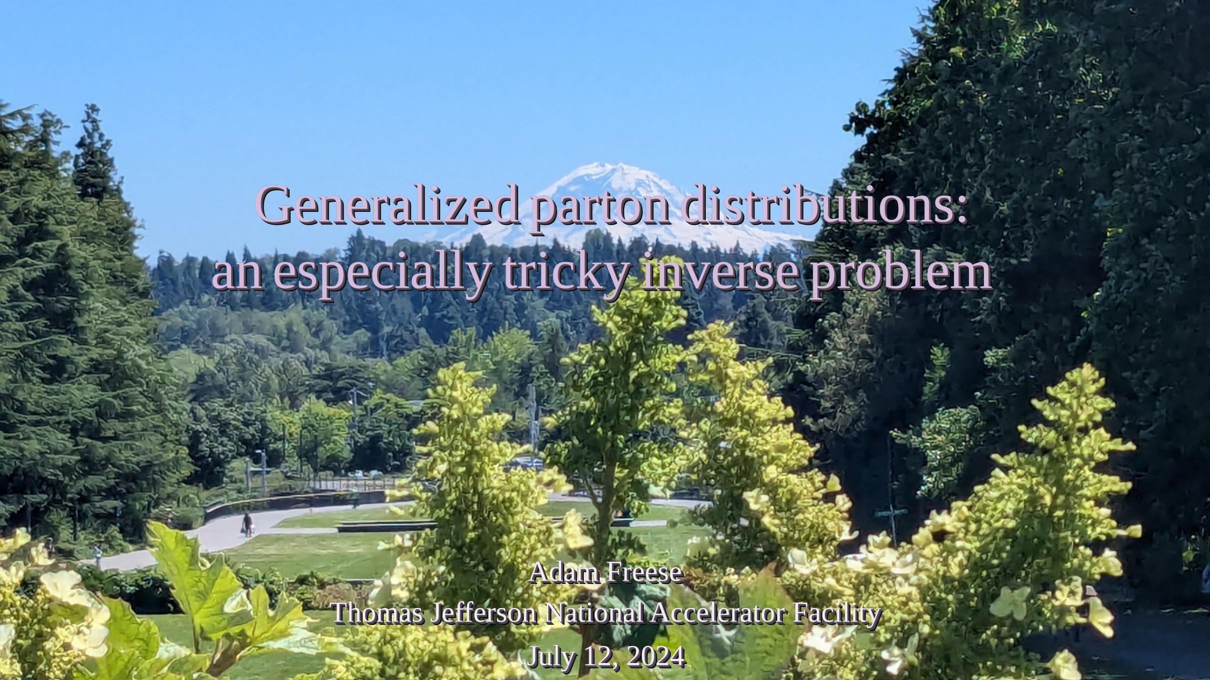


Generalized parton distributions: an especially tricky inverse problem

Adam Freese
Thomas Jefferson National Accelerator Facility

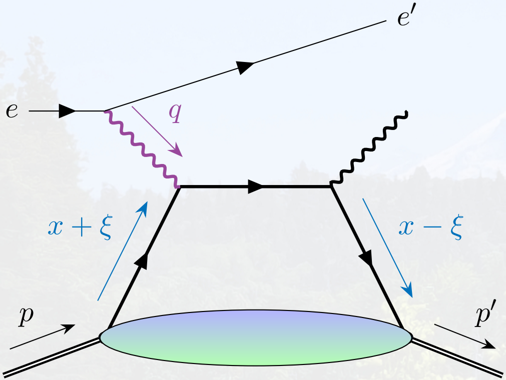
July 12, 2024

An especially tricky inverse problem

- The goal: extract **generalized parton distributions** (GPDs) from **deeply virtual Compton scattering** (DVCS) events.
- The usual caveats of inverse problems apply:
  - Finite data, continuous functions.
  - Epistemic uncertainty (interpolation & extrapolation error).
- The issue is worse for GPDs—the formal inverse *almost* doesn't exist.
  - I'll explain this over the next few slides.
- **Caveats**:
  - This is all a work in progress.
  - I'm just one member of the team—I'll point you to others' slides where they can explain things better.
  - I'm not an expert in inverse problems or uncertainty quantification; if I'm saying or doing anything foolish, feel free to correct me—I'll benefit from being led on the right path!

