ML Techniques & Mass Reconstruction

Siddhartha Gupte

Stony Brook University

May 18, 2026

*This work is supported by **DOE** Grant No. **1185047-1-98541**.*
DarkLight has been supported by DOE, NSF, NSERC, and the Moore Foundation.

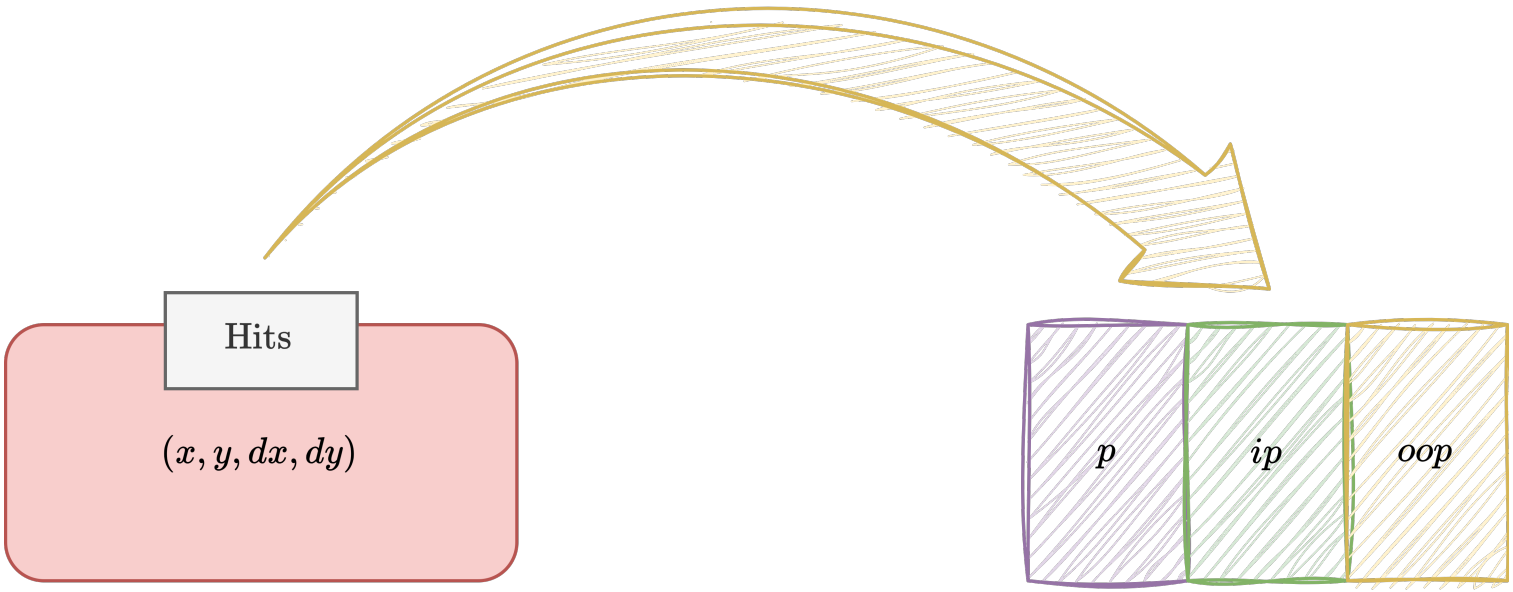
DARKLIGHT @ TRIUMF Advanced Rare Isotope Laboratory (ARIEL)

Stony Brook University

Outline

Kinematic Reconstruction

- **Goal**: Infer particle kinematics at the interaction vertex using GEM detector hits.
- Small uncertainties in momentum and angles $\rightarrow$ sharper mass peaks $\rightarrow$ better sensitivity.



From Detector Observables to Kinematics

- Reconstruction framed as a **supervised regression problem**:

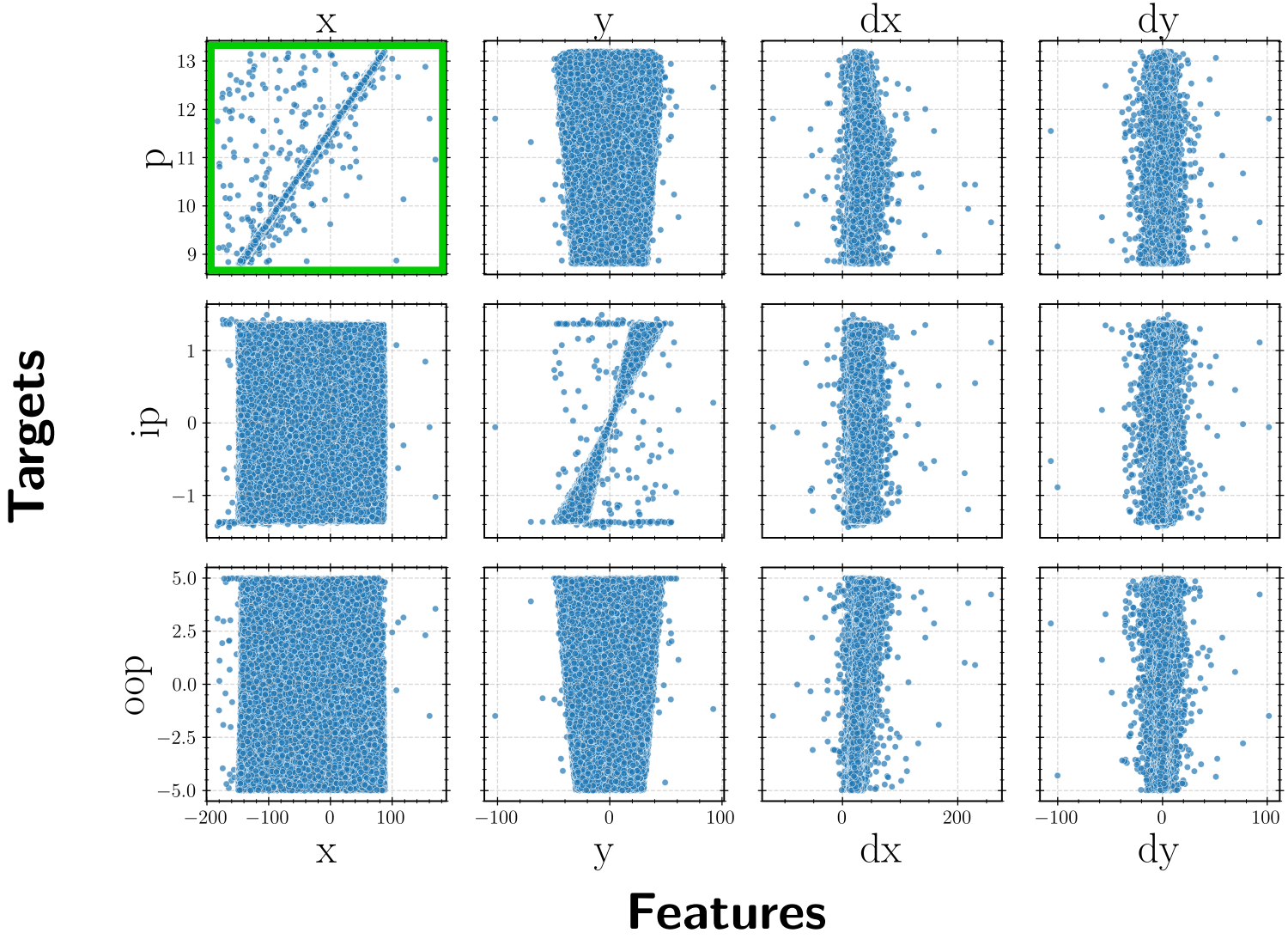
$$f_1 : (x, y, dx, dy) \rightarrow p$$
$$f_2 : (x, y, dx, dy) \rightarrow ip$$
$$f_3 : (x, y, dx, dy) \rightarrow oop$$

- **Inputs:** GEM hit positions (x, y) and "*local hit displacement*" vectors (dx, dy).
- **Targets:** Particle momentum p, in-plane angle ip, out-of-plane angle oop.
- Goal: minimize prediction error (e.g., mean squared error, MSE) between predicted and true kinematics.

