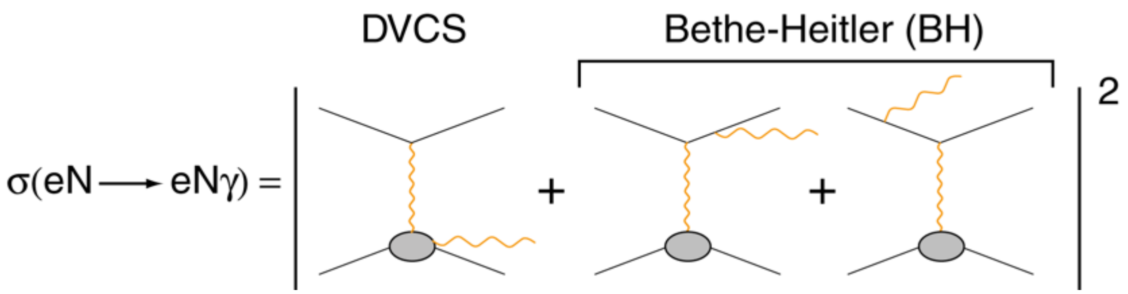
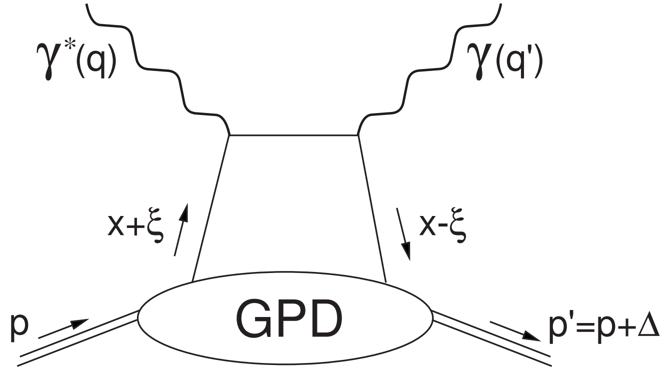
A brief Introduction

Introduction (Some Basics for ANN ML fitting)

- DVCS (Deeply Virtual Compton Scattering)

