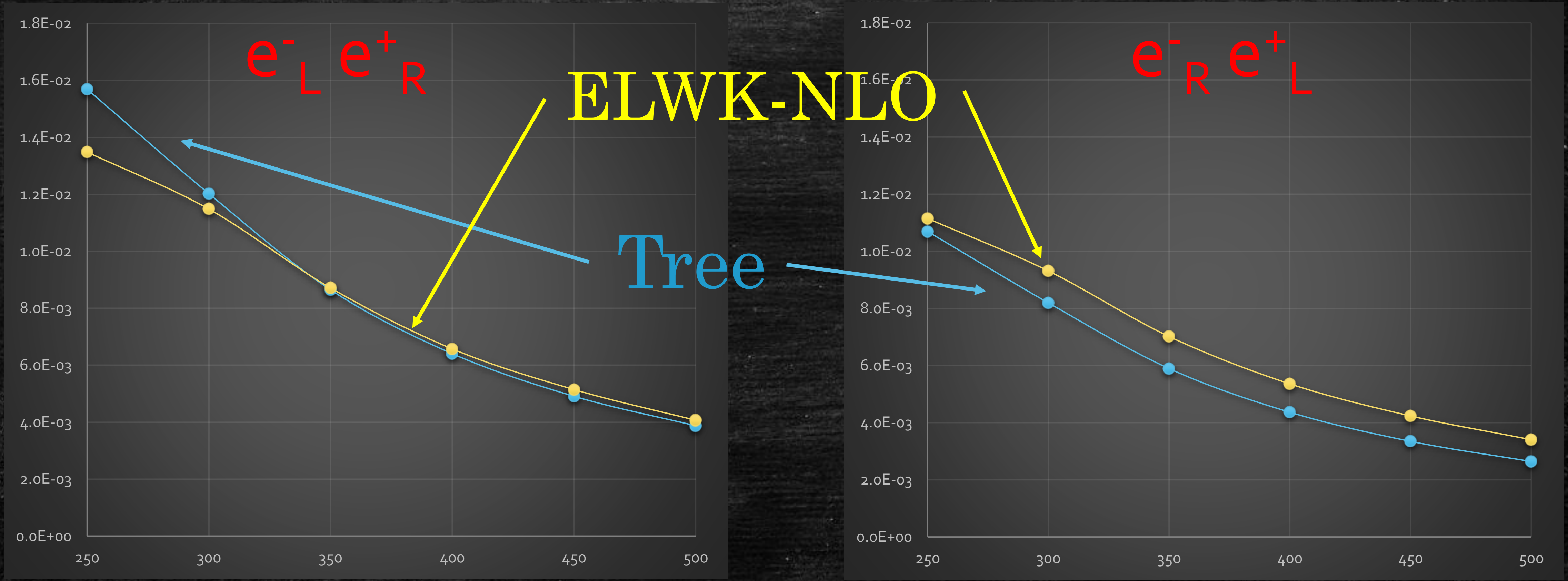


2235 loop diagrams

Physics results: $e^+e^- \to H\mu^+\mu^-$



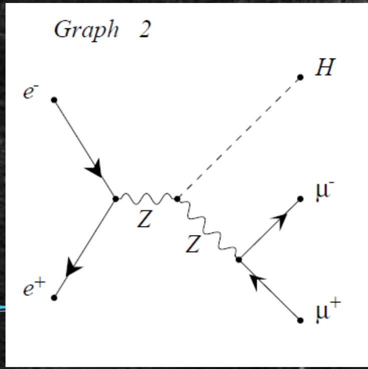
Cross section (pb)

