Exploring Nucleon Resonances with DVCS:

HOW TO MAKE MATHEMATICA DO TENSOR ALGEBRA

Matthew Rumley *(Supervisor: Anthony Thomas)*
The University of Adelaide

The Roper Resonance

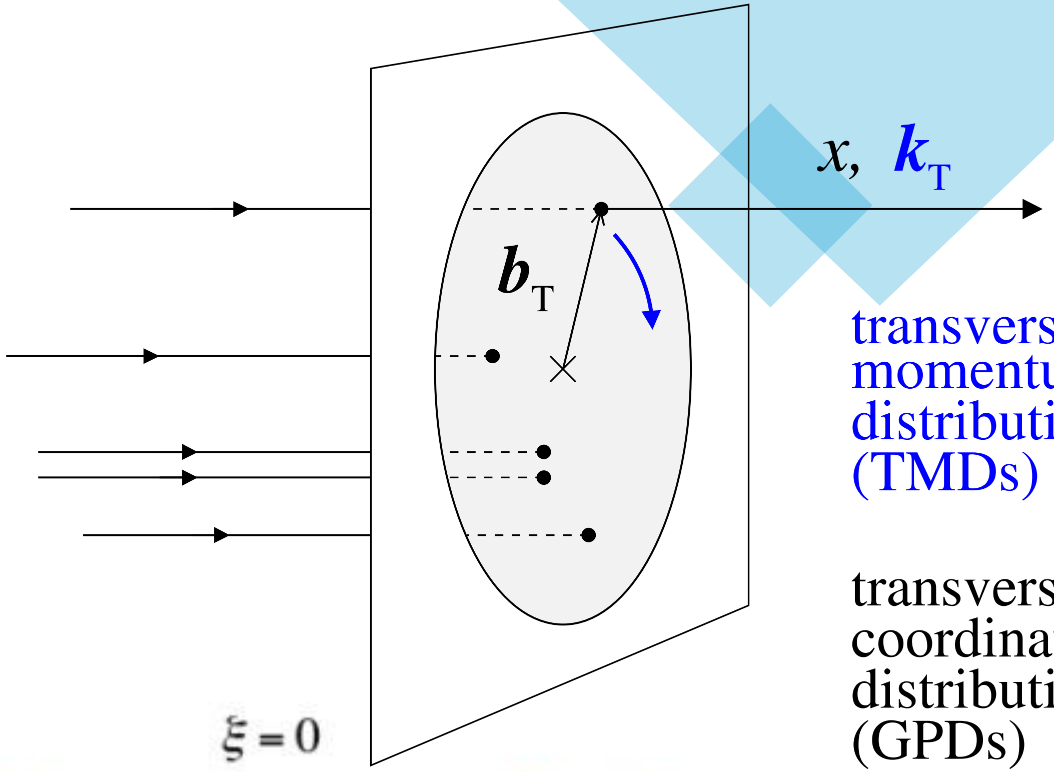
- Nucleon Resonance of mass $M \approx 1440$ and width $\Gamma \approx 300\ \mathrm{MeV}/c^2$
- Total Spin-Parity $J = {}^1/_2{}^+$
- Isospin $I = {}^1/_2$

Can be decomposed as

$$P_{11}(1440)\rangle = \alpha|uud\rangle + \beta|N\pi\rangle + \cdots$$

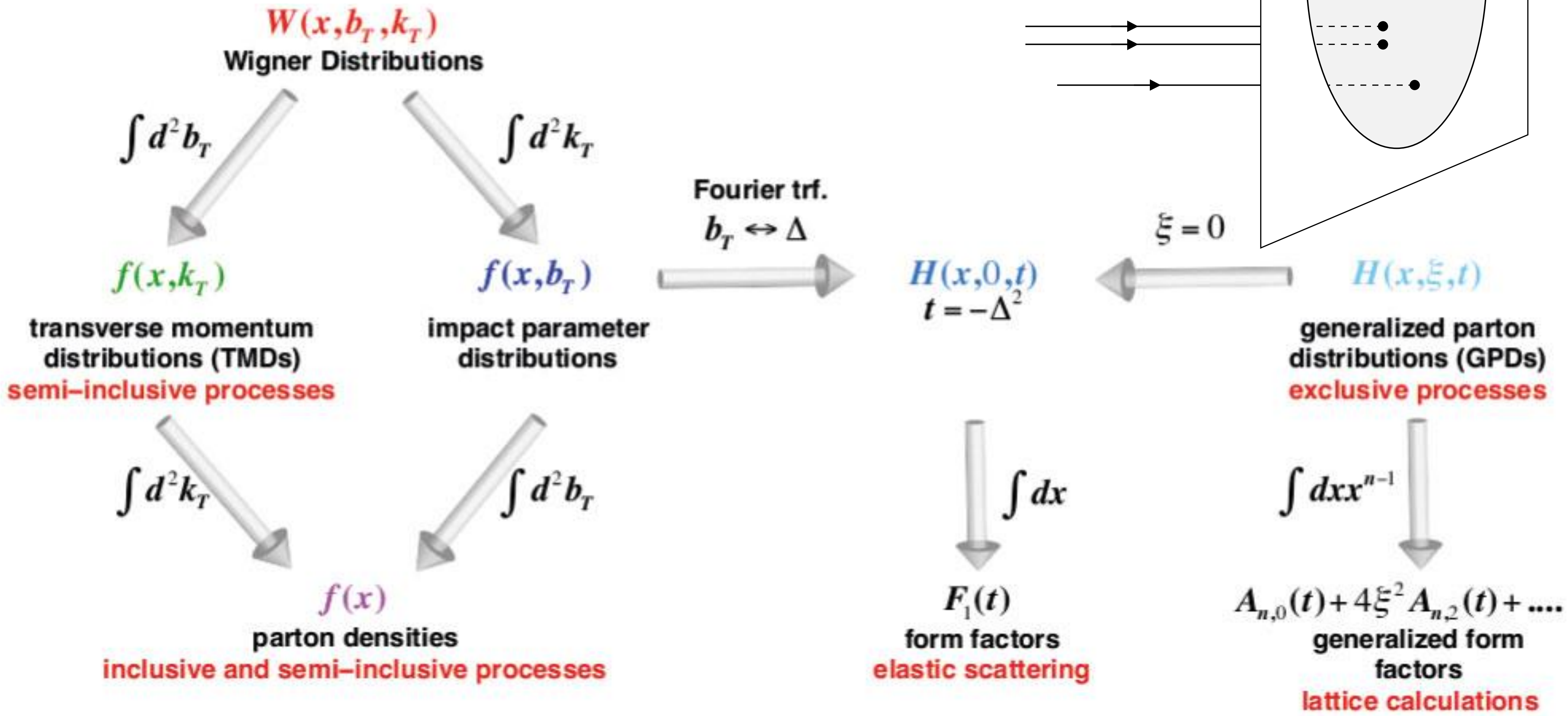
radially-excited three-quark state
$n = 2$ (radial quantum no.)

molecule-like bound state of a
nucleon and a pion



transverse momentum distributions (TMDs)

transverse coordinate distributions (GPDs)

