ANN Input

IO startup

D. Keller

The Real Fit Function

$$\sigma_{UU} = \sigma_{UU}^{DVCS} + \sigma_{UU}^{BH} + \sigma_{UU}^{\mathcal{I}}$$

Fitting parameter

$$\sigma_{UU}^{BH} = \frac{\Gamma}{t}\left[A_{UU}^{BH}\left(F_1^2 + \tau F_2^2\right) + B_{UU}^{BH}\tau G_M^2(t)\right]$$

Elastic Form Factors

$$F_1 = \frac{2.7928}{(1-1.40716\,t)^2} - F_2 \qquad F_2 = \frac{1.7928}{(1-1.40716\,t)^2\left(1-\frac{t}{4M^2}\right)}$$

$$A_{UU}^{BH} = \frac{16\,M^2}{t(kq')(k'q')}\left[4\tau\Big((kP)^2 + (k'P)^2\Big) - (\tau+1)\Big((k\Delta)^2 + (k'\Delta)^2\Big)\right]$$

$$B_{UU}^{BH} = \frac{32\,M^2}{t(kq')(k'q')}\left[(k\Delta)^2 + (k'\Delta)^2\right]$$

$$\sigma_{UU}^{\mathcal{I}} = \frac{\Gamma}{Q^2(-t)}\Big[A_{UU}^{\mathcal{I}}\left(F_1\Re e\mathcal{H} + \tau F_2\Re e\mathcal{E}\right) + B_{UU}^{\mathcal{I}}G_M\left(\Re e\mathcal{H} + \Re e\mathcal{E}\right) + C_{UU}^{\mathcal{I}}G_M\Re e\widetilde{\mathcal{H}}\Big]$$

Compton Form Factors (CFFs)