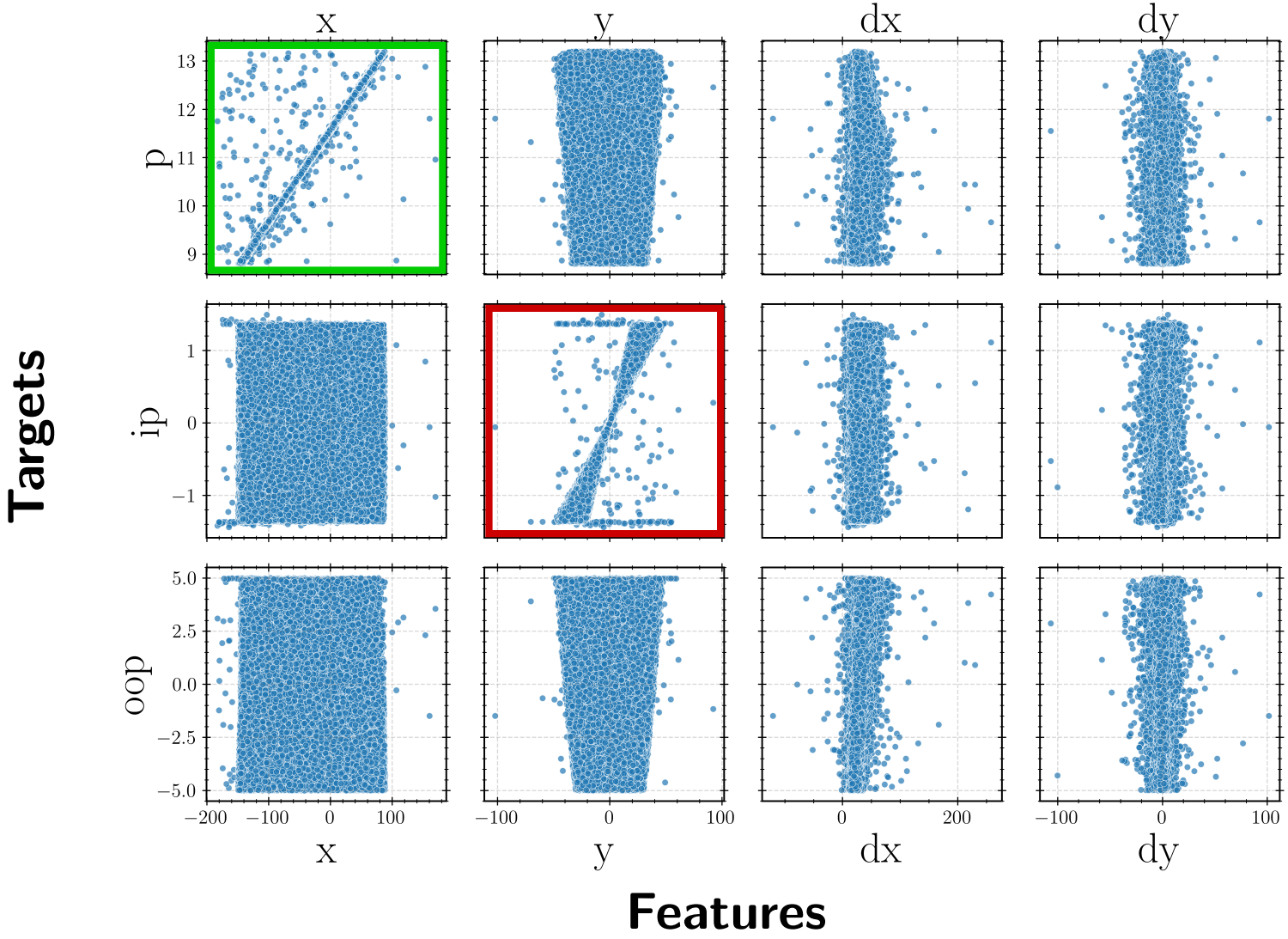
Why Machine Learning?

- Relationship between detector observables (x, y, dx, dy) and kinematics is:
  - Highly **nonlinear** for the angular variables
  - Sensitive to detector-level variations
- Traditional analytic reconstruction becomes challenging.
- **Machine learning** provides a flexible way to learn this mapping directly from data.

Data: Linear Dependency



Data: Linear Dependency



Models Considered

- Gradient boosting methods:
  - **XGBoost**
  - **LightGBM**
- Both models:
  - Handle nonlinear relationships effectively
  - Work well with structured/tabular data
- **Goal**: Identify model with best accuracy and robustness for kinematic reconstruction.

Gradient Boosting: Key Idea

- Ensemble method that builds a model as a **sum of weak learners**:

$$F(x) = \sum_{m=1}^{M} f_m(x)$$

- Models are added **sequentially**, each correcting errors of the previous ones.
- At each step:
  - Fit a new model to the **residuals** (errors)
  - Reduce overall loss (e.g., MSE)
- Typically uses **decision trees** as base learners.

Instead of learning the mapping in one step, boosting builds it incrementally by correcting mistakes.

XGBoost vs LightGBM

- Both are implementations of gradient boosted decision trees for regression.
- **XGBoost:**
  - Level-wise tree growth
  - Strong regularization
  - Robust baseline for structured data
- **LightGBM:**
  - Leaf-wise tree growth (more flexible)
  - Better at capturing complex, nonlinear feature interactions
  - Faster training for large datasets

Key Difference

LightGBM's leaf-wise growth allows more precise modeling of complex detector $\rightarrow$ kinematics mappings.

Loss Function: Huber Loss

- Training performed using **Huber loss**.
- Residual: $r = y - f(x)$
- Combines advantages of:
  - Mean Squared Error (sensitive to small deviations)
  - Mean Absolute Error (robust to outliers)

$$L_\delta(r) = \begin{cases} \frac{1}{2}r^2 & |r| \leq \delta \\ \delta(|r| - \frac{1}{2}\delta) & |r| > \delta \end{cases}$$

Motivation

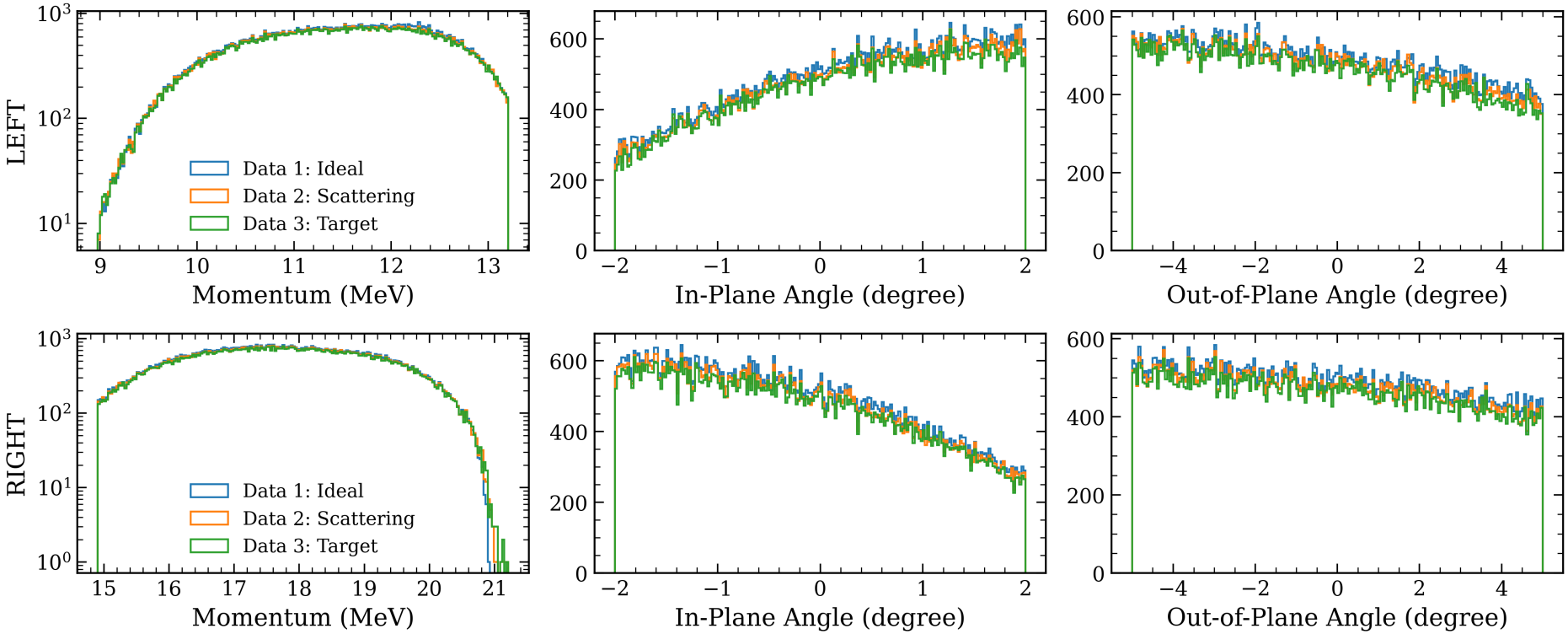
Provides robustness to outliers from detector effects, improving stability of reconstructed kinematics.