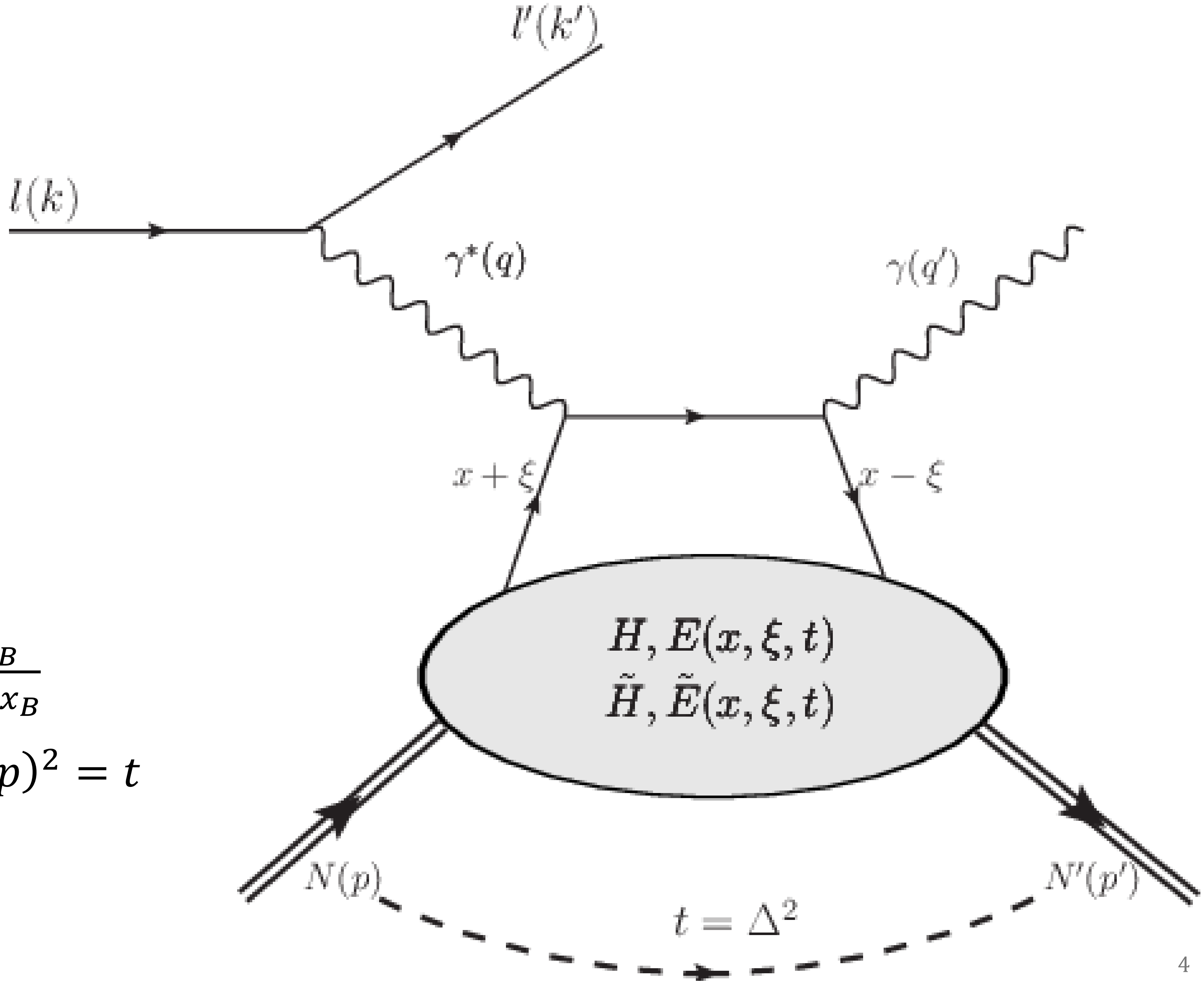


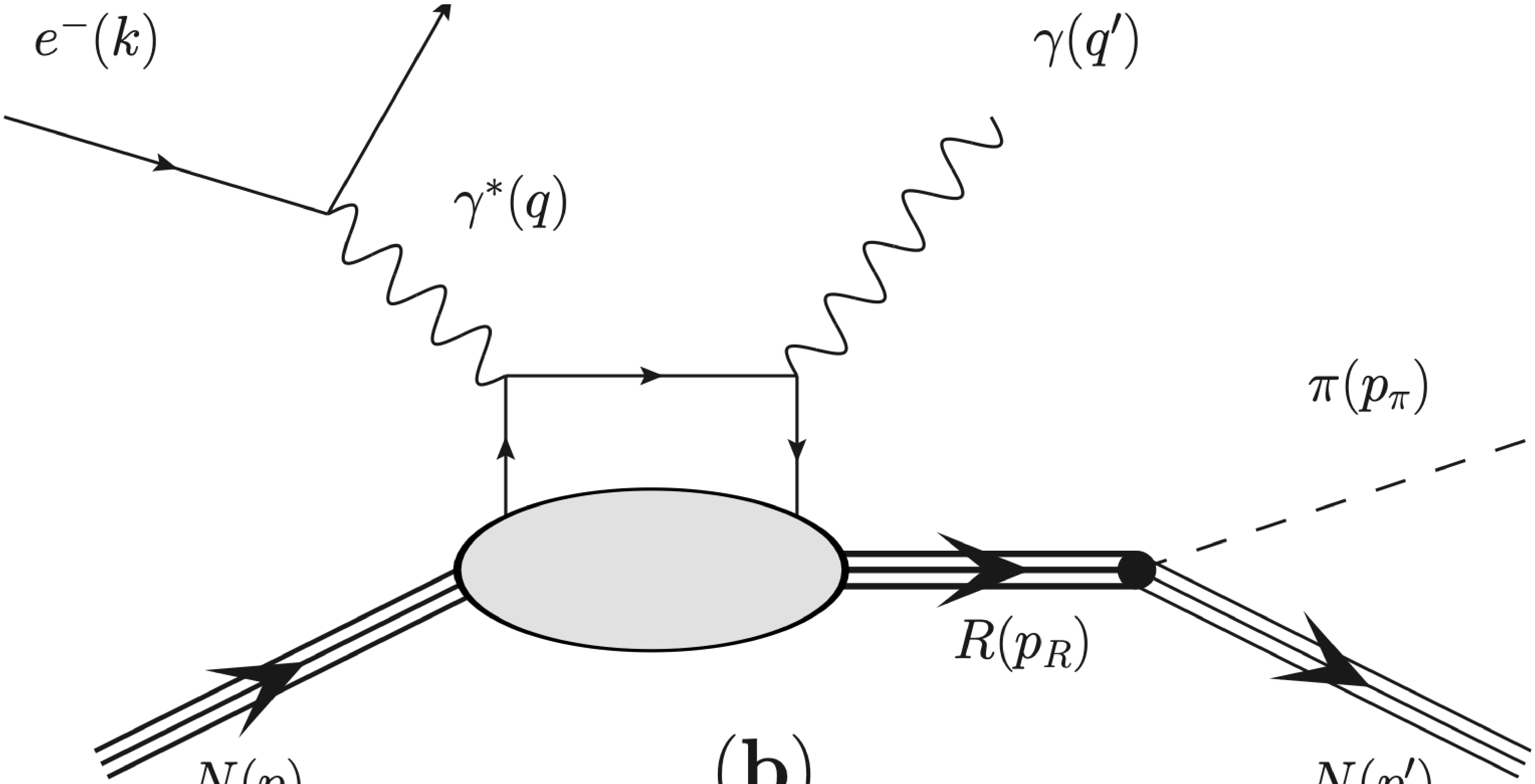
DVCS

$\ell N \to \ell' N' \gamma$

skewness: $\xi = \frac{x_i - x_f}{2} \to \frac{x_B}{2 - x_B}$

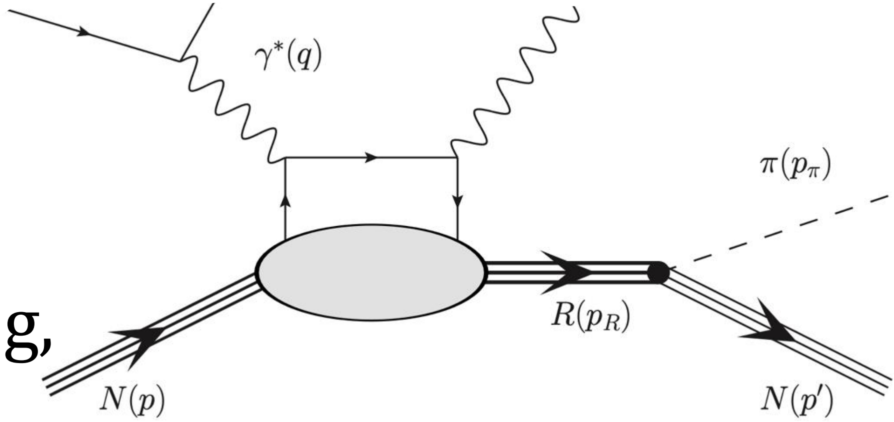
