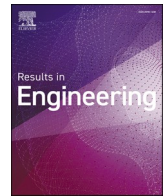




Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering



Research paper

An explainable multitask deep learning framework based on QC multiscale modeling of nanoindentation in thin films

Hasan Ghahremani, Soheil Mohammadi ^{*}

High Performance Computing Laboratory, School of Civil Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran

ARTICLE INFO

Keywords:

Multiscale
Nanoindentation
Quasicontinuum method
Multi-task learning
Nested cross-validation
Deep learning

ABSTRACT

Nanoindentation is a powerful technique for assessing the mechanical properties of materials at the atomic scale, yet its computational modeling remains a challenge due to the complexity of deformation mechanisms. In this study, a deep learning framework is developed to predict the mechanical response of nanoindentation using the data generated via the quasicontinuum (QC) method. The dataset includes simulations of copper thin films with various indenter shapes and sizes, surface roughness levels, and crystallographic orientations.

Three different neural network models are implemented and compared: (1) a multioutput model, which predicts all mechanical parameters simultaneously; (2) a two-network approach, which separates displacement predictions from other mechanical properties to increase accuracy; and (3) a multitask learning (MTL) model, which optimizes shared learning between interdependent variables. The MTL model demonstrates the highest accuracy and generalization capability, effectively capturing the nonlinear relationships between critical loads, load drops, penetration depths, hardness, and absorbed energy.

To validate the proposed framework, two case studies are conducted: (1) a Berkovich indenter on a rough surface with a specific crystallographic orientation and (2) a cylindrical indenter on a smooth surface with a different orientation. The results show strong agreement between the deep learning predictions and the QC simulations, confirming the model robustness. The proposed deep learning approach provides a computationally efficient and accurate tool for predicting the mechanical behavior of materials at the atomic scale, paving the way for further advancements in data-driven nanomechanical analysis.

1. Introduction

Nanoindentation tests are nondestructive techniques used to measure the hardness and mechanical properties of materials at the nanoscale. Its primary advantage lies in its ability to characterize the mechanical properties of materials at the micro- and nanoscale, which is crucial for materials with complex nano or microstructures. Nanoindentation involves processes occurring at multiple scales simultaneously, including atomic-scale interactions, microscopic structural changes, and macroscopic behaviors. This makes it ideal for simulation via multiscale methods [1,2]. Multiscale modeling strategies include sequential approaches, where information flows between scales (such as using molecular dynamics (MD) simulations to inform continuum models), and concurrent approaches, where different scales are solved simultaneously, enabling direct coupling between the atomic and continuum domains [3–6]. Several multiscale methods have been proposed to model material behavior under various conditions [7], such as the

Bridging Domain Method (BDM) [3], Quasicontinuum (QC) [8], Disordered Concurrent Multiscale Method (DCMM) [9], and Variable Node Multiscale Method (VNMM) [10].

Among these, the QC method, introduced by Tadmor et al. [8], offers an efficient balance between atomistic accuracy and computational efficiency. By retaining the atomic resolution only in regions where fine-scale mechanisms (e.g., dislocation nucleation, defect interactions, etc.) dominate, and coarse-graining the remaining domain by a continuum approximation, QC allows for the study of larger systems and longer load steps than MD, while preserving the physical nano responses that may be lost in purely continuum methods. QC has been successfully applied to a range of nanoscale problems, including dislocation nucleation, crack propagation, and surface deformation [11], making it particularly suitable for nanoindentation studies where localized events under the indenter govern the global load–displacement behavior.

Previous studies have primarily examined individual factors influencing nanoindentation behavior, such as surface features, indenter

^{*} Corresponding author.

E-mail address: smoham@ut.ac.ir (S. Mohammadi).

<https://doi.org/10.1016/j.rineng.2025.107729>

Received 13 July 2025; Received in revised form 27 September 2025; Accepted 13 October 2025

Available online 15 October 2025

2590-1230/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

geometry or crystal orientations, often in isolation. For example, Kiely et al. [12] demonstrated that surface defects broaden the onset of plasticity due to localized stress concentrations, while Rodrigo and Iglesias [13] highlighted the role of indenter material and crystal orientation in producing scattering that affects modulus and hardness predictions, with such effects mitigated at higher loads or with softer materials [14]. Multiscale approaches, such as MD-FEM, have confirmed agreement between atomistic and continuum responses, as reported by Wang et al. on Ni thin films [15]. Additional studies addressed the influence of grain size, substrate properties and surface roughness [16], whereas QC-based simulations explored crystal anisotropy, indenter size, and coatings on hardness and deformation [17–19]. Li et al. [17] defined the rectangular indenter size as $4d$, with $d = \frac{l_0}{\sqrt{3}}$, where l_0 ($\approx 4 \text{ \AA}$ for copper) is the lattice constant. The mechanical properties of coatings on substrates were examined using nanoindentation combined with finite element simulations by Akatsu et al. [20]. A new protocol for evaluating the mechanical properties of thin films on substrates using point-sharp nanoindentation was proposed. By employing two weight functions, the method accounts for the indenter tip geometry and provides more accurate measurements of the film's elastic modulus and yield stress compared to conventional approaches. Furthermore, defected geometry was shown to critically affect responses such as stiffness and nanohardness [21–23]. For instance, Zhang et al. [24] demonstrated that triangular surface defects increased hardness more than rectangular ones, while Ghahremani et al. [25] reported that smaller indenters reduced the sensitivity to roughness compared to larger ones, which introduced greater scattering due to their broader contact area. Despite these valuable insights, majority of existing studies focused on isolated single parameters. What remains lacking is a comprehensive investigation that simultaneously considers multiple key factors, such as indenter geometry, surface roughness and crystallographic orientation, and their combined effect on the mechanical response of thin films. The present study aims to address this gap by employing the quasicontinuum method to generate nanoindentation data that captures these interacting effects, followed by developing machine learning models/predictions across diverse conditions.

Although accurate multiscale simulations provide a high degree of precision by capturing atomic details of deformation mechanisms, they are inherently computationally expensive, time-consuming, and require major expertise to implement. While these features make the multiscale methods vital when the goal is to investigate very fine atomic details, machine learning (ML) approaches offer an efficient alternative for rapid generalization from existing datasets, eliminating the need for costly repeated new simulations. Moreover, they are capable of identifying complex nonlinear relationships between multiple input parameters. Machine learning methods increase the feasibility of large-scale parametric studies by providing accurate predictions of average mechanical responses, while significantly reducing the computational cost. They provide a powerful framework for integrating nanoscale insights into broader predictive models of material behavior.

Machine learning (ML) plays a pivotal role in the field of nanoindentation by allowing for analysis and prediction of material behavior with enhanced precision. Xiao et al. [26] developed a hierarchical multiscale method in which molecular dynamics simulations were conducted first to generate a dataset to represent physical phenomena at the nanoscale. The dataset was then used to train a material failure/defect classification model and stress regression models. Park et al. [27,28], introduced a deep learning-based nanoindentation method to simplify the evaluation of mechanical properties of polymers and bulk metallic glass. Nanoindentation simulations based on FE with spherical and Berkovich tips were used to generate a database for model parameters. An inverse analysis method based on artificial neural networks (ANNs) was proposed by Han et al. [29] to determine the free-volume model (FVM) parameters of thin-film metallic glass through nanoindentation. The model was trained based on a database generated by

FEA.

Advanced machine learning algorithms can process complex and noisy experimental data and uncover hidden patterns to predict mechanical properties such as hardness, elastic modulus, and plastic behavior [30]. Jin et al. [31] reported the prediction of mechanical properties of graphene-reinforced metal matrix composites, and Liu et al. [32] adopted four ML models (RF, XGBoost, LightGBM, and SVM) to predict the Vickers hardness of amorphous alloys based on the atomic fraction, structural features and load. The Light Gradient Boosting Machine (LightGBM) model outperformed the other models. A novel micromechanical ML framework was investigated [33] to characterize the material parameters of bulk metallic glasses (BMGs) via nanoindentation simulations with Berkovich and spherical tips. A dataset generated from the Drucker–Prager model coefficients was used to train a nested ML algorithm with a deep neural network as the key estimator. Khoei et al. [34] presented an ML-based atomistic-continuum multiscale framework to model the nonlinear behavior of nanocrystalline structures. Using molecular dynamics simulations, atomistic representative volume element (RVE) data were analyzed to identify key parameters for effective fine-to coarse-scale mechanical property reproduction. An ANN approach optimized the ML process, and the model was validated by predicting RVE responses under various deformations. Recently, increasing attention has been given to the use of interpretable machine learning (IML) techniques, such as SHapley Additive exPlanations (SHAP) [35], to enhance the transparency of ML predictions in nanoindentation. Zaki et al. [36] applied SHAP-based analysis to reveal the dominant role of indenter tip radius on the onset of dislocation pop-in events, thus providing mechanistic insights beyond black-box predictions. Likewise, Trost et al. [37,38], integrated interpretable ML with finite element simulations to estimate nanoindentation tip radii, showing that SHAP values can identify the physical features of load-displacement curves most strongly linked to tip geometry. These studies underline that interpretability not only improves model trustworthiness but also deepens physical understanding, making IML approaches a promising complement to conventional ML frameworks in nanoindentation research.

However, few studies have leveraged QC-generated datasets to train models capable of predicting multiple, interdependent mechanical indicators under varied indentation conditions. Existing works often focused on a single property (e.g., hardness or modulus) or used datasets from FE or MD simulations. Furthermore, when multiple outputs are considered, a standard multi-output regression does not account for the differing degrees of correlation between targets, leading to possible negative transfer effects. To address these gaps, the present work develops a multitask learning (MTL) framework [39] trained on a systematically generated QC dataset spanning 442 combinations of indenter geometry, surface roughness, and crystal orientation. The MTL model is designed to exploit shared feature representations for related targets, while isolating tasks where correlations are weak, thereby improving predictive accuracy for critical loads, load drops, penetration depths, hardness, and dissipated energy. Fig. 1 presents the overall pipeline, integrating QC-based data generation, preprocessing, and deep learning analysis.

The remainder of this paper is structured as follows: Section 2 describes the data generation process, including the quasicontinuum (QC) method for simulation of nanoindentation and the setup for different indenter shapes, surface roughness levels, and crystallographic orientations. Section 3 presents the deep learning framework, detailing the architecture of the neural networks, the optimization process using Optuna, and the nested cross-validation strategy. Section 4 discusses the results of different network configurations, comparing the multioutput, two-network, and multitask learning (MTL) models. Two case studies are analyzed to validate the proposed framework against QC simulations. Finally, Section 5 concludes the study with key findings, implications, and suggestions for future research in data-driven multiscale modeling of nanoindentation.

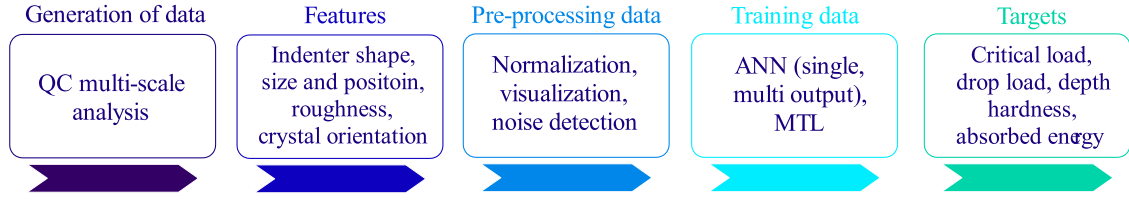


Fig. 1. A data pipeline for QC-based nanoindentation analysis via machine learning.

2. Data generation for nanoindentation

In this study, a 200×100 nm FCC copper thin film, depicted in Fig. 2, is examined. While the general plane stress state in $X-Y$ plane, three-dimensional atomic interaction between neighboring atoms is considered for precise energy calculations. As a result, displacements in the Z direction are induced even under plane stress conditions. Additionally, periodic boundary conditions are imposed along the out-of-plane direction. Previous studies [19,23], have reported that in thin films with protective coating for shallow penetration depths (relative to the coating thickness), the influence of the substrate on the nanoindentation response is negligible. Therefore, the influence of the substrate on the measured hardness and elastic modulus is expected to be negligible. Accordingly, the substrate is not explicitly modeled in the present study due to its negligible effect on the mentioned mechanical properties.

Moreover, since the film thickness (100 nm) is sufficiently large and the depth of dislocation nucleation does not reach the bottom of the model, the lower surface is fixed to ensure mechanical stability and to reduce the computational cost. This boundary condition represents the behavior of a film supported by a rigid substrate rather than a free-standing film, while the use of periodic boundaries in the out-of-plane directions preserves bulk-like behavior in the thickness direction.

The QC method is adopted to generate data from the nanoindentation models [8,11]. It is efficient for studying systems where fine atomic-level details are necessary in some regions, whereas others can be modeled more coarsely with continuum approximations. This method reduces the computational cost while maintaining accuracy in regions of interest [7].

