EXCLUSIVE SCALAR MESON ELECTROPRODUCTION OFF A SCALAR TARGET

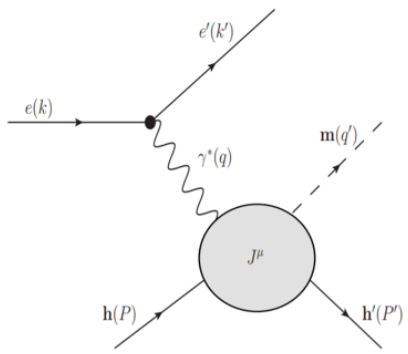
ANDREW LUNDEEN

RESEARCH PRESENTATION, AUGUST 20TH 2021

THEORETICAL MOTIVATIONS AND SUMMARY OF CURRENT STATE OF RESEARCH

Exclusive Meson Electroproduction

$$e(k) + \mathbf{h}(P) \rightarrow e'(k') + \mathbf{h}'(P') + \mathbf{m}(q')$$



$Q^2 = -q^2$

Energy/distance scale of virtual photon probe

$t = (P - P')^2$

Momentum transfer to target

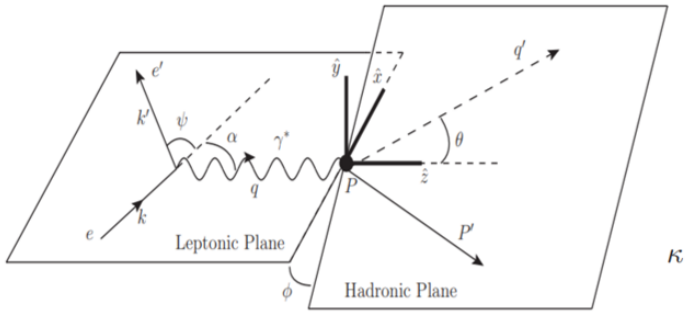
$$x = \frac{Q^2}{2P \cdot q}$$

In the forward limit, momentum fraction of interacting particle

PRELIMINARIES

- Principle: virtual photon scatters off hadron, use as a probe of the hadronic structure by detecting outgoing particles.
- One way of probing is by utilizing factorization theorems which separate the reaction, and the 'soft', long-distance QCD physics can be encapsulated by GPDs, for instance, which are essentially the generalized distributions of the PDFs in DIS.
- More generally one can utilize CFFs: scalar structure functions that tell us contributions from various tensor coefficients. These CFFs are invariant quantities, unlike the GPDs, which rely on 'handbag' dominance. The CFFs should correspond to the GPDs in the forward limit. (The number of CFFs should match the number of GPDs.)

Meson Electroproduction off the Scalar Target: Target Rest Frame (TRF)



$$d\sigma \equiv \frac{d^5\sigma}{dydxdtd\phi_{k'}d\phi_{q'}} = \kappa\langle|\mathscr{M}|^2\rangle$$

$$\kappa \equiv \frac{1}{(2\pi)^5}\frac{yx}{32Q^2\sqrt{1+(\frac{2Mx}{Q})^2}}$$

$$\begin{aligned}\langle|\mathscr{M}|^2\rangle &= \frac{e^4}{q^4}\mathcal{L}^{\mu\nu}\mathcal{H}_{\mu\nu} \\ &= \frac{e^4}{q^4}\left[\frac{2q^2}{\epsilon-1}\langle|\tau_{fi}|\rangle^2 + 2i\lambda\epsilon^{\mu\nu\alpha\beta}k_\alpha k'_\beta J^\dagger_\mu J_\nu\right]\end{aligned}$$

$$J^\mu_{PS} = F_{PS}\epsilon^{\mu\nu\alpha\beta}q_\nu\bar{P}_\alpha\Delta_\beta$$

$$\begin{aligned}J^\mu_S &= F_1(q^2\Delta^\mu - q\cdot\Delta\, q^\mu) \\ &\quad + F_2[(\bar{P}\cdot q + q^2)\Delta^\mu - q\cdot\Delta\,(\bar{P}^\mu + q^\mu)]\end{aligned}$$

COMPTON FORM FACTORS

- The number of CFFs is related to the number of degrees of freedom present in the hadronic interaction. For exclusive scalar meson electroproduction off the scalar target, there is 3 (spin degrees of freedom of the virtual photon), but gauge invariance constrains the current so that there are 2 independent CFFs.
- The advantage of these structure functions lies in their independence of the process in question. The CFFs can be measured experimentally and then be used to predict cross sections for other physics. [See Nagashima, Elementary Particle Physics Vol. 2, pages 325-326 for instance for an intuitive picture of this.]