forwardness: $\Delta^2 = (p' - p)^2 = t$





(b)

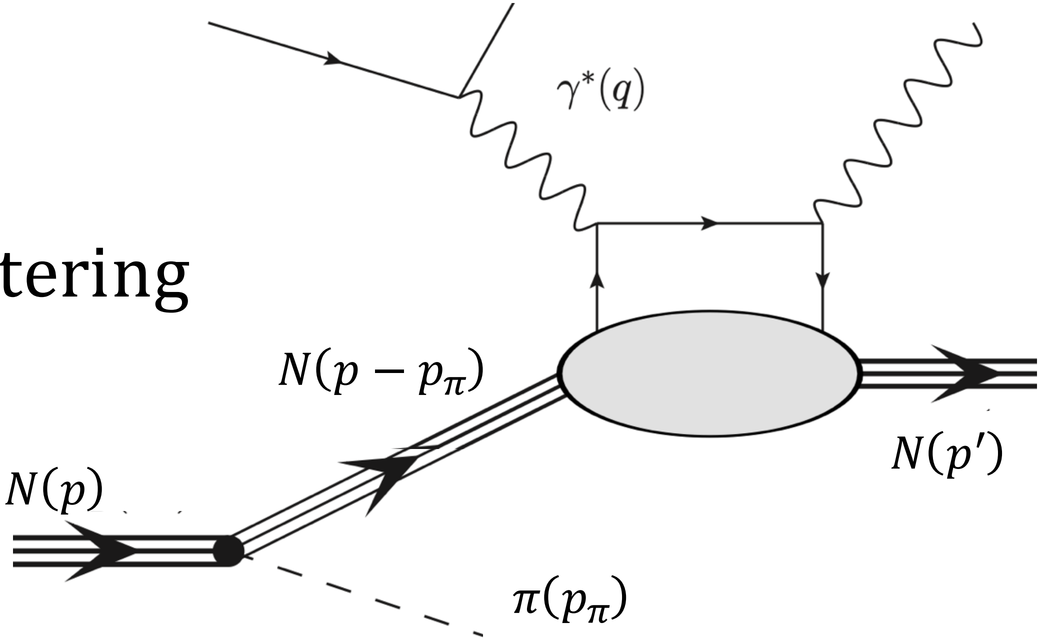
final state $N + \pi$ can be achieved by



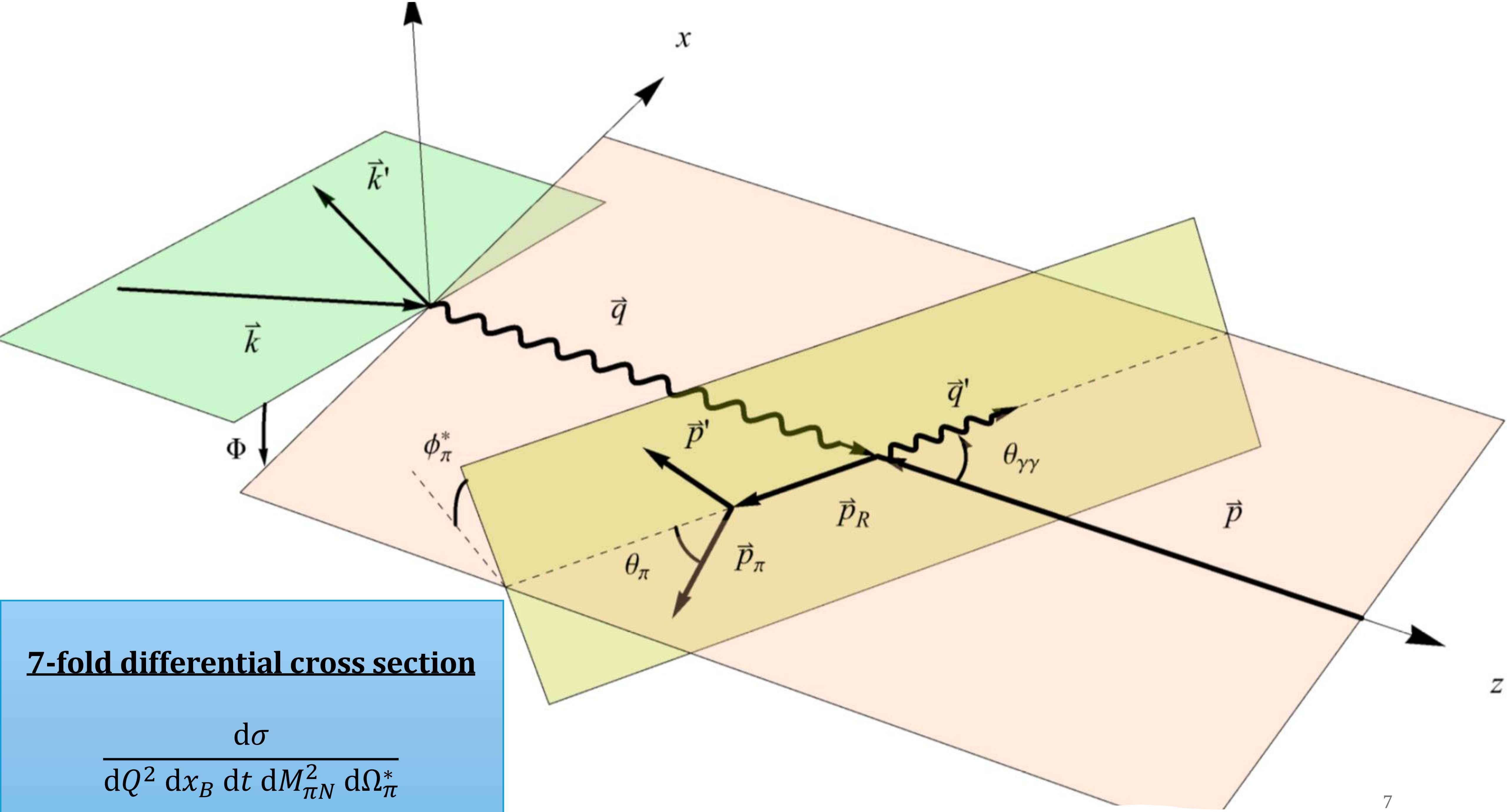
- $N(p) \rightarrow N(p_I) + \pi(k_\pi)$ *before* the lepton scattering,