2.1. Indenter

The indenter, modeled as a rigid body object, penetrates the copper surface at a rate of $0.1 \text{ \AA}$ per step. Available QC simulations employed much smaller radii to capture atomic-scale mechanisms and to reduce the computational costs; for instance, Knap and Ortiz [36] used spherical indenters of 7–70 nm, and Tadmor et al. [8] applied a cylindrical indenter with a radius of $11 \text{ \AA}$. In this study, cylindrical and rectangular indenters are modeled with rounded and flat tips, respectively, while the Berkovich indenter is modeled as ideally sharp. This approach allowed a direct comparison of the influence of rounded, flat, and sharp tip geometries. It should be noted that one of the objectives of the present study is to compare the behavior of the thin copper layer under different indenter geometries. Therefore, a roughly similar range of indenter sizes are chosen for all shapes of indenters. For the cylindrical and rectangular indenters, eight distinct sizes are considered (10, 20, 40, 60, 80, 100, 120, and $140 \text{ \AA}$), corresponding to the tip radius for cylindrical and flat width for rectangular indenters. In contrast, the sharp Berkovich indenter is modeled with angles of 70.3° and 65.3° , as illustrated in Fig. 2. In atomic force microscopy (AFM)-based nanoindentation experiments, the tip radius of indenters is generally below 20 nanometers [40,41], enabling detailed examination of surface mechanical properties. For this study, the indenter size is chosen to align with the nano-scale characteristics of material, ensuring accurate measurement of hardness and elastic modulus.

2.2. Roughness

Surface roughness is a key factor influencing mechanical responses at the micro- and nanoscale. In nanoindentation tests, it affects the initial contact between the indenter and the material surface. This can lead to

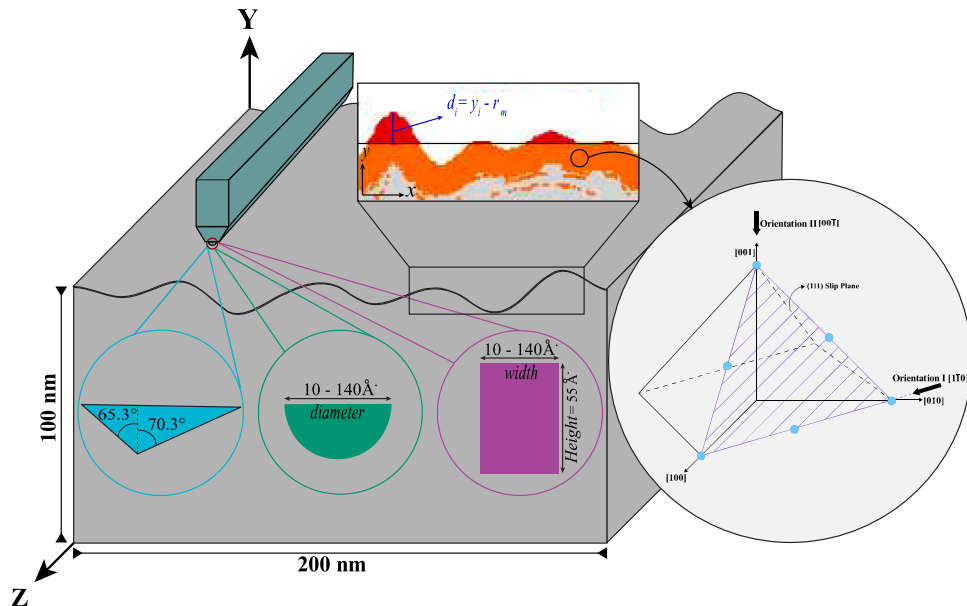


Fig. 2. Nanoindentation model.

variations in measured properties such as hardness and stiffness. Ignoring surface topography may result in inaccurate or non-reproducible results. Therefore, considering surface roughness is essential for reliable material characterization at small scales. The arithmetic average roughness (R_a), as standardized in ISO 4287 for geometric product specifications [42], is frequently used to quantify surface roughness:

$$R_a = \frac{1}{n} \sum_{i=1}^n |d_i| \quad (1)$$

where n represents the total number of points describing the surface, and d_i denotes the deviation of each nodal coordinate y_i from the mean surface coordinate R_m , as illustrated in Fig. 2. The values of y_i are determined using the function introduced by Berke and Massart [43]:

$$y_i(x) = \sum_{i=1}^N \frac{A_i}{2} \sin\left(\frac{2\pi}{\alpha_i} x + \theta_i\right) \quad (2)$$

α_i denotes the maximum scaling factor that adjusts the frequency of the sine wave, θ_i represents the horizontal phase shift along the X-axis, and A_i corresponds to the amplitude of the sine component in the Fourier series. Parameters θ_i , A_i , and α_i are randomly generated in this study.

2.3. Crystallography

The nanoindentation process is significantly affected by crystal orientation. In the present study, two different orientations are examined. In Orientation I, the indenter is applied along the $[1\bar{1}0]$ direction, with the coordinate system defined such that the X-Y axes align with

$[111]$ and $[\bar{1}10]$ directions, respectively. In Orientation II, the indenter is applied along the $[00\bar{1}]$ direction, with the X-Y coordinate axes aligned with $[110]$ and $[001]$ directions, respectively, as depicted in Fig. 2.

2.4. Hardness

Hardness represents a material resistance to penetration by a hard nano-indenter. This key mechanical property is typically quantified by analyzing the relationship between the applied load and the depth of indentation. The Oliver-Pharr (O-P) method, widely recognized for its accuracy, is one of the most commonly used techniques for calculating the hardness from nanoindentation data [44]:

$$H = \frac{P_{\max}}{A} \quad (3)$$

where $P_{\max}$ and A are the peak load and the projected contact area, respectively.

The load-penetration responses of the Berkovich setup with a surface roughness of $13 \text{ \AA}$ for crystallographic directions I and II are depicted in Fig. 3. The "onset of plasticity" is the point at which the first dislocation nucleation occurs, and the load at which it occurs is called the critical load (P_c) [45]. Dislocations are crystallographic line defects that disturb the otherwise perfect arrangement of atoms in a lattice. As illustrated in Fig. 3, the atomic rows are well aligned on both sides, but in the central region, one row of atoms is interrupted, producing a discontinuity in the lattice. This local misalignment forces some atoms to be displaced upward and others downward along the out-of-plane (Z) direction. Such a distortion is a characteristic signature of a dislocation. By analyzing the out-of-plane displacement (U_z), expressed in Angstrom ($\text{\AA}$), these vertical shifts can be clearly detected, which allows us to identify the

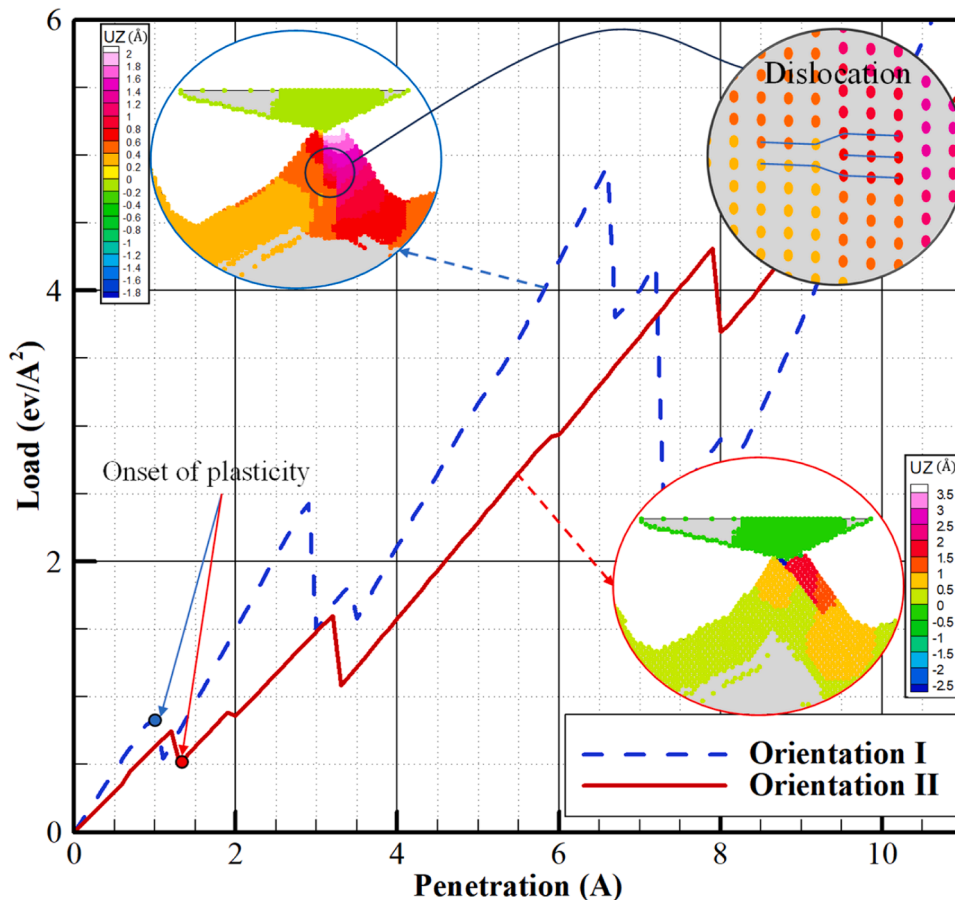


Fig. 3. Load-penetration responses for the Berkovich indenter and $R_a = 13 \text{ \AA}$.

nucleation and evolution of dislocations during the nanoindentation process. The area under the load-penetration response up to the moment of the first dislocation represents the energy absorbed by the material before the occurrence of plastic deformation. This energy, which is mainly stored in the form of elastic deformation in the crystalline structure, leads to permanent changes and crystal slip. The amount of this energy is calculated from the integral of the force with respect to the penetration depth up to the point of the first dislocation and is a measure of the resistance of the material to the onset of irreversible deformation [46]. After the first dislocation occurs in the load-penetration response, a sudden drop in load is usually observed, indicating the release of energy stored in the crystalline structure. This drop in load occurs because of the local failure of the crystalline structure and the onset of slip or movement of dislocations, which corresponds to the phenomenon commonly referred to as a “pop-in” in nanoindentation experiments [47, 48]. The pop-in refers to a sudden change in the load-penetration curve, typically manifested as an abrupt burst of penetration during loading. This phenomenon is generally interpreted as an indication of the initiation of plastic deformation in the material and is often associated with the nucleation and initial motion of dislocations beneath the indenter tip. In this case, plastic deformation dominates, and the material cannot withstand the applied load fully elastically. The amount of load drop depends on factors such as the crystal orientation, grain size, and material type, and indicates the degree of stability of the material against structural changes [49].

In the present study, 13 surface roughnesses between 2 and 13 Å and cylindrical and rectangular indenters with 8 widths/diameters between 10 and 140 Å, and a Berkovich indenter are investigated with two different crystal orientations, I and II. In total, 442 setups are generated and analyzed. In the present simulations, the indentation depth never exceeds 5 % of the film thickness, which is well below the conventional 10 % threshold suggested in the literature [50].

3. Deep learning

Deep learning (DL) is a branch of artificial intelligence techniques in which systems and models gain the ability to learn from data without explicit governing physical formulation [51]. An artificial neural network (ANN) is a machine learning model inspired by the structure of

the human brain and consists of neurons (processing units) organized in layers. Each neuron receives inputs, processes them via weights and biases, and produces an output [52]. The ANN consists of three main parts: the input layer, which is responsible for receiving the initial data; the hidden layers, which perform complex processing on the data; and the output layer, which produces the final result. These models are used because of their high predictive power for complex and nonlinear problems, their adaptability to solve various problems such as classification and regression, and their ability to learn important features from the data. Unlike traditional algorithms, neural networks may automatically identify hidden patterns in the data and provide accurate results [53,54].

This study presents the implementation and optimization of an ANN for predicting multiple continuous variables via the PyTorch library [55]. The process includes data preprocessing, defining the network architecture, and optimizing parameters with Optuna [56], and evaluating the model via nested cross-validation. The results demonstrate that the use of appropriate optimization and validation techniques can significantly improve the model performance.

3.1. Defining the models

3.1.1. Multioutput model

Fig. 4 shows the general architecture of the artificial neural network. The input layers, which are the indenter shape, indenter size, roughness, crystallography orientation, and indenter position (x_1, x_2, x_3, x_4, x_5), are passed through weights (green lines) to the neurons in the hidden layer. At each neuron, a weighted sum of inputs is computed, and a bias (blue lines) is added. This sum is then passed through an activation function to introduce nonlinearity to the model. The output layers are the critical load (P_{cr1}, P_{cr2}), penetration depth (D_1, D_2), load drop (P_{dr1}, P_{dr2}) in the first and second dislocations, hardness (H), and absorbed energy (U) ($y_1, y_2, y_3, \dots, y_8$).

3.1.2. Multitask learning model

Fig. 5 illustrates the architecture for multitask learning. MTL is a technique in machine learning where multiple related tasks are trained simultaneously via a shared model. The key idea is to leverage shared information across tasks to improve generalizability and performance.