SIDE NOTE

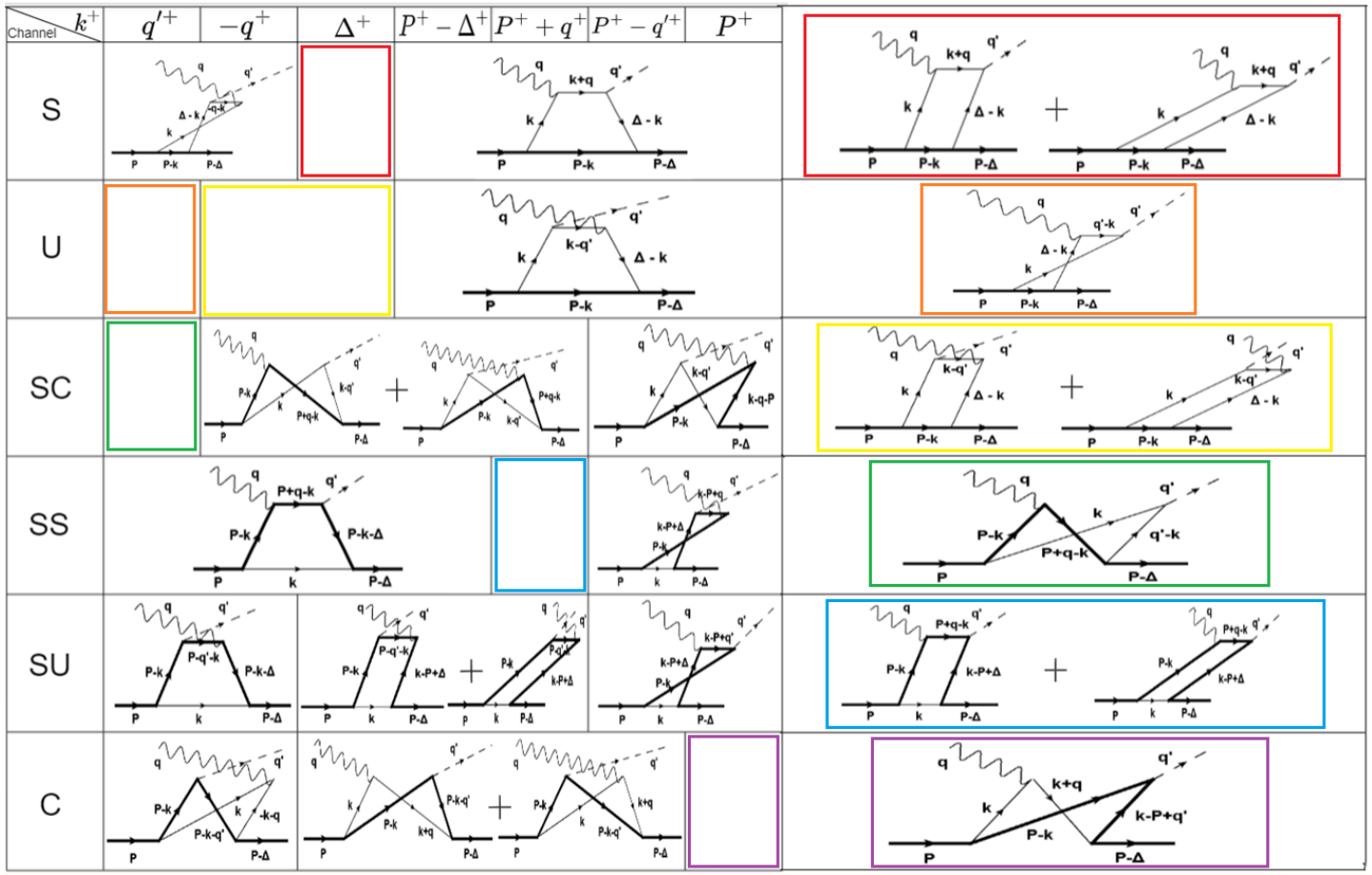
- GPDs are essentially generalized PDFs from DIS. They rely on factorization for their validity, wherein the interacting parton is struck on a timescale too short for it to interact with the rest of the partons again. Thus, the amplitude is expressed as a convolution of the hard and soft parts.
- Specific contributions to the amplitude from particular partons can be established, namely from quark GPDs and gluon GPDs.
- However, the GPDs are not invariant quantities as aforementioned. CFFs are invariant, but without factorization it is not possible to separate out particular partonic contributions as the CFFs assess the *most general* structure of the hadron target. Yet this 'frees up' the acceptable kinematics, and as the GPDs and CFFs must correspond in the forward limit, both can be used to extract information about the hadronic physics, and comparing and contrasting the CFFs and GPDs can establish the theoretical uncertainties underpinning the GPDs due to their lack of invariance.