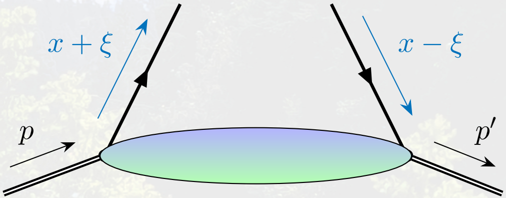

Generalized parton distributions

Generalized parton distributions



Deeply virtual Compton scattering

$\mathcal{H}(\xi, t; Q^2)$

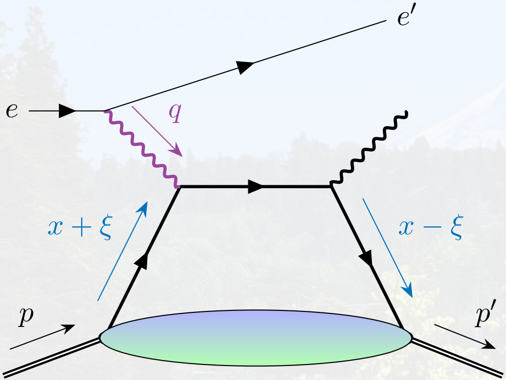


Generalized parton distribution

$H(x, \xi, t; Q^2)$

- **Generalized parton distributions** are 4-variable functions.
- Probed in processes such as **deeply virtual Compton scattering** (DVCS).
- Exciting because they encode **spatial distributions** of quarks and gluons.

The GPD variables



$$x = \frac{(k + k') \cdot n}{(p + p') \cdot n}$$

$$\xi = \frac{(p - p') \cdot n}{(p + p') \cdot n}$$

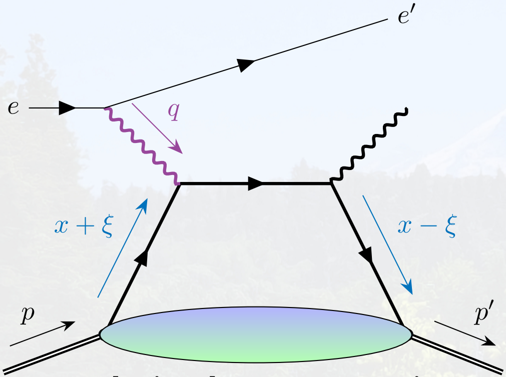
$$t = (p' - p)^2$$

$$Q^2 = -q^2$$

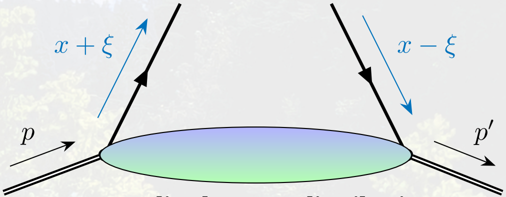
n defines the reference frame

- x is *average* momentum fraction of struck parton.
- 2ξ is the **skewness**: momentum fraction lost by struck parton.
- t is the invariant momentum transfer.
- GPDs also depend on **resolution scale** Q^2.

DVCS and GPDs



Deeply virtual Compton scattering
$\mathcal{H}(\xi, t; Q^2)$



Generalized parton distribution
$H(x, \xi, t; Q^2)$

- Loop in diagram: x is integrated out
- Integrated quantities seen in experiment: **Compton form factors**

$$\mathcal{H}(\xi, t; Q^2) = \int_{-1}^{1} \mathrm{d}x \, C(x, \xi) H(x, \xi, t; Q^2) \overset{\text{LO}}{=\!=} \int_{-1}^{1} \mathrm{d}x \left[\frac{1}{\xi - x - i0} \mp \frac{1}{\xi + x - i0} \right] H(x, \xi, t; Q^2)$$

