ID ML Project
Technical Overview

Carla Takahashi

ML ID Project Overview

Background

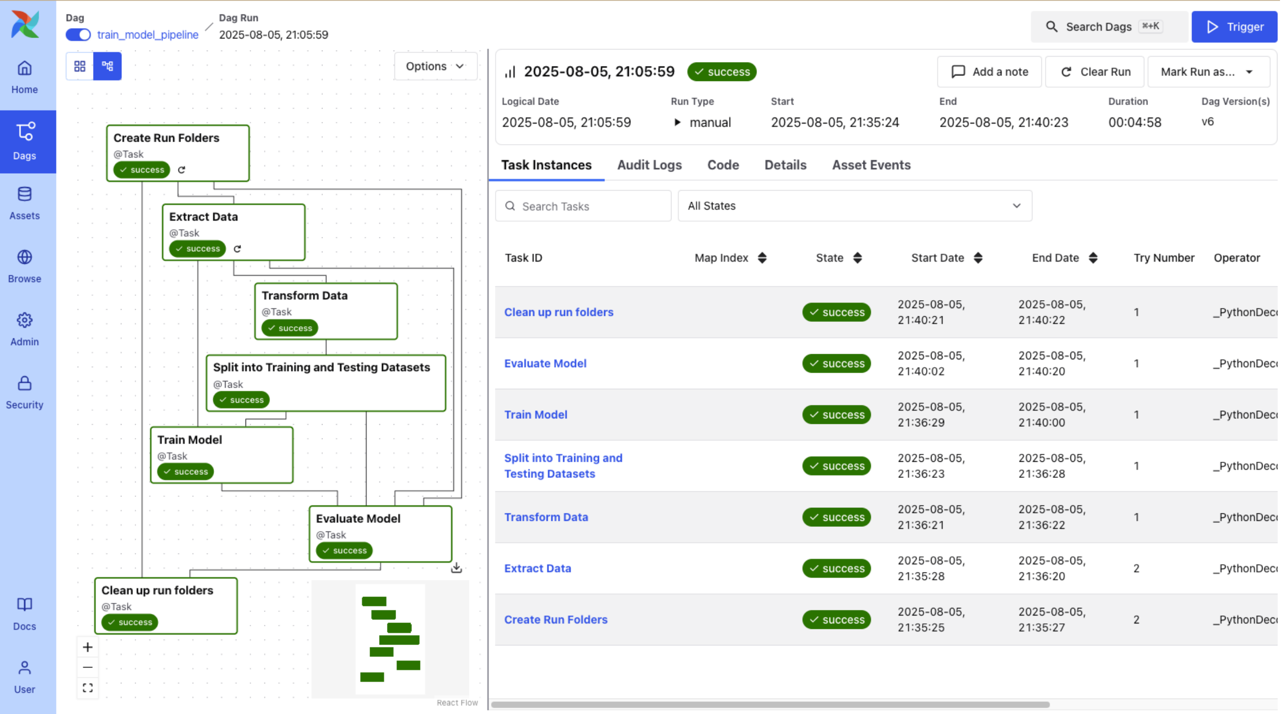
- Joint initiative between Insertion Devices group and Software Group, driven by the ID team as a CPO project.
- First fully fledged project carried out with Machine Learning as main tool.