"GPDs are constrained by sum rules and reduction formulae to the forward kinematics, which give us a guideline on how to model them. One may hope that these constraints, once implemented into a parametrization, provide a realistic order of magnitude estimate for cross sections and asymmetries. However, we should emphasize that our evaluations (as well as by other authors) are strongly affected by the model ambiguities involved. Experimental data will necessarily constrain GPDs via theoretical formulae. Therefore, *it is important to understand the (i) theoretical uncertainties* and (ii) influence of the model parameters on predictions."

- *Theory of deeply virtual Compton scattering on the nucleon,* A.V. Belitsky, D. M¨uller, A. Kirchner, 2002, pages 28-29. Emphasis added by me.

SIMPLE BENCHMARK CALCULATION

- We wish to access kinematics where the BSA is non-vanishing, as this tells us interesting information about some GPDs and highlights the advantage of measuring CFFs
- To that end, I've been working on a simple model calculation. I am looking to provide a bare bones structure which is sufficient to find significant BSA regions
- Lack of BH process means I must find at least one CFF that takes on a non-vanishing imaginary component



CURRENT STATE OF RESEARCH

- Neutral target + scalar coupling numerical work satisfies gauge invariance and recent (this week!) work shows consistency with the “E1” boost
- My usual set of parameters has consistently resulted in one form factor dominating, resulting in very small BSA results. Will experiment with changing mass parameters as F2 contains contributions from $P = p + p'$
- Derivative coupling contribution, as well as charged target channels are essentially ready to go, I just need to code it up

E1 BOOST

- The E1 boost rotates the coordinate system around the y-axis and boosts in the x-direction. For any given 4-momentum, this boost preserves the light-front time component (+) and transverse y-component.

$$(P^0)^2 - \vec{\mathbf{P}}^2 = \left(M + \frac{\vec{\mathbf{P}}_\perp^2}{2M}\right)^2 - \vec{\mathbf{P}}_\perp^2 - \left(-\frac{\vec{\mathbf{P}}_\perp^2}{2M}\right)^2 = M^2$$

[See *Interpolating helicity spinors between the instant form and the light-front form*, by Li, An, Ji, 2015, PRD, page 11]

NUMERICAL INTEGRATION IN *MATHEMATICA*

NUMERICAL INTEGRATION BASICS

- Recall 1D Reimann sums. The important feature: more sampled integrand points means better integral estimate, and the more oscillatory the integrand in a local region the more important to have a dense set of sampling points in that region
- Easily extend this to higher dimensions, but how do you know how many sampling points you need?

NUMERICAL INTEGRATION ALGORITHMS

Hence an adaptive strategy has these components [MalcSimp75]:

(i) an integration rule to compute the integral and error estimates over a region;

(ii) a method for deciding which elements of a set of regions $\{V_i\}_{i=1}^{n}$ to partition/subdivide;

(iii) stopping criteria for deciding when to terminate the adaptive strategy algorithm.

Global Adaptive Strategy

A global adaptive strategy reaches the required precision and accuracy goals of the integral estimate by recursive bisection of the subregion with the largest error estimate into two halves, and computes integral and error estimates for each half.

The global adaptive algorithm for NIntegrate is specified with the Method option value "GlobalAdaptive":

```
In[32]:= NIntegrate[1/Sqrt[x], {x, 0, 1}, Method -> "GlobalAdaptive"]

Out[32]= 2.
```

option name	default value	
Method	Automatic	integration rule used to compute integral and error estimates over each subregion
"SingularityDepth"	Automatic	number of recursive bisections before applying a singularity handler
"SingularityHandler"	Automatic	singularity handler
"SymbolicProcessing"	Automatic	number of seconds to do symbolic preprocessing

"GlobalAdaptive" options.