Fitting parameters

~small

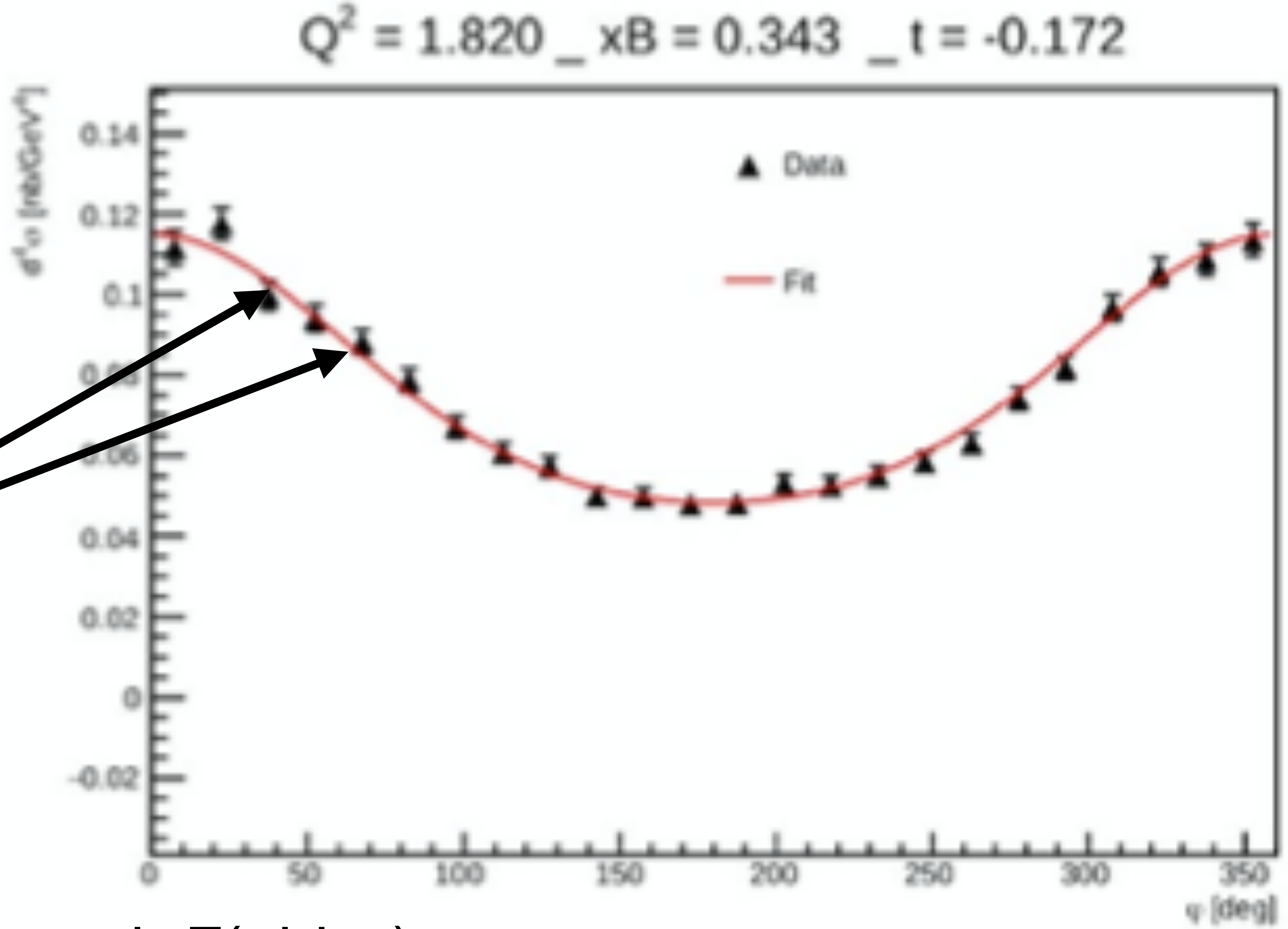
$$\begin{aligned}A_{UU}^{\mathcal{I}} = -\frac{4}{(kq')(k'q')}\Big\{&(Q^2+t)\Big[2((kP)+(k'P))(kk)_T + (Pq)(kq')_T + 2(k'P)(kq')\\&-2(kP)(k'q') + (k'q')(kP)_T + (kq')(k'P)_T - 2(kk')(kP)_T\Big]\\&+(Q^2-t+4(k\Delta))\Big[(Pq')((kk')_T + (kq')_T - 2(kk'))\\&+2(kk')(Pq')_T - (k'q')(kP)_T - (kq')(k'P)_T\Big]\Big\}\cos\phi\end{aligned}$$

$$\begin{aligned}B_{UU}^{\mathcal{I}} = \frac{2\xi}{(kq')(k'q')}\Big\{&(Q^2+t)\Big[2(kk)_T((k\Delta)+(k'\Delta)) + (kq')_T\Big((q\Delta) - (kq') - (k'q')\\&+2(kk')\Big) + 2(kq')(k'\Delta) - 2(k'q')(k\Delta)\Big] + (Q^2-t+4(k\Delta))\Big[((kk)_T\\&-2(kk'))(q'\Delta) - (kk')\Delta_T^2 - 2(k\Delta)_T(kq')\Big]\Big\}\cos\phi\end{aligned}$$

The Cross Section

$$\frac{Acos^2(\phi) + Bcos(\phi) + C}{Dcos^2(\phi) + Ecos(\phi) + F} \quad cos(\phi)$$