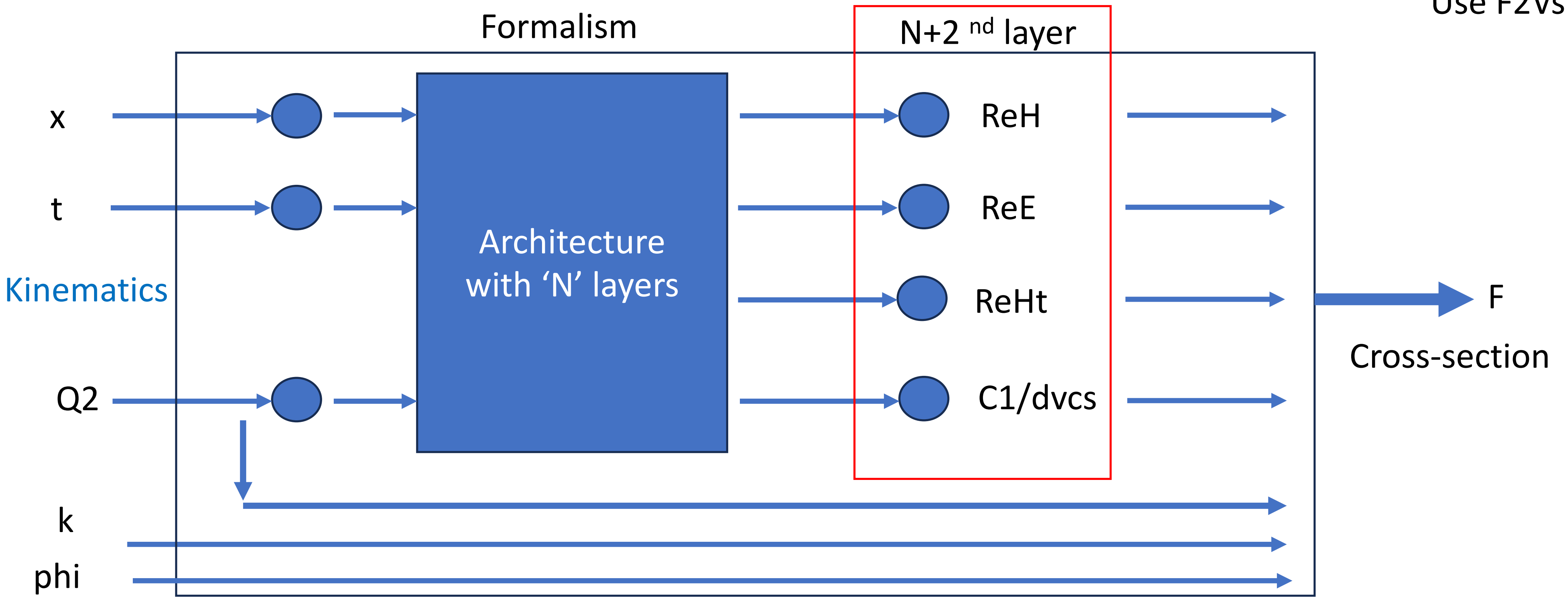




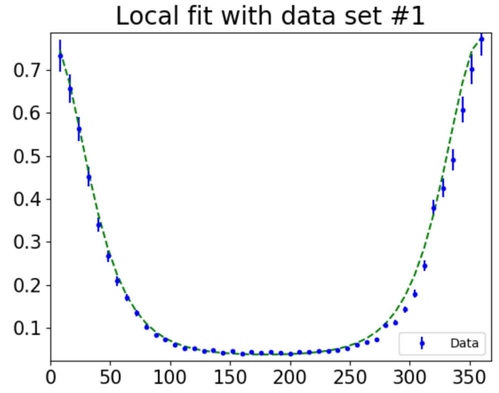
https://arxiv.org/pdf/1903.05742.pdf

- GitHub page : https://github.com/uva-spin/DNN-CFFs
- Steps:
  Generate pseudodata with ‘known’ or ‘True’ Compton Form Factors (CFFs)
  Perform a ‘Fit’ and obtain the fitted CFFs and compare with the ‘True’ CFFs

Local multivariate Inference (LMI) Method



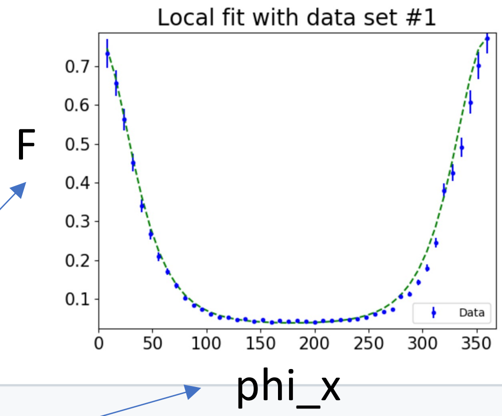
Use F2VsPhi to make plot for F



- Inputs to the "Framework/Formalism" : Kinematics
- Output from the "Framework/Formalism": Total Cross-Section (F)
- Compton Form Factors (CFFs) are considered as outputs (4) from a DNN that takes only 'x', 't', and 'Q2' as inputs.
- A typical data set can be represented as F vs phi while rest of the all kinematics are fixed.
- In the LMI method, we will use multiple data sets with a sparse kinematic phase-space to train the DNN.

See https://confluence.its.virginia.edu/display/twist/The+DNN+Extraction+Approach for more details.