"GlobalAdaptive" is the default integration strategy of NIntegrate. It is used for both one-dimensional and multidimensional integration. "GlobalAdaptive" works with both Cartesian product rules and fully symmetric multidimensional rules.

"GlobalAdaptive" uses a data structure called a "heap" to keep the set of regions partially sorted, with the largest error region being at the top of the heap. In the main loop of the algorithm the largest error region is bisected in the dimension that is estimated to be responsible for most of its error.

Each region has a record of how many bisections are made per dimension in order to produce it. When a region has been produced through too many bisections a singularity flattening algorithm is applied to it; see "Singularity Handling".

"GlobalAdaptive" stops if the following expression is true:

$$\text{globalError} \le \text{globalIntegral}\, 10^{-pg} \vee \text{globalError} \le 10^{-ag},$$

where pg and ag are precision and accuracy goals.

The strategy also stops when the number of recursive bisections of a region exceeds a certain number (see "MinRecursion and MaxRecursion"), or when the global integration error oscillates too much (see "MaxErrorIncreases").

Theoretical and practical evidence show that the global adaptive strategies have, in general, better performance than the local adaptive strategies [MalcSimp75][KrUeb98].

MinRecursion and MaxRecursion

The minimal and maximal depths of the recursive bisections are given by the values of the options MinRecursion and MaxRecursion.

If for any subregion the number of bisections in any of the dimensions is greater than MaxRecursion then the integration by "GlobalAdaptive" stops.

Setting MinRecursion to a positive integer forces recursive bisection of the integration regions before the integrand is ever evaluated. This can be done to ensure that a narrow spike in the integrand is not missed. (See "Tricking the Error Estimator".)

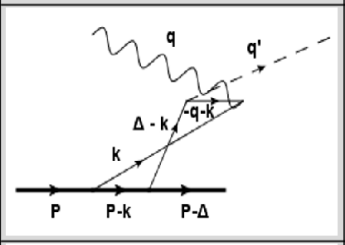
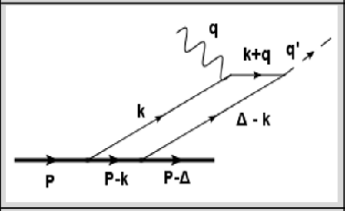
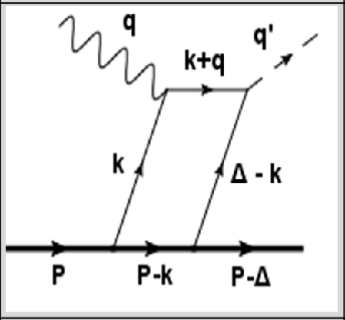
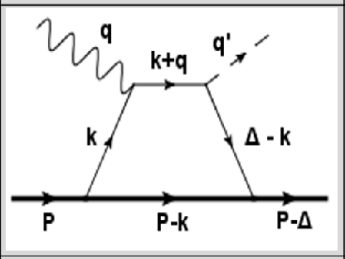
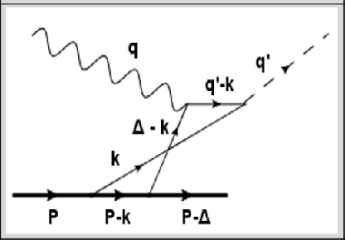
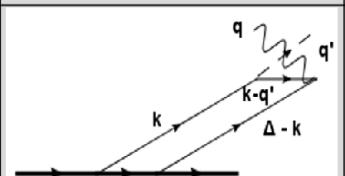
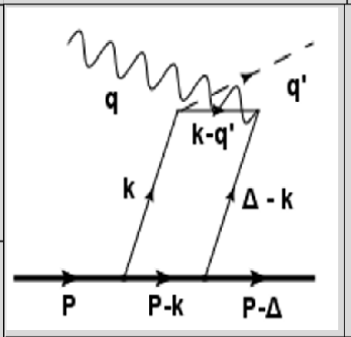
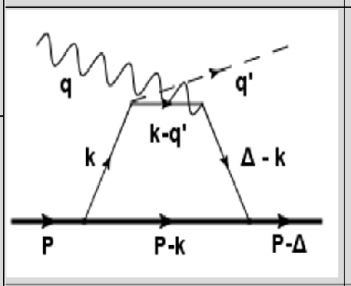
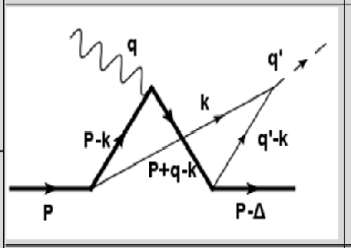
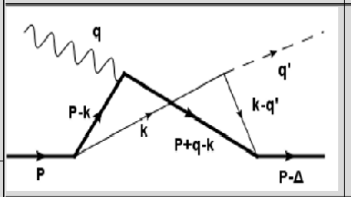
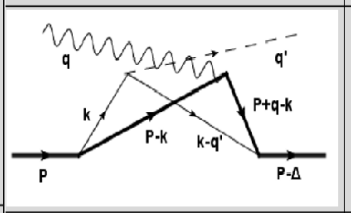
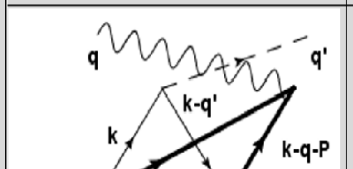
For multidimensional integration an effort is made to bisect in each dimension for each level of recursion in MinRecursion.