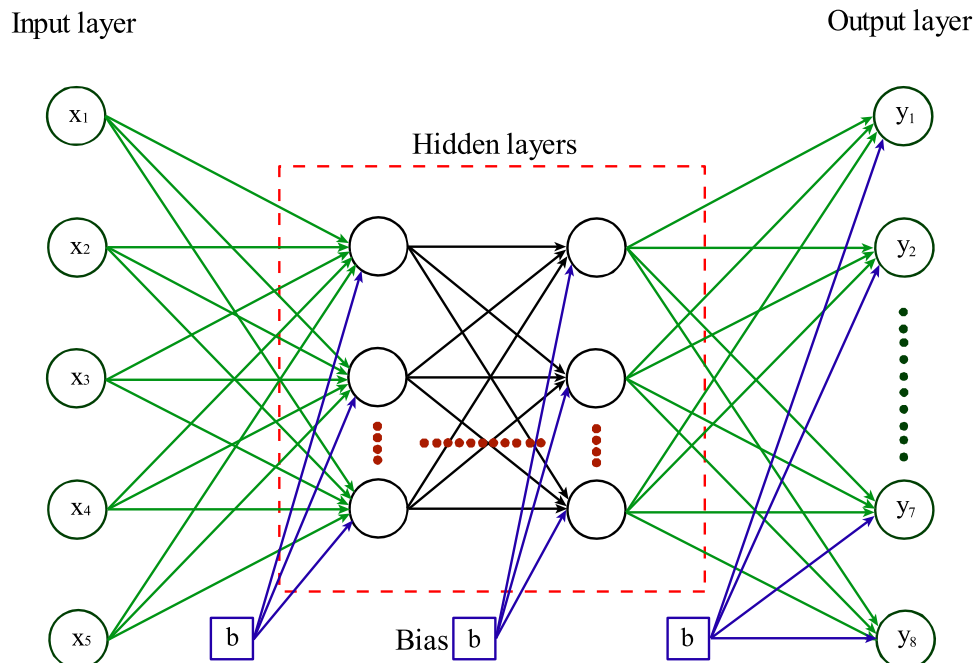


Fig. 4. Architecture of the ANN model.

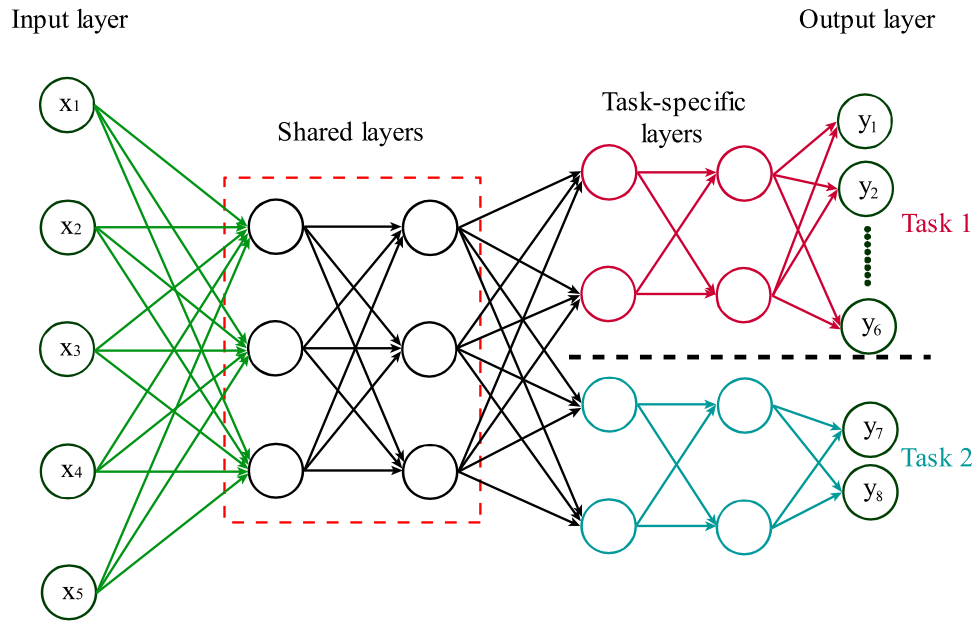


Fig. 5. Architecture of the MTL model.

Unlike single-task learning, where models are trained separately for each task, MTL promotes the sharing of representations, enabling tasks to benefit from one another [57].

The middle section of Fig. 5, enclosed in the red box, represents the shared layer. This layer extracts general features common to all tasks. Sharing this representation allows the model to transfer knowledge across tasks, improving the overall performance. On the right, task-specific layers are tailored to the individual needs of each task. Task 1 has six output nodes and uses a unique processing path from the shared layers. Task 2, with two output nodes, follows a distinct processing path from the shared layers.

3.2. Data preprocessing

The data are divided into training and test sets, which allows for training the model on the training data and independent evaluation of its performance on the test data. Ninety percent of the data is separated for training, and 10 percent is reserved for testing. The input features were normalized using the StandardScaler from scikit-learn [58], which would transform each feature to have zero mean and unit variance. To avoid any potential data leakage, the dataset was first split into training and test sets prior to any preprocessing. The normalization procedure was implemented within a Pipeline object, ensuring that the scaling parameters (mean and standard deviation) were computed exclusively from the training set during the fitting stage and subsequently applied to the test set during transformation. This approach guarantees that no information from the test set influenced the model training process.

3.3. Model definition and parameter optimization

The neural network architecture and its parameters are optimized via Optuna. The following steps are performed:

3.3.1. Defining the network architecture

Hyperparameter tuning is the process of optimizing the settings of a machine learning model to improve its performance. Hyperparameters are external configurations that must be set manually or via automated processes. Examples include the learning rate, number of layers, number of neurons, activation functions, dropout rates, optimizers, and regularization parameters [59,60]. Hyperparameter tuning is performed via Optuna, a powerful optimization framework [61].

3.3.2. Defining hyperparameters

Number of layers, number of neurons in each layer, activation functions, and dropout rates are optimized via the Optuna library to find the best network architecture from the following selections for the present study:

- The number of layers is between 3 and 7.
- The number of neurons in each layer is between 6 and 200.
- The activation function of each layer is selected from ReLU, sigmoid, and tanh.
- The optimizer of each layer is chosen from Adam, SGD, and AdamW.
- The learning rate of optimization is between $1e-6$ and $1e-2$.
- The weight decay is chosen from $1e-5$ to $1e-2$.
- The amount of dropout for each layer is selected between 0.1 and 0.3.

3.3.3. Parameter optimization with optuna

In ML model optimization, the main goal is to minimize the loss function, which is achieved by updating the weights and biases in the neural network. This update is based on the gradient of the cost function with respect to the parameters, which is usually performed via gradient descent methods. In this process, the weights are updated in small steps in the negative direction of the gradient to minimize the cost function. The simplest optimizer is the gradient descent, in which the weights are updated in fixed steps in the negative direction of the gradient at each step. The stochastic gradient descent (SGD) [62] uses a small sample or group of data at each step instead of using the entire dataset, which speeds up the calculations. Adaptive moment estimation (Adam) [63] is a combination of momentum and RMSProp, which adjusts the learning rate adaptively and is usually used for deep neural networks. RMSProp uses the mean square gradient to adjust the learning rate dynamically [64].

In the neural network training process, the backpropagation algorithm is first used to calculate the gradients of the cost function with respect to the weights and biases. These gradients are propagated from the output layers to the input layers to obtain the necessary changes to update the parameters. Then, when an optimizer such as Adam or SGD is used, the weights and biases are gradually adjusted to minimize the error:

$$w_{t+1} = w_t - \eta \nabla L(w_t) = w_t - \eta \frac{\partial L}{\partial w_t} \quad (4)$$

$$b_{t+1} = b_t - \eta \nabla L(b_t) = b_t - \eta \frac{\partial L}{\partial b_t} \quad (5)$$

where w_{t+1} and b_{t+1} are the weights and biases at step $t + 1$, respectively, and η and ∇L are the learning rate and the gradient of the loss function, respectively [65]. The learning rate is a hyperparameter that determines the rate at which the weights change at each update step. Choosing the right learning rate for any model is crucial [66]. If the learning rate is too high, the model may not be able to achieve correct optimization; if it is too low, the learning process is slow, and the optimization may be incomplete.

Regularization or weight decay technique is an effective way to reduce overfitting in optimization algorithms. It encourages the model to adjust weights in a way that prevents them from growing too large, thereby simplifying the model. Weight decay is chosen as a hyperparameter for the optimizer. In weight decay, a small value called the regularization parameter (λ) is defined and added to the loss function:

$$L = \text{Loss} + \lambda \sum_i w_i^2 \quad (6)$$

3.4. Model training and performance evaluation

The model is trained by a nested cross-validation approach, and its performance is evaluated via the R^2 score. The following steps are performed.

3.4.1. Nested cross-validation

Nested cross-validation is an extension of the k-fold cross-validation technique. It is used to estimate the performance of a machine learning model while simultaneously selecting the best hyperparameters. In a standard k-fold cross-validation, the training and testing sets are fixed, and the model is trained and evaluated on each fold. This method helps in minimizing the effect of data selection and improves the assessment of the generality of ML model [67].

In the nested cross-validation, however, the training process involves two levels of cross-validation:

Inner cross-validation is used to tune the model hyperparameters. The inner loop is used for model selection by training the model on the training data in each fold and validating it on the validation data within the training set.

Outer cross-validation is used to evaluate the generalization error of the model. Its folds are used to evaluate how well the model generalizes to unseen data for final testing [68].

In this study, 10 % of the original data is allocated for testing, and 90 % is considered for the main network with the help of nested cross-validation. Nested cross-validation is implemented via two layers of k-

fold cross-validation. The outer cross-validation (with $n_splits = 3$) is used to estimate the model performance on unseen data, whereas the inner cross-validation (with $n_splits = 2$) is responsible for hyperparameter tuning within each fold of the outer loop. The outer loop generates training and testing splits, and the inner loop performs the model selection process without leaking information from the outer test set, as depicted in Fig. 6. The results from the outer folds are then aggregated to provide a final performance estimate, which ensures reliable evaluation by preventing overfitting.

3.4.2. Early stopping

Early stopping is used to allow the model to halt training, once it has reached its optimal performance, to prevent overfitting. If the model performance on the validation set stops improving after several epochs, training is terminated.

3.4.3. Mini-batch

Using mini-batch gradient descent can help reduce overfitting. Accordingly, instead of using all the data or just one sample per step, a small subset of data is used, which makes the model less sensitive to specific data points. As a result, it helps to prevent the learning of random or anomalous features in the training data. With mini-batches, there are more fluctuations in the learning process, which helps the model to avoid getting stuck in local minima and allows it to move towards a global optimum [69].

3.4.4. Model performance evaluation

The model performance is evaluated by the R^2 score on the validation and test sets. The R^2 score is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variable(s) in a regression model. In simpler terms, it indicates how well the model fits the observed data.

4. Results

4.1. Multi-output model with all targets

The first multi-output model in Table 1 is trained to predict all the targets together: P_{cr1} , P_{cr2} , P_{dr1} , P_{dr2} , D_1 , D_2 , U and H . It consists of three hidden layers with 75, 109, and 81 neurons, and it uses the tanh activation function. The Adam optimizer is used with a learning rate of 0.00230 and a weight decay of 0.0035. This model achieves a coefficient of determination (R^2) of 0.936.

Fig. 7 provides insights into the performance of the multi-output model. Scatter plots in Fig. 7 compare the model predictions with the actual values for different output variables. While some variables show good agreement with the exact values (such as H and U), others exhibit more scattering (P_{cr1} , P_{dr2}).

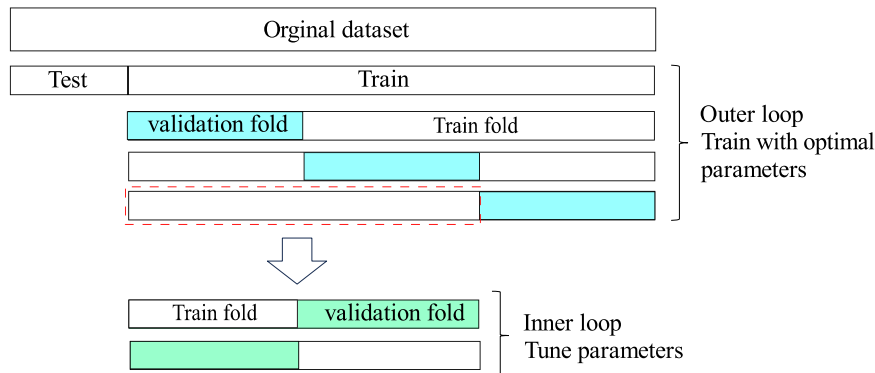


Fig. 6. Schematic presentation of nested cross-validation technique.

Table 1
Features description and information.