Datasets for Reconstruction

- Three datasets considered in this study:
  - **Ideal**: No detector effects, no scattering
  - **Scattering**: Includes multiple scattering effects
  - **Target**: Includes full detector and target-related effects
- For model comparison, we focus on the **Ideal dataset**
  - Provides a controlled environment
  - Allows direct evaluation of model capability

Data Distributions

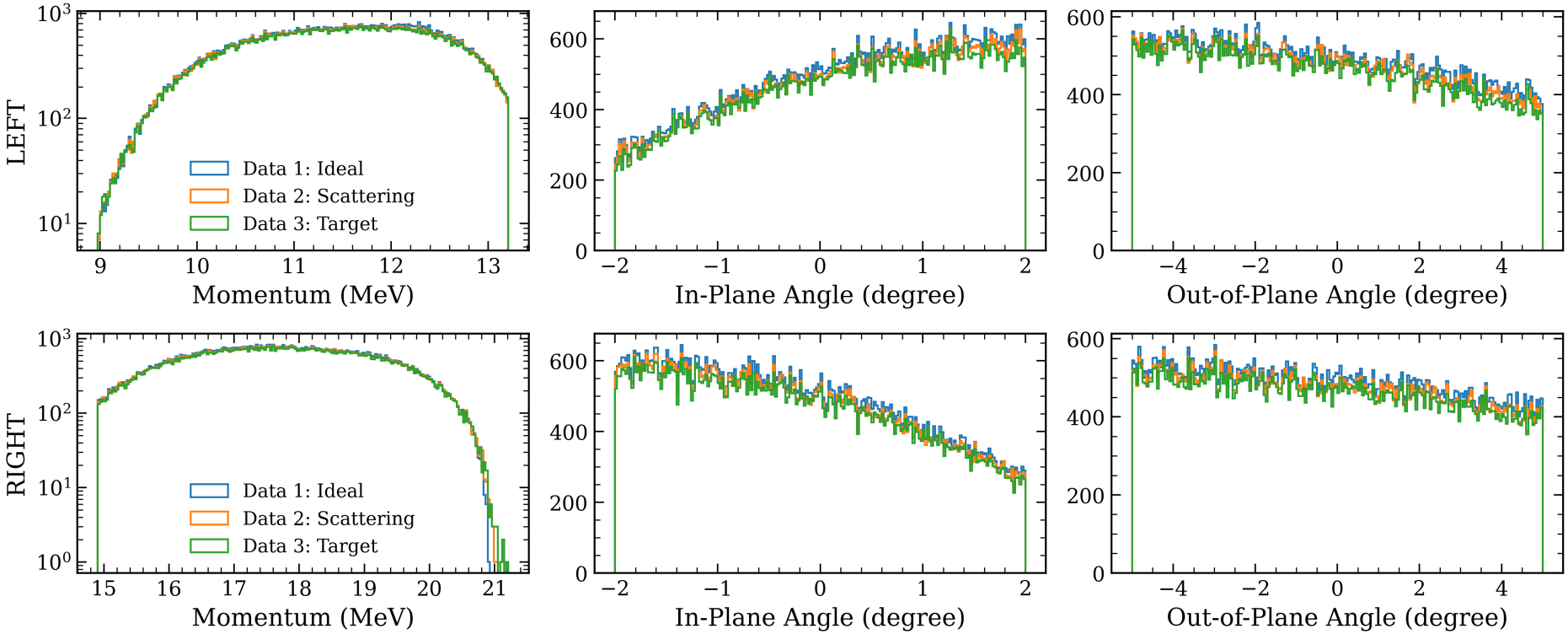
- Comparison of target variable distributions across datasets of increasing realism.



Distribution of target variables

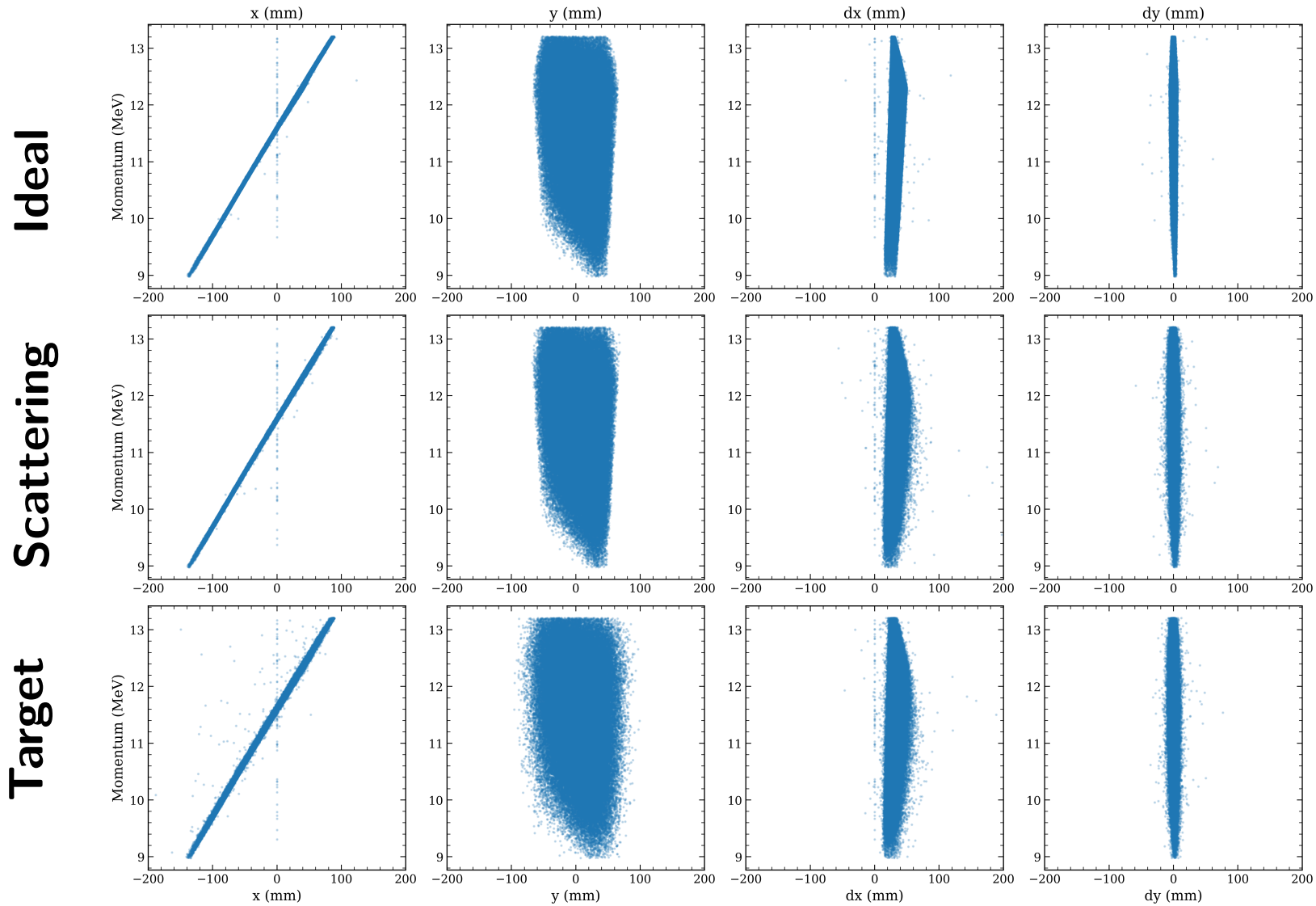
Data Distributions

- Comparison of target variable distributions across datasets of increasing realism.
- While 1D distributions look similar, pairwise relationships reveal degradation in realistic datasets.

