Machine Learning–Based X-Ray Line Profile Analysis for HCP Nanostructure Characterization

Semester 2025/26 II Report

Nguyen Khac Huy
MSc Applied Mathematics / ELTE

May 2026

1 Introduction

This report documents the implementation work carried out during the second semester of 2025/26, building on the theoretical foundations established in the previous semester. Whereas the first-semester report consisted of a literature study with no code produced, the present semester delivered a complete, working software pipeline that extends the Machine Learning–based X-ray Line Profile Analysis (ML-XLPA) framework from cubic (FCC) crystal structures to hexagonal close-packed (HCP) materials.

The project builds on two prior works:

1. Nagy et al. (2022) [1], who introduced the ML-XLPA methodology for FCC structures using XGBoost regression, and
2. Kaszás et al. (2024) [2], who published the `DifFault` library for simulating XRD patterns of faulted cubic crystals.

The full source code is publicly available at: https://github.com/khachuyz/ML_XLPA.

2 Relation to Prior Work

It is important to clarify what is reused from the existing `DifFault` library [2] (https://github.com/balintkaszas/DifFault) and what constitutes new work.

Reused from DifFault (theoretical framework). The fundamental peak-broadening model is inherited from `DifFault`: each diffraction peak is computed as the inverse FFT of the product of three Fourier amplitude contributions,

$$A(L) = A_S(L) \cdot A_d(L) \cdot A_f(L), \tag{1}$$

where A_S accounts for crystallite-size broadening (log-normal spherical distribution, Eq. 4 of [2]), A_d for dislocation broadening via the Wilkens (1970) model [3], and A_f for planar-fault broadening via the Warren (1969) model [4]. The Wilkens restriction function $f(\eta)$ and its numerical evaluation are also based on the same approach.

New contributions in this project. Table 1 summarises the key differences. All items in the right column were implemented from scratch during this semester.

The HCP contrast factor follows Dragomir & Ungár (2002) [5]:

$$C_{hkl} = C_a\big(1 + q_1\,\Gamma + q_2\,\Gamma^2\big), \qquad \Gamma = \frac{(h^2+k^2+hk)\,l^2}{\big[(h^2+k^2+hk) + (3a^2/4c^2)\,l^2\big]^2}. \tag{2}$$

Default parameters correspond to titanium: $a = 0.295\,\text{nm}$, $c = 0.468\,\text{nm}$, $C_a = 0.28$, $q_1 = -0.35$.

Table 1: Comparison of `DifFault` (cubic) and `ML_XLPA` (HCP).

Aspect	**DifFault (cubic)**	**ML_XLPA (HCP, this work)**
Crystal system	FCC / BCC / SC	HCP
d-spacing	$1/d^2 = (h^2+k^2+l^2)/a^2$	$1/d^2 = \frac{4}{3}\frac{h^2+hk+k^2}{a^2} + \frac{l^2}{c^2}$
Reflection rules	FCC/BCC extinctions	$(h{+}2k)$ mod 3=0 & l odd $\rightarrow$ absent
Contrast factors	Cubic: $C_{hkl} = C_{h00}(1-qH)$	HCP: $C_{hkl} = C_a(1+q_1\Gamma+q_2\Gamma^2)$
Fault model	FCC subreflections	HCP basal faults (Warren)
Predicted params	4: m, σ, ρ, β	6: $m, \sigma, \rho, R^*, \alpha, \beta$
ML models	None (forward model only)	XGBoost + 1D-CNN/MLP
Pipeline	Library only	End-to-end: generate $\rightarrow$ train $\rightarrow$ evaluate

3 Software Architecture

The project consists of 10 Python modules organised as a sequential pipeline. Table 2 lists each module and its role.

Table 2: Modules of the ML_XLPA pipeline.