Data file structure (generic)



#Set	index	k	QQ	x_b	t	phi_x	F	sigmaF	varF	F1	F2	ReH	ReE	ReHTilde	dvcs
1.00000	0.00000	5.75000	1.82000	0.34300	-0.17200	7.50000	0.12424	0.00576	0.05000	0.68309	1.09312	-2.56442	2.21195	1.39564	0.03159
1.00000	1.00000	5.75000	1.82000	0.34300	-0.17200	22.50000	0.10715	0.00554	0.05000	0.68309	1.09312	-2.56442	2.21195	1.39564	0.03159
1.00000	2.00000	5.75000	1.82000	0.34300	-0.17200	37.50000	0.10739	0.00517	0.05000	0.68309	1.09312	-2.56442	2.21195	1.39564	0.03159
1.00000	3.00000	5.75000	1.82000	0.34300	-0.17200	52.50000	0.08818	0.00472	0.05000	0.68309	1.09312	-2.56442	2.21195	1.39564	0.03159

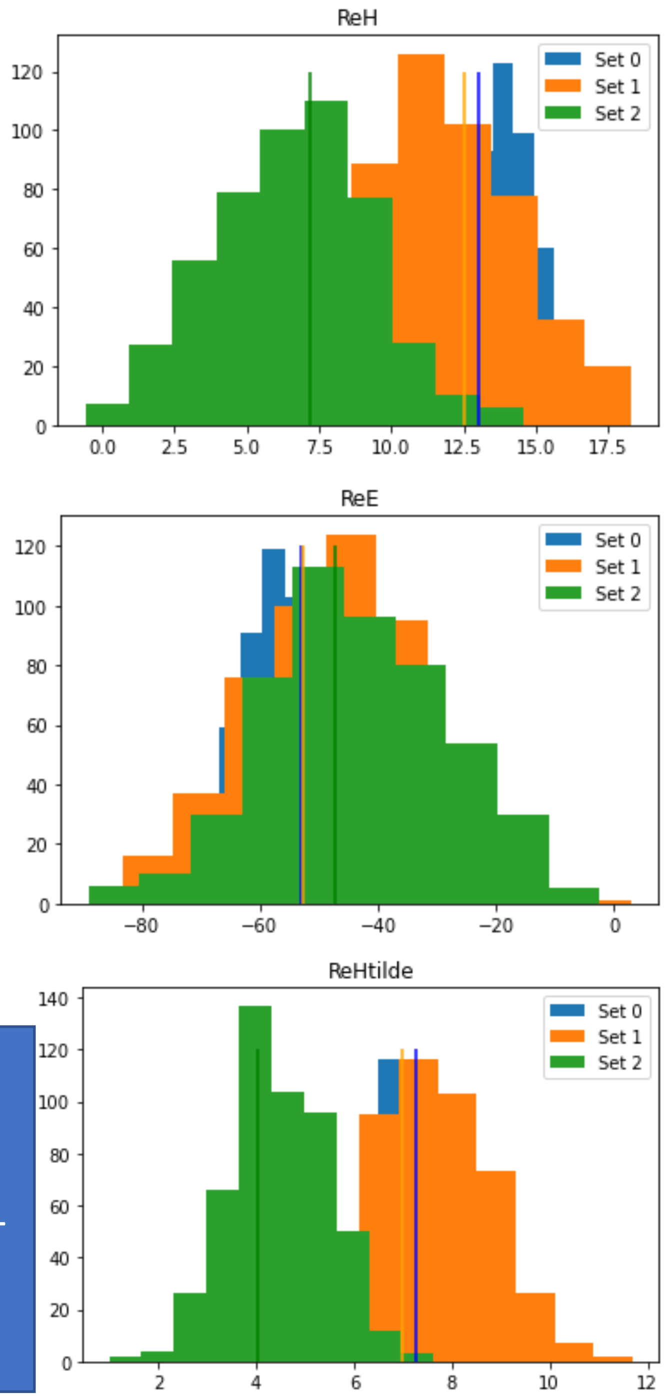
Kinematics

Compton Form Factors (CFFs)

Generate Pseudo-data

- Using the code ‘generate_pseudodata.py’ you can generate a single file of pseudodata (as a .csv file) for a given set of CFFs and kinematic variables.
  You can also vary the number of ‘phi’ bins as a user input in the code so it will determine how many ‘rows’ (or data points) of ‘F’ (total cross-sections) with respect to the angle (‘phi’).
- Similarly, you can generate multiple pseudodata files (.csv files), mimicking multiple experiments.
- For the error/uncertainty of ‘F’ (cross-section), it is better to use values in the order of real experimental measurements.