OR

- $R(p_R) \rightarrow N(p') + \pi(k_\pi)$ *after* the lepton scattering



where $p' + p_\pi = p_R$



7-fold differential cross section

$$\frac{\mathrm{d}\sigma}{\mathrm{d}Q^2\,\mathrm{d}x_B\,\mathrm{d}t\,\mathrm{d}M^2_{\pi N}\,\mathrm{d}\Omega^*_\pi}$$

Tensor Algebra using xAct

- Written for the Wolfram Language in 2004 primarily by José M. Martín-García.
- Often used for General Relativity.

- Handles really complex Tensor Algebra reasonably intuitively.
- SYMBOLIC!!!

Define Manifolds, Metrics, and Tensors

no. of dimensions

indices linked to manifold

chart labels for manifold

```
In[•]:= (* Define 4D Minkowski Manifold and Chart *)
(* also defines tangent vector bundle *)
DefManifold[M, 4, {μ, ν, α, β, ρ, λ, τ, ω, κ}];
DefChart[Mink, M, {0, 1, 2, 3}, {t[], x[], y[], z[]}];

(* Define a metric tensor for 4D Flat Minkowski Space *)
η = CTensor[DiagonalMatrix[{1, -1, -1, -1}], {-Mink, -Mink}];
SetCMetric[η, Mink, SignatureOfMetric → {1, 3, 0}];
MetricCompute[η, Mink, All];
```

$$g_{\mu\nu} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

```
(* Define 4D Euclidean Manifold - For Matrix Multiplication *)
DefManifold[MEuc, 4, {AA, BB, CC, DD, EE, FF, GG, HH, II, JJ, KK, LL}];
DefChart[Mat, MEuc, {1, 2, 3, 4}, {a[], b[], c[], d[]}];

(* Define a metric tensor for 4D Euclidean Space *)
g = CTensor[IdentityMatrix[4], {-Mat, -Mat}];
SetCMetric[g, Mat, SignatureOfMetric → {4, 0, 0}];
MetricCompute[g, Mat, All];
```

$$I_{AA\,BB} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Initialise Tensors

```
In[•]:= (* Initialise Particle Momentum Vectors *)
DefTensor[p[-μ], M, PrintAs → "p"];
DefTensor[ke[-μ], M, PrintAs → "kₑ"];
DefTensor[kePrime[-μ], M, PrintAs → "k
DefTensor[qPrime[-μ], M, PrintAs → "q
DefTensor[k[-μ], M, PrintAs → "kπ"];

(* Initialise Local Momentum Vectors
DefTensor[pF[-μ], M, PrintAs → "pf"];
DefTensor[pR[-μ], M, PrintAs → "pR"];
DefTensor[q[-μ], M, PrintAs → "q"];

(* Initialise GPD Vectors *)
DefTensor[pTilde[-μ], M, PrintAs → "p
DefTensor[n[-μ], M, PrintAs → "n"];
DefTensor[Δ[-μ], M, PrintAs → "Δ"];
DefTensor[PBar[-μ], M, PrintAs → "PBa

(* Initialise Polarization Vectors - Photon *)
DefTensor[ϵConj[μ], M, PrintAs → "ϵ*"];
```

Minkowski Vectors

DefTensor[*label*[*index*], *manifold*, *options*]

$$p_\mu = \begin{pmatrix} \frac{E}{c} & p_x & p_y & p_z \end{pmatrix}$$

```
(* Initialise Spinors *)
DefTensor[ueSpin[-AA], MEuc];
DefTensor[uNSpin[-AA], MEuc];

(* Initialise Barred Spinors *)
DefTensor[ueBarSpin[AA], MEuc];
DefTensor[uNBarSpin[AA], MEuc];

(* Initialise Derivatve of Scalar Field *)
DefTensor[PartialPhi[-μ], M, PrintAs → "∂ϕ"];
```

Bispinors

(Euclidean vectors)

`DefTensor[label[index], manifold]`

e.g, u_{AA} could be $(1 \quad 0 \quad 0 \quad 1)^{\mathrm{T}}$

Dirac γ Matrices

(a Minkowski 4-vector of 4×4 Euclidean matrices)

`DefTensor[label[index1,index2,…], {manifold1, manifold2,…}]`

$$(\gamma^{\mu})_{AA}{}^{BB} = \begin{bmatrix} \gamma^0{}_{4\times4} \\ \gamma^1{}_{4\times4} \\ \gamma^2{}_{4\times4} \\ \gamma^3{}_{4\times4} \end{bmatrix}$$

```
(* Define Gamma Matrices *)
DefTensor[γ[μ, -AA, BB], {M, MEuc}];
DefTensor[γ5[-AA, BB], {MEuc}];
```

```
(* Define Sigma Matrices *)
DefTensor[σ[μ, ν, -AA, BB], {M, M, MEuc, MEuc}];
```

Relativstic Spin Matrices

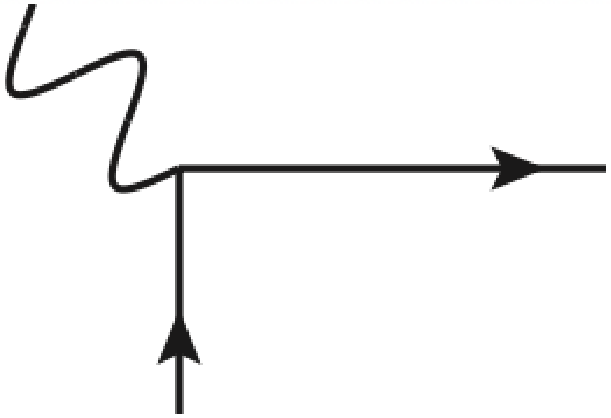
(a 4 × 4 Minkowski matrix of 4 × 4 Euclidean matrices)