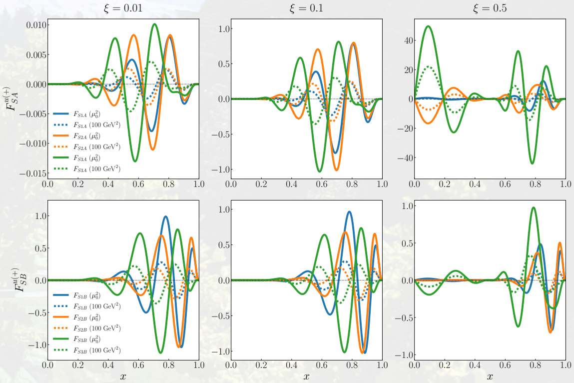
- Need to invert the relationship:

$$\mathcal{H}(\xi, t; Q^2) = \int_{-1}^{1} \mathrm{d}x\, C(x, \xi) H(x, \xi, t; Q^2)$$

- For *fixed* Q^2, the inverse doesn't exist!
- Multiple solutions encoded by **shadow GPDs**:

$$\int_{-1}^{1} \mathrm{d}x\, C(x, \xi) \mathfrak{h}(x, \xi, t; Q_0^2) = 0$$

  - $H(x, \xi, t, Q_0^2)$ and
  - $H(x, \xi, t, Q_0^2) + \mathfrak{h}(x, \xi, t, Q_0^2)$ give the same physical amplitude.
  - Bertone, *et al.*, PRD103 (2021) 114019
- Akin to inverting a 4×3 matrix.
- But *this is not the end*.



Examples of shadow GPDs.

Evolution equations

- GPDs obey **evolution equations** for Q^2 dependence:

$$\frac{\mathrm{d}H(x,\xi,t,Q^2)}{\mathrm{d}\log(Q^2)} = \int_{-1}^{+1} \mathrm{d}y\, K(x,y,\xi,Q^2) H(y,\xi,t,Q^2) \equiv K \otimes H$$

  - **Kernel** $K(x,y,\xi,Q^2)$ known theoretically (up to NLO).
  - Basically a generalization of DGLAP evolution.
- Only need 3D GPD at one scale Q_0^2 to fix 4D GPD at all Q^2.

$$H(x,\xi,t,Q^2) = \left(\frac{Q^2}{Q_0^2}\right)^{K\otimes} H(y,\xi,t,Q_0^2)$$

Please excuse the horrendous abuse of notation!

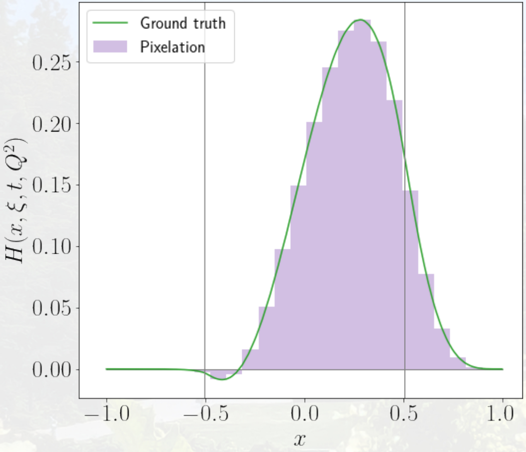
- Shadow GPDs evolve into non-shadows:

$$\mathcal{H}_{\text{shadow}}(\xi,t,Q^2) = \int_{-1}^{1} \mathrm{d}x\, C(x,\xi) \left(\frac{Q^2}{Q_0^2}\right)^{K\otimes} \mathfrak{h}(y,\xi,t,Q_0^2) \neq 0$$

  - Three variable function $\rightarrow$ three variable function.
  - We now have a proper inverse problem.
  - The inverse exists, but how well can we find it—given finite data, with uncertainties?