Module	**Purpose**
`hcp_generator.py`	HCP forward model: parameters $\rightarrow$ synthetic XRD pattern
`generate_dataset.py`	Batch generation with parameter sampling
`augmentation.py`	Noise, peak jitter, intensity perturbation
`preprocessor.py`	ARPLS baseline subtraction + max-normalisation
`xgb_model.py`	XGBoost multi-output regressor + Optuna search
`cnn_model.py`	1D-CNN (PyTorch) with MLP fallback (sklearn)
`train_xgb.py`	XGBoost training script with CLI
`train_cnn.py`	CNN/MLP training script with grid search
`evaluate.py`	Metrics, scatter plots, residuals, model comparison
`run_pipeline.py`	End-to-end pipeline runner

In addition, two visualisation scripts were developed:

- `plot_paper_scatter.py` — publication-quality scatter plots (Predicted vs. Real) for all six parameters, styled after Figure 5 of [1].
- `plot_example_diffractograms.py` — synthetic diffractogram overview and parameter sensitivity analysis.

The pipeline operates as follows:

Sampling $\rightarrow$ HCP generation $\rightarrow$ Augmentation $\rightarrow$ Preprocessing $\rightarrow$ ML training $\rightarrow$ Evaluation

4 Parameter Space and Data Generation

Synthetic training data is generated by uniformly sampling six microstructural parameters (Table 3) and computing the corresponding HCP diffractogram via the forward model.

The generator produces patterns on a κ-grid with 4096 Fourier coefficients ($\kappa_{\max} = 14\,\mathrm{nm}^{-1}$), which are then resampled to 1024 points on $\kappa \in [1, 14]\,\mathrm{nm}^{-1}$ after preprocessing. Augmentation adds Gaussian noise (relative level 1%), random peak jitter ($\pm 1.5\%$), and intensity perturbation ($\pm 2\%$) to improve robustness. The HCP generator identifies 21 allowed reflections for Ti-like lattice parameters within the simulated κ-domain.

Table 3: Parameter sampling ranges for synthetic data generation.

Symbol	Description	Unit	Range	Distribution
m	Crystallite size median	nm	8–60	Uniform
σ	Log-normal variance (sqrt)	—	0.01–0.8	Log-uniform
ρ	Dislocation density	nm^{-2}	0.001–0.06	Uniform
R^*	Effective outer cut-off	nm	3–25	Uniform
α	Stacking fault probability	—	0–0.08	Uniform
β	Twin fault probability	—	0–0.05	Uniform

5 Experiments and Results

5.1 Diffractogram Sensitivity Analysis

To understand the information content available to the ML models, we generated diffractograms while sweeping each parameter individually (Figure 1). Key observations:

- σ produces the most distinctive spectral signature: at very low values ($\sigma = 0.05$), clear Fourier oscillations appear from the near-monodisperse size distribution; at high σ these smooth out entirely.
- m and ρ primarily affect the overall decay rate of the pattern envelope.
- R^* causes visible fan-out at low κ, reflecting changes in dislocation strain extent.
- α and β produce the subtlest effects — slight envelope modifications that are harder for ML to distinguish.

These observations explain why σ is the easiest parameter to predict and α, β are the most challenging.

5.2 ML Model Training and Evaluation

We trained an XGBoost multi-output regressor on $N = 100\,000$ synthetic patterns (80/20 train/test split, 300 estimators, max depth 6, learning rate 0.1, early stopping after 30 rounds). The resulting predicted-vs-real scatter plots are shown in Figure 2.

Table 4: Test-set results with $N = 100\,000$ training patterns.

Parameter	R^2	RMSE	Target R^2	Interpretation
m [nm]	0.114	14.18	> 0.88	Weak trend visible, high scatter
σ	0.756	0.100	> 0.90	Best parameter — distinctive signal
ρ [nm^{-2}]	0.084	0.016	> 0.80	Similar envelope effect as m
R^* [nm]	0.085	6.084	> 0.75	Correlated with ρ broadening
α	0.145	0.021	> 0.85	Subtle, beginning to separate
β	0.010	0.014	> 0.85	Nearly unlearned at current setup

The results reveal that σ is by far the most learnable parameter ($R^2 = 0.76$), consistent with the sensitivity analysis: it is the only parameter that produces a qualitatively distinct spectral signature (Fourier oscillations). The remaining parameters primarily affect the overall decay slope of the pattern envelope in similar ways, making them difficult to disentangle from the raw spectrum alone.