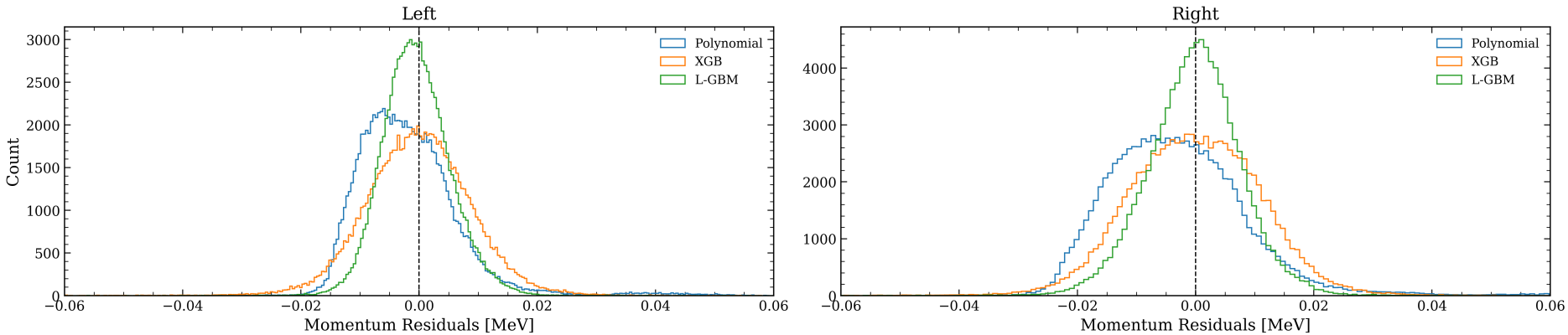


Distribution of target variables

Target 1: Momentum

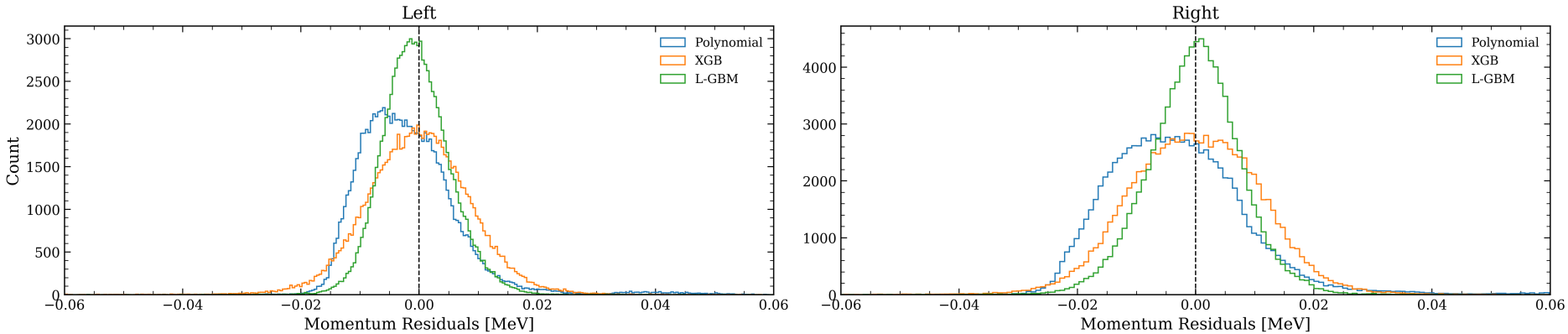


Target 1: Momentum



Residual Distribution on **Ideal** dataset

Target 1: Momentum

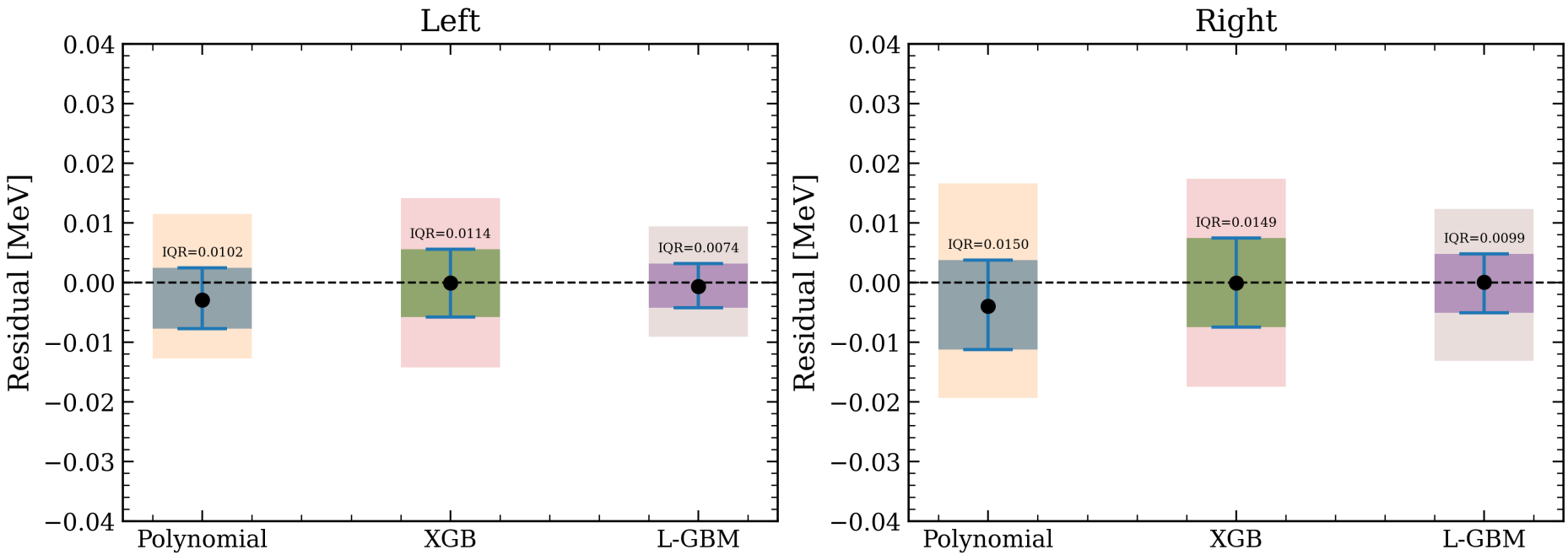


Residual Distribution on **Ideal** dataset

LEFT	MSE	R^2
Polynomial	0.000657	0.999212
XGB	0.000938	0.998875
LGB-B	0.000756	0.999093

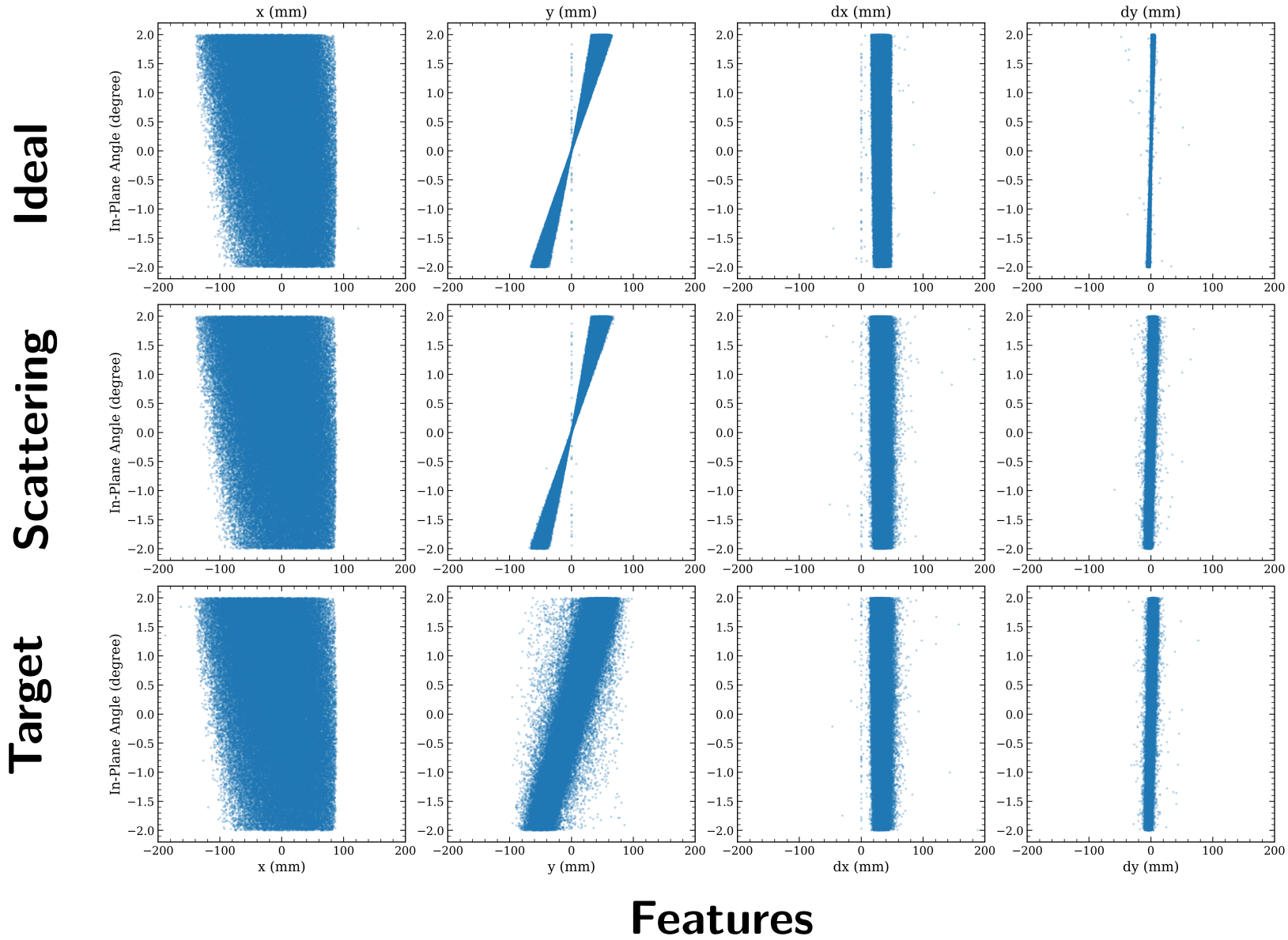
RIGHT	MSE	R^2
Polynomial	0.001589	0.999100
XGB	0.001391	0.999212
LGB-B	0.001157	0.999345

