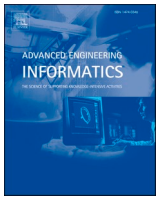
Advanced Engineering Informatics 73 (2026) 104600

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Full length article

A generalizable neural operator for full-field deformation prediction in robotic design

Zhiwei Zhao [a], Yan Jin [a,*], Caitlin Sands [a], Paul Goodall [b], Andrew West [b], Mark Price [a]

[a] School of Mechanical and Aerospace Engineering, Queen's University Belfast, United Kingdom
[b] Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, United Kingdom

ARTICLE INFO

Keywords:
Robotic design
Deformation field prediction
Neural operator
Graph neural network

ABSTRACT

Robotic design and optimization often require extensive simulation-based iterations, particularly for predicting static deformations under varying postures, external loads, and gravitational conditions. Conventional finite element simulations can achieve high accuracy but are computationally expensive, limiting their use in large-scale design optimization. Data-driven approaches have emerged to accelerate prediction; however, most existing methods are restricted to simplified scenarios, such as fixed geometries with varying poses or fixed poses with varying loads, and typically predict only partial deformation at points rather than the full deformation field. To enable fast and generalizable deformation prediction, a new neural operator architecture named Robot-NO is developed. A geometric–load graph based on the finite element mesh connectivity, where each node encodes spatial coordinates with local load conditions is constructed. A graph neural network is then employed to extract a unified global representation that captures both geometric structure and external force distribution. A query–decoder module combines global features with arbitrary query-coordinates to predict the full deformation field. This query-based formulation enables accurate continuous deformation prediction across varying robot geometries, postures, and load conditions. Experimental results demonstrate that the proposed model achieves a root mean square error of 0.91 μm in deformation prediction and is 6000 times faster than conventional finite element simulations. This significant improvement enables rapid design, analysis and optimization for robotic mechanisms.

1. Introduction

Robots offer significant advantages such as high flexibility [1], large workspaces [2], low weight and relatively low cost [3], which make them valuable in applications including surgical operations [4], manufacturing[5], and assembly [6]. However, the mechanical stiffness of robots strongly depends on their geometry and posture [7], and substantial structural deformation can occur under varying gravitational and loading conditions [8]. Such deformation significantly degrades the positioning accuracy of the end-effector and adversely affects task performance [9]. For example, in ophthalmic microsurgery, even a small deformation of the robot structure can lead to loss of positioning precision [10], while in robotic milling, elastic deformation can distort the intended tool path and surface quality [11].

To mitigate these effects, designers often optimize both the robot structure and its task trajectory based on predicted deformation behavior [12]. Accurate deformation modeling is therefore essential for achieving high-precision and reliable robotic performance [13]. However, predicting deformation in robots is inherently complex, as the deformation field varies with both the geometric configuration and spatial posture [14]. In practical design processes, thousands of simulations are often required to evaluate candidate designs, configurations, and loading conditions. Finite element analysis (FEA) can provide accurate deformation estimates but incurs prohibitive computational costs, making it unsuitable for iterative or large-scale design optimization. Consequently, there is a pressing need for fast and accurate deformation field prediction methods that can efficiently support the design and optimization of robotic systems [15].

Existing approaches for robotic deformation field modeling and prediction can generally be categorized into mechanics-based methods and data-driven methods. Mechanics-based methods explicitly construct physical or analytical models of the robot structure to estimate deformation under external loads. Representative techniques include the finite element method (FEM) [16,17], the matrix structural analysis

* Corresponding author.
E-mail address: y.jin@qub.ac.uk (Y. Jin).

https://doi.org/10.1016/j.aei.2026.104600
Received 7 January 2026; Received in revised form 21 February 2026; Accepted 11 March 2026
Available online 18 March 2026

(MSA) [18,19], and the virtual joint method (VJM) [20,21]. FEM provides high-fidelity results by discretizing the robot structure into finite elements and solving the governing equilibrium equations. It can accurately capture stress and strain distributions but requires substantial computational resources, especially when analyzing different postures and geometric configurations [22]. MSA simplifies the structure into beam and joint elements, improving computational efficiency at the cost of accuracy and flexibility [23]. The VJM further abstracts the structure by modeling compliance through virtual springs at the joints, enabling analytical formulation of end-effector deflection [24]. However, these simplified methods rely on linear elasticity assumptions and predefined structural parameters, making them unsuitable for predicting full-field deformations across varying geometries and loading conditions [22].