Several factors likely contribute to the gap between current and target performance. First, the HCP diffractograms — particularly for small crystallites with high defect densities — produce heavily overlapping peaks that merge into smooth envelopes, reducing the discriminative information available to the model. Second, the current approach feeds 1024 raw intensity values to

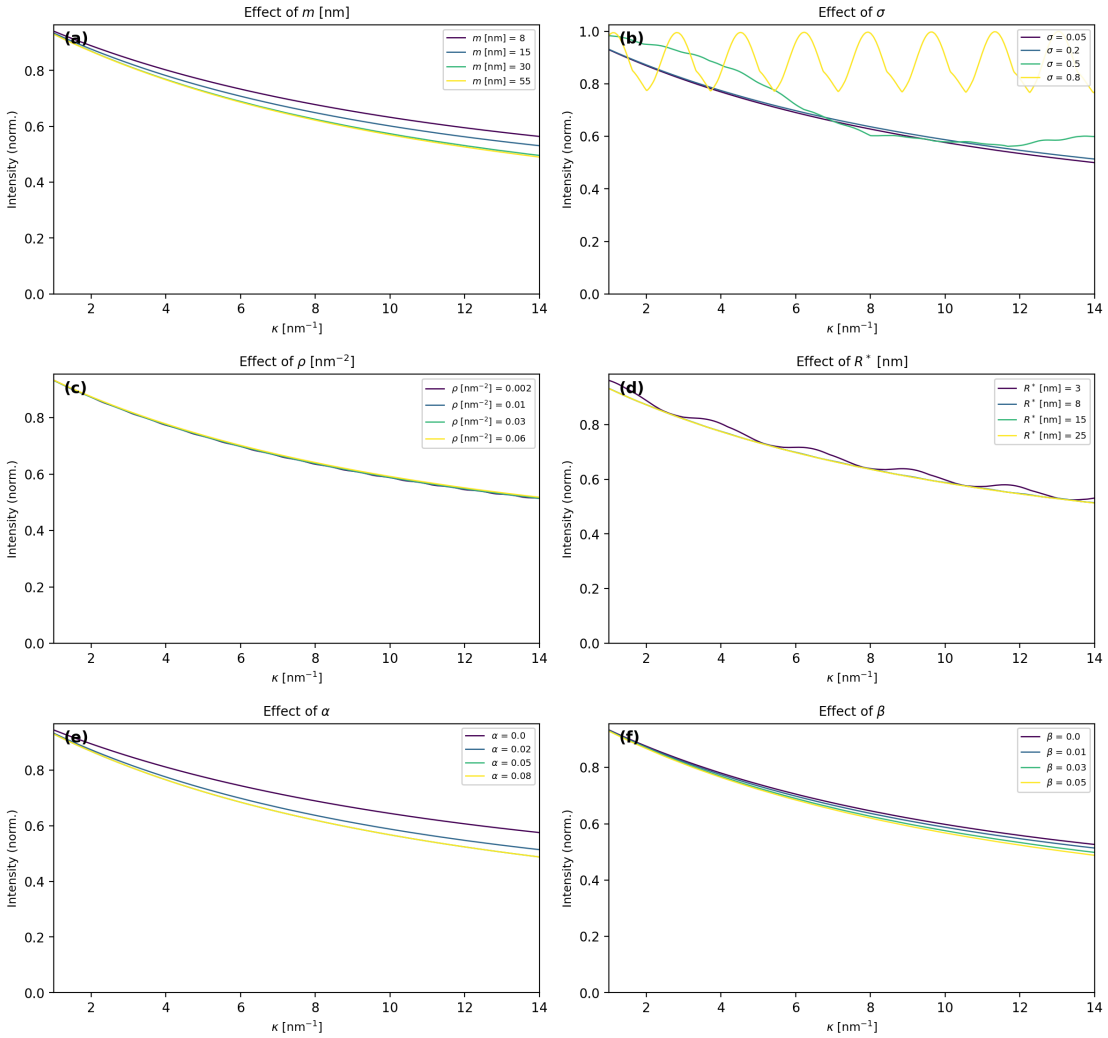

Figure 1: Sensitivity of synthetic HCP diffractograms to each microstructural parameter. In each panel, one parameter is varied while the others are held at baseline values ($m = 25\,\text{nm}$, $\sigma = 0.2$, $\rho = 0.015\,\text{nm}^{-2}$, $R^* = 10\,\text{nm}$, $\alpha = 0.02$, $\beta = 0.01$).