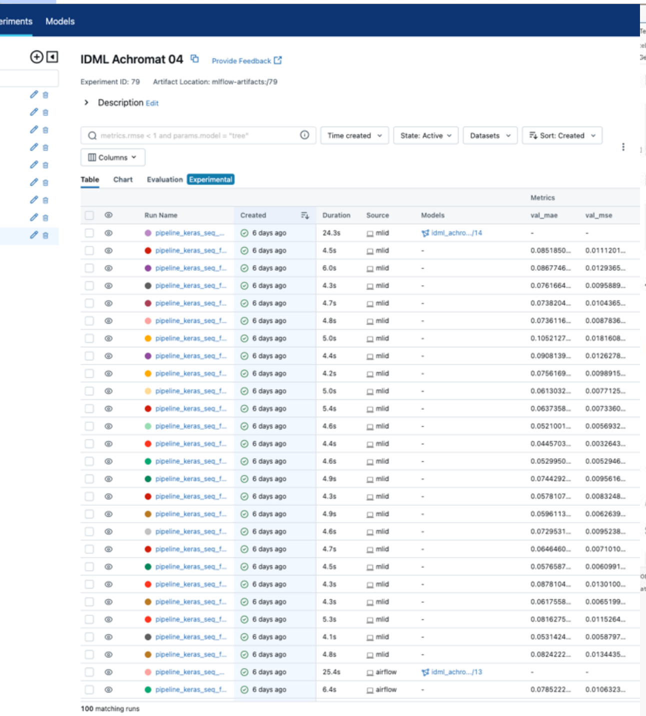
Motivation

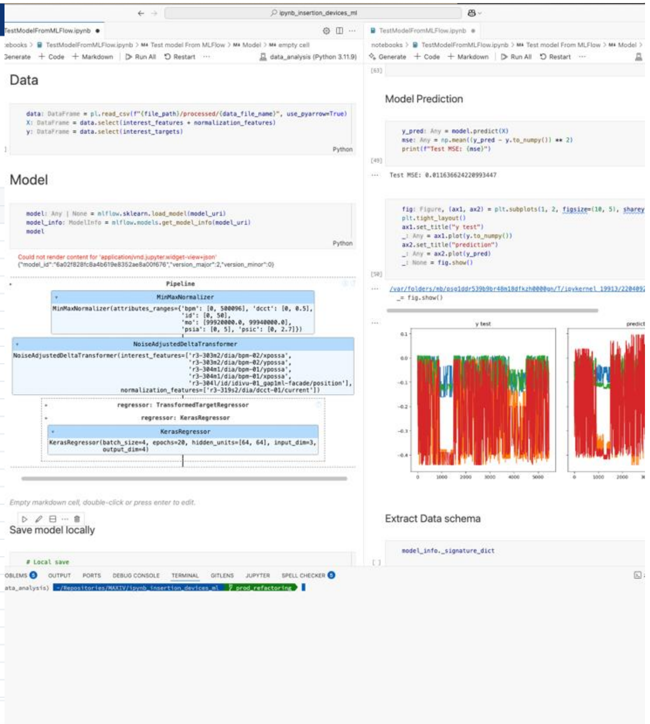
- Beam orbit distortions by Insertion Devices currently corrected with measured feed-forward tables
  - Very time consuming
- Tables might be invalidated over time, or with different machine parameters (e.g. machine upgrades)
- New modes of ID operation can require several hundreds of measurement points e.g. universal mode.
- Prototyping of ML models that effectively replace measured feed-forward tables

Beam Orbit Distortion from Insertion Devices

- **Current Challenge**
  - Insertion Devices (IDs) introduce **orbit distortions** due to residual field integrals.
  - Compensation relies on **feed-forward tables**, which are:
    - Time-consuming to measure
    - Sensitive to changes in accelerator settings
    - Difficult to scale for complex ID configurations (e.g., Universal Mode)
- **Motivation for ML**
  - Automate compensation without extensive measurements
  - Improve maintainability and adaptability of orbit correction
  - Prepare for future operational scenarios and optical changes
- **Approach**
  - Use **archived operational data** (ID gaps, BPMs, correctors)
  - Train **neural network models** to replicate and eventually improve feed-forward behavior
  - Embed ML model into control loop via Tango Device Server