In contrast, data-driven approaches bypass explicit structural modeling by learning a direct mapping from inputs such as external loads or robot postures to the resulting deformation field [25]. Early efforts based on machine learning employed surrogate models such as response surface methods [26] and Gaussian process regression [27,28] to approximate the deformation or error response over design spaces. These methods can accelerate optimization but are constrained by their dependence on parameterized design variables and cannot generalize to new geometries. Recent advances in deep learning have introduced neural networks capable of learning complex nonlinear mappings from load and pose parameters to deformation or stiffness outputs [15,29–31]. Although these models achieve faster predictions than mechanics-based methods, most are trained for fixed geometric configurations or specific postures and often predict deformation only at discrete points rather than across the continuous field.

Recently, neural operators have emerged as a powerful paradigm for learning mappings between continuous physical fields, enabling end-to-end field-to-field prediction [32]. Unlike traditional neural networks that approximate discrete input–output relationships, neural operators learn the underlying functional mapping that governs system behavior, thereby allowing inference at arbitrary spatial resolutions. Representative models such as DeepONet [33] and the Fourier Neural Operator (FNO) [34] have demonstrated remarkable success in solving partial differential equations and predicting complex physical phenomena, including fluid flow solid mechanics, and heat transfer [35–39]. These approaches provide a promising foundation for predicting deformation fields in robotic structures, where the mapping from geometry and external load to the deformation field can naturally be formulated as a function-to-function problem.

However, most existing neural operator frameworks are designed for fixed spatial domains, where the geometric configuration remains unchanged across training and inference. This limitation constrains their ability to generalize to systems with varying geometries or boundary conditions. To address this issue, several recent studies have proposed theoretically grounded extensions such as diffeomorphic neural operators [40,41], and transformer-based operator networks [42–44], which incorporate spatial coordinate transformations or attention-based representations to handle geometric variations. These methods have achieved progress in domains such as aerodynamic flow prediction and manufacturing deformation modeling, where the geometry varies moderately within a constrained shape family.

Despite these advances, robotic deformation field prediction remains particularly challenging. Unlike components with limited geometric variability, robots exhibit large-scale geometric changes across different link geometries, postures and configurations, resulting in deformation fields on varying domains and a wide workspace. Moreover, the coupling between joint kinematics, gravity, and external loads introduces highly nonlinear and configuration-dependent deformation behavior. Consequently, developing a neural operator that can generalize deformation field prediction across diverse robot geometries, postures, and loading conditions remains an open challenge.

To address these challenges, Robot-NO, a novel neural operator framework designed for fast and generalizable prediction of robotic deformation fields across varying geometries, postures, and loading conditions, is proposed in this paper. Note that the geometry herein is referred to as the joint and link dimensions rather than the robotic architecture. The proposed method integrates geometric and load information through a geometric–load graph and employs a Graph Neural Network (GNN) to extract a unified structural representation in a low-dimensional embedding space. A query-based decoder is then used to predict continuous deformation fields at arbitrary spatial locations, enabling resolution-independent inference, as shown in Fig. 1. This formulation allows Robot-NO to overcome the limitations of fixed-domain neural operators and conventional data-driven models, achieving both high accuracy and strong generalization across the configuration space of a robot. Through experiments on diverse robot structures and working conditions, we demonstrate that Robot-NO delivers acceptable accuracy while outperforming finite element analysis by four orders of magnitude in computational efficiency. This study provides an effective and scalable deformation prediction approach that supports real-time and large-scale robotic design and optimization.

2. Problem formulation

The deformation field considered in this study is governed by static structural mechanics, where the robot experiences elastic deformation under the combined influence of gravity and external concentrated loads. For a robot structure with varying geometries and postures, the resulting deformation depends on the spatial distribution of material properties, the gravitational field defined by the current configuration, and the external forces applied at specific nodes or surfaces.