`DefTensor[label[index1,index2,…], {manifold1, manifold2,…}]`

$$(\sigma^{\mu\nu})_{AA}{}^{BB} = \begin{bmatrix} \sigma^{00}{}_{4\times4} & \sigma^{01}{}_{4\times4} & \sigma^{02}{}_{4\times4} & \sigma^{03}{}_{4\times4} \\ \sigma^{10}{}_{4\times4} & \sigma^{11}{}_{4\times4} & \sigma^{12}{}_{4\times4} & \sigma^{13}{}_{4\times4} \\ \sigma^{20}{}_{4\times4} & \sigma^{21}{}_{4\times4} & \sigma^{22}{}_{4\times4} & \sigma^{23}{}_{4\times4} \\ \sigma^{30}{}_{4\times4} & \sigma^{31}{}_{4\times4} & \sigma^{32}{}_{4\times4} & \sigma^{33}{}_{4\times4} \end{bmatrix}$$

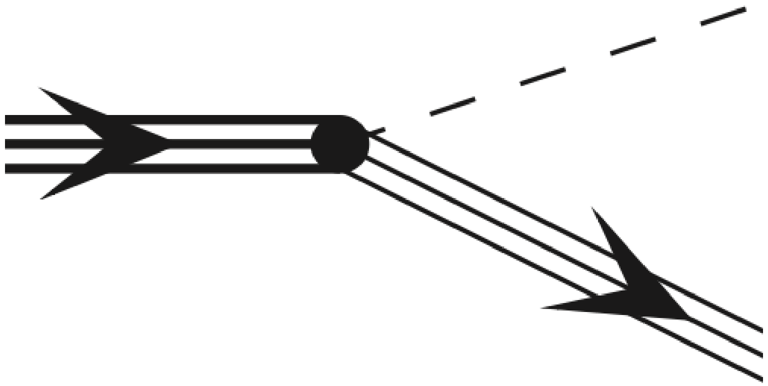
Define Vertex Rules

```
(* Fermion/Fermion/Photon Vertex: k_e flowing into photon momentum q, leaving with k_e' *)
ffP[μ_, AA_, BB_] := -I*gf*e*γ[μ, AA, BB];
```



$$= -i g_f e\, \gamma^\mu$$

```
(* Nucleon-Pion Vertex: p flowing into vertex, pion emitted with k *)
Nπ[AA_, BB_] := I*PartialPhi[-ρ] × γ[ρ, AA, II] × γ5[-II, BB] // ReplaceDummies;
```



$$= i\, \partial_\rho \phi\, \gamma^\rho\, \gamma^5$$

Define Vertex Rules

```
(* Fermion Propagator: Momentum q *)
Sf[q_, m_, AA_, BB_] := I (q[-λ] × γ[λ, AA, BB] + m * Id[AA, BB]) / (Scalar[q[-λ] × q[λ]] - m^2);
```

$$= i \frac{p_\lambda \gamma^\lambda + m \, \mathrm{I}_{4\times4}}{p^2 - m^2}$$

Define Vertex Rules

```
(* Define Spinors and Adjoint Spinors *)
ue[AA_, q_, m_] := (q[-ω] × γ[ω, AA, II] + m * Id[AA, II]) / Sqrt[Scalar[EVec[μ] × q[-μ]] + m] ueSpin[-II] // ReplaceDummies;
ueBar[AA_, q_, m_] := ueBarSpin[JJ] (q[-λ] × γ[λ, -JJ, AA] + m * Id[-JJ, AA]) / Sqrt[Scalar[EVec[μ] × q[-μ]] + m] // ReplaceDummies;
uN[AA_, q_, m_] := (q[-ω] × γ[ω, AA, II] + m * Id[AA, II]) / Sqrt[Scalar[EVec[μ] × q[-μ]] + m] uNSpin[-II] // ReplaceDummies;
uNBar[AA_, q_, m_] := uNBarSpin[JJ] (q[-λ] × γ[λ, -JJ, AA] + m * Id[-JJ, AA]) / Sqrt[Scalar[EVec[μ] × q[-μ]] + m] // ReplaceDummies;
```

Spinors

$$\text{e.g. } u_e(p,m) = \frac{p_\omega\,\gamma^\omega + m\,\mathrm{I}_{4\times 4}}{\sqrt{p_0+m}}\begin{bmatrix}1\\0\\0\\1\end{bmatrix}$$

DVCS Tensor $H_{NR}^{\mu\nu}$

$$
\begin{aligned}
H_{NR}^{\mu\nu} &= \frac{1}{2}(-g_{\perp}^{\mu\nu}) \int_{-1}^{1} dx \left[\frac{1}{x-\xi+i\epsilon} + \frac{1}{x+\xi-i\epsilon}\right] \int \frac{d\lambda}{2\pi} e^{i\lambda x} \sum_q e_q^2 \langle R(p_R, s_R)| \bar{q}\left(-\frac{\lambda n}{2}\right) \gamma\cdot n\, q\left(\frac{\lambda n}{2}\right) |N(p, s_N)\rangle \\
&+ \frac{i}{2}(\varepsilon_{\perp}^{\mu\nu}) \int_{-1}^{1} dx \left[\frac{1}{x-\xi+i\epsilon} - \frac{1}{x+\xi-i\epsilon}\right] \int \frac{d\lambda}{2\pi} e^{i\lambda x} \sum_q e_q^2 \langle R(p_R, s_R)| \bar{q}\left(-\frac{\lambda n}{2}\right) \gamma\cdot n\gamma_5\, q\left(\frac{\lambda n}{2}\right) |N(p, s_N)\rangle,
\end{aligned}
$$

$$
\begin{aligned}
&\int \frac{d\lambda}{2\pi} e^{i\lambda x} \sum_q e_q^2 \langle R(p_R, s_R)| \bar{q}\left(-\frac{\lambda n}{2}\right) \gamma\cdot n\, q\left(\frac{\lambda n}{2}\right) |N(p, s_N)\rangle \\
&= H_1^{pP_{11}}(x,\xi,\Delta^2)\, \bar{R}(p_R, s_R) \left(n^\nu - \frac{n\cdot\Delta}{\Delta^2}\Delta^\nu\right) \gamma_\nu N(p, s_N) + H_2^{pP_{11}}(x,\xi,\Delta^2)\, \bar{R}(p_R, s_R) \frac{i\sigma_{\nu\kappa} n^\nu \Delta^\kappa}{M_R + M_N} N(p, s_N).
\end{aligned}
$$

$$
\begin{aligned}
&\int \frac{d\lambda}{2\pi} e^{i\lambda x} \sum_q e_q^2 \langle R(p_R, s_R)| \bar{q}\left(-\frac{\lambda n}{2}\right) \gamma\cdot n\,\gamma_5\, q\left(\frac{\lambda n}{2}\right) |N(p, s_N)\rangle \\
&= \tilde{H}_1^{pP_{11}}(x,\xi,\Delta^2)\, \bar{R}(p_R, s_R)\, \gamma\cdot n\gamma_5 N(p, s_N) + \tilde{H}_2^{pP_{11}}(x,\xi,\Delta^2)\, \bar{R}(p_R, s_R) \frac{\Delta\cdot n}{M_R + M_N} \gamma_5 N(p, s_N).
\end{aligned}
$$