Feature	Range	Information
Indenter shape	Berkovich, rectangular, and cylindrical	Three different shapes (cylindrical, rectangular, and Berkovich) are used to investigate the effect of shape on stiffness and load-displacement response.
Indenter size	From 10 to 140 Å	For the rectangular indenter, the width of the flat punch tip and for the cylindrical indenter, the radius of the round tip are investigated, eight distinct sizes: 10, 20, 40, 60, 80, 100, 120 and 140 angstroms. For the Berkovich indenter, a fully sharp geometry is modeled.
Roughness	From 2 to 13 Å	13 surface profiles with different roughnesses are investigated to evaluate the effect of roughness on the indentation results.
Crystallography orientation	I and II	Two specific crystallographic orientations of the substrate are studied for anisotropic behavior.
Indenter position	i, ii, and iii	Indenter position at the moment of contact with the substrate, which can be at three locations: peak, slope, or valley.

4.2. Multi-output model with two separate groups of targets

To improve performance, two separate multi-output models are now trained instead of one unified model. The targets are divided into two groups: one model is trained on P_{cr1} , P_{cr2} , P_{dr1} , P_{dr2} , U and H , and the other is focused only on D_1 and D_2 . This separation is motivated by the observation that D_1 and D_2 have a weak correlation with P_{dr1} and P_{dr2} but a strong correlation with other targets, suggesting that their inclusion in a single model with all targets might negatively impact the learning.

The D_1 and D_2 models achieve a coefficient of determination (R^2) of 0.961, which is significantly greater than when they are included in the full multi-output model (0.936). This finding suggests that training D_1 and D_2 independently leads to better feature extraction and generalization for these two targets. On the other hand, the model that predicts P_{cr1} , P_{cr2} , P_{dr1} , P_{dr2} , U and H has a R^2 score of 0.934, which is nearly identical to the coefficient of determination of the single multioutput model. This indicates that removing D_1 and D_2 did not significantly affect the learning of the other targets, reinforcing the idea that D_1 and D_2 may have interfered with the learning process in the full multi-output model (see Table 2).

According to Fig. 8, which illustrates prediction performance of P_{cr1} , P_{cr2} , P_{dr1} , P_{dr2} , U and H , the model is less accurate than the second model. This is likely because the key variables D_1 and D_2 are removed from the target outputs. Since D_1 and D_2 are correlated with some target variables, their absence makes it more difficult for the model to accurately learn and predict the remaining outputs. The scatter plots show that while H and U are reasonably well predicted, the predictions for P_{cr1} , P_{cr2} , P_{dr1} and P_{dr2} have noticeable errors, indicating that the model struggles to capture their underlying relationships without D_1 and D_2 .

In contrast, Fig. 9, which focuses only on predicting D_1 and D_2 , shows significantly better accuracy. The scatter plots reveal a strong alignment between the predicted and exact values. This suggests that D_1 and D_2 are easier to model and they play crucial roles in defining the relationships among all the outputs.

Despite the improved accuracy for D_1 and D_2 , this two-model approach introduces inefficiencies. Training separate models prevents shared learning and leads to redundant computations, which is not ideal. Instead of completely separating the tasks, an MTL approach can be used to retain the benefits of task separation while still allowing for shared feature learning, leading to a more efficient and potentially more ac-

curate model, as discussed in Section 4.3.

4.3. Multitask learning

To address the limitations of training separate multi-output models, a multitask learning approach is implemented. It is expected to preserve the benefits of task separation while enabling shared learning. In this model, a shared network is defined to extract common features, followed by two task-specific branches: Task 1 for P_{cr1} , P_{cr2} , P_{dr1} , P_{dr2} , U , H and Task 2 for D_1 and D_2 . This design allows each task to focus on its relevant features while still benefiting from shared representations learned in the common layers.

According to Table 3, the MTL model achieves an overall R^2 score of 0.9573, which is greater than that of the single multioutput model (0.936) and the separate multioutput models (0.934 for P_{cr1} , P_{cr2} , P_{dr1} , P_{dr2} , U , H and 0.961 for D_1 and D_2). This suggests that MTL successfully leverages shared learning while mitigating the interference that occurs in the single multi-output model. By explicitly defining two tasks, it is ensured that D_1 and D_2 are not negatively influenced by their weakly correlated targets (P_{dr1} , P_{dr2}) while still benefiting from shared knowledge with other strongly correlated targets.

Additionally, MTL is computationally more efficient than training two independent models, as it optimizes both tasks simultaneously rather than redundantly processing shared features in two separate models. The use of an SGD optimizer instead of Adam may also have contributed to better generalization, as SGD often helps prevent overfitting by introducing more noise into the optimization process. Overall, MTL outperforms both the single and split multioutput models, confirming its effectiveness in handling targets with varying correlations while maintaining computational efficiency.

The model quickly learns the main data patterns within the first 20 epochs, showing a sharp drop in both training and validation losses. After that, the losses continue to decrease smoothly and stabilize around epoch 100. The close alignment of the two curves without divergence indicates successful training with no overfitting, as shown in Fig. 10.

Fig. 11 presents the results of the MTL model trained to predict multiple outputs simultaneously. The first eight subplots show scatter plots for comparing the predicted values (y-axis) with the exact values (x-axis) for each target variable, including P_{cr1} , P_{cr2} , P_{dr1} , P_{dr2} , D_1 , D_2 , U and H . Each plot includes a red dashed line representing the ideal 1:1 relationship, where perfect predictions would lie. The points are generally well aligned along this line, suggesting that the model has learned meaningful patterns, but some scatters indicate variability in prediction accuracy across different targets.

To interpret the effect of features on the predictions of complex machine learning models, the SHAP (Shapley Additive explanations) method was employed [70]. SHAP quantitatively and comparatively expresses the contribution of each feature to the model's prediction. The effect of each feature is computed by considering all possible combinations of features, thereby providing a fair estimate of its actual contribution to the model's output. This approach enables a precise understanding of how and to what extent each feature influences predictions, even in complex and nonlinear models such as neural networks or MTL models. In the present study, SHAP values were calculated and visualized for two representative features, P_{cr1} and H , to assess their impact on the MTL model's predictions. The resulting SHAP plots illustrate how variations in each feature can drive the model toward higher or lower predicted values (Fig 12).

4.4. Case studies

4.4.1. Example 1

A nanoindentation problem with a Berkovich indenter, surface roughness of 6.5Å, and crystallographic orientation I is solved. To evaluate the accuracy of the deep learning model solutions, the results of

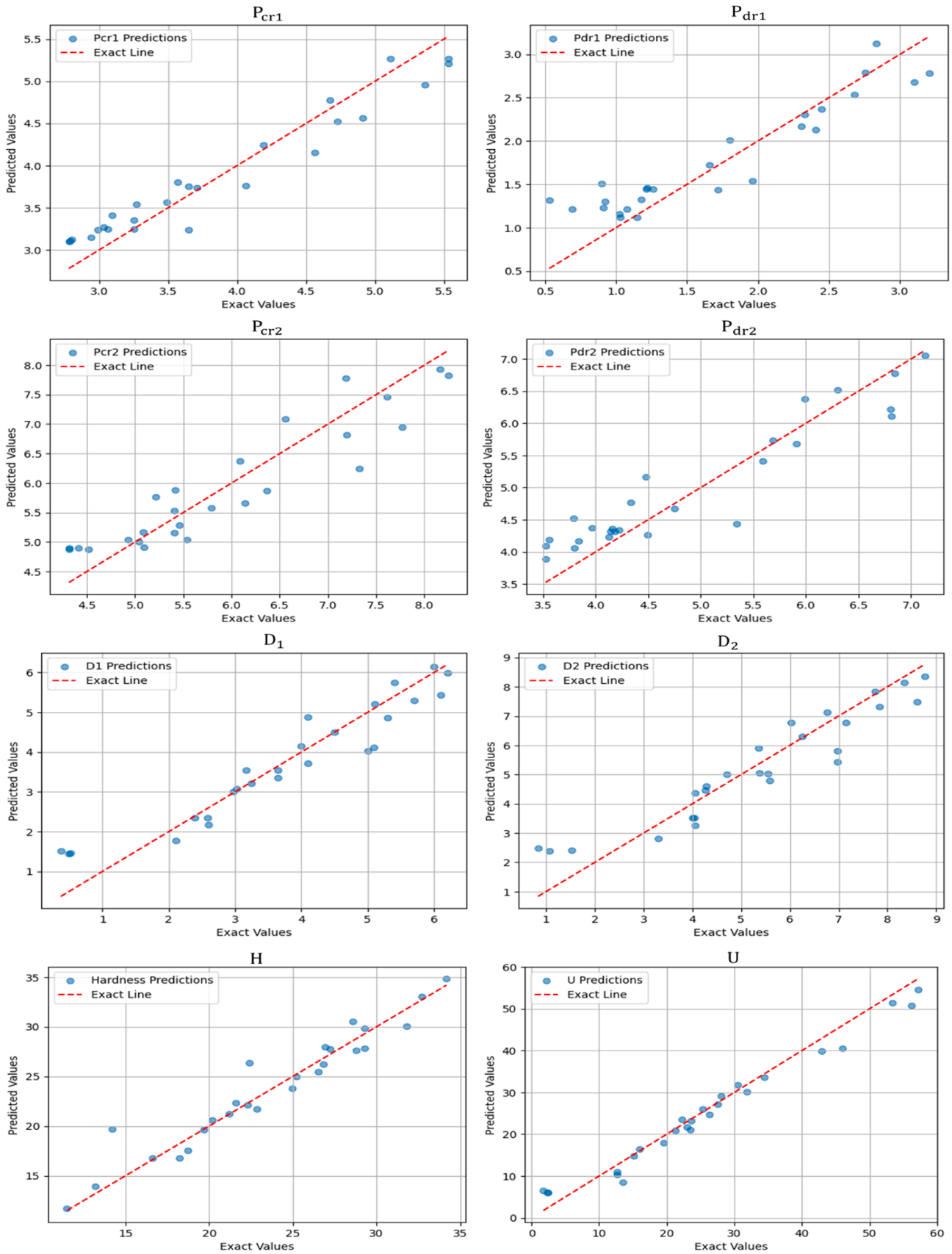


Fig. 7. Prediction performance and loss convergence of the multioutput model with all targets.

Table 2
Multi-output model: single network for all targets.

Model	Targets	N-layer	N-neurons	Activation	Drop-out	Optimizer	Learning rate	Weight-decay	coefficient of determination (R^2)
Multi output	P_{cr1} , P_{cr2} , P_{dr1} , P_{dr2} , D_1 , D_2 , U , H	3	75, 109, 81	Tanh	0.21, 0.11, 0.14	Adam	0.00230	0.0035	0.936

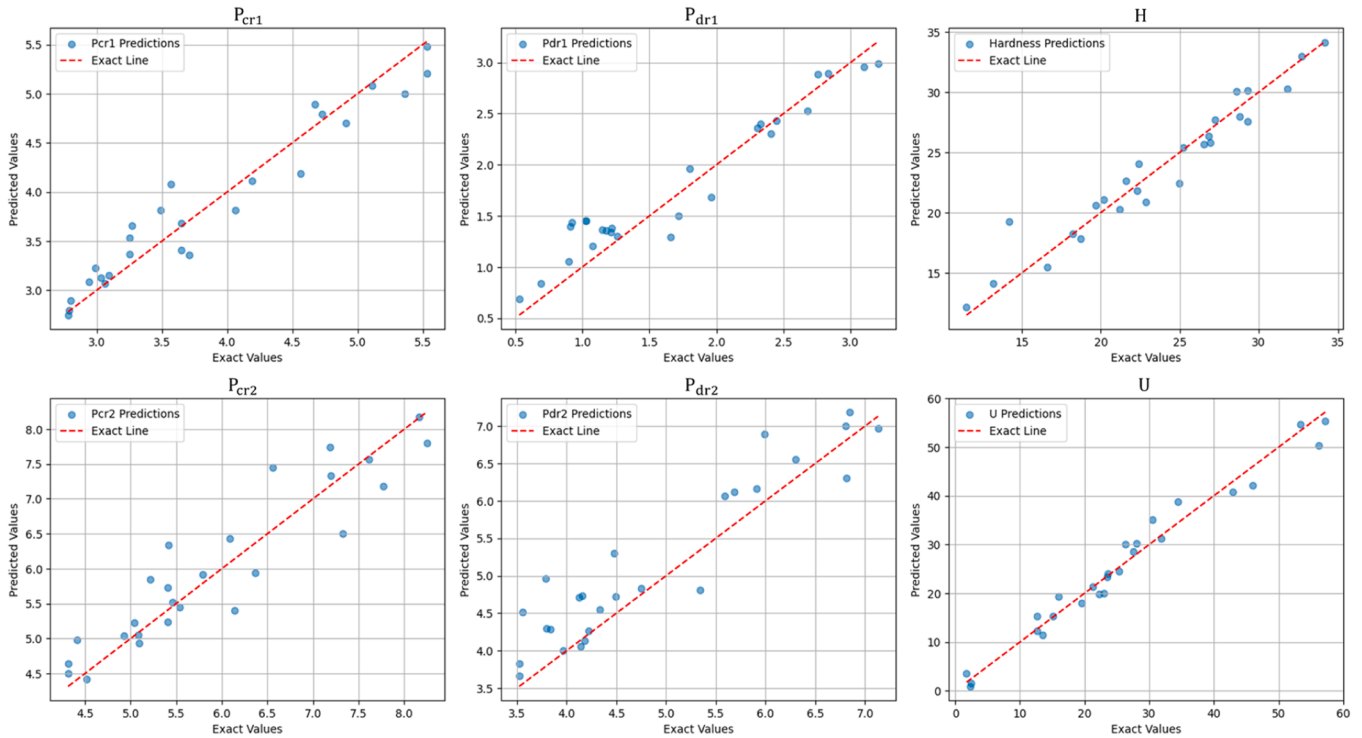


Fig. 8. Prediction performance and training convergence of the multi-output model without D_1 and D_2 .