The framework

Pixelation



- We **pixelate** the GPD:

$$H(x, \xi, t, Q^2) \to H_{ijkl}$$

- Avoids biases of functional forms.
- Meshes well with finite element methods.
- Number of needed pixels furnishes a resolution. (A kind of uncertainty quantification?)
- Integrals become tensor contraction:

$$\int \mathrm{d}y\, C(y) H(y, \xi, t, Q^2) \to \sum_i C_i H_{ijkl}$$

  - **Fast and differentiable!**

Notes on pixelation

- Pixel widths & placement constitute a covert model dependence.
- Linear vs. logarithmic spacing is motivated by expected functional behavior.
- Allowing pixel widths to float being explored—talk by Daniel Adamiak (see QR code).

Interpixels

- **Interpixels** (**interp**olated **pixel**): interpolation basis functions.
  - Exploit linearity of polynomial (e.g., Newton) interpolation:

$$N[y_1 + y_2](x) = N[y_1](x) + N[y_2](x)$$

  - GPD pixelation is a sum of pixels:

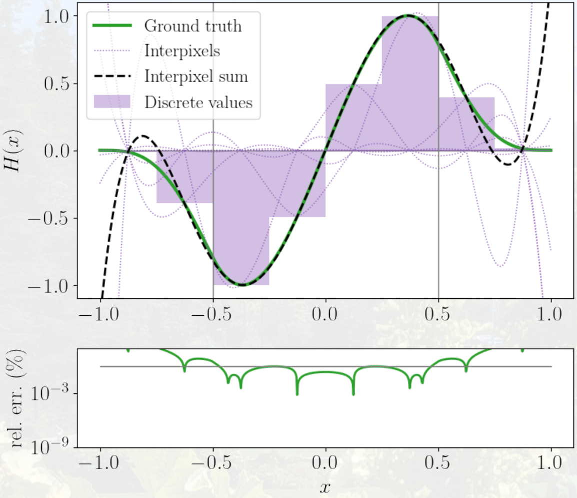
$$\boldsymbol{H} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{bmatrix} = H_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + H_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \ldots + H_n \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \equiv H_1 \hat{e}_1 + H_2 \hat{e}_2 + \ldots + H_n \hat{e}_n$$

  - Interpolated pixelation is a sum of interpixels!

$$N[\boldsymbol{H}](x) = H_1 N[\hat{e}_1](x) + H_2 N[\hat{e}_2](x) + \ldots + H_n N[\hat{e}_n](x)$$

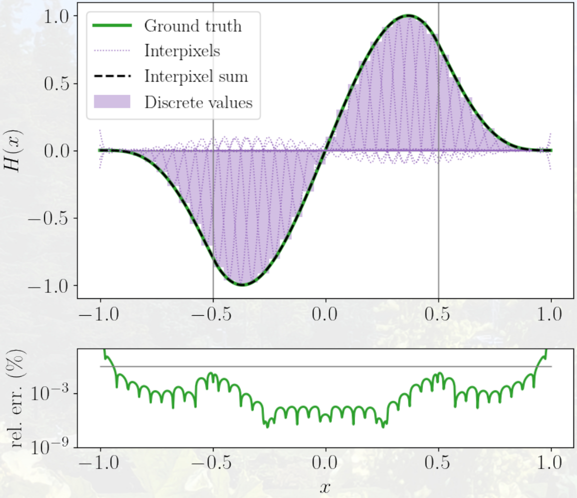
- Basically a shoddy **finite element method**.
  - I just learned this Wednesday that this can be done better.
- Get convolution matrices by putting $H[\hat{e}_j](x)$ into integrals.