Target 1: Momentum

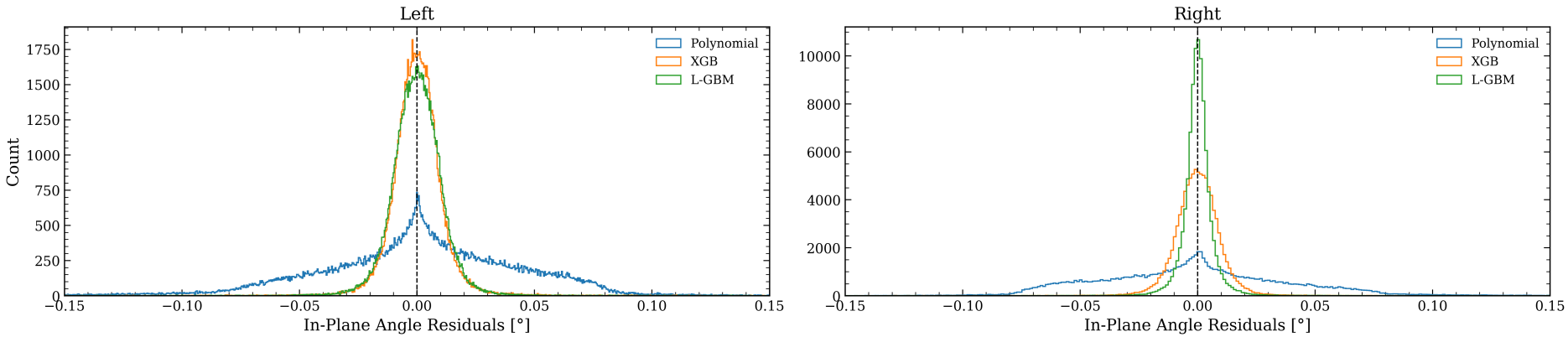


Interquartile range of residuals on **Ideal** dataset

Target 2: In-Plane Angle

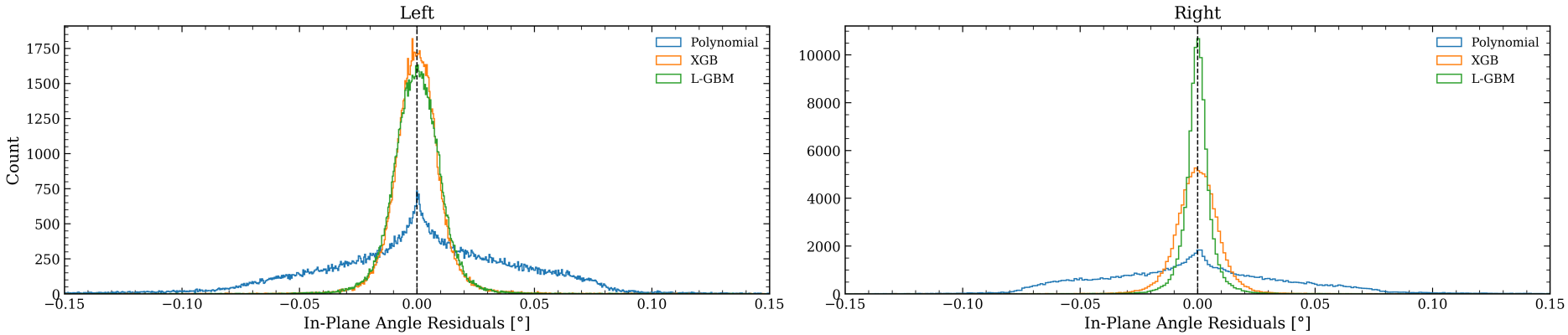


Target 2: In-Plane Angle



Residual Distribution on **Ideal** dataset

Target 2: In-Plane Angle

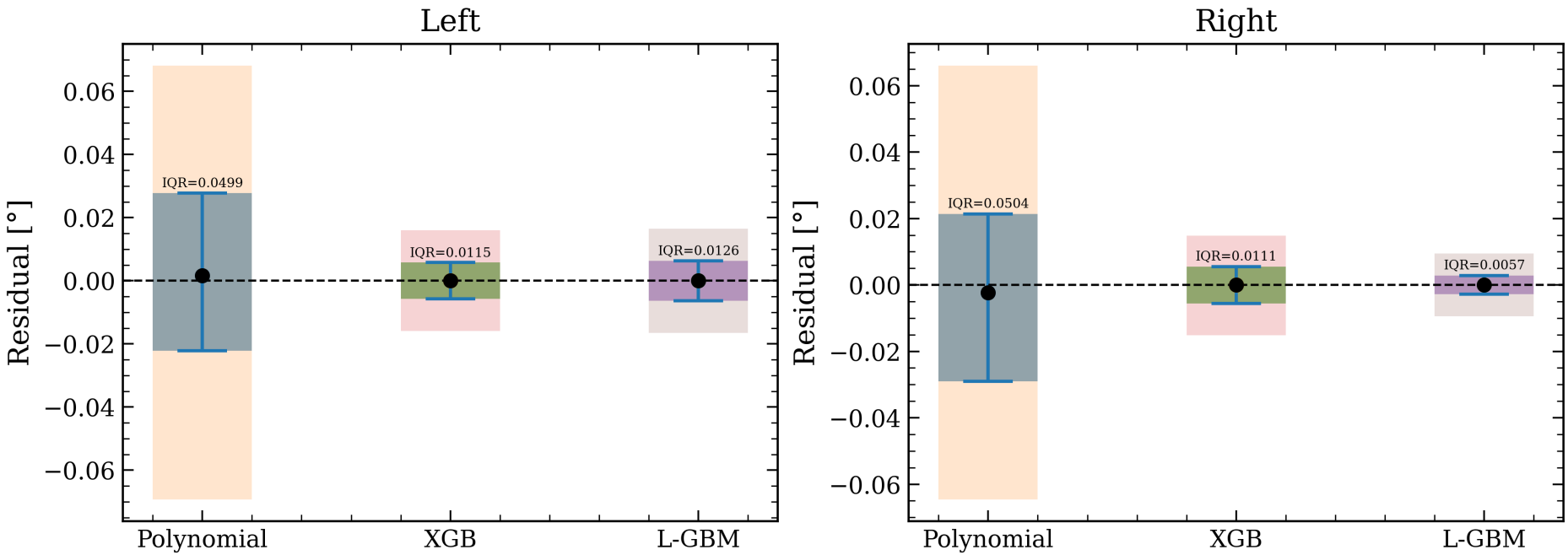


Residual Distribution on **Ideal** dataset

LEFT	**MSE**	**R^2**
Polynomial	0.002787	0.997722
XGB	0.0009768	0.999202
LGB-B	0.000817	0.999331

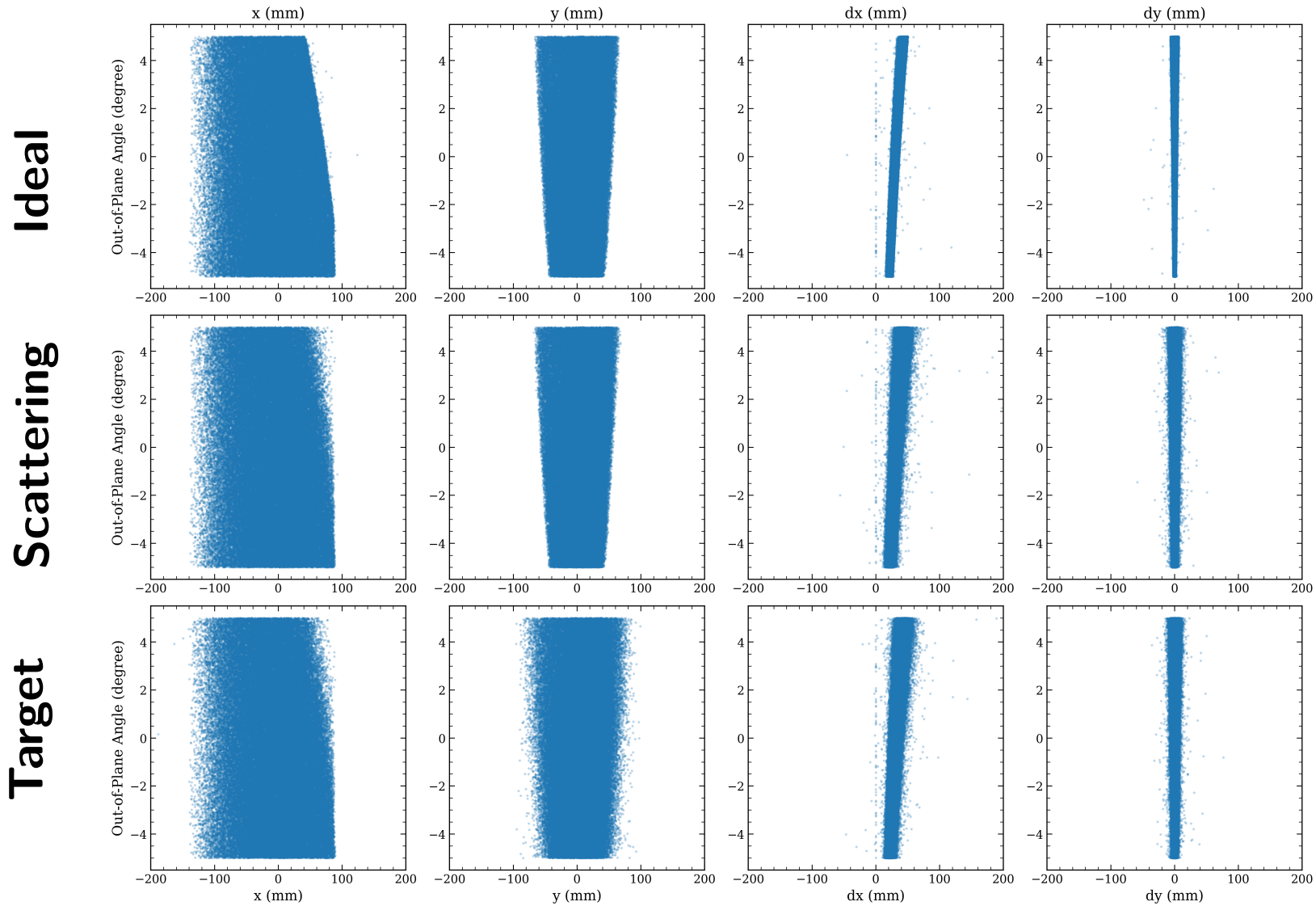
RIGHT	**MSE**	**R^2**
Polynomial	0.002970	0.997571
XGB	0.000981	0.999198
LGB-B	0.001789	0.998537