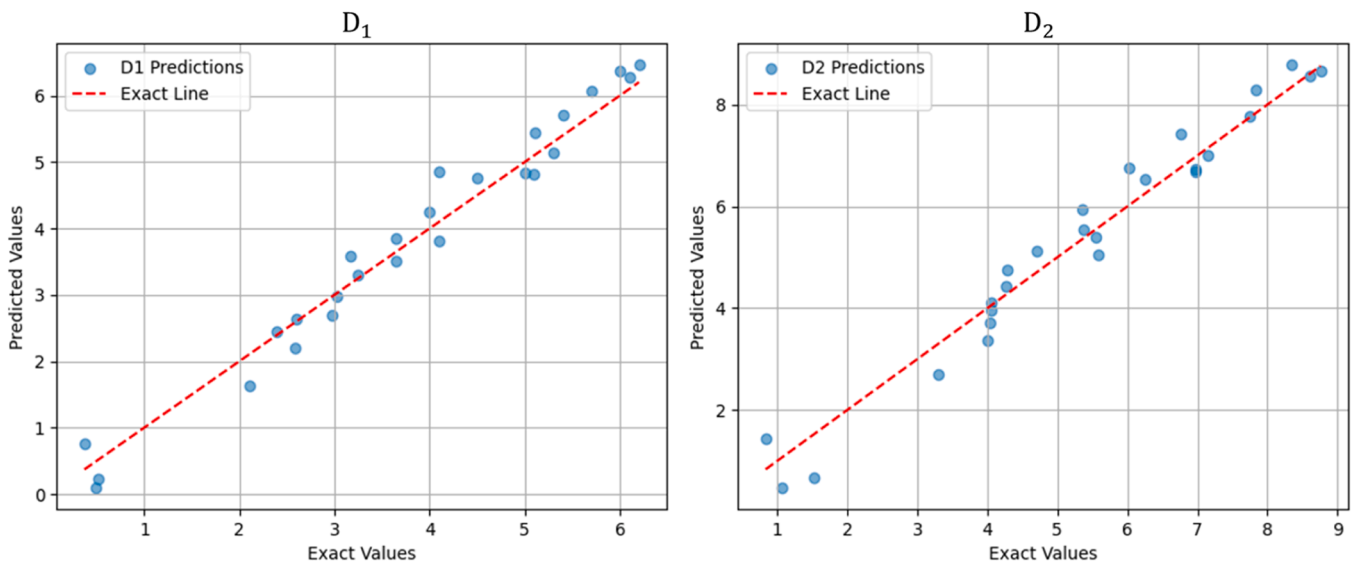


Fig. 9. Prediction performance and training loss convergence of the multi-output model for predicting only D_1 and D_2 .

the MTL model are compared with those of the QC model, as shown in Fig. 13.

The predicted load-penetration response from the MTL model closely matches that of the QC model across the entire range of penetration depths. The critical loads (P_{cr1} , P_{cr2}) at the first and second dislocations

are almost captured by the MTL model (points A and C), with a relative error of $<1.2\%$. The MTL model provides a hardness value close to that of the QC model, reflecting the accuracy of the deep learning model in capturing the effects of surface roughness and crystallographic orientation. Moreover, the energy (U) absorbed up to the second dislocation is

Table 3

Multi-output model: separate networks for different target groups.

Model	Targets	N-layer	N-neurons	Activation	Drop-out	Optimizer	Learning rate	Weight-decay	coefficient of determination (R^2)
Multi output	$P_{cr1}, P_{cr2}, P_{dr1}, P_{dr2}, U, H$	3	65,70,82	Tanh, Tanh, Sigmoid	0.27, 0.17, 0.14	Adam	0.00221	0.0030	0.934
Multi output	D_1, D_2	3	43,16,89	Tanh	0.21, 0.11, 0.15	Adam	0.00240	0.0035	0.961

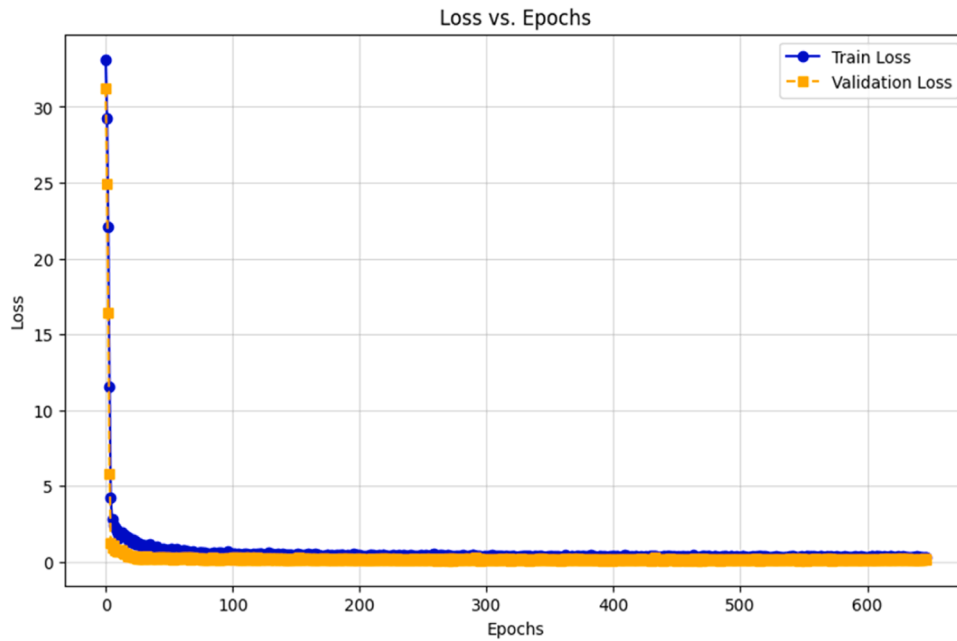


Fig. 10. Training and validation losses convergence over epochs for the MTL model.

approximately predicted by the model, with a negligible relative error. This highlights the model robustness in estimating energy-based mechanical metrics, which are critical for material deformation analysis. After the onset of plasticity, the MTL model effectively captures the drops in load (P_{dr1}, P_{dr2}), which are indicative of dislocation motion and slip initiation in the crystalline structure (points B and D).

Table 4 presents a comparison between the QC and MTL models in predicting H, U, and the required computational time. The MTL model demonstrates excellent accuracy, with only about 2.2 % error in predicting hardness and energy compared to the QC model. More importantly, it significantly reduces the computational time, from 45 h for the QC model to just 4.2 h for the MTL model training.

4.2.2. Example 2

An indentation test with a cylindrical indenter, smooth surface, and crystallographic orientation II is solved by both the QC method and the MTL model.

This load-penetration diagram in Fig. 14 compares the results of the QC and MTL models. Points A and C represent the first and second critical loads (P_{cr1}, P_{cr2}), where dislocations occur, whereas B and D correspond to the load drops (P_{dr1}, P_{dr2}) following these critical events. The zoomed-in atomic structures illustrate deformation at these points, showing the initial dislocation formation at P_{cr1} and the second dislocation at P_{cr2} . The area under the curve up to D represents U, whereas the hardness is calculated as P_{cr1} divided by the contact area between the indenter and the substrate. Comparing the two models, the MTL model closely follows the QC model, with only minor deviations at critical points and a relative error of <1.8 %, indicating slight variations in predicting dislocation events and material response.

Considering that a surface roughness of zero has not been included in

the training samples, evaluating the model performance under this condition is particularly important. As shown in Table 5, the MTL model is able to predict H and U with reasonable accuracy despite this limitation. The prediction errors are 2.5 % for hardness and 4.5 % for energy. Moreover, the MTL model significantly reduces the computational time, requiring only 4.2 h compared to 21 h for the QC model (Table 6).

It is noted that while the training time of the MTL model network is approximately 4 h, predicting new samples after training is very fast and takes only a few seconds. These results indicate that the MTL model not only maintains high predictive accuracy but also offers a highly efficient alternative to computationally expensive multiscale methods such as QC.

4.4.3. Critical force assessment

Fig. 15 illustrates the variation of the first critical load (P_{cr1}) as a function of surface roughness for the Berkovich indenter. It is worth noting that the selected roughness values (5, 7, 9, and 11 Å) are not explicitly present in the original dataset. However, they fall within the range of [2–13] Å, over which the model is trained. Hence, the predictions in this range are reliable and can be reasonably generalized to intermediate roughness values.

As observed, the model predicts a physically consistent trend. With increasing surface roughness, the magnitude of the critical load required for the nucleation of the first dislocation decreases monotonically. This behavior can be rationalized by considering the stress concentration at asperities of a rough surface. Higher roughness generates localized weak points where dislocation nucleation is facilitated, thereby reducing the effective resistance of the material against indentation. Consequently, lower values of (P_{cr1}) are expected. This finding highlights that the machine learning model not only captures the training data accurately

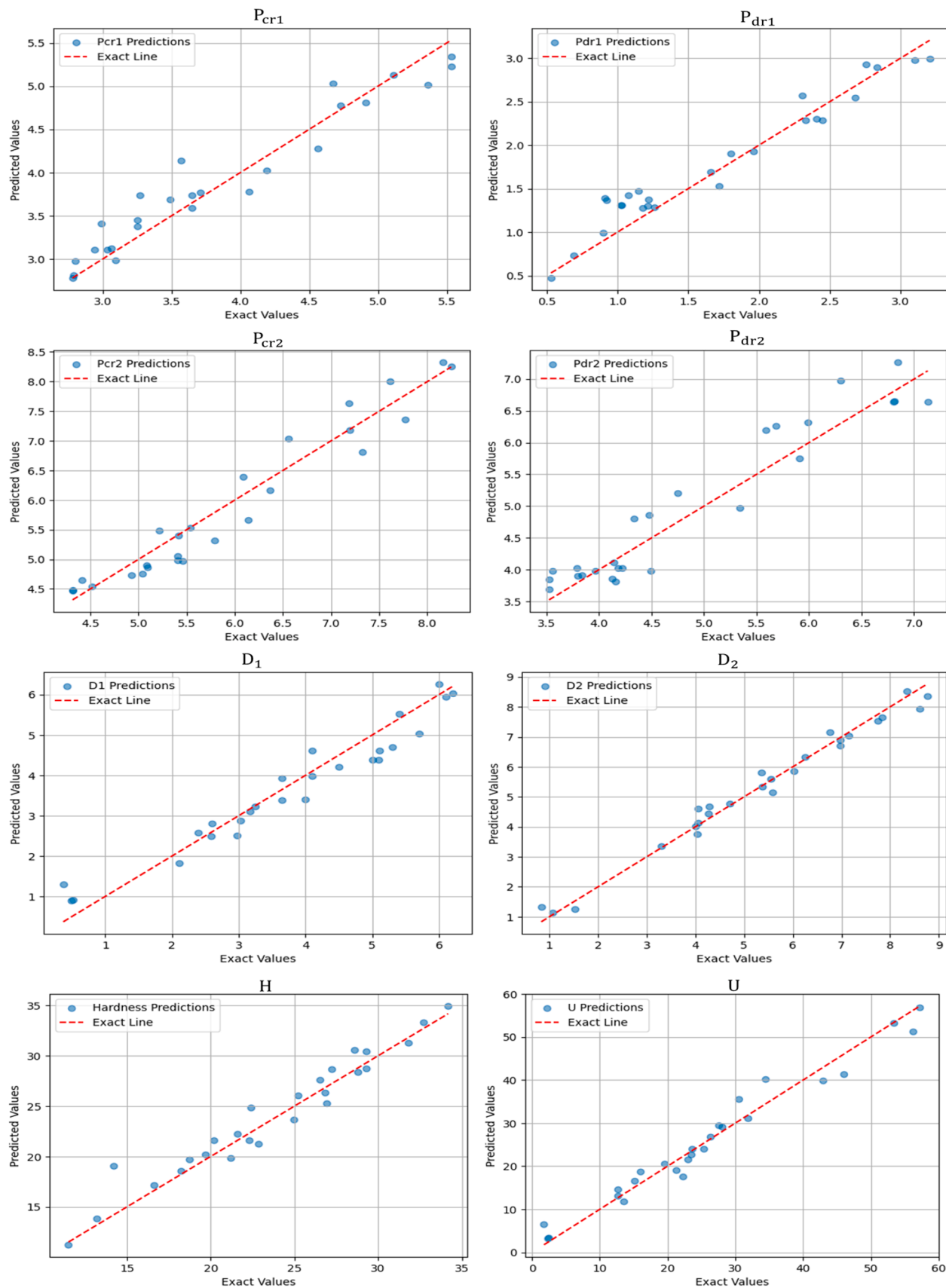


Fig. 11. Prediction performance of the MTL model.