- See github: BHDVCS.py



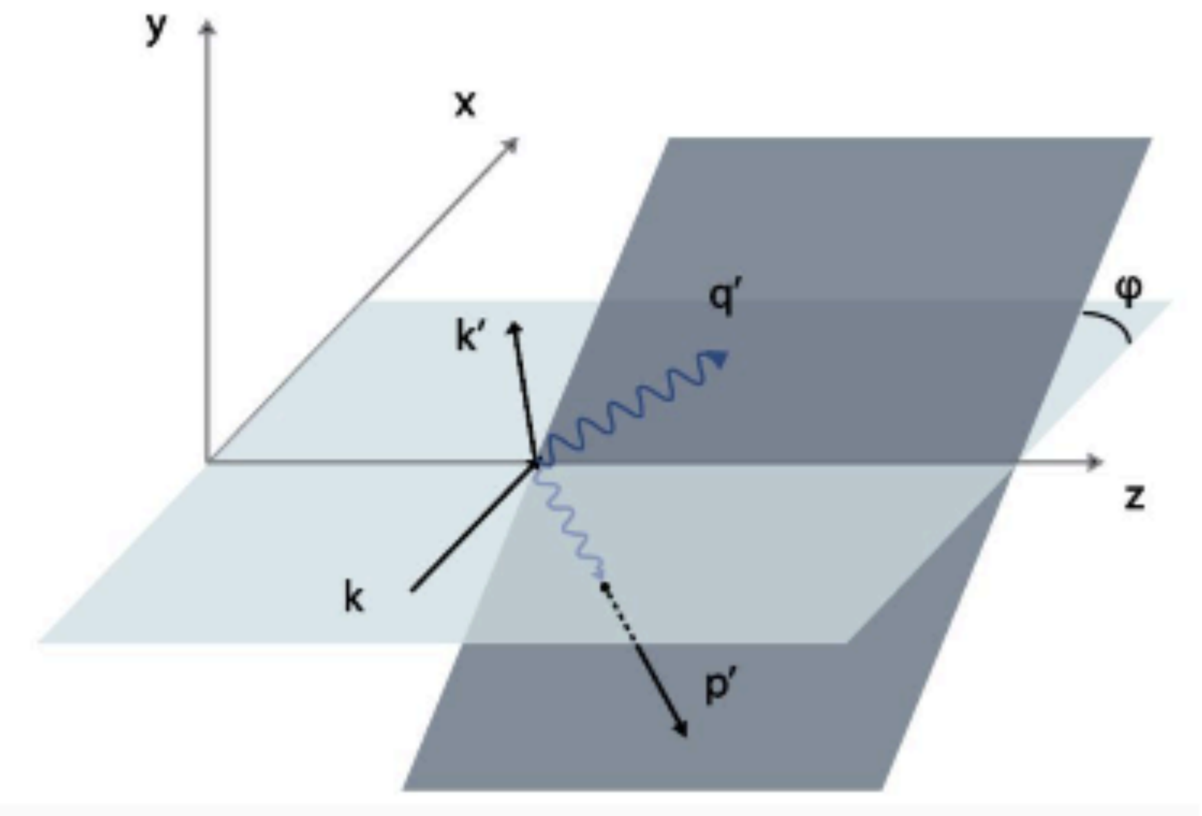
- F(phi_x)

- errF: error at each F(phi_x)

Inputs

Kinematic settings

$$ep \to e'p'\gamma$$



4-momentum vectors

- **k** incoming electron
- **k'** outgoing electron
- **q** virtual exchange photon
- **q'** outgoing photon
- **p'** outgoing proton

$$d\sigma(k, x_B, t, Q^2, \phi)$$

- k Energy of the incoming electron
- Q^2 Electron squared momentum transfer
- t Squared momentum transfer to the proton
- x_B Bjorken variable

$$x_B = \frac{Q^2}{2(pq)}$$

Determines the momentum fraction of the quark or gluon on which the photon scatters.

- �� Azimuthal angle between the hadron plane formed by the outgoing proton and photon and the lepton plane

Elastic Form Factors

$$F_1 = \frac{2.7928}{(1 - 1.40716\,t)^2} - F_2 \qquad F_2 = \frac{1.7928}{(1 - 1.40716\,t)^2 \left(1 - \frac{t}{4M^2}\right)}$$

- k
- QQ
- t
- x_b
- phi_x
- F and errF
- F1 and F2

In/Out

Starting point

- Use k, QQ, x_b, t as inputs to F(phi_x) for several phi_x points
- errF is the Gaussian error in F at those several phi_x points
- So first start with fixed kinematics to produce a fit: a minimization in errF using F(k,QQ,x_b,t,phi_x) for fixed k, QQ, x_b, t over 0-360 in phi_x
- F1 and F2 are simply calculated values that go into the function and are the same for fixed kinematics
- After this step compare your results to a chi^2 minimization fit of the same thing