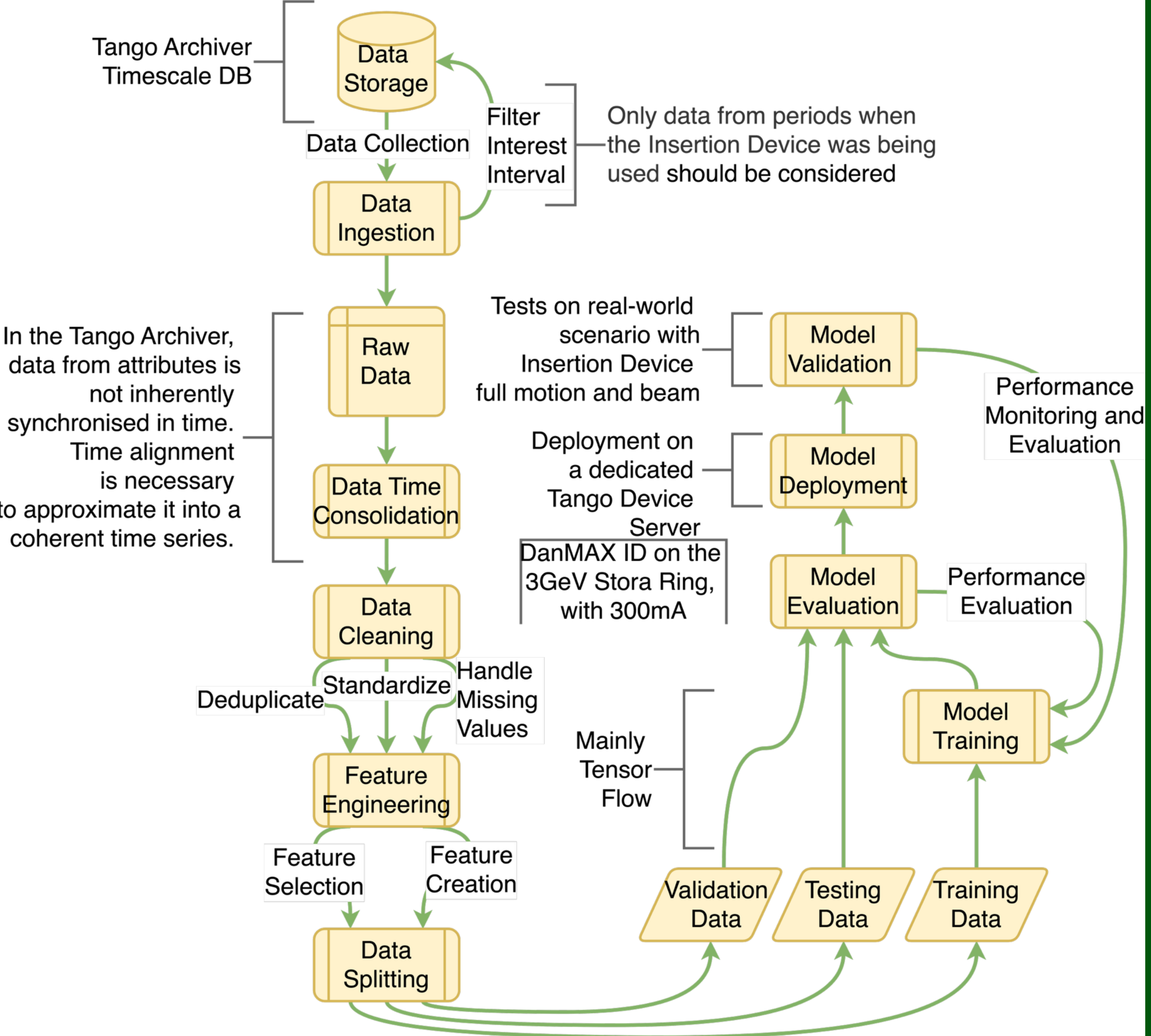






Infrastructure for Scalable and Trackable ML Development

- MLFlow for Experiment Tracking and Model Registry
  - Used to log training runs, parameters, metrics, and model versions
  - Enables reproducibility and collaboration
- Hosted on OKD (OpenShift)
  - Initial deployment for testing and integration
  - Provides containerized environment for scalable ML workflows
- Airflow for ETL and Training Pipelines
  - Currently running in a local Docker image for prototyping
  - Plans to deploy on OKD for production-grade orchestration
  - Supports automated data extraction, preprocessing, training, and model registration