Generate replicas for the statistical uncertainty



Training Process for LMI method

- In the training process, we need to ensure to feed all 'data sets' that cover a sparce kinematic range.
  Note: If you use only one set of kinematics, then it is called the 'Local Fit'
- Each data set file contains uncertainty column for 'F' (total cross-section)
- Generating replicas = sampling the 'F' value in each row within its uncertainty range. Therefore, you can generate a replica set by dynamically samply 'F' within 'sigmaF'.
- Each 'replica' can be treated as a 'job' and each training of a replica will provide a trained DNN model with evaluated training loss and validation loss. Save those replica DNN models.
- Once you have multiple replica DNN-models, then you can evaluate the statistical uncertainty from the replicas which propagated the uncertainty to the CFFs from 'sigmaF'.

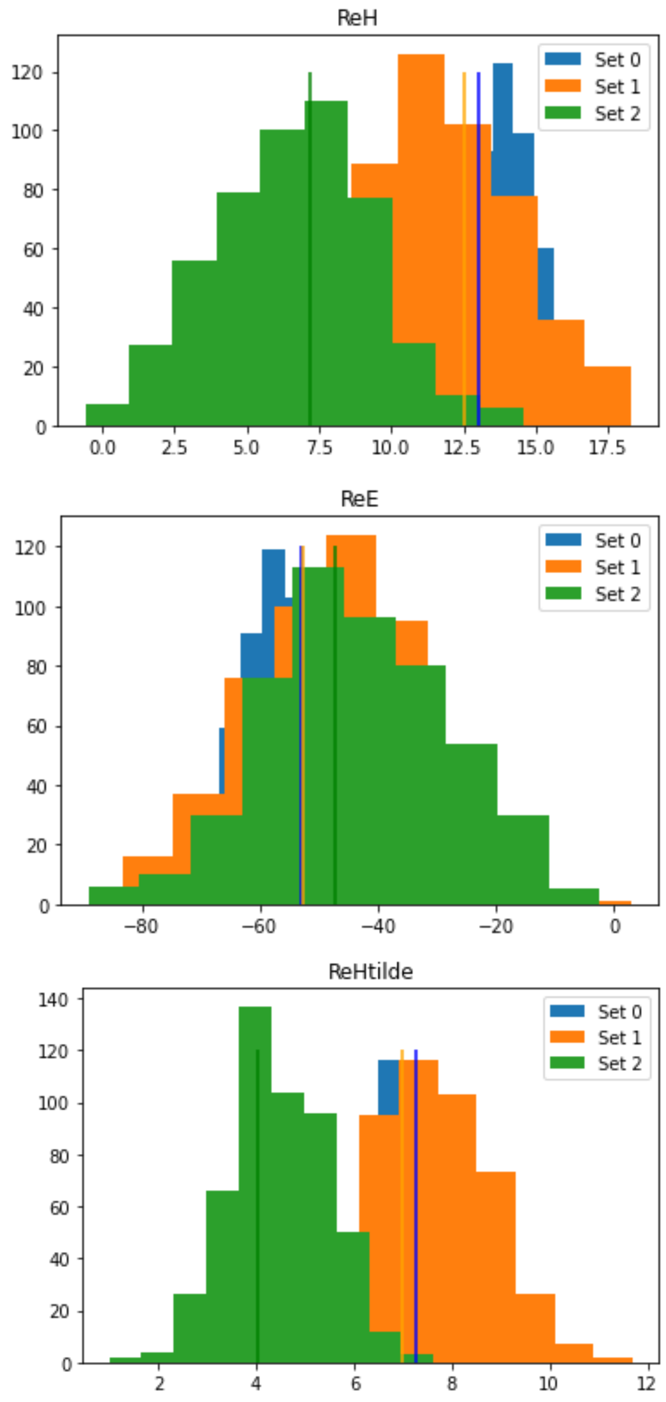
Accuracy

Definition:
How far away the mean of the replicas from the “truth” value ?

$$Accuracy = \left(1 - \frac{|True\ - Mean|}{True}\right)100\%$$

Accuracy is a “quantity” that you can use to evaluate the “improvement” of your architecture (or the configuration of hyperparamters that you ran with)

- You can calculate the “Accuracy” of each CFF for a given set (kinematic-set)
- You can be creative to develop a code to compare Accuracy of CFFs between kinematic sets, within the kinematics (with respect to angle), etc. Think… and propose you ideas..