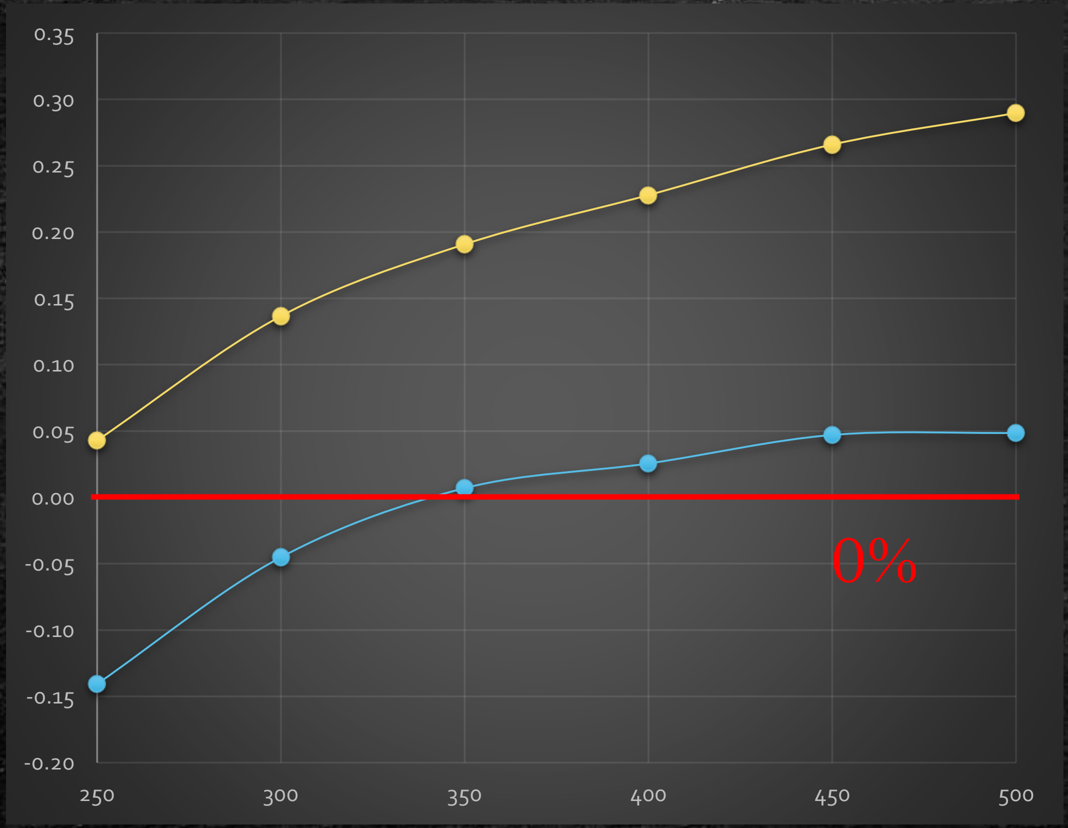


√s GeV

√s GeV

Physics results: $e^+e^- \to H\mu^+\mu^-$





δ_{NLO}

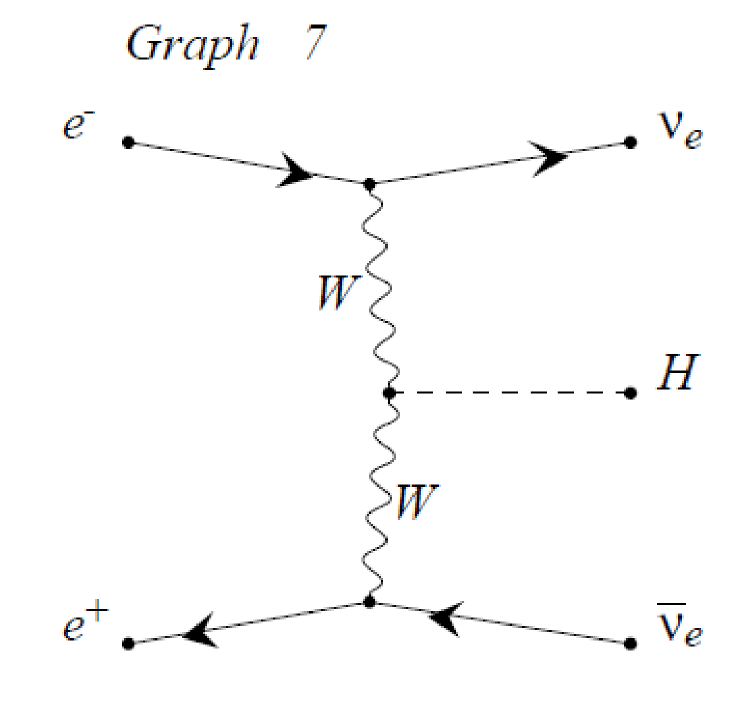
$\sqrt{s}$ GeV

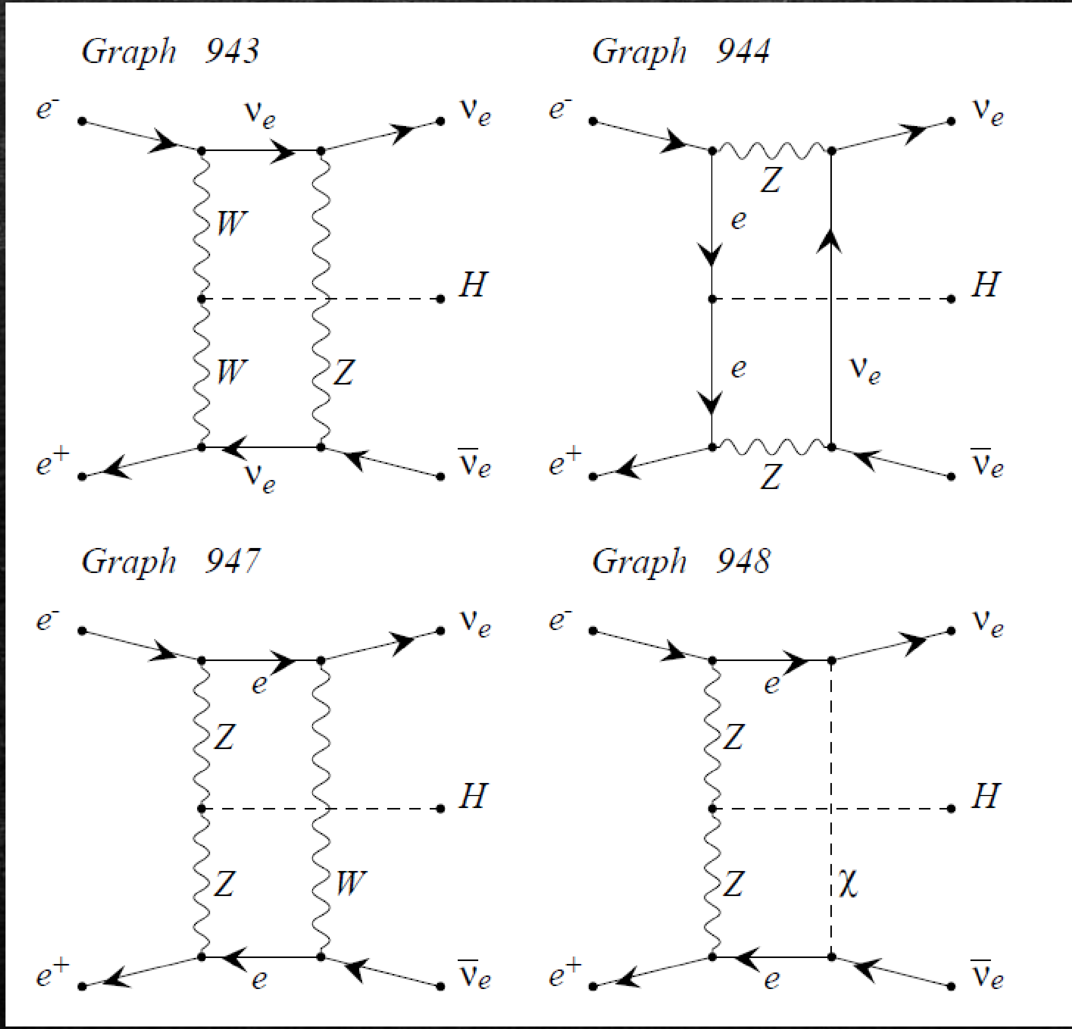
$e^-_R\, e^+_L$

$e^-_L\, e^+_R$

$\delta_{NLO}=(\sigma_{NLO}-\sigma_{Tree})/(\sigma_{NLO}+\sigma_{Tree})$

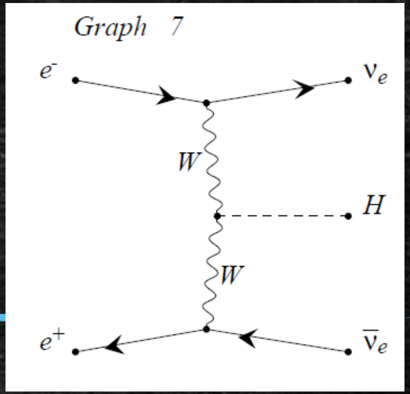
Physics results: $e^+e^- \to H\nu_e \bar{\nu}_e$