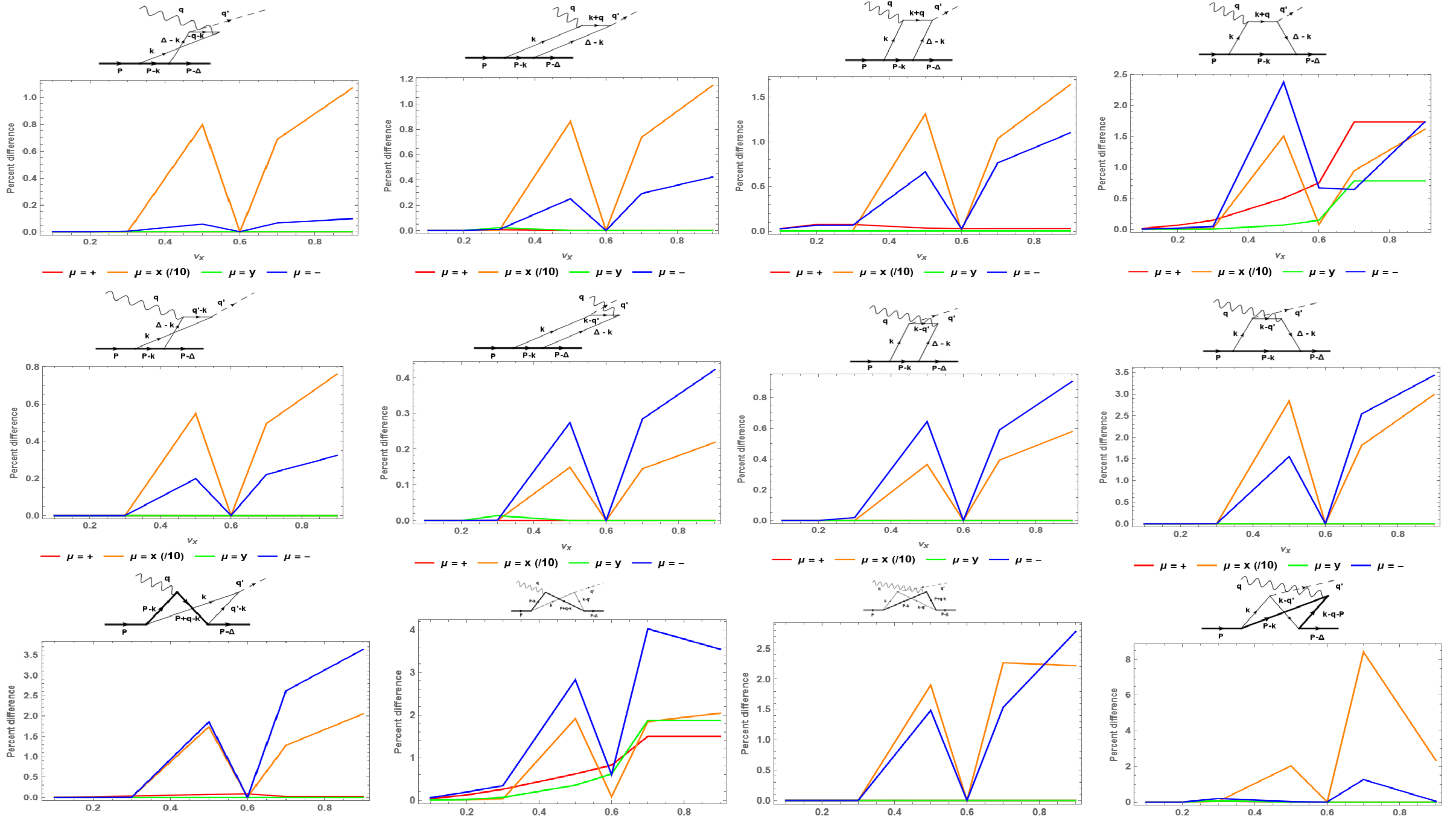
- Looking at the function documentation, the default minimum recursion depth is ZERO! Increasing the WorkingPrecision isn't going to help with this, as the same points are still sampled, just more precisely. Increasing MaxErrorIncreases only tells Mathematica that IF the integral subregion error sum estimates keep going up, how many times this can happen before it gives up. Again, this isn't going to help an integral with small values and rapidly-changing behavior due to a complicated integrand. GlobalAdaptive is already the best strategy for this kind of integral, and increasing the symbolic preprocessing isn't going to do much for a complicated integrand like this, that only in principle decreases the evaluation time, anyway. Increasing MinRecursion to, say, 4 splits each dimension of the integral domain 2^4 = 16 times, however. *Mathematica* is now forced to do 16^2 integrals and is MUCH more likely to find small contributions and get better error estimates. Another possibility is to increase your PrecisionGoal, but the effective precision you get is about half what your setting is for WorkingPrecision, and WorkingPrecision must be at least that of PrecisionGoal, and computation times increase dramatically beyond MachinePrecision (~16). I found more success by just increasing my MinRecursion.