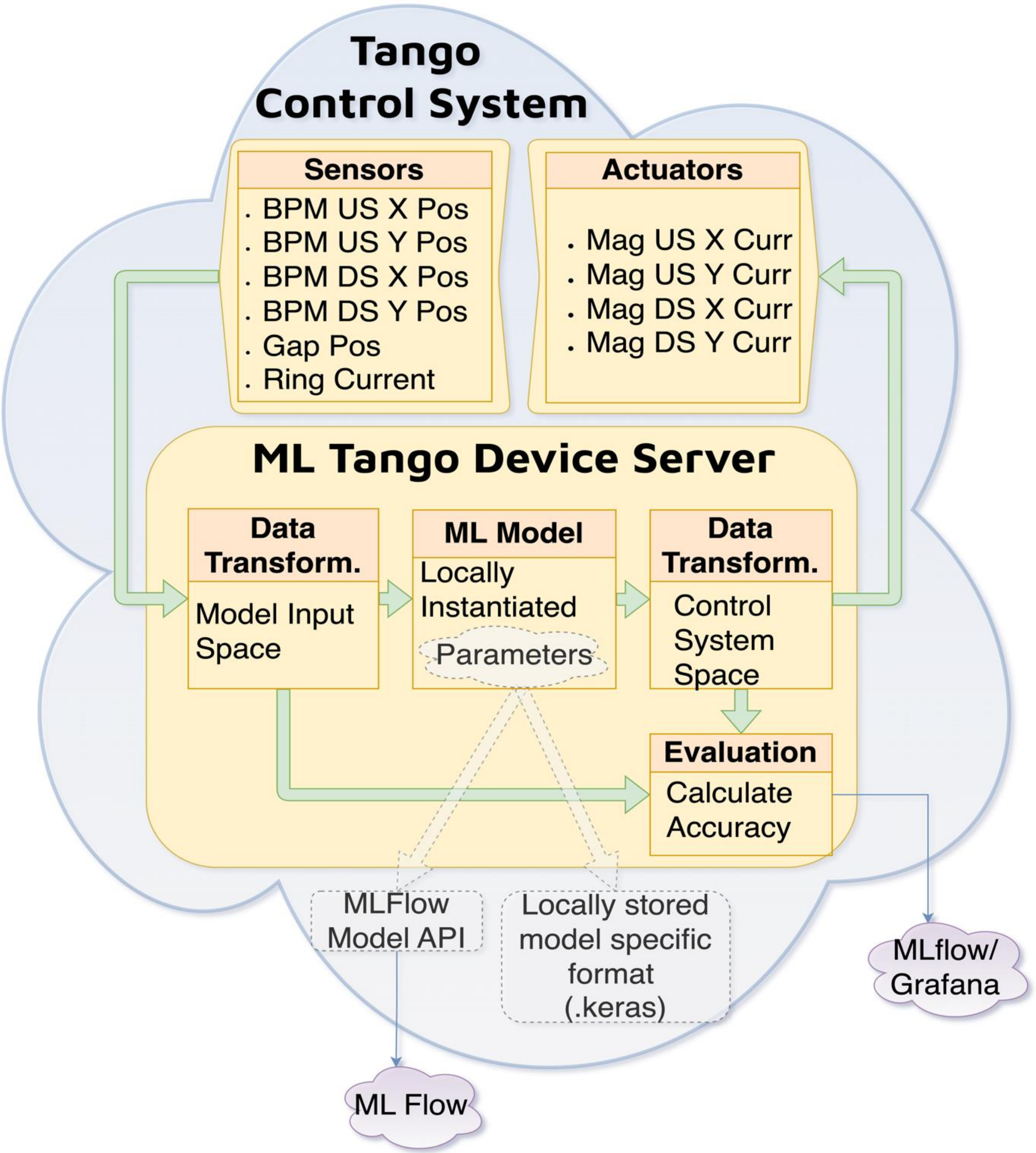


Retrieving Archived Data for ML Training

- **Data Source**
  - Historical data from MAX IV Archiver (TimescaleDB)
  - Includes: BPM readings, corrector currents, ID gaps, ring current
- **Filtering Strategy**
  - Focused on intervals when the ID was actively in use
  - Detected motion and position thresholds in ID gap time series
  - Reduced dataset size for efficient training and retrieval
- **Preprocessing Pipeline**
  - Time consolidation to align asynchronous attribute timestamps
  - De-duplication, completeness filtering, normalization
  - Feature engineering: e.g., BPM readings normalized by ring current to reduce noise
- **Outcome**
  - Enabled non-disruptive training using existing operational data
  - Avoided need for dedicated measurement campaigns