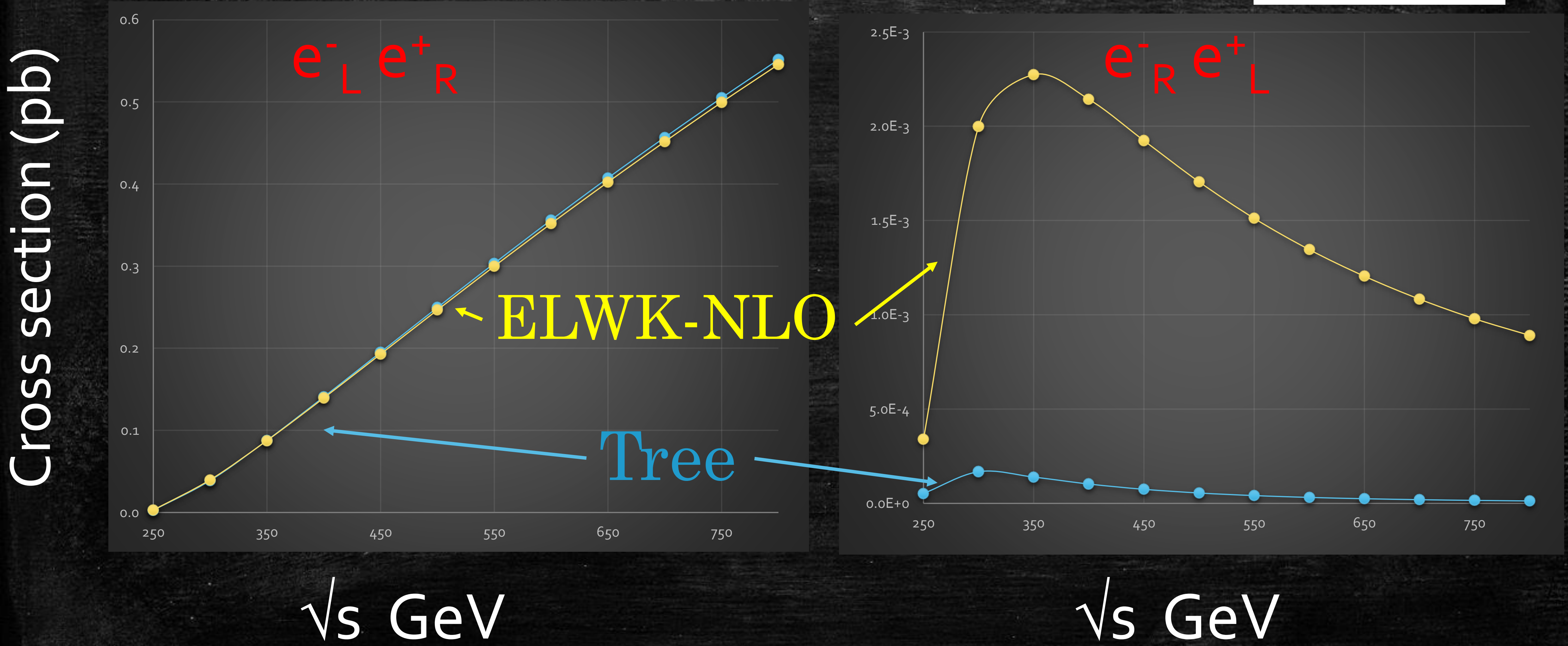




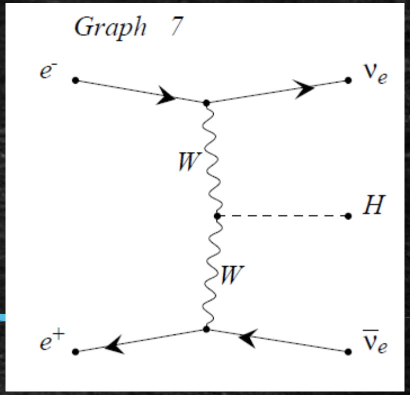
1350 loop diagrams