DVCS Tensor $H^{\mu\nu}_{NR}$

$$\int \frac{d\lambda}{2\pi} e^{i\lambda x} \sum_q e_q^2 \langle R(p_R, s_R)| \bar{q}\left(-\frac{\lambda n}{2}\right) \gamma\cdot n\, q\left(\frac{\lambda n}{2}\right) |N(p, s_N)\rangle$$

$$= H_1^{pP_{11}}(x,\xi,\Delta^2)\, \bar{R}\,(p_R, s_R) \left(n^\nu - \frac{n\cdot\Delta}{\Delta^2}\Delta^\nu\right) \gamma_\nu N\,(p, s_N) + H_2^{pP_{11}}(x,\xi,\Delta^2)\, \bar{R}\,(p_R, s_R) \frac{i\sigma_{\nu\kappa} n^\nu \Delta^\kappa}{M_R + M_N} N\,(p, s_N)\,.$$

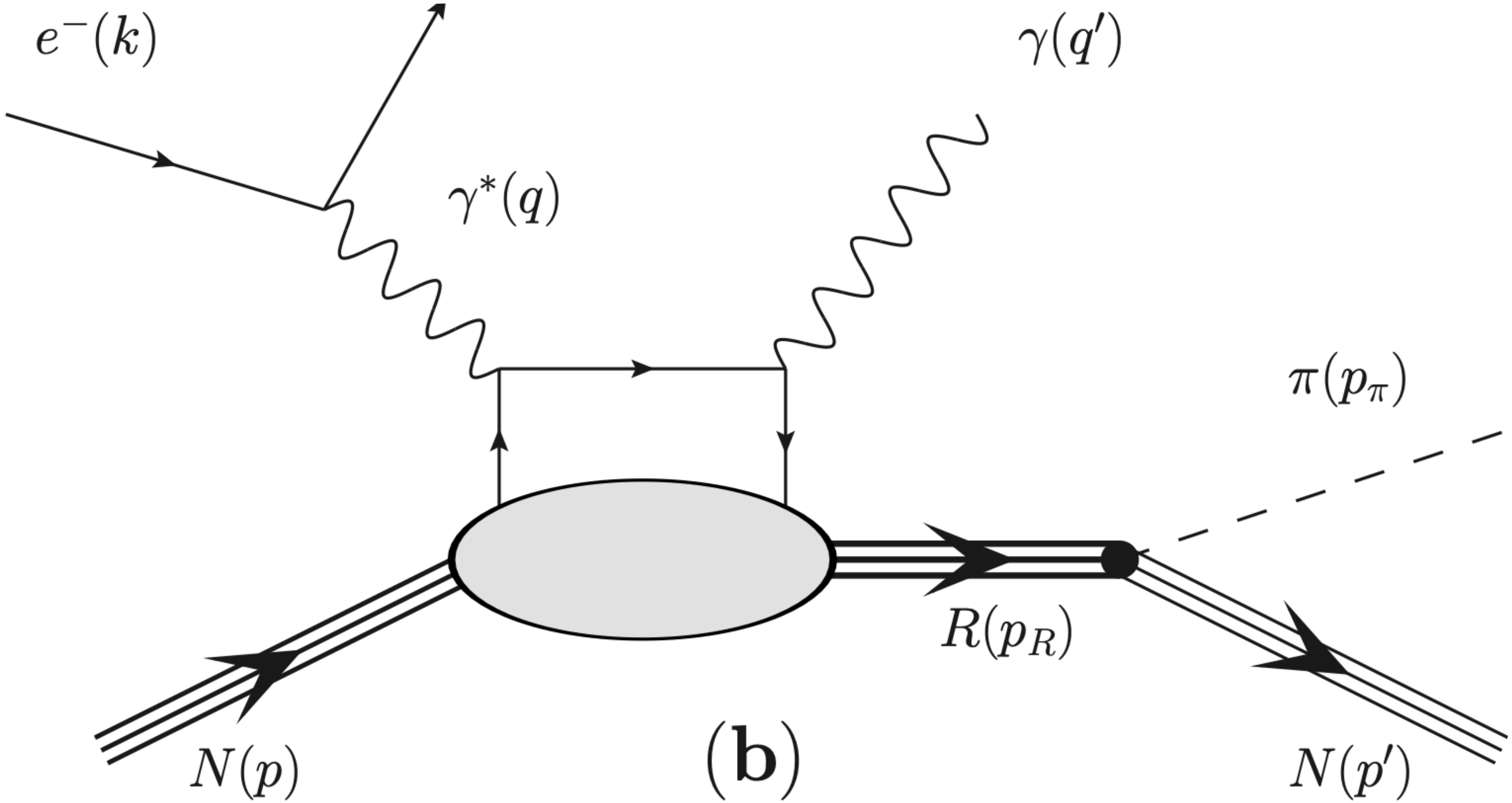
$$\int \frac{d\lambda}{2\pi} e^{i\lambda x} \sum_q e_q^2 \langle R(p_R, s_R)| \bar{q}\left(-\frac{\lambda n}{2}\right) \gamma\cdot n\, \gamma_5\, q\left(\frac{\lambda n}{2}\right) |N(p, s_N)\rangle$$

$$= \tilde{H}_1^{pP_{11}}(x,\xi,\Delta^2)\, \bar{R}\,(p_R, s_R)\, \gamma\cdot n \gamma_5 N\,(p, s_N) + \tilde{H}_2^{pP_{11}}(x,\xi,\Delta^2)\, \bar{R}\,(p_R, s_R) \frac{\Delta\cdot n}{M_R + M_N} \gamma_5 N\,(p, s_N)\,.$$

```
(* GPD Factorisation *)
VectorBilinear[α_, β_, pTilde_, n_, Δ_, m1_, m2_, AA_, BB_] :=
  1/2 * (-η[α, β] + pTilde[α] × n[β] + pTilde[β] × n[α]) (H1 (n[ν] - Scalar[n[-ω] × Δ[ω]] / Scalar[Δ[-λ] × Δ[λ]] Δ[ν]) γ[-ν, AA, BB] + H2 (I * σ[-ν, -κ, AA, BB] × n[ν] × Δ[κ]) / (m1 + m2)) // ReplaceDummies;
AxialVectorBilinear[α_, β_, pTilde_, n_, Δ_, m1_, m2_, AA_, BB_] :=
  I/2 ϵ[α, β, λ, ρ] × n[-λ] × pTilde[-ρ] (H1Tilde * γ[-ω, AA, II] × n[ω] * γ5[-II, BB] + H2Tilde Scalar[n[-ω] × Δ[ω]] / (m1 + m2) γ5[AA, BB]) // ReplaceDummies;
GPD[α_, β_, pTilde_, n_, Δ_, m1_, m2_, AA_, BB_] = VectorBilinear[α, β, pTilde, n, Δ, m1, m2, AA, BB] + AxialVectorBilinear[α, β, pTilde, n, Δ, m1, m2, AA, BB];
```