The Function

Branch: master ▾ ptgroupneuralnet / Compton_FF_Code / BHDVCS.py / <> Jump to ▾

Find file | Copy path

tvallar Added untracked files e5999e5 on Sep 29, 2019

1 contributor

235 lines (191 sloc) | 12.5 KB

Raw | Blame | History

```python

import numpy as np
import Lorentz_Vector as lv

class BHDVCS(object):

```

```python

    def TotalUUXS(self, angle, par):
        phi = angle[0]
            # Set QQ, xB, t and k and calculate 4-vector products
        #print(par)
        self.SetKinematics( par[0], par[1], par[2], par[3] )
        self.Set4VectorsPhiDep(phi)
        self.Set4VectorProducts(phi)

        xsbhuu   = self.GetBHUUxs(par[4], par[5])
        xsiuu    = self.GetIUUxs(phi, par[4], par[5], par[6], par[7], par[8])

        tot_sigma_uu = xsbhuu + xsiuu + par[9] # Constant added to account for DVCS contribution
        #print(xsbhuu, " ", xsiuu, " ", tot_sigma_uu)
        return tot_sigma_uu

```

In/Out

The Data Set (this is testing data)

Branch: master ▾ ptgroupneuralnet / Compton_FF_Code / **DVCS_cross.csv**

Find file | Copy path

tvallar Added untracked files

e5999e5 on Sep 29, 2019

1 contributor

3205 lines (3205 sloc) | 339 KB

Raw | Blame | History

Search this file...

Index	k	QQ	x_b	t	phi_x	F	errF	F1	F2	ReH	ReE	ReHTilde
0	2.75	1.72395	0.355194	-0.205368	4.6426	10.4022	1.14418	0.631324	0.95597	0.631324	0.95597	0.852383
1	2.75	1.72395	0.355194	-0.205368	12.2379	10.158	1.11654	0.631324	0.95597	0.631324	0.95597	0.852383
2	2.75	1.72395	0.355194	-0.205368	24.5827	9.22548	1.01274	0.631324	0.95597	0.631324	0.95597	0.852383
3	2.75	1.72395	0.355194	-0.205368	37.8822	7.74295	0.852762	0.631324	0.95597	0.631324	0.95597	0.852383
4	2.75	1.72395	0.355194	-0.205368	44.7016	6.51672	0.71587	0.631324	0.95597	0.631324	0.95597	0.852383
5	2.75	1.72395	0.355194	-0.205368	52.1727	7.56232	0.832748	0.631324	0.95597	0.631324	0.95597	0.852383
6	2.75	1.72395	0.355194	-0.205368	69.6681	6.12349	0.6707	0.631324	0.95597	0.631324	0.95597	0.852383

Before first steps

Sanity Check

- Just do a out of the box chi^2 minimization fit
- Lets choose a single fix kinematics for Liliet to fit
- Compare results and make sure we are all on the same page
- Use resulting error and the true CFF for those points to check results
- Then use Teddys utilities (Numpy, TensorFlow) and check the same
- Then: see if you can improve on Teddys and/or make your own

Start with first fixed setting in DVCS_cross_fixed.csv

After First Steps

Now things get interesting

- Then try additional kinematics
- Full scope of data
- Use ANN (or any ML) to train on dynamic set
- Impose addition linear constraints and use simultaneous fitting
- Multivariate Optimization
- Use real experimental data
- ——> Then next phase

Additional Info

- We can meet weekly (same meeting same time)
- Everyone should have a Github or something
- Help each other and work together
- make sure your hours reflect productivity
- Project site: confluence.its.virginia.edu/display/twist/ANN+Fitting+Project
- Liliet: lc2fc@virginia.edu
- me: dustin@virginia.edu