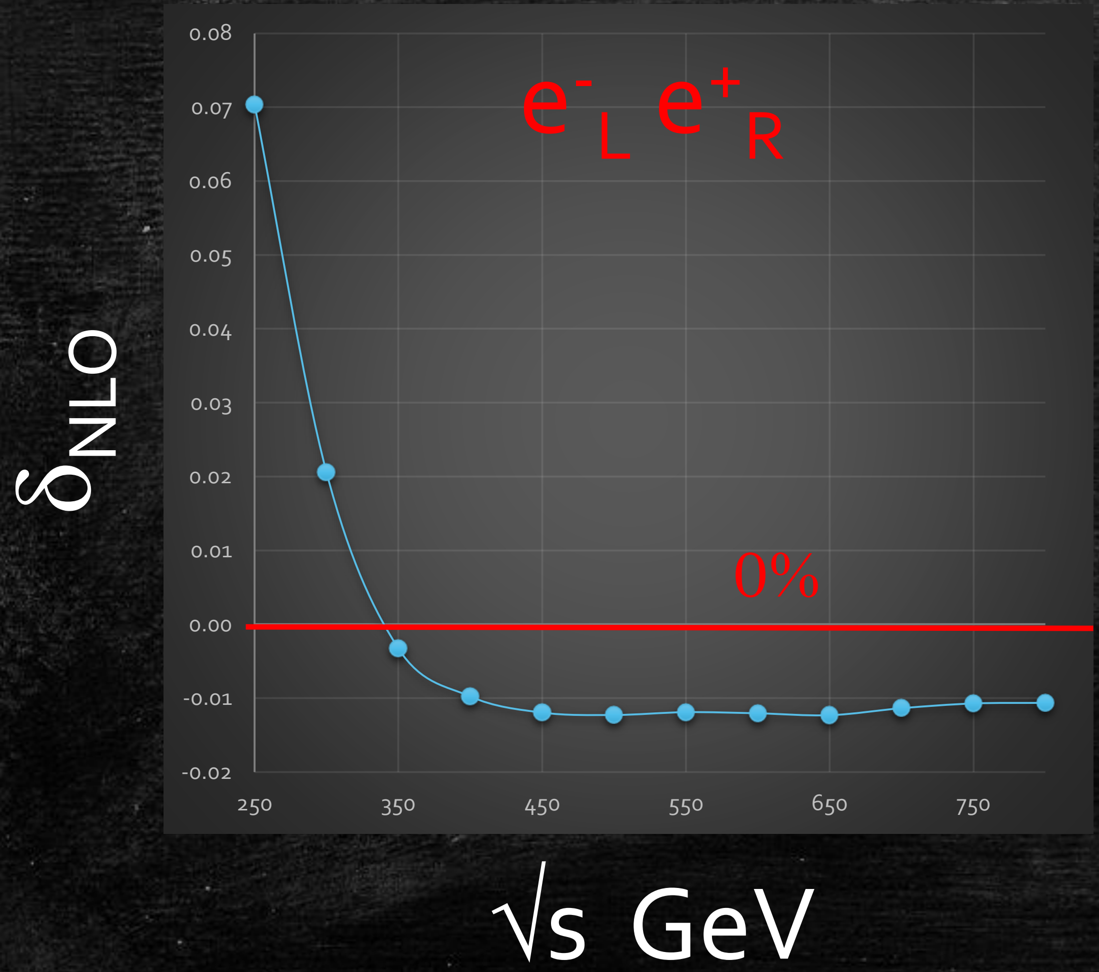
Physics results: $e^+e^- \rightarrow H\nu_e \bar{\nu}_e$