Target 2: In-Plane Angle

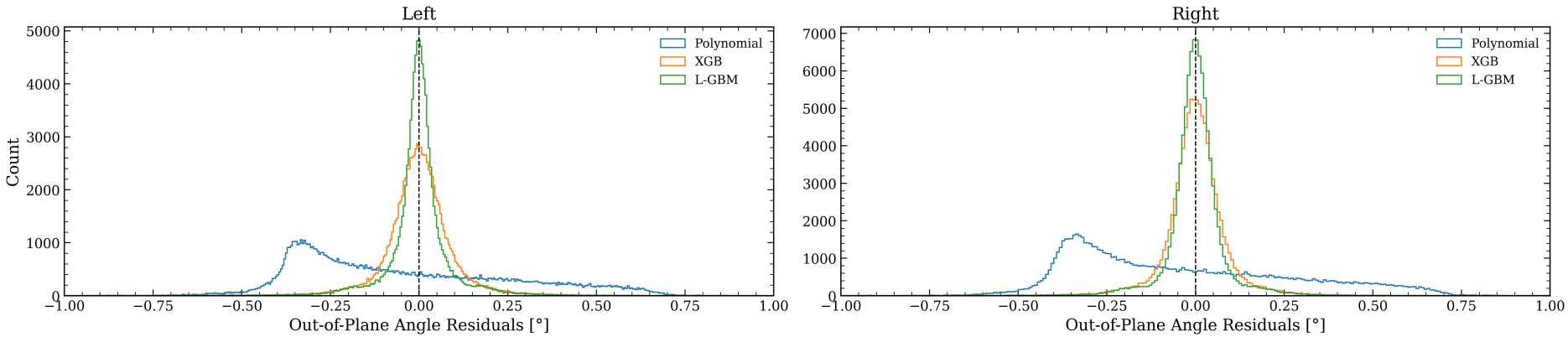


Interquartile range of residuals on **Ideal** dataset

Target 3: Out-of-Plane Angle

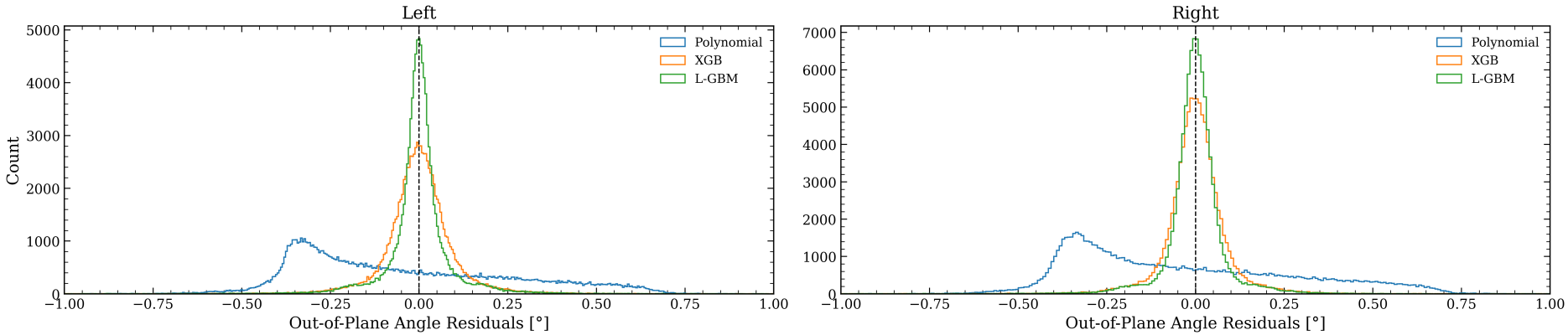


Target 3: Out-of-Plane Angle



Residual Distribution on **Ideal** dataset

Target 3: Out-of-Plane Angle

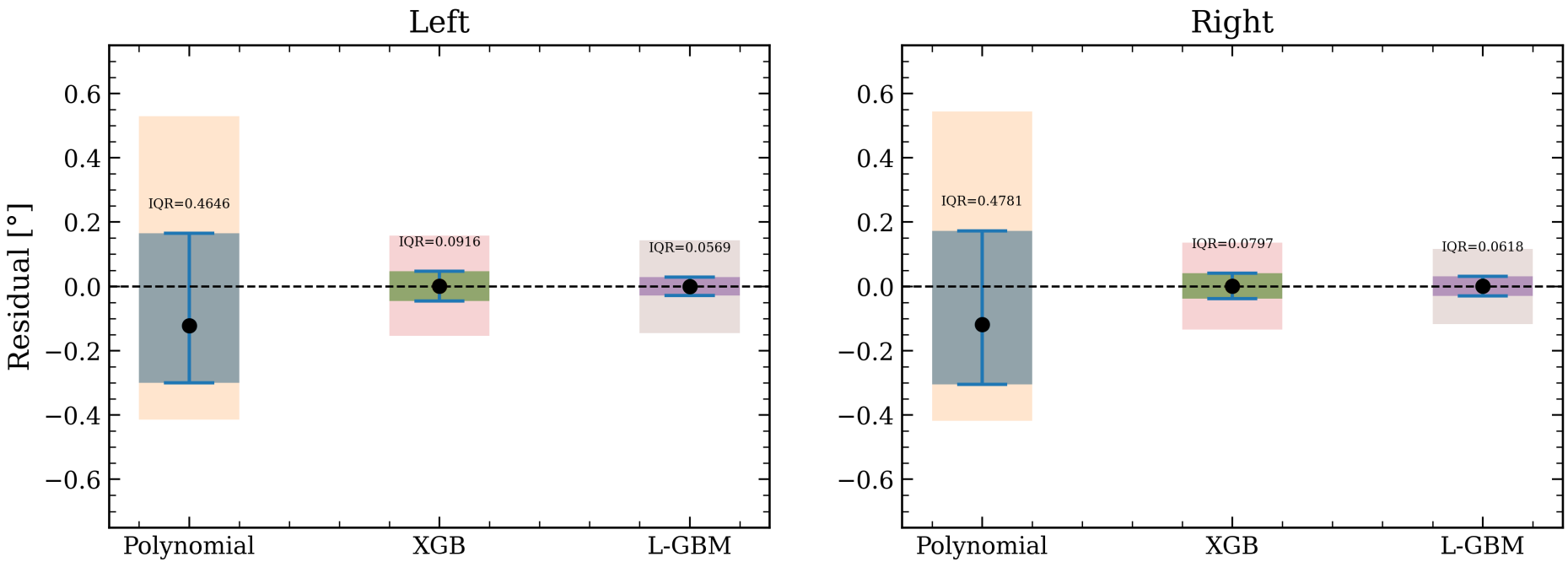


Residual Distribution on **Ideal** dataset

LEFT	MSE	R^2
Polynomial	0.284487	0.964983
XGB	0.040702	0.994990
LGB-B	0.040896	0.994966

RIGHT	MSE	R^2
Polynomial	0.265673	0.967613
XGB	0.032941	0.995984
LGB-B	0.031704	0.996135

