TwinSpec: A Data-Driven Digital Twin Framework for Interpretable GIWAXS Analysis

Tajah Trapier
Department of Materials Science and Engineering, North Carolina State University

1. Background and Motivation

Grazing-incidence wide-angle X-ray scattering (GIWAXS) is widely used to study molecular packing, orientation, and texture in thin-film materials. In practice, however, GIWAXS measurements are almost exclusively performed at synchrotron facilities, where access is limited by competitive beamtime proposals, long wait times, travel costs, and unpredictable instrument availability. Even awarded beamtime may be lost due to last-minute maintenance or scheduling changes, reducing opportunities for hands-on learning and iterative experimentation.

As a result, GIWAXS analysis is often performed post-hoc, and the connection between instrument geometry, acquisition parameters, and observed scattering patterns can be difficult to develop intuitively. While simulation and analysis tools exist, they are rarely integrated with interactive instrument-level reasoning or designed to operate on historical or literature-derived datasets.

This work introduces **TwinSpec**, a modular digital-twin framework for GIWAXS that decouples interpretive reasoning and method development from beamtime availability. TwinSpec emphasizes reproducibility, explicit assumptions, and interpretability rather than full physical simulation.

2. System Architecture

TwinSpec is composed of six loosely coupled components with clear separation of concerns:

2.1 TwinSpec Lakehouse

A provenance-first, local-first data lakehouse stores raw literature assets (PDFs, cropped figures), digitized numeric data, and structured metadata. GIWAXS figures are digitized into long-form CSVs (`qz, qxy, intensity`) and standardized into “core” numeric surfaces. Metadata is stored as method, experiment, and characterization JSON files, enabling traceability from every numeric value to its source and declared assumptions.

2.2 Data Digital Twin

The **data digital twin** (`twinspec-data-twin`) converts static lakehouse assets into deterministic, instrument-aware outputs. Given a dataset identifier, it loads digitized GIWAXS core data, bins it onto a fixed `(qz, qxy)` grid, applies robust normalization, and emits a canonical 2D pattern representation along with scalar summaries, warnings, and provenance metadata. The data twin does not perform physical simulation or structure inference; it exposes the structure already present in the data under explicit assumptions.

2.3 Operator Console

The **TwinSpec console** is an operator-style web application that serves as the single source of truth for instrument state. Users adjust geometry and acquisition parameters, select datasets, and observe immediate updates to data-twin outputs. The console presents 2D GIWAXS patterns, derived summaries, warnings, and logs in a layout inspired by laboratory control software.

2.4 Unity Visual Twin

The **Unity visual twin** provides a state-driven, three-dimensional schematic of GIWAXS instrument geometry. Implemented as a kinematic rig with explicit pivots (sample tilt, azimuthal rotation, detector distance and tilt), it deterministically maps geometry state into visual motion. The visual twin intentionally does not render GIWAXS patterns, avoiding conflation between visualization and measurement. Real-time kinematic coupling is implemented at the code level, with final physical bindings under validation.

2.5 Processing-Aware FeatureML

A lightweight **FeatureML** module demonstrates ML readiness by extracting physically motivated descriptors from digitized GIWAXS maps. Integrated intensities are computed within fixed scattering regions (lamellar, π–π, backbone), along with a background proxy. These descriptors form an ML-ready feature table used to generate a two-dimensional PCA embedding and nearest-neighbor relationships.

3. Methodology

GIWAXS figures for P(NDI2OD-T2) thin films under varied annealing conditions were digitized from literature sources. One fully digitized 2D GIWAXS map and a controlled series of 1D linecuts were incorporated into the lakehouse, each linked to experiment-level metadata.

Digitized 2D data were binned onto a fixed grid and normalized via quantile clipping to ensure consistent dynamic range across datasets. Feature extraction was performed directly on the binned grids using fixed regions of interest, avoiding peak fitting or model-dependent assumptions.

The interactive system can be found at: https://www.twinspec.org

4. Results

The TwinSpec console enables interactive manipulation of geometry parameters with deterministic updates to data-twin outputs, accompanied by explicit warnings and provenance. The Unity visual twin provides complementary geometric intuition aligned with the same instrument state.

The FeatureML pipeline produces a non-collapsed PCA embedding, with approximately 89% of the variance captured in the first two components for the test dataset. This indicates that the extracted descriptors preserve processing-dependent structural variation and are suitable for downstream ML tasks such as similarity retrieval.