Interpixel demo



$$n_x = 8$$

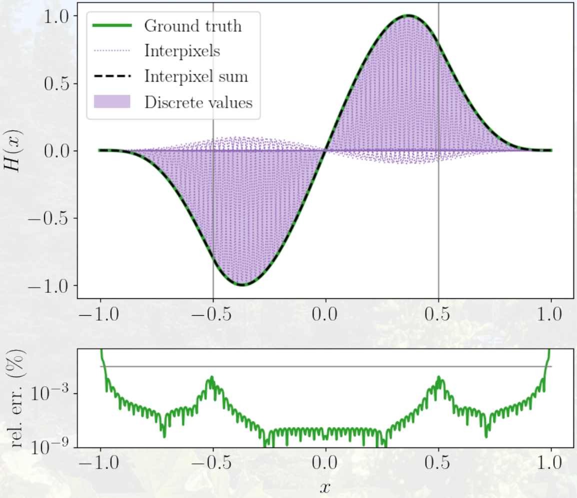
- Interpixel is a *piecewise* polynomial.
  - Of fixed order.
  - Avoids Runge phenomennon.
- Knots on the discrete x grid.
- Each interpixel is oscillatory.
- Oscillations cancel in sum.
- Improvement at high N_x.



$n_x = 40$

- Interpixel is a *piecewise* polynomial.
  - Of fixed order.
  - Avoids Runge phenomennon.
- Knots on the discrete x grid.
- Each interpixel is oscillatory.
- Oscillations cancel in sum.
- Improvement at high N_x.

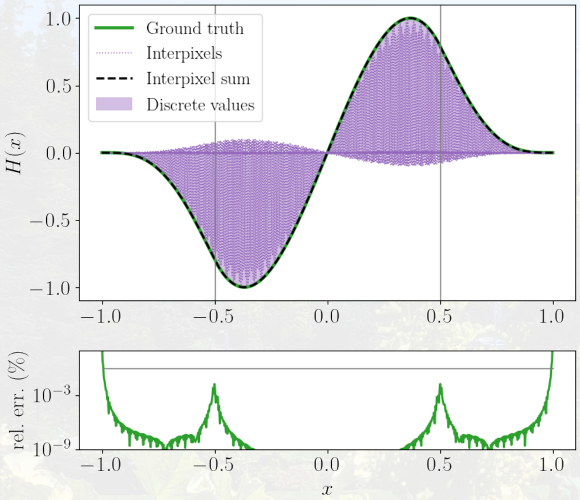
Interpixel demo



$$n_x = 100$$

- Interpixel is a *piecewise* polynomial.
  - Of fixed order.
  - Avoids Runge phenomennon.
- Knots on the discrete x grid.
- Each interpixel is oscillatory.
- Oscillations cancel in sum.
- Improvement at high N_x.

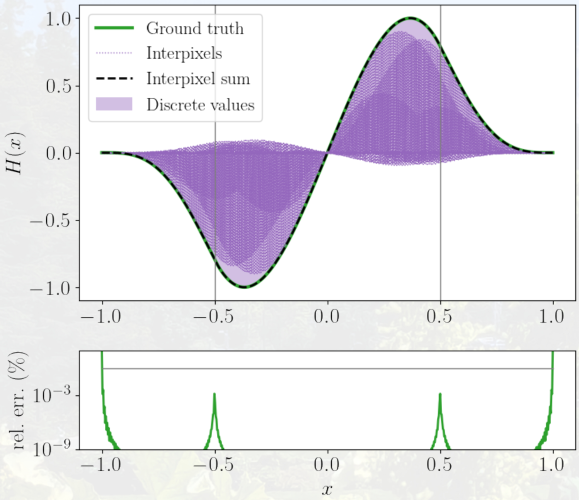
Interpixel demo



$$n_x = 300$$

- Interpixel is a *piecewise* polynomial.
  - Of fixed order.
  - Avoids Runge phenomennon.
- Knots on the discrete x grid.
- Each interpixel is oscillatory.
- Oscillations cancel in sum.
- Improvement at high N_x.

Interpixel demo



$$n_x = 1000$$

- Interpixel is a *piecewise* polynomial.
  - Of fixed order.
  - Avoids Runge phenomennon.
- Knots on the discrete x grid.
- Each interpixel is oscillatory.
- Oscillations cancel in sum.
- Improvement at high N_x.