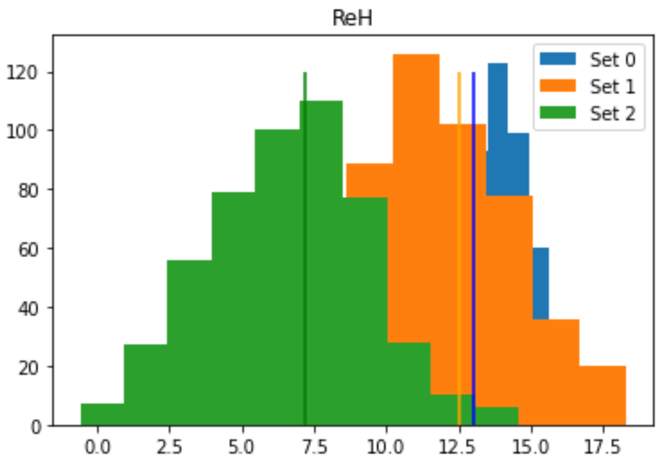


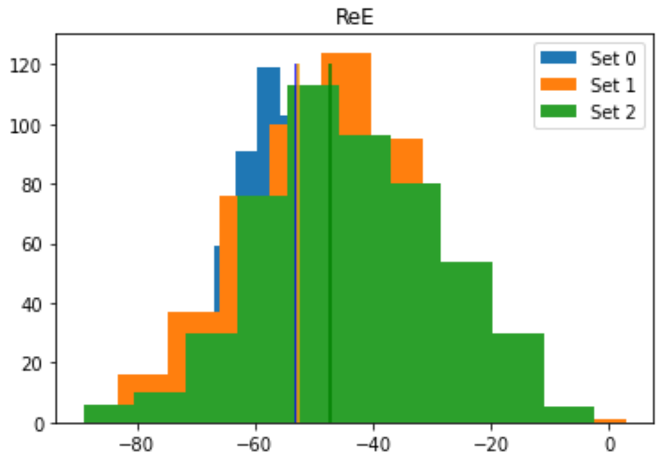
Precision

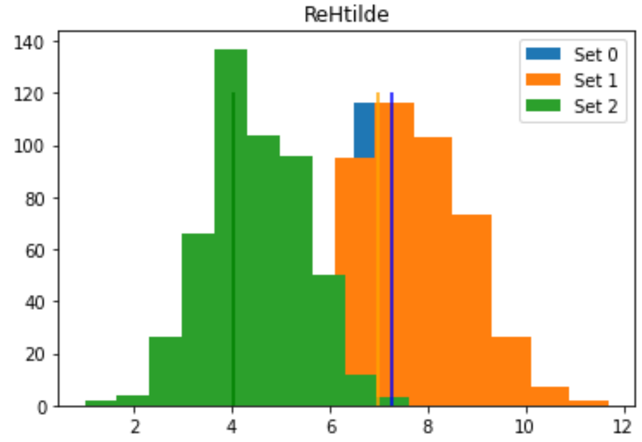
Definition:
How precise the extracted CFF is (standard deviation of the CFFs from all replicas)?

$$Precision = \frac{1}{N\_replicas} \sum_{n=1}^{N\_replica} (Replica(i) \; - Mean)^2$$

Precision is a "quantity" that you can use to evaluate the "improvement" in terms of the statistical uncertainty of your architecture
(or the configuration of hyperparamters that you ran with)







Some resources

- Introduction from Professor Keller
  https://confluence.its.virginia.edu/display/twist/Introduction
- Running jobs on Rivanna
  https://confluence.its.virginia.edu/display/twist/Running+ANN+jobs+in+UVA-Rivanna
- Overleaf
- Discord channel
- Running on ‘Shannon’ (with the options of parallelizing jobs)