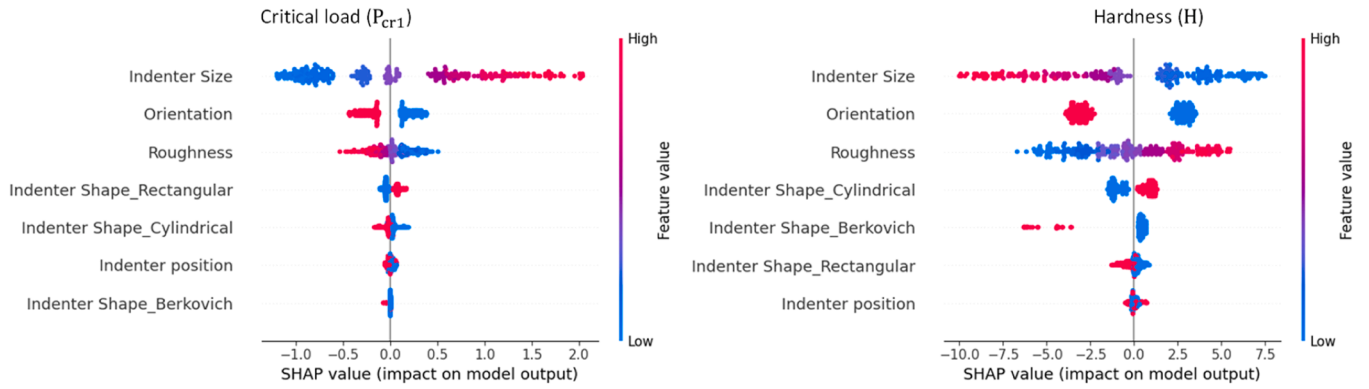


Fig. 12. Feature contribution analysis using SHAP: Critical load (P_{cr1} and Hardness (H) in the MTL model.

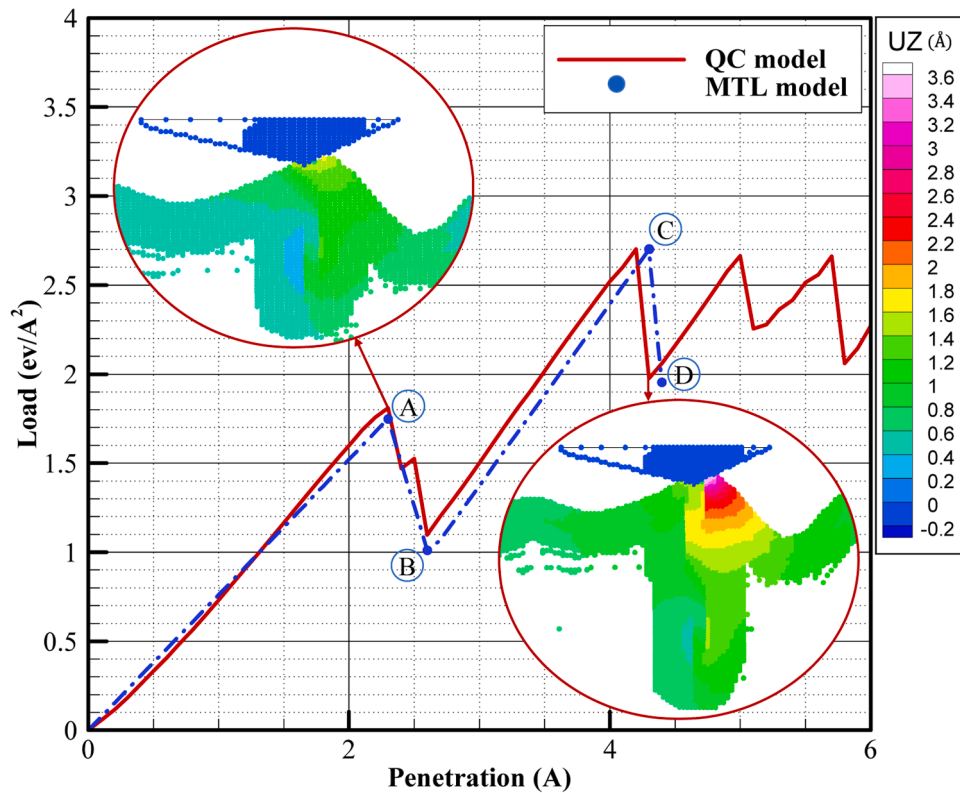


Fig. 13. Comparison between the results of the QC and MTL models for the Berkovich indenter and $R_a = 6.5 \text{ Å}$, the blue dashdotted lines are depicted only for better visualization.

Table 4
Multitask learning model.

Model	Targets		N-layer	N-neurons	Activation	Drop-out	Optimizer	Learning rate	Weight-decay	coefficient of determination (R^2)
MTL	$P_{cr1}, P_{cr2}, P_{dr1}, P_{dr2},$	Shared	2	59,91	Relu	0.15, 0.10	SGD	0.0043	0.0015	0.9573
	U, H	Task1	2	36,31	Tanh	0.19, 0.24				
	D_1, D_2	Task2	2	19,18	Tanh	0.17, 0.2				

but also reproduces the underlying physical trends governing the nanoindentation response.

The variation of the first critical load (P_{cr1}) as a function of surface roughness is depicted in Fig. 16 for three different indenter geometries: Cylindrical, Rectangular, and Berkovich, each considered under two crystallographic orientations (Orientation I and Orientation II).

When comparing the three geometries, the Rectangular indenter

consistently exhibits the highest values of P_{cr1} . This behavior can be explained by its larger and more uniform contact area, which distributes the applied stress more evenly and thereby delays the onset of dislocation nucleation. In contrast, the Berkovich indenter produces the lowest P_{cr1} values. The sharp geometry of this indenter leads to severe local stress concentration at the tip, which promotes easier nucleation of dislocations. The Cylindrical indenter shows an intermediate response,

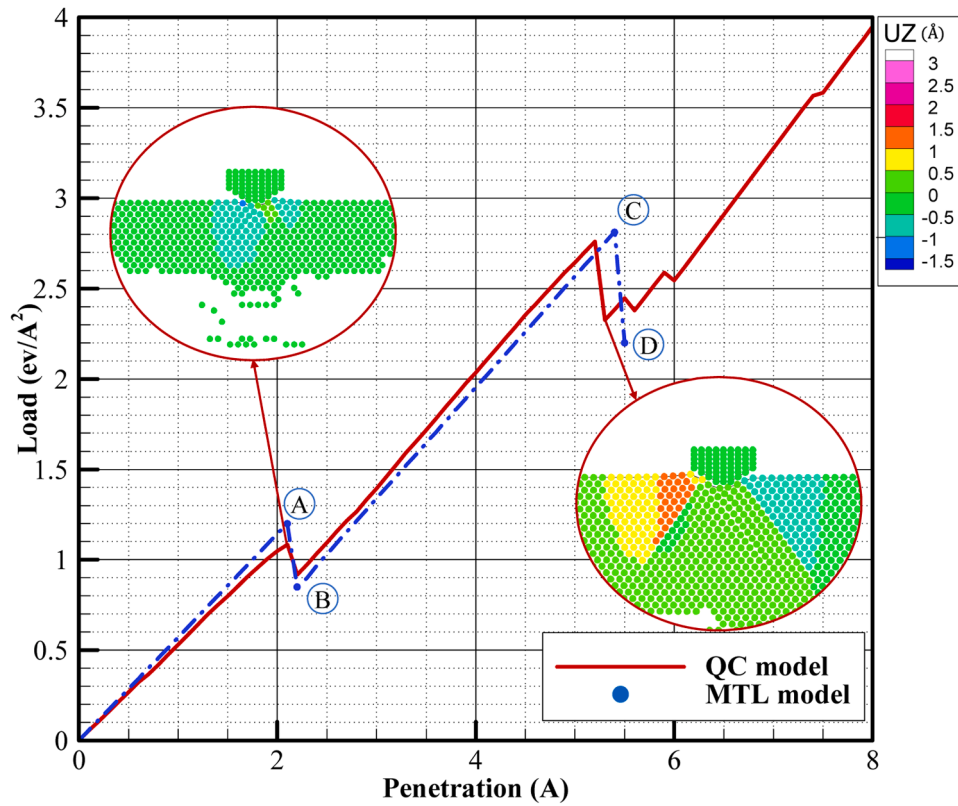


Fig. 14. Comparison between the results of QC and MTL models for the cylindrical indenter and smooth surface; the blue dashdotted lines are depicted only for better visualization.

Table 5

Comparison of QC and MTL models for the Berkovich indenter on a rough surface.

	H (Gpa)	U (ev)	Run time (h)
QC model	21.70	6.25	45
MTL model	21.23	6.11	4.2
Error (%)	2.16	2.24	—

Table 6

Comparison of QC and MTL models for the spherical indenter on a smooth surface.

	H (Gpa)	U (ev)	Run time (h)
QC model	18.66	7.95	21
MTL model	19.13	8.31	4.2
Error (%)	2.51	4.52	—

with P_{cr1} values lying between those of the rectangular and Berkovich geometries.

Furthermore, a consistent difference between the two crystallographic orientations is evident. For all indenter geometries, Orientation I yields higher values of P_{cr1} compared to Orientation II. This difference arises from crystallographic anisotropy: in Orientation I, the crystal lattice offers greater resistance to deformation and dislocation nucleation, whereas in Orientation II, slip systems are more easily activated, requiring lower critical loads.

Fig. 17 presents the predictions of the model for a rectangular indenter with a surface roughness of 9 Å. As observed, P_{cr1} increases monotonically with indenter size for both crystallographic orientations, indicating that larger indenters require a higher critical load to initiate the first instability or slip event. This trend reflects a clear size effect, whereby an increased contact area enhances the overall resistance of the

system against deformation. Furthermore, a consistent difference is evident between the two orientations: for all indenter sizes, P_{cr1} values corresponding to Orientation I remain higher than those of Orientation II. This implies that Orientation I exhibits greater hardness and requires more energy to trigger instability, whereas Orientation II is comparatively softer and yields more readily.

Overall, these figures highlight three key physical insights that are well captured by the model: (1) the decrease of P_{cr1} with increasing roughness, (2) the strong dependence of the response to indenter geometry, and (3) the significant role of crystallographic orientation in governing the onset of dislocation nucleation.

5. Conclusions

This study presents a deep learning framework for predicting the mechanical response of nanoindentation using the data generated by the quasicontinuum method. By leveraging neural networks, complex nonlinear relationships between key nanoindentation parameters, including critical loads (P_{cr1} , P_{cr2}), load drops (P_{dr1} , P_{dr2}), indentation depths (D_1 , D_2), hardness (H), and energy dissipation (U) are successfully modeled. The results confirm that the developed deep learning model can effectively capture material deformation mechanisms at the atomic scale while significantly reducing computational costs compared with conventional QC simulations.

One of the key contributions of this work is the evaluation of three different machine learning models. The multioutput model, which predicts all indentation parameters simultaneously, provides a useful baseline but struggles with weakly correlated features. The two separate model approach, where indentation depths (D_1 , D_2) are separated from other mechanical properties, which improves prediction accuracy by reducing interference between unrelated outputs. However, the multi-task learning (MTL) model outperforms both alternatives, achieving the highest accuracy and lowest computational cost by leveraging shared

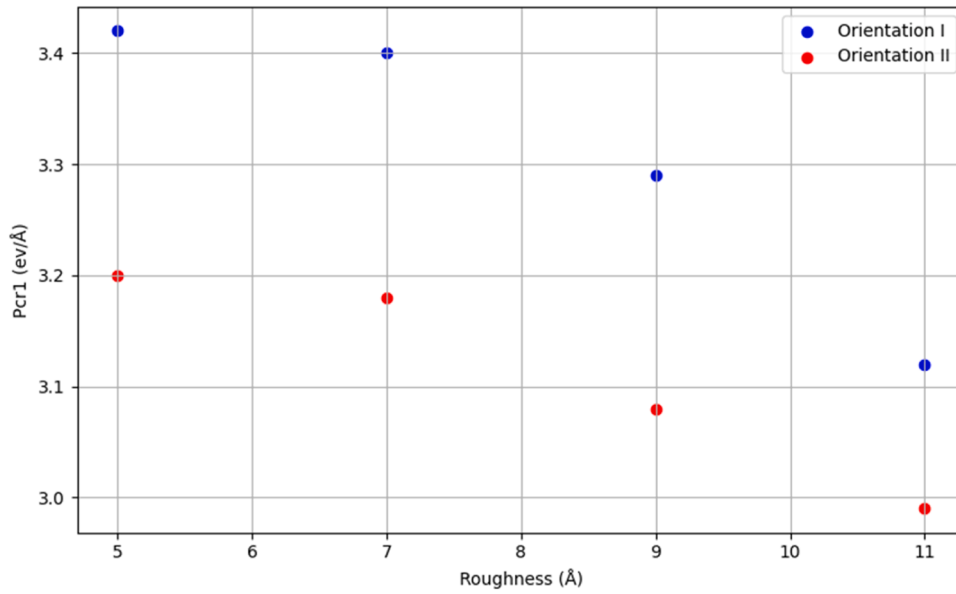


Fig. 15. Variation of the first critical load (P_{cr1}) with surface roughness for the Berkovich indenter in two crystallographic orientations.