Convolutions as tensor contractions

- A pixelated GPD is a rank-4 tensor:

$$H(y, \xi_j, t_k, Q_l^2) \approx \sum_{i=1}^{N_x} H_i(\xi_j, t_k, Q_l^2)\hat{e}_i(y)$$

- Put inside an integral—e.g., for the Compton form factor:

$$\mathcal{H}(\xi_j, t_k, Q_l^2) = \int_{-1}^{+1} \mathrm{d}y\, C(y, \xi_j) H(y, \xi_j, t_k, Q_l^2) = \sum_{i=1}^{N_x} \underbrace{\left(\int_{-1}^{+1} \mathrm{d}y\, C(y, \xi_j)\hat{e}_i(y)\right)}_{C_i} H_i(\xi_j, t_k, Q_l^2)$$

  - Getting the Compton form factor entails a loss of rank:

$$\mathcal{H}_{jkl} = \sum_{i=1}^{N_x} C_i H_{ijkl}$$

  - Of course this operation isn't invertible.

Evolution as a gain of rank

- Evolution of pixelated GPDs (deferring Q^2 discretization):

$$\frac{\mathrm{d}H_{ijk}(Q^2)}{\mathrm{d}\log Q^2} = \sum_{i'=1}^{N_x} K_{ii'jk}(Q^2) H_{i'jk}(Q^2)$$

- Solution:

$$H_{ijkl} = \sum_{i'=1}^{N_x} M_{ii'jkl} H_{i'jk}(Q_0^2)$$

  - Numerically implemented via RK4.
- Compton form factors in terms of model scale GPD:

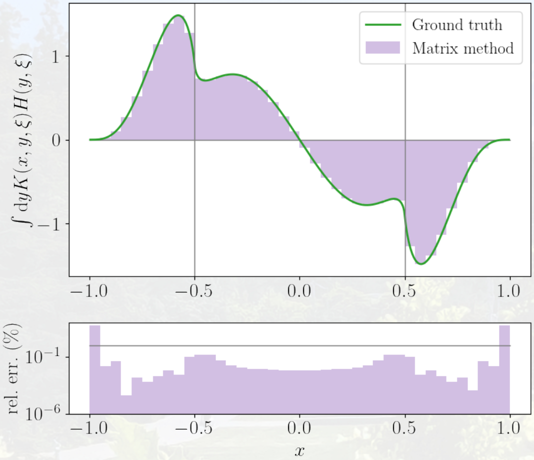
$$\mathcal{H}_{jkl} = \sum_{i'=1}^{N_x} \underbrace{\left(\sum_{i=1}^{N_x} C_i M_{ii'jkl} \right)}_{\mathcal{M}_{li'}(\xi_j, t_k)} H_{i'jk}(Q_0^2)$$

  - Effectively matrix multiplication ($x_{i'}$ dependence $\to$ Q_l^2 dependence).
  - **Inverse problem**: invert $\mathcal{M}$ (in terms of x and Q^2 indices).

Talk by me on GPD evolution

Evolution accuracy benchmark (non-singlet evolution)

$n_x = 40$

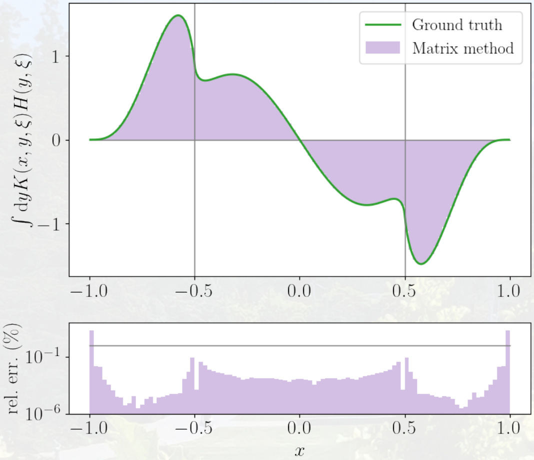


- "Ground truth" determined by adaptive integration of model function.
- Error represents error from both pixelation & interpolation.
- Sub-percent error even at $n_x = 40$.

- Probably a source of epistemic uncertainty in extractions.