Let the robot structure occupy a spatial domain $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$. The deformation field is described by the displacement vector $u(x) \in \mathbb{R}^3$ at spatial coordinate $x \in \Omega$. Under the assumption of linear elasticity, the static deformation obeys the equilibrium equation,

$$\nabla \cdot \sigma(x) + f(x) = 0 \text{ in } \Omega, \tag{1}$$

where $\sigma(x)$ is the Cauchy stress tensor and $f(x)$ represents the body force field. In this work, $f(x)$ includes the gravity-induced force determined by the robot posture. The stress tensor is related to the strain tensor $\varepsilon(x)$ through the constitutive equation,

$$\sigma(x) = \mathbb{C}(x)\,\varepsilon(x), \tag{2}$$

where $\mathbb{C}(x)$ is the elasticity tensor. The strain is defined as,

$$\varepsilon(x) = \frac{1}{2}\left(\nabla u(x) + \nabla u(x)^T\right). \tag{3}$$

Boundary conditions specify the mechanical constraints on the robot structure. In robotic systems, boundary constraints typically arise at the robot base, mounting surfaces, or joint interfaces. These constraints are commonly modeled as fixed boundary conditions, where the displacement at the constrained locations is set to zero. Let $\Gamma_D \subseteq \partial\Omega$ denote the set of nodes under fixed constraints. The Dirichlet boundary condition can be written as,

$$u(x) = 0 \text{ on } \Gamma_D. \tag{4}$$

External concentrated loads are applied on nodes or surfaces such as end-effectors, tools, or structural attachment points. Let $\Gamma_N \subseteq \partial\Omega$ denote the load-bearing boundary, where the Neumann condition is expressed as,

$$\sigma(x)n(x) = p(x) \text{ on } \Gamma_N, \tag{5}$$

where $n(x)$ is the outward normal vector and $p(x)$ the applied load.

For robots, both the domain Ω and the body force field $f(x)$ vary with the joint configuration and geometric parameters. The deformation field can therefore be expressed as a functional mapping,

$$u(x) = \mathscr{F}(\Omega,\ f(x),\ p(x)\,), \tag{6}$$

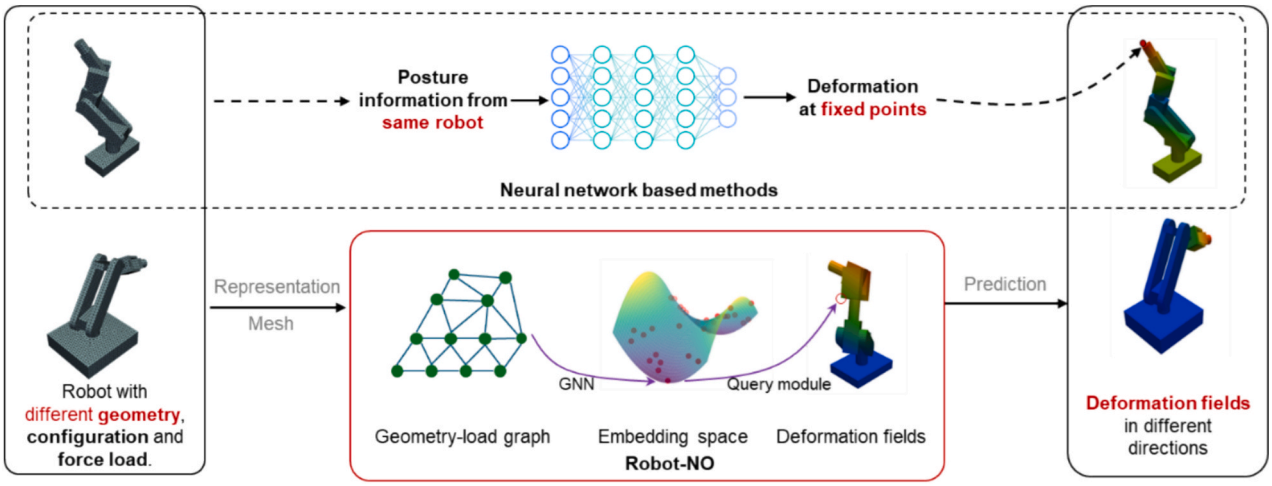


Fig. 1. Overview of Robot-NO for deformation fields prediction.

where $\mathscr{T}$ denotes the solution operator of the static elasticity equations. More generally, the relationship between geometry, gravity, external loads, and the resulting displacement field can be written in operator form as,

$$\mathscr{T} : (\Omega, f(x), p(x)) \rightarrow u(x), \tag{7}$$

Together with the body force field $f(x)$ induced by gravity and configuration-dependent orientation, these boundary conditions fully describe the static equilibrium of the robot under given posture and loading settings.

Efficiently approximating this operator $\mathscr{T}$ across diverse robot geometries, postures, and loading conditions is the objective of this work. The proposed neural operator framework aims to learn a generalizable and accurate approximation of $\mathscr{T}$, enabling fast prediction of continuous deformation fields in robotic design and optimization.