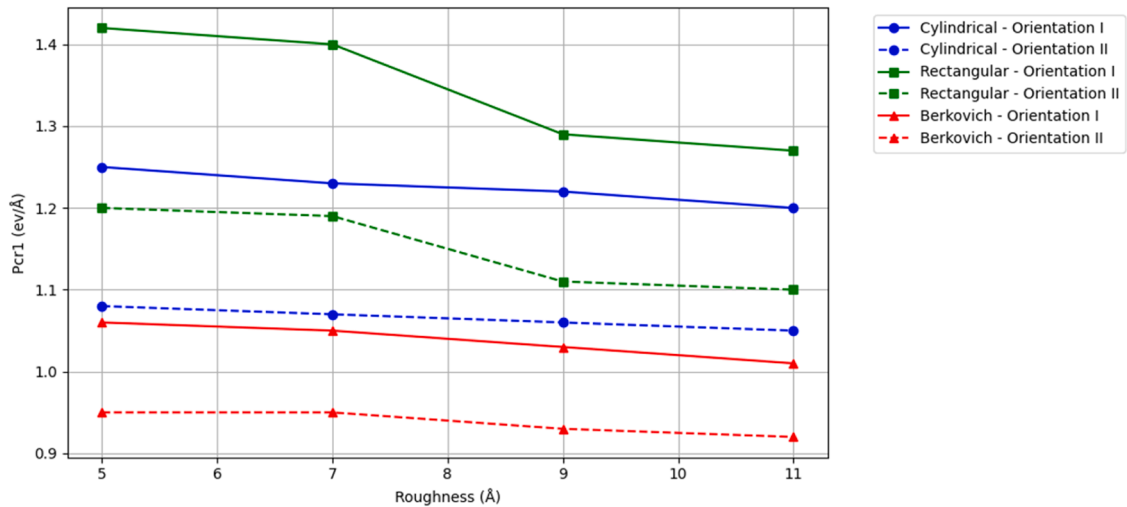


Fig. 16. Comparison of the first critical load among Cylindrical, Rectangular, and Berkovich indenters as a function of surface roughness in two crystallographic orientations.

learning between related variables.

The superiority of the MTL approach is due to its ability to capture interdependencies between mechanical properties while preventing interference from weakly correlated variables. Unlike the simple multioutput model, where each target variable is treated independently, MTL explicitly organizes the learning process into two related tasks: one focused on penetration (D_1 , D_2) and the other on critical loads, load drops, hardness, and energy dissipation. This structured learning process allows the model to benefit from shared feature representations, where relevant information from one task can improve predictions for another. By preventing unrelated outputs from interfering with each other, MTL effectively reduces prediction errors and enhances model robustness. Additionally, MTL reduces the risk of overfitting by enforcing regularization through task sharing, leading to improved generalizability to unseen data.

To validate the performance of the models, two case studies were conducted: (1) a Berkovich indenter on a rough copper surface with crystallographic orientation I and (2) a cylindrical indenter on a smooth surface with orientation II. In both scenarios, the deep learning

predictions closely matched the QC simulations, with minimal errors in critical load and hardness predictions. This confirmed that the proposed framework could reliably model material behavior across different nanoindentation conditions.

The proposed deep learning framework has demonstrated its ability to successfully capture the influence of key physical features, such as indenter geometry, surface roughness and crystallographic orientation, on the nanoindentation response. The predicted trends are in close agreement with the underlying atomic-scale physics: the model correctly identifies the reduction of the critical load with increasing surface roughness, the strong dependence of the response on indenter geometry, and the anisotropic behavior associated with crystallographic orientation. These findings highlight that the model not only reproduces the training data but also generalizes well to unseen cases, providing physically consistent predictions that align with established mechanisms of dislocation nucleation.

A crucial aspect of this study was overfitting; a common challenge in machine learning models trained on high-dimensional physical data. Nested cross-validation was used to ensure robust hyperparameter

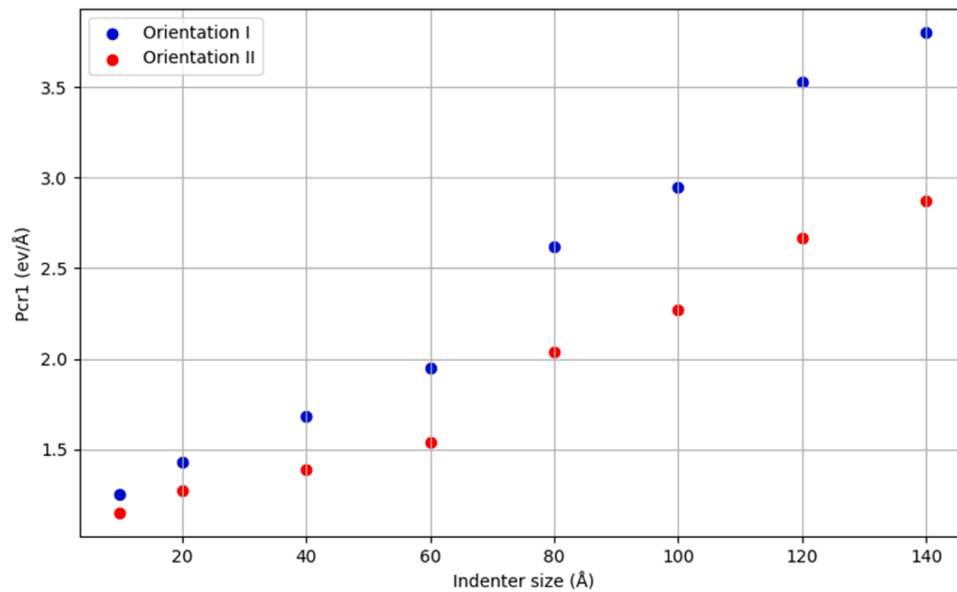


Fig. 17. Variation of the first critical load (P_{cr1}) with indenter size for a rectangular indenter at a surface roughness of 9 Å in two crystallographic orientations.

tuning by separating the model selection process from the final performance evaluation, preventing information leakage between the training and validation sets and the mini-batch. Additionally, regularization techniques such as weight decay and dropout were applied to control model complexity weight decay penalized large weights for stability, whereas dropout randomly deactivated neurons to encourage robust feature learning. Finally, early stopping was employed to halt training once the validation performance plateaued, avoiding unnecessary overfitting. These combined strategies ensured that the trained neural networks maintained high accuracy and reliability across unseen data.

Another advantage of the proposed deep learning framework is its computational efficiency. While traditional QC simulations are computationally demanding and require significant processing time for each indentation scenario, the trained neural networks provide near-instantaneous predictions. This efficiency makes the model well-suited for high-performance studies, where multiple indentation conditions need to be analyzed rapidly. The ability to predict material behavior with minimal computational cost opens new possibilities for large-scale material screening and optimization.

The proposed MTL framework provides a promising foundation for future advancements in AI-driven nanomechanics, enabling more efficient material characterization and design. As deep learning continues to evolve, its integration with multiscale modeling and experimental data is expected to further enhance our understanding of nanoscale mechanical phenomena.

While the present study focused on copper thin films, the proposed MTL-QC framework is not limited to this specific material system. Owing to its flexible structure, the framework can be readily extended to other crystalline metals, as long as appropriate simulation parameters are provided. Moreover, with suitable scaling of boundary conditions and loading configurations, the methodology can be adapted to investigate bulk materials as well. This generalizability highlights the potential of the MTL-QC approach as a versatile tool for predicting nanoscale mechanical properties across a wide range of materials and geometries, beyond the specific case investigated here.

CRedit authorship contribution statement

Hasan Ghahremani: Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization.
Soheil Mohammadi: Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

The authors declare that they have no known competing/conflicting financial interests or personal relationships that could have appeared to influence the work reported in this paper. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to express their sincere appreciation for the technical support provided by the High-Performance Computing Laboratory (HPC Lab) at the University of Tehran. We also wish to extend our gratitude to Dr. O. Alizadeh and Dr. H. Moslemzadeh for their assistance and guidance in the earlier stage of QC analysis.

Data availability

Data will be made available on request.

References

- [1] S. Huang, C. Zhou, Modeling and simulation of nanoindentation, *Miner. Met. Mater. Soc.* (2017), <https://doi.org/10.1007/s11837-017-2541-1>.
- [2] L. Nipa, H.R. Siller, M. Sharma, B. Rout, R.A. Mirshams, Investigating the effects of hot isostatic pressing and proton irradiation on L-PBF stainless steels via nanoindentation across different strain rates, *Results Eng.* 27 (2025) 106637, <https://doi.org/10.1016/j.rineng.2025.106637>. Sep.
- [3] J. Fish, *Multiscale Methods*, Oxford University Press/Oxford, 2009, <https://doi.org/10.1093/acprof:oso/9780199233854.001.0001>.
- [4] G. Lu and E. Kaxiras, "An overview of multiscale simulations of materials," 2004.
- [5] X. Zeng, X. Wang, J.D. Lee, Y. Lei, Multiscale modeling of nano/micro systems by a multiscale continuum field theory, *Comput. Mech.* 47 (2) (2011) 205–216, <https://doi.org/10.1007/s00466-010-0538-5>.
- [6] O. Alizadeh, S. Mohammadi, A review of concurrent multiscale methods for the analysis of fine scale discontinuity problems, *J. Comput. Methods Eng.* 42 (2) (2024), <https://doi.org/10.47176/jcme.42.2.1013>. Mar.
- [7] S. Mohammadi, *Multiscale Biomechanics*, Wiley, 2023, <https://doi.org/10.1002/9781119033714>.
- [8] E.B. Tadmor, M. Ortiz, R. Phillips, Quasicontinuum analysis of defects in solids, *Philos. Mag. A* 73 (6) (1996) 1529–1563, <https://doi.org/10.1080/01418619608243000>. Jun.