XGBoost, whereas physics-informed features (e.g., individual peak widths, FWHM ratios across reflections, peak area ratios) could provide more targeted input. Third, a 1D-CNN architecture may better capture local peak-shape features that tree-based models miss in a high-dimensional flattened vector. These directions are discussed as next steps in Section 6.

6 Summary and Plan for Next Semester

Summary. This semester delivered a complete ML-XLPA pipeline for HCP crystal structures, progressing from zero implementation to a working GitHub repository with 12 Python modules. The HCP forward model, data generation, augmentation, preprocessing, two ML architectures (XGBoost and 1D-CNN), and publication-quality evaluation tools were all implemented and tested. Training on $N = 100\,000$ synthetic patterns demonstrated that the pipeline is fully functional, with σ achieving $R^2 = 0.76$. The remaining parameters (m, ρ, R^*, α, β) showed limited learnability ($R^2 < 0.15$), which the sensitivity analysis traced to their similar effects on the pattern envelope.

Plan for next semester.

1. **Real data acquisition:** obtain experimental synchrotron XRD patterns from HCP materials (e.g., Ti or Zr alloys processed by severe plastic deformation), with reference microstructural parameters determined by traditional CMWP fitting.
2. **Domain adaptation:** bridge the gap between synthetic training data and real measurements by incorporating realistic instrumental broadening, background profiles, and noise characteristics observed in experimental patterns.
3. **Validation against CMWP:** apply the trained ML-XLPA models to real diffractograms and compare the predicted microstructural parameters (m, σ, ρ, R^*, α, β) with those obtained from traditional CMWP pattern fitting, following the validation approach of [1].
4. **Microstructural mapping:** if combinatorial or spatially resolved HCP samples are available, generate composition–microstructure maps to demonstrate the speed advantage of ML-XLPA over point-by-point CMWP evaluation.
5. **Model refinement:** based on discrepancies observed with real data, refine the forward model, preprocessing, and ML architecture (e.g., physics-informed features, CNN, hyperparameter tuning) to improve prediction accuracy on experimental patterns.on-smooth backgrounds containing secondary-phase contributions.

References

[1] P. Nagy, B. Kaszás, I. Csabai, Z. Hegedűs, J. Michler, L. Pethö, and J. Gubicza, "Machine learning-based characterization of the nanostructure in a combinatorial Co-Cr-Fe-Ni compositionally complex alloy film," *Nanomaterials*, vol. 12, no. 24, p. 4407, 2022.

[2] B. Kaszás, P. Nagy, and J. Gubicza, "DifFault: Simulation of diffraction patterns of faulted crystals," *SoftwareX*, vol. 27, p. 101860, 2024.

[3] M. Wilkens, "The determination of density and distribution of dislocations in deformed single crystals from broadened X-ray diffraction profiles," *Phys. Status Solidi (a)*, vol. 2, pp. 359–370, 1970.

[4] B. E. Warren, *X-Ray Diffraction*. Dover, 1969.

[5] I. C. Dragomir and T. Ungár, "Contrast factors of dislocations in the hexagonal crystal system," *J. Appl. Cryst.*, vol. 35, pp. 556–564, 2002.

[6] J. Gubicza, *X-Ray Line Profile Analysis in Materials Science*. IGI Global, 2014.

[7] S. J. Baek, A. Park, Y. J. Ahn, and J. Choo, "Baseline correction using asymmetrically reweighted penalized least squares smoothing," *Analyst*, vol. 140, no. 1, pp. 250–257, 2015.

[8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD*, pp. 785–794, 2016.