5. Conclusions and Outlook

TwinSpec demonstrates a reproducible, interpretable approach to GIWAXS digital twinning that emphasizes explicit assumptions and modular design. By combining a provenance-first lakehouse, a data digital twin, an operator console, a state-driven visual twin, and physics-aware ML descriptors, TwinSpec enables interactive reasoning about GIWAXS measurements without requiring immediate access to synchrotron facilities.

Future work will expand dataset coverage, refine Unity physical bindings, and incorporate additional characterization modalities and analysis methods. More advanced learning modules can be layered onto the existing architecture without restructuring the system.

The data infrastructure, visualization components, and ML prototype were developed and implemented during the 2025 ML/AI in Microscopy Hackathon; no pre-existing project code was reused, aside from standard numerical and web libraries where appropriate. The designing of the system architecture and scaffolding and initial building of the lake house begun prior to the hackathon. Code was developed with the assistance of ChatGPT 5.1 and 5.2.

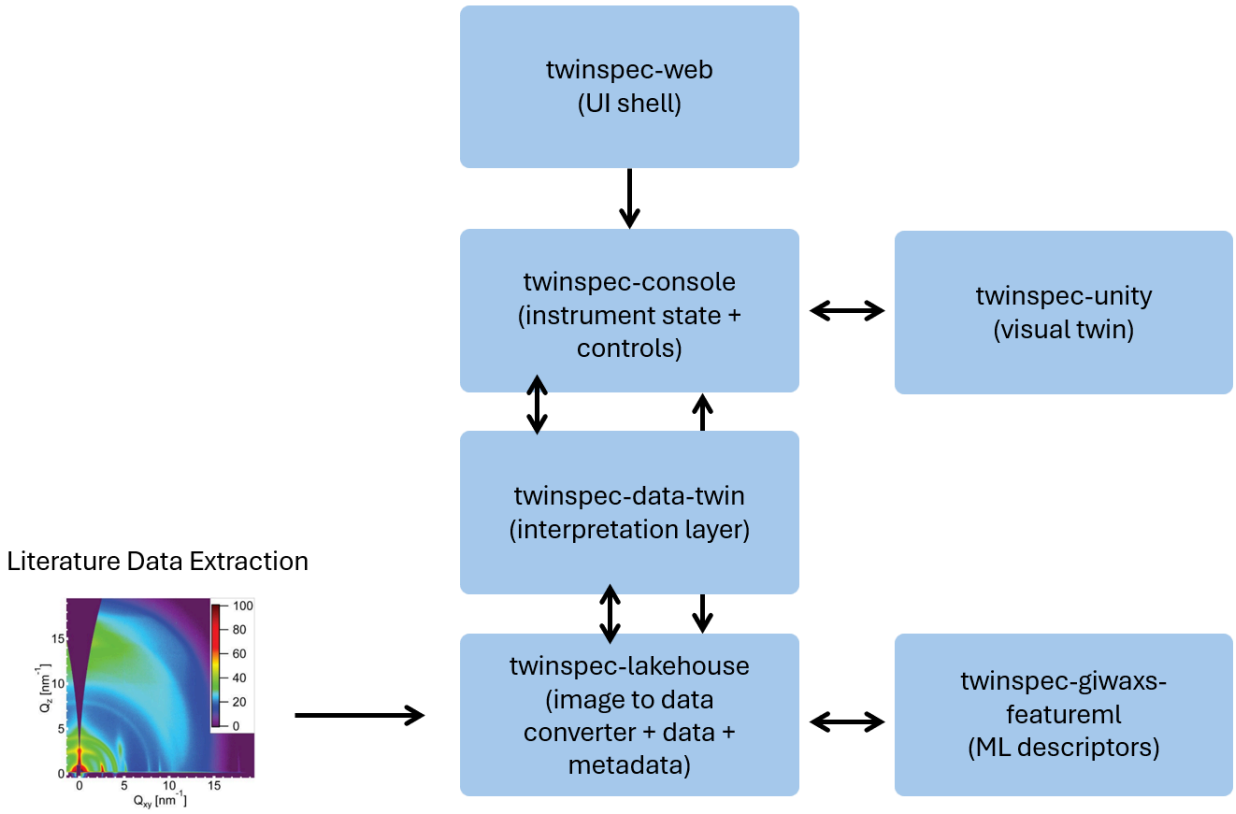

Figure 2: TwinSpec system architecture showing separation between data storage (lakehouse), interpretation (data twin), interaction (console and web UI), visualization (Unity), and downstream ML descriptors.