- [9] H. Moslemzadeh, S. Mohammadi, An atomistic entropy based finite element multiscale method for modeling amorphous materials, *Int. J. Solids. Struct.* 256 (2022) 111983, <https://doi.org/10.1016/j.ijsolstr.2022.111983>. Dec.
- [10] O. Alizadeh, S. Mohammadi, The variable node multiscale approach: coupling the atomistic and continuum scales, *Comput. Mater. Sci.* 160 (2019) 256–274, <https://doi.org/10.1016/j.commatsci.2019.01.003>. Apr.
- [11] R.E. Miller, E.B. Tadmor, The Quasicontinuum method: overview, applications and current directions, *J. Comput.-Aided Mater. Des.* 9 (3) (2002) 203–239, <https://doi.org/10.1023/A:1026098010127>.
- [12] J.D. Kiely, R.Q. Hwang, J.E. Houston, Effect of surface steps on the plastic threshold in nanoindentation, *Phys. Rev. Lett.* 81 (20) (1998) 4424–4427, <https://doi.org/10.1103/PhysRevLett.81.4424>. Nov.
- [13] R. Iglesias, E.P.M. Leiva, Two-grain nanoindentation using the quasicontinuum method: two-dimensional model approach, *Acta Mater.* 54 (10) (2006) 2655–2664, <https://doi.org/10.1016/j.actamat.2006.02.004>. Jun.
- [14] C. Walter, T. Antretter, R. Daniel, C. Mitterer, Finite element simulation of the effect of surface roughness on nanoindentation of thin films with spherical indenters, *Surf. Coat. Technol.* 202 (4–7) (2007) 1103–1107, <https://doi.org/10.1016/j.surfcoat.2007.07.038>. Dec.
- [15] C.-T. Wang, S.-R. Jian, J.S.-C. Jang, Y.-S. Lai, P.-F. Yang, Multiscale simulation of nanoindentation on Ni (100) thin film, *Appl. Surf. Sci.* 255 (5) (2008) 3240–3250, <https://doi.org/10.1016/j.apsusc.2008.09.034>. Dec.
- [16] A.V. Bolesta, V.M. Fomin, Molecular dynamics simulation of sphere indentation in a thin copper film, *Phys. Mesomech.* 12 (3–4) (2009) 117–123, <https://doi.org/10.1016/j.physme.2009.07.003>. May.
- [17] J. Li, Y. Ni, H. Wang, J. Mei, Effects of crystalline anisotropy and indenter size on nanoindentation by multiscale simulation, *Nanoscale Res. Lett.* 5 (2) (2010) 420–432, <https://doi.org/10.1007/s11671-009-9500-x>. Feb.
- [18] J. Li, H. Lu, Y. Ni, J. Mei, Quasicontinuum study the influence of misfit dislocation interactions on nanoindentation, *Comput. Mater. Sci.* 50 (11) (2011) 3162–3170, <https://doi.org/10.1016/j.commatsci.2011.05.045>. Oct.
- [19] O. Alizadeh, G.Tolooei Eshlaghi, S. Mohammadi, Nanoindentation simulation of coated aluminum thin film using quasicontinuum method, *Comput. Mater. Sci.* 111 (2016) 12–22, <https://doi.org/10.1016/j.commatsci.2015.08.044>. Jan.
- [20] T. Akatsu, H. Inuzuka, Y. Shinoda, F. Wakai, Weight functions for point-sharp nanoindentation to evaluate the mechanical properties of a film/coating on a substrate, *Results Eng.* 24 (2024) 103447, <https://doi.org/10.1016/j.rineng.2024.103447>. Dec.
- [21] A. Zhu, D. He, R. He, C. Zou, Nanoindentation simulation on single crystal copper by quasi-continuum method, *Mater. Sci. Eng.: A* 674 (2016) 76–81, <https://doi.org/10.1016/j.msea.2016.07.091>. Sep.
- [22] J. Jin, P. Yang, J. Cao, S. Li, Q. Peng, Quasicontinuum simulation of the effect of Lotus-type nanocavity on the onset plasticity of single crystal Al during nanoindentation, *Nanomaterials* 8 (10) (2018) 778, <https://doi.org/10.3390/nano8100778>. Sep.
- [23] H. Moslemzadeh, O. Alizadeh, S. Mohammadi, Quasicontinuum multiscale modeling of the effect of rough surface on nanoindentation behavior, *Meccanica* 54 (3) (2019) 411–427, <https://doi.org/10.1007/s11012-019-00956-x>. Feb.
- [24] Z. Zhang, C. Wang, X. Hu, Y. Ni, Shape effect of surface defects on nanohardness by quasicontinuum method, *Micromachines* 11 (10) (2020) 909, <https://doi.org/10.3390/mi11100909>. Sep.
- [25] H. Ghahremani, O. Alizadeh, H. Moslemzadeh, and S. Mohammadi, “Evaluation of nanohardness of rough surfaces in nanoindentation problems by the QC multiscale method”, submitted.
- [26] S. Xiao, R. Hu, Z. Li, S. Attarian, K.-M. Björk, A. Lendasse, A machine-learning-enhanced hierarchical multiscale method for bridging from molecular dynamics to continua, *Neural Comput. Appl.* 32 (18) (2020) 14359–14373, <https://doi.org/10.1007/s00521-019-04480-7>. Sep.
- [27] S. Park, J.H. Fonseca, K.P. Marimuthu, C. Jeong, S. Lee, H. Lee, Determination of material properties of bulk metallic glass using nanoindentation and artificial neural network, *Intermetallics* 144 (2022) 107492, <https://doi.org/10.1016/j.intermet.2022.107492>. May.
- [28] S. Park, K.P. Marimuthu, G. Han, H. Lee, Deep learning based nanoindentation method for evaluating mechanical properties of polymers, *Int. J. Mech. Sci.* 246 (2023) 108162, <https://doi.org/10.1016/j.ijsolstr.2023.108162>. May.
- [29] G. Han, K.P. Marimuthu, H. Lee, Evaluation of thin film material properties using a deep nanoindentation and ANN, *Mater. Des.* 221 (2022) 111000, <https://doi.org/10.1016/j.matdes.2022.111000>. Sep.
- [30] E.S. Puchi-Cabrera, E. Rossi, G. Sansonetti, M. Sebastiani, E. Bemporad, Machine learning aided nanoindentation: a review of the current state and future perspectives, *Curr. Opin. Solid State Mater. Sci.* 27 (4) (2023) 101091, <https://doi.org/10.1016/j.cossms.2023.101091>. Aug.
- [31] W. Jin, J. Pei, P. Xie, J. Chen, H. Zhao, Machine learning-based prediction of mechanical properties and performance of nickel–Graphene nanocomposites using molecular dynamics simulation data, *ACS Appl. Nano Mater.* 6 (13) (2023) 12190–12199, <https://doi.org/10.1021/acsanm.3c01919>. Jul.
- [32] X. Liu, Z. Long, L. Peng, Prediction of Vickers hardness of amorphous alloys based on interpretable machine learning, *J. Non. Cryst. Solids.* 602 (2023) 122095, <https://doi.org/10.1016/j.jnoncrysol.2022.122095>. Feb.
- [33] M. Sahnounne Chaouche, H.K. Al-Mohair, S. Askar, B.S. Abdullaeva, N.A. Hussien, A.H. Alawadi, A micromechanical nested machine learning model for characterizing materials behaviors of bulk metallic glasses, *J. Non. Cryst. Solids.* 625 (2024) 122733, <https://doi.org/10.1016/j.jnoncrysol.2023.122733>. Feb.
- [34] A.R. Khoei, M.R. Seddighian, A.R. Sameti, Machine learning-based multiscale framework for mechanical behavior of nano-crystalline structures, *Int. J. Mech. Sci.* 265 (2024) 108897, <https://doi.org/10.1016/j.ijsolstr.2023.108897>. Mar.
- [35] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions [Online]. Available, <https://arxiv.org/abs/1705.07874>, 2017.
- [36] M. Zaki, S. Kasimuthumanian, S. Sahoo, N.N.G. Jayadeva, N.M.A. Krishnan, Interpretable machine learning approach for identifying the tip sharpness in atomic force microscopy, *Scr. Mater.* 221 (2022) 114965, <https://doi.org/10.1016/j.scriptamat.2022.114965>. Dec.
- [37] C.O.W. Trost, et al., Explainable machine learning and feature engineering applied to nanoindentation data, *Mater. Des.* 253 (2025) 113897, <https://doi.org/10.1016/j.matdes.2025.113897>. May.
- [38] C.O.W. Trost, S. Zak, S. Schaffer, C. Saringer, L. Exl, M.J. Cordill, Bridging fidelities to predict nanoindentation tip radii using interpretable deep learning models, *JOM* 74 (6) (2022) 2195–2205, <https://doi.org/10.1007/s11837-022-05233-z>. Jun.
- [39] S. Bardeeniz, C. Panjapornpon, M.A. Hussain, V. Varabuntoonvit, K. Jitapunkul, Enhancing industrial sustainability in complex production systems through energy hotspot identification: a multi-task learning with layer-wise relevance propagation approach, *Results Eng.* 23 (2024) 102818, <https://doi.org/10.1016/j.rineng.2024.102818>. Sep.
- [40] T. Csanádi, M. Bfanda, A. Duszová, P. Hvizdoš, J. Dusza, Nanoindentation and AFM studies on tungsten carbide crystals in WC-Co hardmetal, *Key Eng. Mater.* 606 (2014) 107–110, <https://doi.org/10.4028/www.scientific.net/KEM.606.107>. Mar.
- [41] K. Sangwal, P. Gorostiza, J. Servat, F. Sanz, Atomic force microscopy study of nanoindentation deformation and indentation size effect in MgO crystals, *J. Mater. Res.* 14 (10) (1999) 3973–3982, <https://doi.org/10.1557/JMR.1999.0537>. Oct.
- [42] “ISO 21920-2, (en), ‘Geometrical product specifications (GPS) — Surface texture: profile — Part 2: terms, definitions and surface texture parameters, Int. Organ. Stand. (2021).”
- [43] P. Berke and T.J. Massart, “Coupled friction and roughness surface effects in shallow spherical nanoindentation,” 2011, pp. 269–289. doi: 10.1007/978-3-642-22167-5_15.
- [44] W.C. Oliver, G.M. Pharr, An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments, *J. Mater. Res.* 7 (6) (1992) 1564–1583, <https://doi.org/10.1557/JMR.1992.1564>. Jun.
- [45] A. Gouldstone, K.J. Van Vliet, S. Suresh, Simulation of defect nucleation in a crystal, *Nature* 411 (6838) (2001), <https://doi.org/10.1038/35079687>, 656–656Jun.
- [46] W. Zhang, Y. Gao, Y. Xia, H. Bei, Indentation Schmid factor and incipient plasticity by nanoindentation pop-in tests in hexagonal close-packed single crystals, *Acta Mater.* 134 (2017) 53–65, <https://doi.org/10.1016/j.actamat.2017.05.058>. Aug.
- [47] T. Ohmura, M. Wakeda, Pop-In phenomenon as a fundamental plasticity probed by nanoindentation technique, *Materials* 14 (8) (2021) 1879, <https://doi.org/10.3390/ma14081879>. Apr.
- [48] Y. Sato, S. Shinzato, T. Ohmura, T. Hatano, S. Ogata, Unique universal scaling in nanoindentation pop-ins, *Nat. Commun.* 11 (1) (2020) 4177, <https://doi.org/10.1038/s41467-020-17918-7>. Aug.
- [49] P.H. J. L. Peter M. Anderson, *Theory of Dislocations*, Cambridge University Press, 2017.
- [50] S. Zak, C.O.W. Trost, P. Kreiml, M.J. Cordill, Accurate measurement of thin film mechanical properties using nanoindentation, *J. Mater. Res.* 37 (7) (2022) 1373–1389, <https://doi.org/10.1557/s43578-022-00541-1>. Apr.
- [51] K.P. Murphy, *Machine learning : a Probabilistic Perspective*, MIT Press, 2012.
- [52] J. Heaton, Ian Goodfellow, Yoshua Bengio, and Aaron Courville: deep learning, *Genet. Program. Evolvable Mach.* 19 (1–2) (Jun. 2018) 305–307, <https://doi.org/10.1007/s10710-017-9314-z>.
- [53] S.S. Haykin, *Neural Networks and Learning Machines*, Prentice Hall/Pearson, 2009.
- [54] C.M. Bishop, *Pattern recognition and machine learning*, *J. Electron. Imaging* 16 (4) (2007) 049901, <https://doi.org/10.1117/1.2819119>. Jan.
- [55] A. Paszke, et al., PyTorch: an imperative style, high-performance deep learning library, in: *Advances in Neural Information Processing Systems*, 32, NeurIPS, 2019.
- [56] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA: ACM, 2019, pp. 2623–2631, <https://doi.org/10.1145/3292500.3330701>. Jul.
- [57] R. Caruana, L. Pratt, S. Thrun, *Multitask Learning **, Kluwer Academic Publishers, 1997.
- [58] O. Kramer, “Scikit-learn,” 2016, pp. 45–53. doi: 10.1007/978-3-319-33383-0_5.
- [59] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, in: *Proceedings of the 26th International Conference on Neural Information Processing Systems 2*, 2012, pp. 2951–2959. NIPS’12. Red Hook, NY, USA: Curran Associates Inc.
- [60] J. Bergstra, J.B. Ca, Y.B. Ca, Random search for hyper-parameter optimization Yoshua Bengio [Online]. Available, <http://scikit-learn.sourceforge.net>, 2012.
- [61] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA: ACM, 2019, pp. 2623–2631, <https://doi.org/10.1145/3292500.3330701>. Jul.
- [62] L. Bottou, “Stochastic gradient descent tricks,” 2012, pp. 421–436. doi: 10.1007/978-3-642-35289-8_25.
- [63] J. Yang, Q. Long, a modification of adaptive moment estimation (Adam) for machine learning, *J. Ind. Manag. Optim.* 20 (7) (2024) 2516–2540, <https://doi.org/10.3934/jimo.2024014>.
- [64] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, Dec, <http://arxiv.org/abs/1412.6980>, 2014.

- [65] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. *Cognitive Modeling*, The MIT Press, 2002, pp. 213–222, <https://doi.org/10.7551/mitpress/1888.003.0013>.
- [66] L. Bottou, “Stochastic gradient descent tricks,” 2012, pp. 421–436. doi: 10.1007/978-3-642-35289-8_25.
- [67] H. Babaei, M. Zamani, S. Mohammadi, The impact of data splitting methods on machine learning models: A case study for predicting concrete workability, *ML-CS&E* 1 (1) (2025) 21, <https://doi.org/10.1007/s44379-025-00021-3>.
- [68] S. Varma, R. Simon, Bias in error estimation when using cross-validation for model selection, *BMC Bioinform.* 7 (1) (2006) 91, <https://doi.org/10.1186/1471-2105-7-91>. Dec.
- [69] M. Smelyanskiy, N. Shirish Keskar, D. Mudigere, J. Nocedal, P. Tak, and P. Tang, “On large-batch training for deep learning: generalization gap and sharp minima,” 2016, doi: 10.48550/arXiv.1609.04836.
- [70] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, *Adv. Neural Inf. Process. Syst.* (2017) 4766–4775, 2017-Decem, no. Section 2.