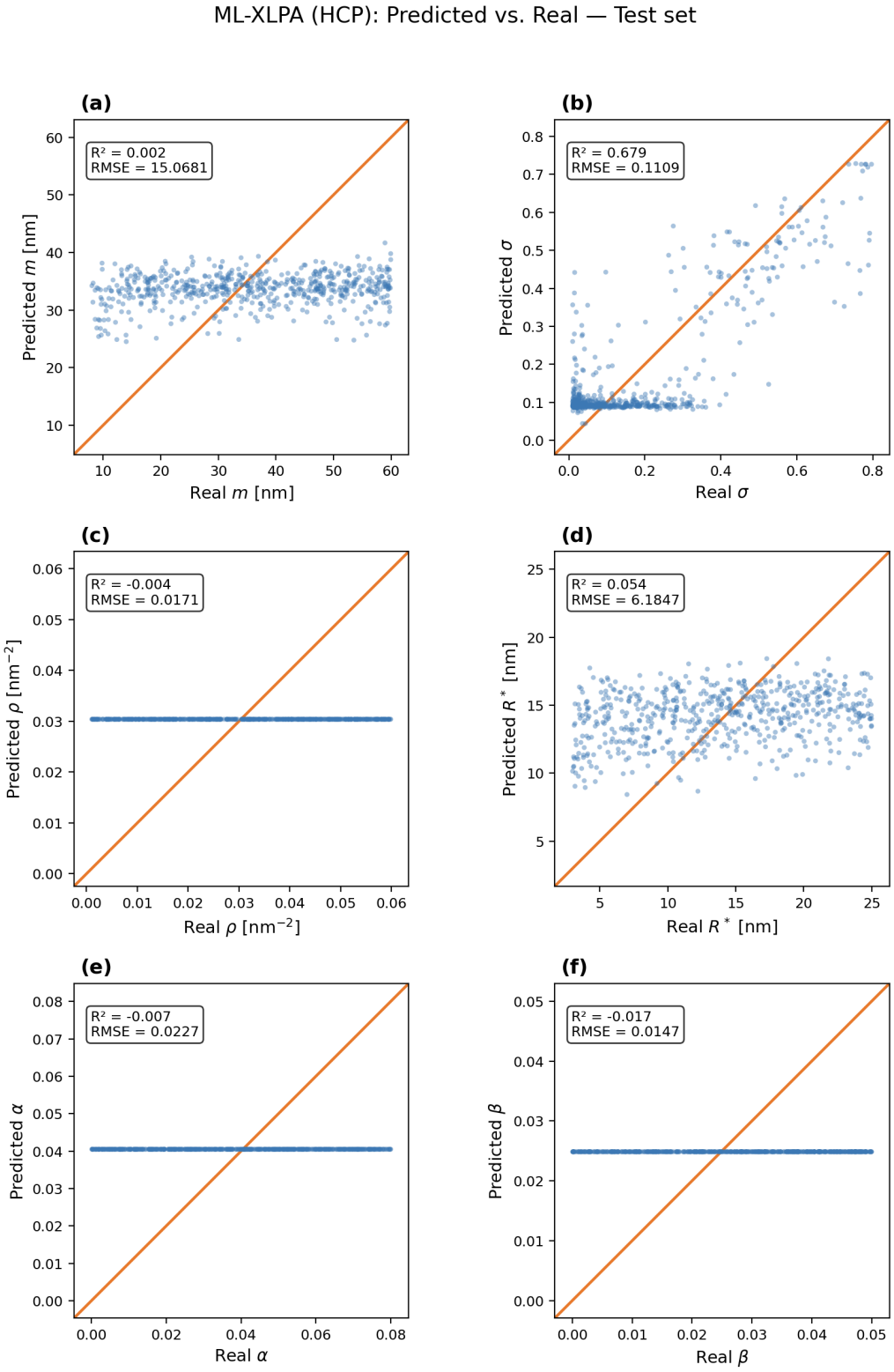

Figure 2: Predicted vs. real values on the test set for all six parameters ($N = 100\,000$ training patterns). The parameter σ achieves the best prediction ($R^2 = 0.76$), while the remaining parameters show limited learnability with the current approach.