Define Matrix Element and Apply the Bjorken Limit

```
initM = ueBar[AA, kePrime, me] × ffP[β, -AA, BB] × ue[-BB, ke, me] × ϵConj[α]
        uNBar[CC, pF, mN] × Nπ[-CC, DD] × Sf[pR, mN, -DD, EE]
        GPD[-α, -β, pTilde, n, Δ, mN, mR, -EE, FF] × uN[-FF, p, mN];
```



Define Matrix Element and Apply the Bjorken Limit

```
initM = ueBar[AA, kePrime, me] × ffP[β, -AA, BB] × ue[-BB, ke, me] × ϵConj[α]
            uNBar[CC, pF, mN] × Nπ[-CC, DD] × Sf[pR, mN, -DD, EE]
            GPD[-α, -β, pTilde, n, Δ, mN, mR, -EE, FF] × uN[-FF, p, mN];
```

```
Bjorken = {ξ → (xB / 2) / (1 - (xB / 2))};

momCons = {q[μ_] :> ke[μ] - kePrime[μ], Δ[μ_] :> pR[μ] - p[μ], pF[μ_] :> pR[μ] - k[μ]};

M = initM /. GPDrr1 /. GPDrr2 /. Bjorken /. {xB :> Q^2 / (2 mN Scalar[p[μ] × q[-μ]])} /. momCons;
```

Everything up until now has been entirely general: no frame dependence

Set Up Vectors in chosen frame

```
In[•]:= (* Momentum Vectors *)
        pVec = {mN, 0, 0, 0};
        pRVec = {mN + Ee - EPrime - EGamma,
            -EGamma * Cos[ϕ] * Sin[θγ],
            -EGamma * Sin[ϕ] * Sin[θγ],
            (2 * (Ee - EPrime) ^2 + Q^2 - 2 * EGamma * (Ee - EPrime) * Cos[θγ]) / (2 (Ee - EPrime))};
```

```
(* Photon Polarization Vectors *)
(* Conjugated since outgoing *)
(* Phase factor of Exp[ +- I ϕ] omitted *)
ϵpVec = 1/Sqrt[2] {0, -Cos[θγ] Cos[ϕ] - I Sin[ϕ], -Cos[θγ] Sin[ϕ] + I Cos[ϕ], Sin[θγ]};
ϵmVec = -1/Sqrt[2] {0, -Cos[θγ] Cos[ϕ] + I Sin[ϕ], -Cos[θγ] Sin[ϕ] - I Cos[ϕ], Sin[θγ]}

(* Pion vectors - in rest frame - will need to boost *)
kRestVec = {(ERStar^2 + mπ^2)/(2 ERStar), Sqrt[((ERStar^2 - mπ^2 + mN^2)/(2 ERStar))^2 - mN^2] Cos[ϕπ] Sin[θπ], Sqrt[(
```

Boost Necessary Vectors into Chosen Frame

```
In[•]:= (* Define the difference between frames by creating a boost velocity *)
```

$$vXX = \frac{EGamma * Cos[\phi] * Sin[\theta\gamma]}{mN + Ee - EPrime - EGamma};$$

$$vYY = \frac{EGamma * Sin[\phi] * Sin[\theta\gamma]}{mN + Ee - EPrime - EGamma};$$

$$vZZ = -\frac{2 * (Ee - EPrime)^2 + Q^2 - 2\, EGamma * (Ee - EPrime) * Cos[\theta\gamma]}{2 * (Ee - EPrime) * (mN + Ee - EPrime - EGamma)};$$

```
vBoost = {vXX, vYY, vZZ};
vCoba = CTensor[vBoost, {Vec}];
vSquared = Scalar[vCoba[-XX] × vCoba[XX]];
```

$$Lorentz = \frac{1}{Sqrt[1 - vSquared]};$$

$$\Lambda Mat = \{\{Lorentz, -Lorentz * vXX, -Lorentz * vYY, -Lorentz * vZZ\},$$
$$\{-Lorentz * vXX, 1 + (Lorentz - 1)\frac{vXX^2}{vSquared}, (Lorentz - 1)\frac{vXX * vYY}{vSquared},$$
$$\{-Lorentz * vYY, (Lorentz - 1)\frac{vXX * vYY}{vSquared}, 1 + (Lorentz - 1)\frac{vYY^2}{vSquared},$$
$$\{-Lorentz * vZZ, (Lorentz - 1)\frac{vXX * vZZ}{vSquared}, (Lorentz - 1)\frac{vYY * vZZ}{vSquared}, 1 +$$

```
kBoostedVec = ΛMat.kRestVec;
```

```
(* Set Up Gamma Matrices in Dirac Basis *)
γ0 = {{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, -1, 0}, {0, 0, 0, -1}}
γ1 = {{0, 0, 0, 1}, {0, 0, 1, 0}, {0, -1, 0, 0}, {-1, 0, 0, 0}}
γ2 = {{0, 0, 0, -I}, {0, 0, I, 0}, {0, I, 0, 0}, {-I, 0, 0, 0}}
γ3 = {{0, 0, 1, 0}, {0, 0, 0, -1}, {-1, 0, 0, 0}, {0, 1, 0, 0}}
γ5Expansion = -I * γ0.γ1.γ2.γ3;

(* γ Matrices as a Vector of Matrices *)
γVec = {γ0, γ1, γ2, γ3};

(* Set Up Sigma Matrices in Dirac Basis *)
σ00 = {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};
σ11 = {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};
σ22 = {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};
σ33 = {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};

σ01 = I * γ0.γ1;
σ02 = I * γ0.γ2;
σ03 = I * γ0.γ3;

(* σ Matrices as a Matrix of Matrices *)
σMat = {{σ00, σ01, σ02, σ03}, {σ10, σ11, σ12, σ13}, {σ20, σ21, σ22, σ23}, {σ30, σ31, σ32, σ33}};
```

```
(* Calculate all possible spin combinations independently *)
MUpUpUpUpPlus = M /. ϵp /. ue1r /. ue1rBar /. uN1r /. uN1rBar /. rr;
MUpUpUpUpMinus = M /. ϵm /. ue1r /. ue1rBar /. uN1r /. uN1rBar /. rr;
MUpUpUpDownPlus = M /. ϵp /. ue1r /. ue1rBar /. uN1r /. uN2rBar /. rr;
MUpUpUpDownMinus = M /. ϵm /. ue1r /. ue1rBar /. uN1r /. uN2rBar /. rr;

MUpUpDownUpPlus = M /. ϵp /. ue1r /. ue1rBar /. uN2r /. uN1rBar /. rr;
MUpUpDownUpMinus = M /. ϵm /. ue1r /. ue1rBar /. uN2r /. uN1rBar /. rr;
MUpUpDownDownPlus = M /. ϵp /. ue1r /. ue1rBar /. uN2r /. uN2rBar /. rr;
MUpUpDownDownMinus = M /. ϵm /. ue1r /. ue1rBar /. uN2r /. uN2rBar /. rr;

(* Conjugate them all *)
ResultUpUpUpUpPlus = MUpUpUpUpPlus * Dagger[MUpUpUpUpPlus];
ResultUpUpUpUpMinus = MUpUpUpUpMinus * Dagger[MUpUpUpUpMinus];
ResultUpUpUpDownPlus = MUpUpUpDownPlus * Dagger[MUpUpUpDownPlus];
ResultUpUpUpDownMinus = MUpUpUpDownMinus * Dagger[MUpUpUpDownMinus];

ResultUpUpDownUpPlus = MUpUpDownUpPlus * Dagger[MUpUpDownUpPlus];
ResultUpUpDownUpMinus = MUpUpDownUpMinus * Dagger[MUpUpDownUpMinus];
ResultUpUpDownDownPlus = MUpUpDownDownPlus * Dagger[MUpUpDownDownPlus];
ResultUpUpDownDownMinus = MUpUpDownDownMinus * Dagger[MUpUpDownDownMinus];
```