Target 3: Out-of-Plane Angle

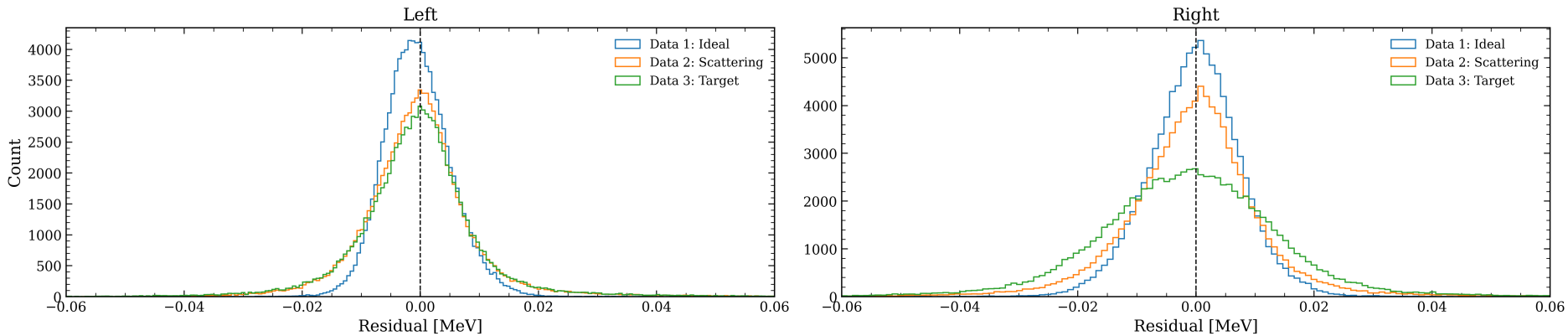


Interquartile range of residuals on **Ideal** dataset

Model Selection

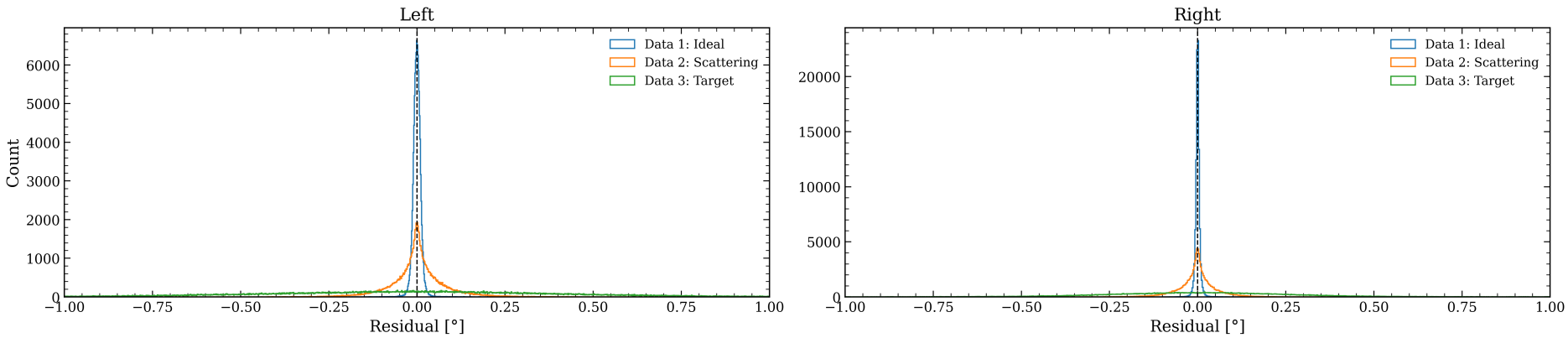
- LightGBM achieves the best balance of:
  - Lower MSE
  - Higher R^2
  - More consistent performance across (p, ip, oop)
- Better captures nonlinear detector $\rightarrow$ kinematics relationships

Model performance on the three datasets



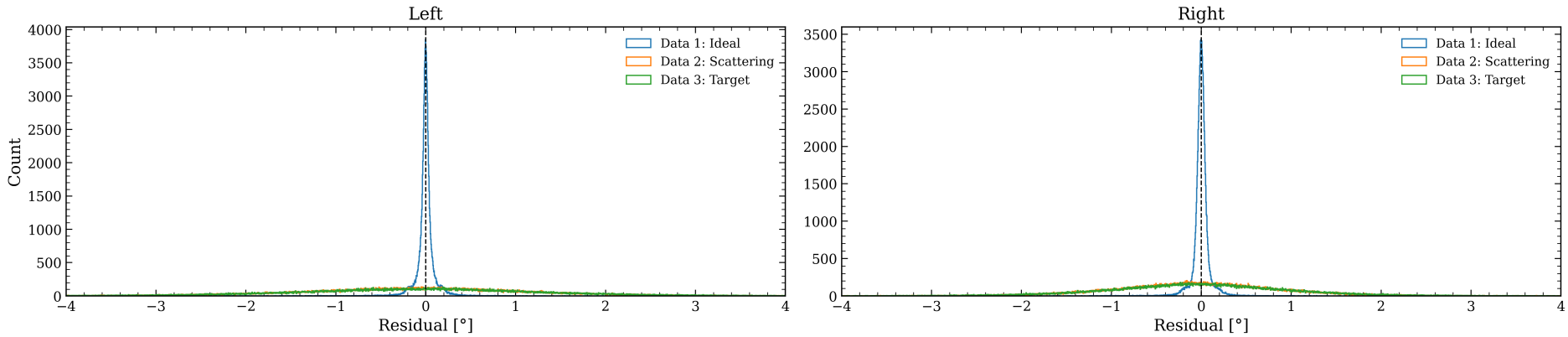
Momentum

Model performance on the three datasets



In-Plane Angle

Model performance on the three datasets

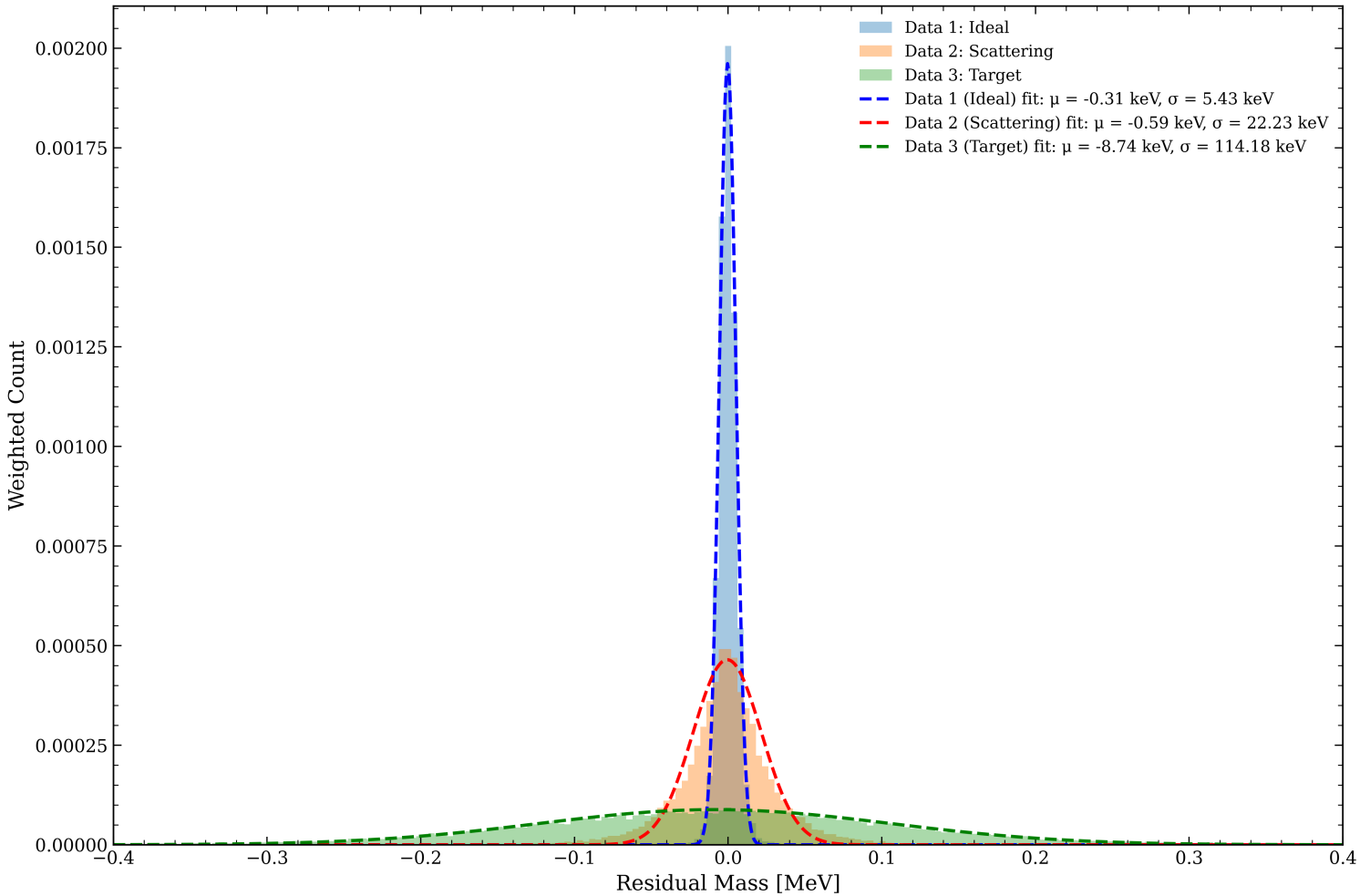


Out-of-Plane Angle

Summary of ML Reconstruction Results

- LightGBM selected for kinematic reconstruction.
- Demonstrated high accuracy and precision across all kinematic variables (p, ip, oop).
- Performance decreases significantly for more realistic datasets, highlighting *scattering* and *target* effects.