Evolution accuracy benchmark (non-singlet evolution)

$n_x = 100$

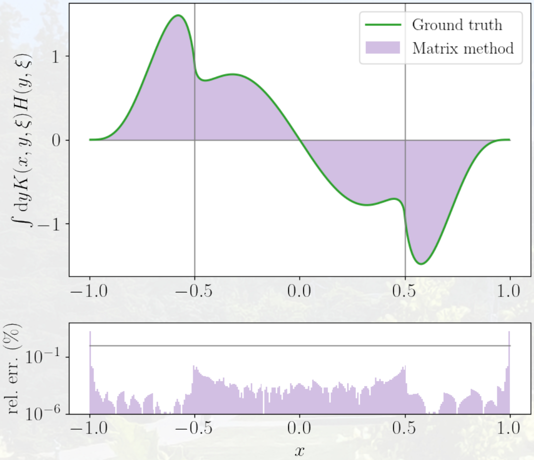


- "Ground truth" determined by adaptive integration of model function.
- Error represents error from both pixelation & interpolation.
- Sub-percent error even at $n_x = 40$.

- Probably a source of epistemic uncertainty in extractions.

Evolution accuracy benchmark (non-singlet evolution)

$n_x = 300$

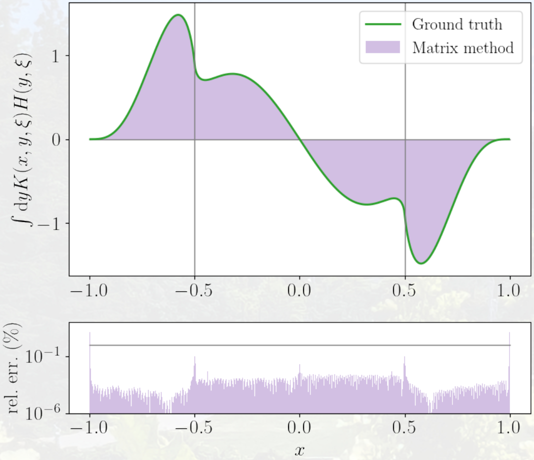


- "Ground truth" determined by adaptive integration of model function.
- Error represents error from both pixelation & interpolation.
- Sub-percent error even at $n_x = 40$.

- Probably a source of epistemic uncertainty in extractions.

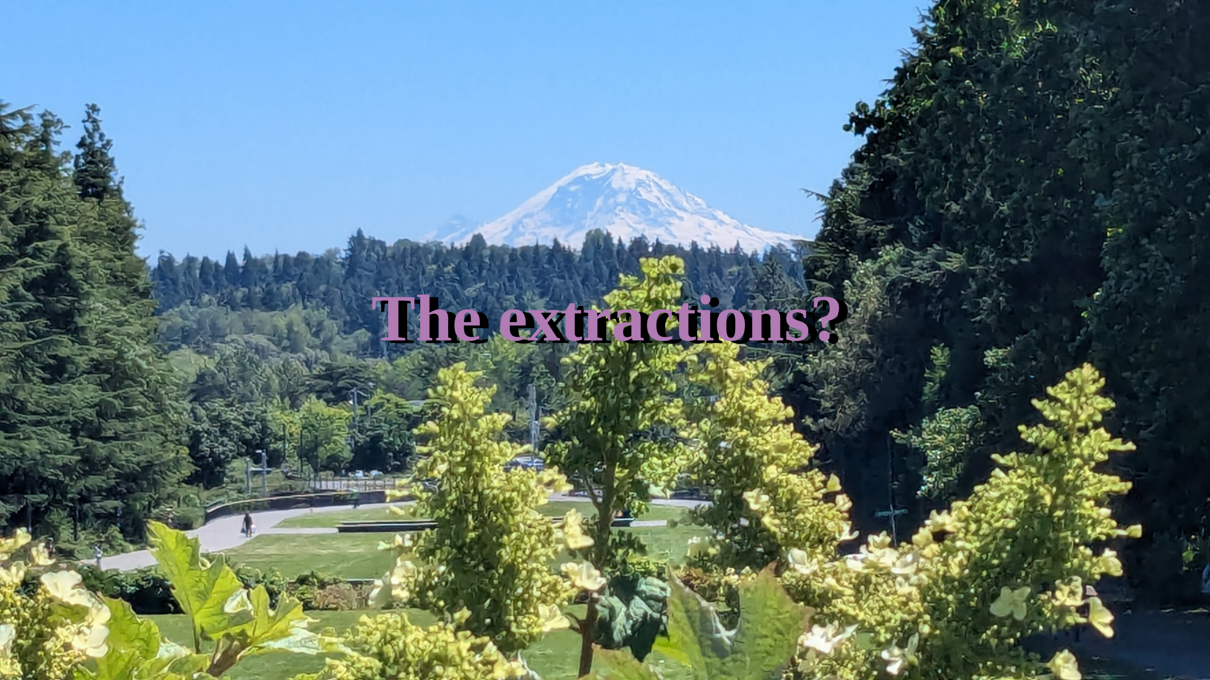
Evolution accuracy benchmark (non-singlet evolution)