MY INTEGRANDS ARE 'PATHOLOGICAL'

- You should realize that with sufficiently pathological functions, the algorithms used by NIntegrate can give wrong answers.

- This is the importance of checking the results given! Fortunately, the E1 boost and gauge invariance checks together appear to be working!

Diagram	% Diff. Re[F1]	% Diff. Re[F2]	% Diff. Im[F1]	% Diff. Im[F2]	F1	F2
	0.00379856	0.00375759	0.	0.	−0.0168445	−0.000872031
	0.00897444	0.000877477	0.	0.	0.0166401	0.000947423
	0.0167613	0.0273646	7.75291×10^{-7}	6.83434×10^{-8}	−0.0218629 + 0.00104279 i	−0.00279239 + 0.00017036 i
	0.00228116	0.00134154	0.000120246	1.17589×10^{-7}	0.0777919 + 0.0698001 i	0.0514137 − 0.0225406 i
	8.54234×10^{-5}	0.0000834713	0.	0.	−0.00902736	−0.000531295
	0.000810242	0.000497499	0.	0.	0.010236	0.000869374
	0.01415	0.0128435	0.	0.	0.00508443	−0.00028635
	0.00120314	0.0000455675	0.	0.	0.112515	−0.0281499
	0.00256287	0.00240514	5.18271×10^{-9}	1.16911×10^{-10}	0.00402116 − 0.000108698 i	0.000694937 − 0.0000431301 i
	0.0534726	0.00215519	0.000260944	9.63893×10^{-7}	−0.0281504 − 0.0208832 i	−0.0536614 − 0.022626 i
	7.51892×10^{-6}	1.01028×10^{-6}	0.	0.	0.0464539	0.0356634
	0.188937	0.174627	0.	0.	0.00257042	0.000145213



q
q'
k
k+q
Δ - k
-q-k
q'-k
k-q'
P+q-k
k-q-P
P
P-k
P-Δ
Percent difference
v_x
μ = +
μ = x (/10)
μ = y
μ = −

WHY IS THE X-COMPONENT SO BAD?

- The minus component also has errors, but the error is smaller, despite these integrands being more “pathological”.
- I hypothesize this is due to the x-component being small before being boosted. Fortunately I only need the + and y components to solve for the CFFs.

STUFF FOR ME TO DO NOW!

- Improve the x-component of the current, and I'll throw more computer power into the evaluation. Compare individual diagram CFF contributions using the - and x-components instead.
- Vary the mass parameters to try to find a kinematic region where both CFFs contribute significantly.