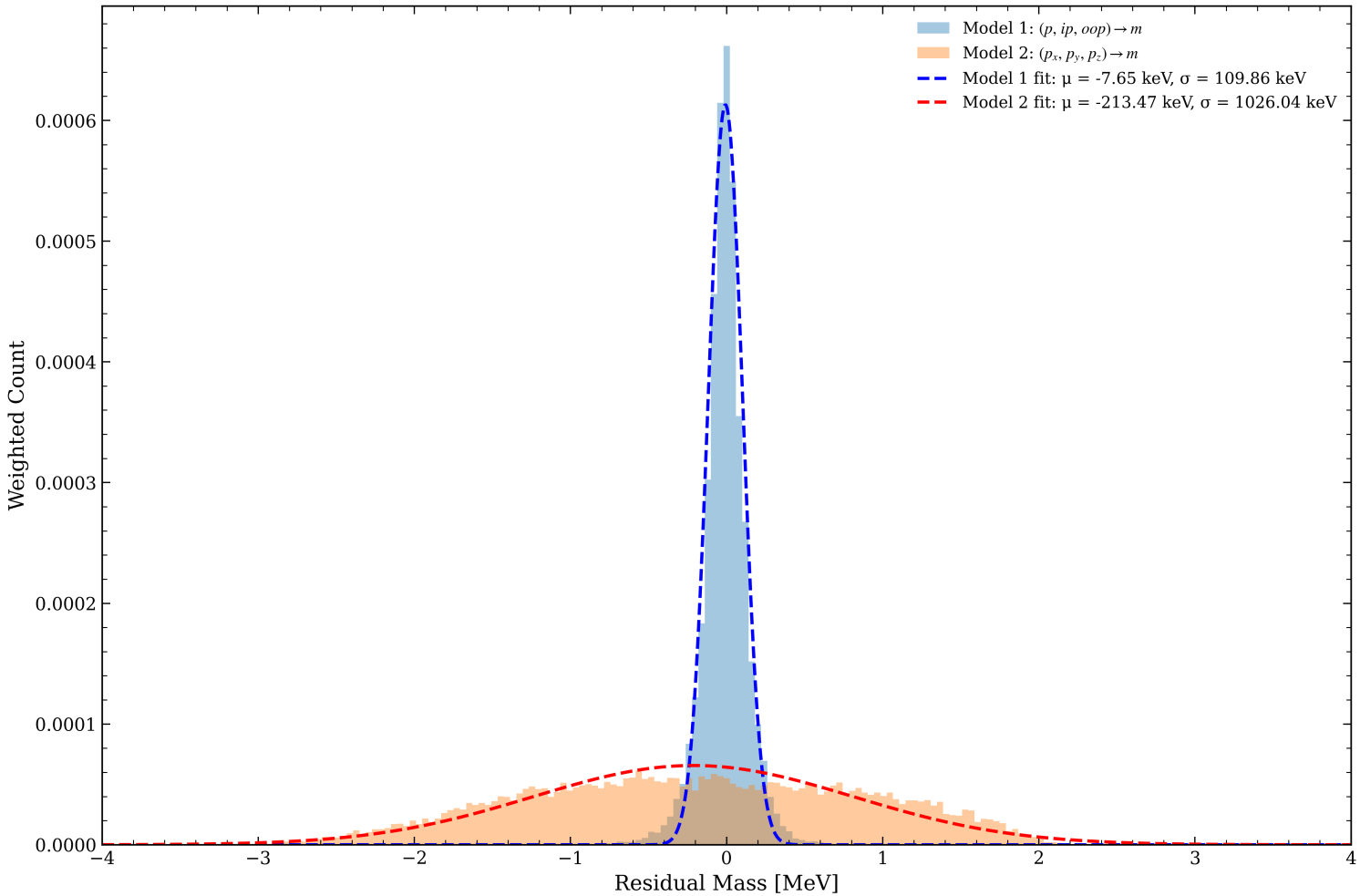
Mass Reconstruction



Kinematic Reconstruction Using Cartesian Components

- Previous reconstruction targeted particle momentum p and angles (ip, oop) at the interaction vertex.
- After interacting with the target, tracing these variables becomes difficult due to scattering and detector effects.
- To improve mass reconstruction, we retrain models to predict Cartesian momentum components: p_x, p_y, p_z.
- These predictions are then used to calculate the invariant mass of the system.

Mass Reconstruction



Event-Level Perturbation Study

- Assess sensitivity of reconstructed invariant mass to small systematic shifts in detector observables.
- For each event, detector-level observables are perturbed individually:

$$z_j \to z_j + \delta$$

- Recompute invariant mass:

$$m_{\text{pert}} = f(\text{perturbed } \mathbf{z})$$

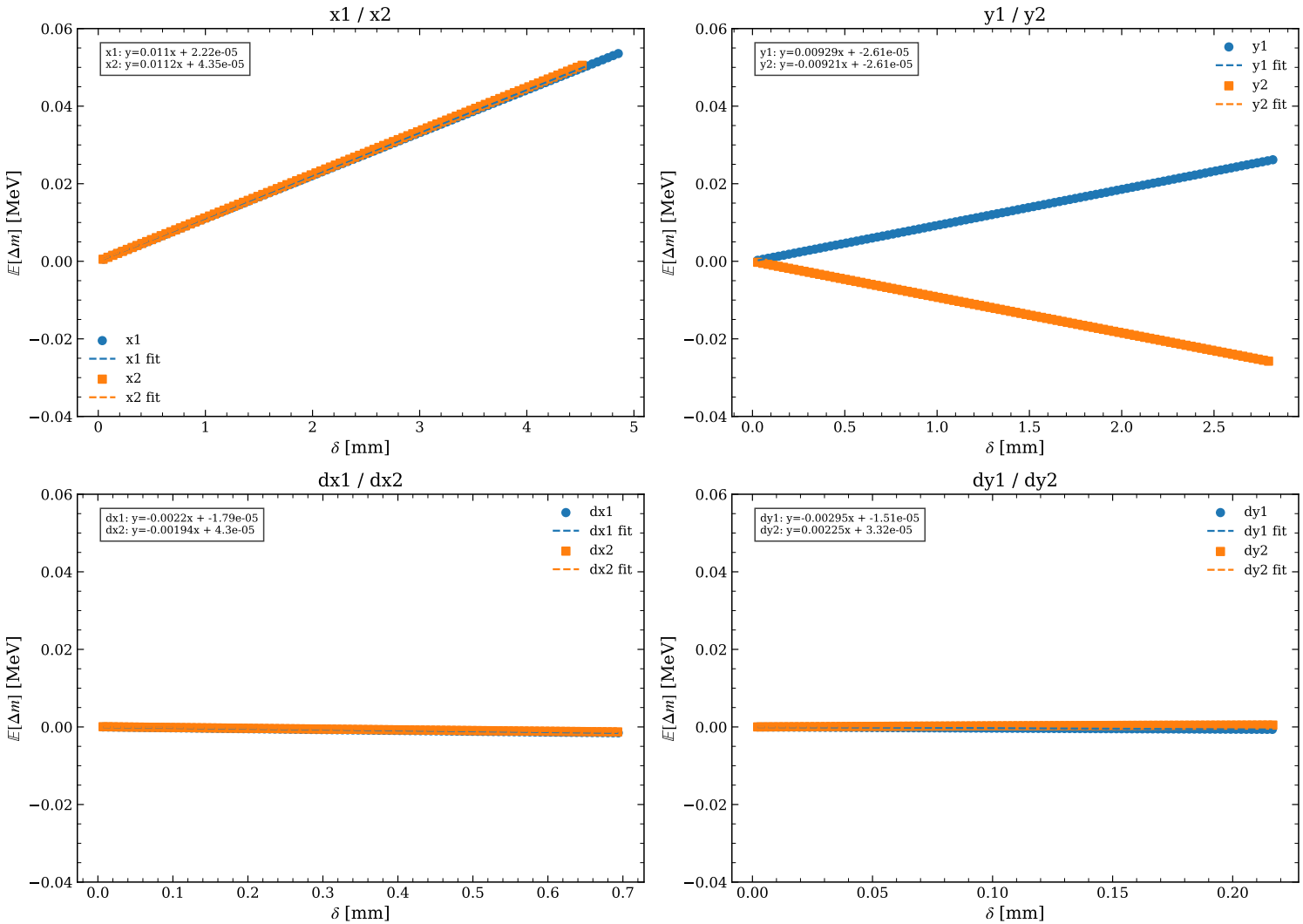
and calculate the mass change:

$$\Delta m = m_{\text{pert}} - m_{\text{base}}$$

- Repeat for each observable and a range of δ values to quantify sensitivity.

Mass Sensitivity to Detector-Level Perturbations

- To evaluate the robustness of our reconstruction, we systematically perturbed each detector observable for all events.
- The resulting change in reconstructed invariant mass, $\Delta m = m_{\text{pert}} - m_{\text{base}}$, quantifies the sensitivity to each observable.
- This allows us to identify which detector measurements have the largest effect on mass reconstruction.
- The next slide shows these results visually for all observables.


x1 / x2
x1: y=0.011x + 2.22e-05
x2: y=0.0112x + 4.35e-05
x1
x1 fit
x2
x2 fit
$\mathbb{E}[\Delta m]$ [MeV]
δ [mm]
y1 / y2
y1: y=0.00929x + -2.61e-05
y2: y=-0.00921x + -2.61e-05
y1
y1 fit
y2
y2 fit
dx1 / dx2
dx1: y=-0.0022x + -1.79e-05
dx2: y=-0.00194x + 4.3e-05
dx1
dx1 fit
dx2
dx2 fit
dy1 / dy2
dy1: y=-0.00295x + -1.51e-05
dy2: y=0.00225x + 3.32e-05
dy1
dy1 fit
dy2
dy2 fit

Feature Sensitivities

Feature	**Sensitivity (MeV/mm)**	**Sensitivity (MeV/σ)**
x_1	0.011034	0.535547
x_2	0.011183	0.505016
y_1	0.009288	0.261797
y_2	-0.009212	-0.257508
dx_1	-0.002197	-0.015246
dx_2	-0.001937	-0.013426
dy_1	-0.002947	-0.006380
dy_2	0.002247	0.004870