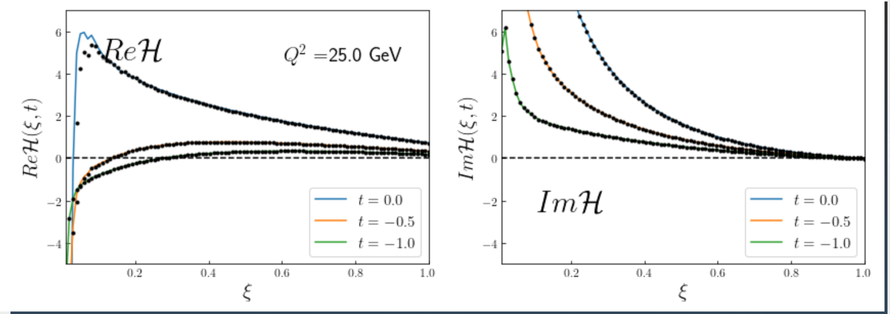
$n_x = 1000$



- "Ground truth" determined by adaptive integration of model function.
- Error represents error from both pixelation & interpolation.
- Sub-percent error even at $n_x = 40$.

- Probably a source of epistemic uncertainty in extractions.

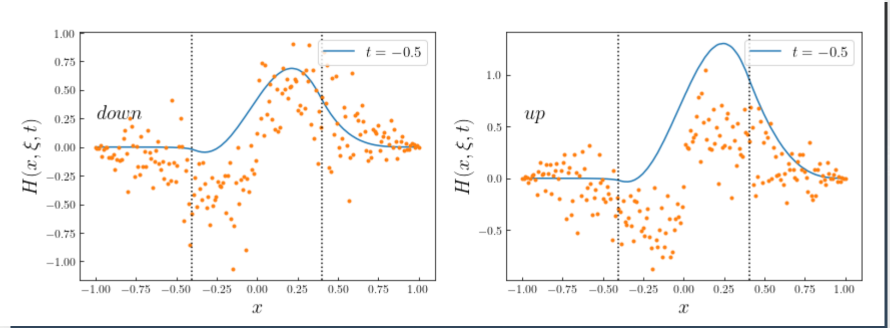

The extractions?



Talk by Marco Zaccheddu

- A “neural network” (one linear layer) fit.
- Compton form factors can be “extracted” quite accurately …
  - …with lots of arbitrarily precise data.
- **Preliminary toy extraction**

Extraction: generalized parton distributions



Talk by Marco Zaccheddu

- A “neural network” (one linear layer) fit.
- This inverse problem appears unsolved by evolution.
- **Preliminary toy extraction**
- **Remaining work**: uncertainty quantification, constructing/exploring latent space.

Credits

- Daniel Adamiak
- Ian Cloët
- Chris Cocuzza
- Adam Freese
- Nobuo Sato
- Marco Zaccheddu

Thank you for your time!