What's remaining?

One really long SCALAR function of our kinematic variables AND the GPDs

$$H_1(x,\xi,\Delta^2), \qquad H_2(x,\xi,\Delta^2), \qquad \widetilde{H_1}(x,\xi,\Delta^2), \qquad \widetilde{H_2}(x,\xi,\Delta^2).$$

We need a parameterisation of these GPDs.

Use a product of the transition form factor and a ξ-independent valence-quark parameterisation.

$$H_{1,2}^{pP_{11}}(x,\xi,\Delta^2) = N\left[F_{1,2}^{pP_{11}}(\Delta^2) + \frac{2}{3}F_{1,2}^{nP_{11}}(\Delta^2)\right]x^r(1-x)^s,$$

where N is chosen such that the first moments correctly give the FFs and r and s can be fine-tuned to incoming data.

Need some Transition FFs: use MAID data!

$$A_{1/2} = \sqrt{\frac{2\pi\alpha}{k_R}} \frac{1}{(M_R + M_N)} \left(F_1(Q^2) + \frac{(M_R - M_N)}{2M_N} F_2(Q^2) \right)$$

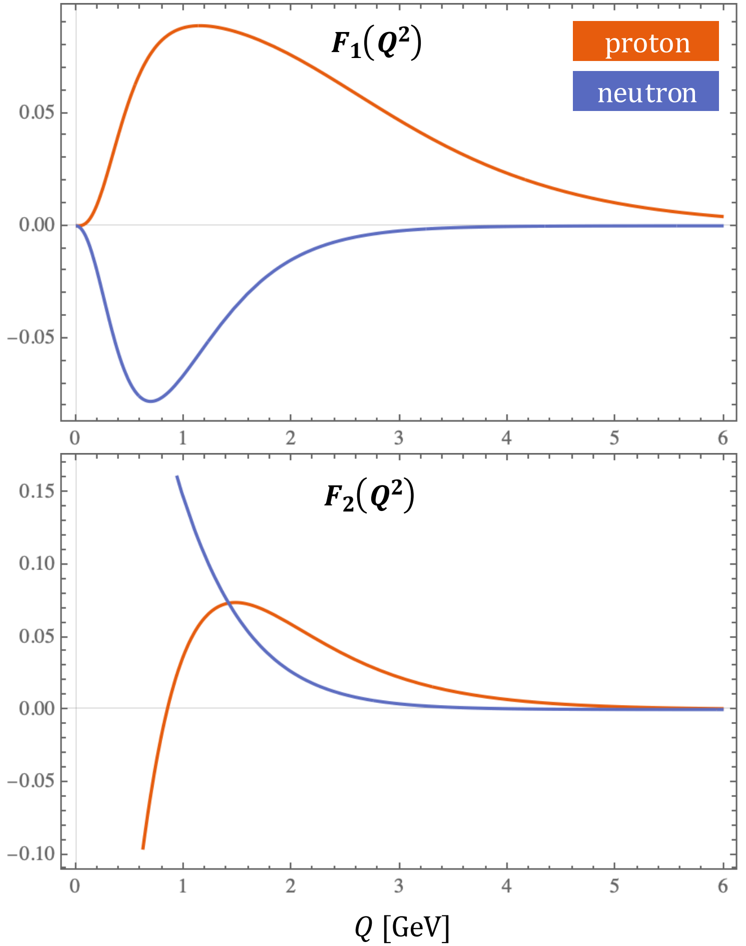
$$S_{1/2} = \sqrt{\frac{2\pi\alpha}{k_R}} \frac{\sqrt{Q^2}}{(M_R + M_N)} \left(F_1(Q^2) - \frac{(M_R + M_N)}{2M_N} F_2(Q^2) \right)$$

$$\bar{A}_\alpha(Q^2) = \bar{A}_\alpha(0)(1 + a_1 Q^2 + a_2 Q^4 + a_3 Q^6 + a_4 Q^8)\, e^{-b_1 Q^2}.$$

$N^*,\ \Delta^*$		$\bar{A}_\alpha(0)$	a_1	a_2	a_4	b_1
$P_{11}(1440)p$	$A_{1/2}$	-61.4	0.871	-3.516	-0.158	1.36
	$S_{1/2}$	4.2	$40.$	0	1.50	1.75

N^*		$\bar{A}_\alpha(0)$	a_1	b_1
$P_{11}(1440)n$	$A_{1/2}$	54.1	0.95	1.77
	$S_{1/2}$	-41.5	2.98	1.55

THE CURRENT STAGE



Then what???

1. Integrate over the solid angle to get a total cross-section.
2. Repeat the process for the pion-emitted first and Bethe-Heitler interference terms.
3. Compare the cross-sections.

4. Repeat everything for other resonances or extend to other meson-baryon molecules

- Program can be converted into a script that takes as input the kinematics, resonance, and meson and returns a total cross-section.

compute time ≈ 6 minutes

Exploring Nucleon Resonances with DVCS:

HOW TO MAKE MATHEMATICA DO TENSOR ALGEBRA

Matthew Rumley *(Supervisor: Anthony Thomas)*
The University of Adelaide

Further Reading

- Drechsel, D., Kamalov, S.S., Tiator, L. 2007 *Unitary Isobar Model - MAID2007*. Eur.Phys.J.A, 34 69-97 (2007). https://doi.org/10.1140/epja/i2007-10490-6
- Tiator, L., Drechsel, D., Kamalov, S.S. et al. *Electromagnetic excitation of nucleon resonances.* Eur. Phys. J. Spec. Top. 198, 141–170 (2011). https://doi.org/10.1140/epjst/e2011-01488-9
- Semenov-Tian-Shansky, K.M., Vanderhaedhen, M. *Deeply virtual Compton process* $e^{-}N \rightarrow e^{-}\gamma\pi N$ *to study nucleon to resonance transitions*. Phys. Rev. D 108, 034021. https://doi.org/10.1103/PhysRevD.108.034021