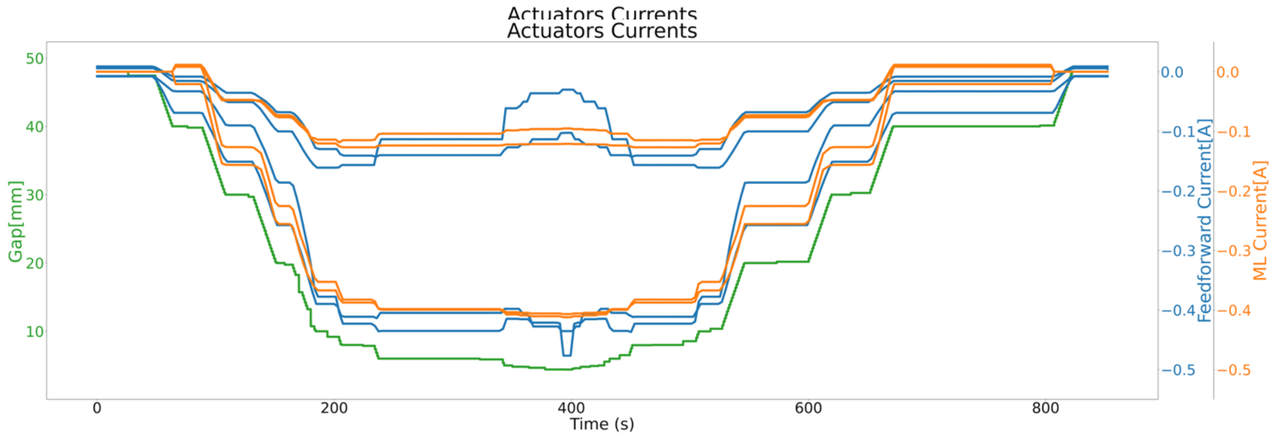
Embedding ML into the Orbit Control Loop

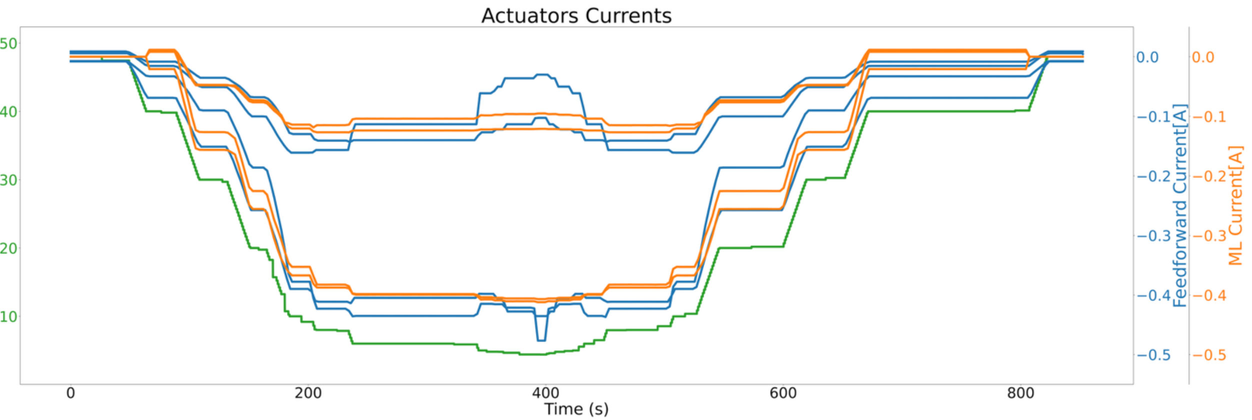
- **Control System Integration**
  - ML model hosted in a Tango Device Server
  - Interfaces with MAX IV control system for real-time inference
- **Inputs & Outputs**
  - Inputs: BPM positions (up/downstream X/Y), ID gap, ring current
  - Outputs: Corrector magnet currents (up/downstream X/Y)
- **Functionality**
  - Synchronizes and transforms incoming data
  - Performs periodic inference
  - Translates predictions back to control system space
  - Logs predictions and computes evaluation metrics(TBD)
- **Model Management**
  - Models loaded from local files or MLFlow registry(Need refactoring)
  - Supports flexible versioning and deployment



Experimental Results

- **Experimental Results**
  - ML model closely approximates feedforward correction across the ID gap range
  - Shows reduced disturbance peaks at smaller gaps → improved local adaptation
  - Some divergence at larger gaps (>30 mm), likely due to limited training data
- **Impact**
  - Demonstrates feasibility of ML-based orbit compensation
  - Indicates potential for adaptive control and reduced noise





Toward Robust and Adaptive ML-Based Orbit Control

- **Live Evaluation Metrics (TBD)**
  - Integrate real-time performance metrics into monitoring tools
  - Track model accuracy, prediction drift, and orbit stability
  - Enable alerts for performance degradation
- **Retraining and Deployment Policies (TBD)**
  - Define triggers for retraining:
    - Data shift detection
    - Degeneration metrics
    - Orbit deviation thresholds
  - Establish rollback mechanisms and fallback models
  - Automate model registration and deployment via MLFlow
- **Goal**
  - Build a **resilient control loop** that adapts to changing conditions
  - Ensure **safe and maintainable integration** of ML in operations