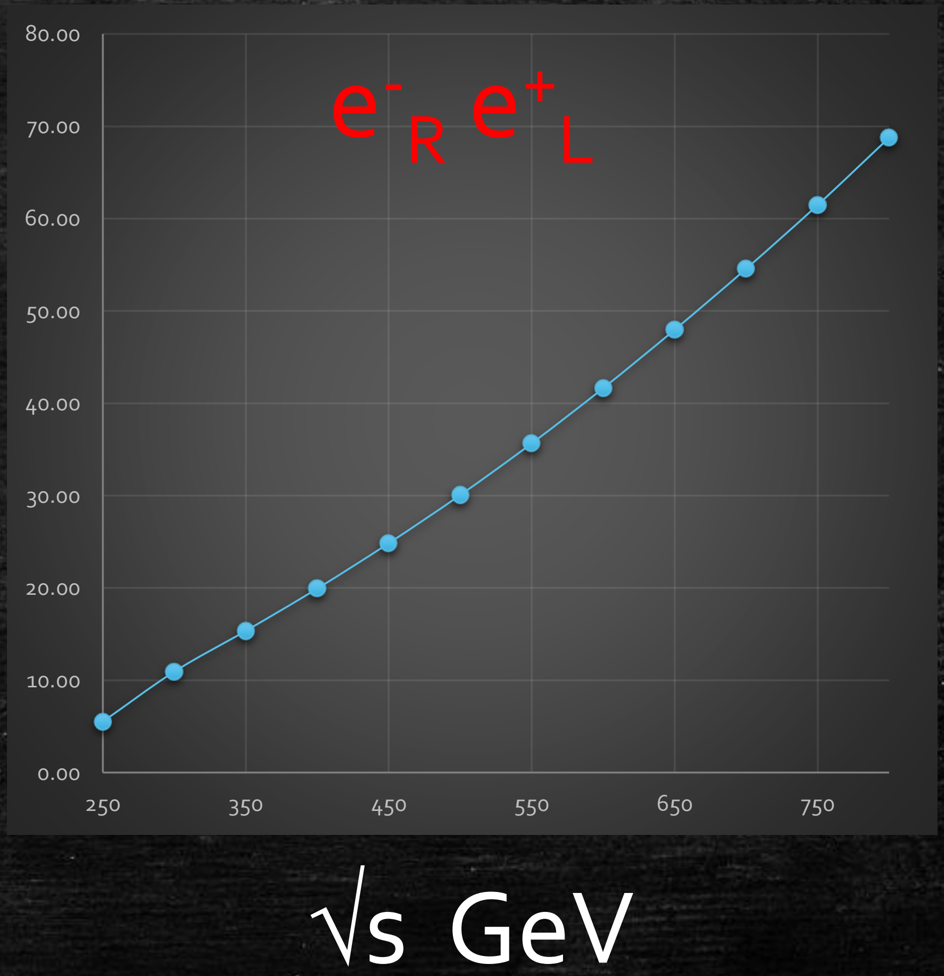




Physics results: $e^+e^- \to H\nu_e \nu_e$







Summary

Summary

- GRACE:
  - Full $O(\alpha)$ electroweak corrections w/ Beam polarization effects
    - Code optimization by FORM: object module ~1/10
    - Z-Higgs production
    - W-fusion