3. Robot-NO framework

This section presents the proposed Robot-NO framework for generalizable deformation field prediction across diverse robotic geometries, postures, and loading conditions. Building upon the operator formulation introduced in the previous section, the proposed framework learns an approximation of the deformation operator by combining geometric representation, load encoding, and continuous field decoding. The method consists of three main components. First, the robot geometry and external load information are converted into a geometry–load graph that characterizes the spatial domain and force distribution. Second, a GNN extracts a unified structural representation in a low-dimensional embedding space from this graph, capturing the relationship between geometry, postures, and applied loads. Finally, a query-based decoder receives arbitrary spatial coordinates together with the global structural representation and predicts the corresponding deformation values, enabling continuous and resolution-independent field inference, as shown in Fig. 2. The details of each component are described in the following subsections.

3.1. Geometric–load graph construction

To represent the robot geometry and loading conditions in a form suitable for operator learning, the spatial domain is discretized into a mesh and converted into a geometry–load graph. Let the robot geometry be sampled into a set of points,

$$V = \{\boldsymbol{x}_i \in \mathbb{R}^3 | i = 1, \cdots, N\}, \tag{8}$$

where each $\boldsymbol{x}_i = (x, y, z)$ denotes a node in the discretized domain. Connectivity between neighboring points is defined based on the mesh structure, forming an edge set,

$$E = \{(i,j) | \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are adjacent}\}. \tag{9}$$

The geometric domain is therefore represented as an undirected graph,

$$G = (V, E). \tag{10}$$

Each node $\boldsymbol{x}_i$ is associated with a feature vector h_i that encodes both geometric information and external load conditions. The geometric attribute is simply the node coordinate,

$$\boldsymbol{g}_i = \boldsymbol{x}_i. \tag{11}$$

External concentrated loads are expressed as three-dimensional

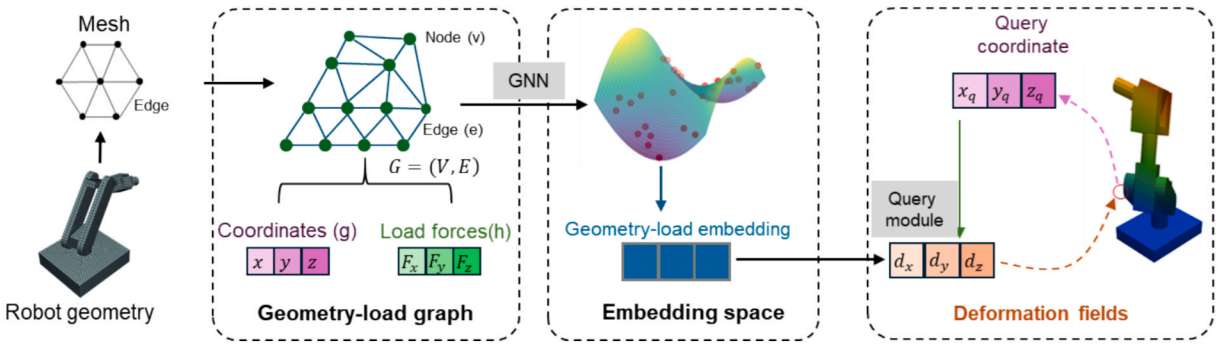


Fig. 2. Architecture of Robot-NO.

vectors,

$$\boldsymbol{p_i} = \left(p_i^x, p_i^y, p_i^z\right), \tag{12}$$

where p_i^x, p_i^y, and p_i^z denote the load components along the x, y, and z axes of the Euclidean coordinate frame. For nodes without external loads, $\boldsymbol{p_i} = (0, 0, 0)$.

The node feature is then defined as,

$$\boldsymbol{h_i} = [\boldsymbol{g_i}, \boldsymbol{p_i}]. \tag{13}$$

Gravity is treated as a uniform body force whose direction is determined entirely by the robot posture. Since geometry implicitly determines both the spatial domain and the orientation-dependent contribution of gravity. Thus, the posture is encoded in the spatial coordinates $\{\boldsymbol{x_i}\}$, the gravitational effect does not require explicit node-level encoding.

The geometric–load graph provides a unified representation of the robot's shape, posture, and external concentrated forces. This graph serves as the input to the subsequent GNN. For clarity, the complete input representation can be written as,

$$G = (V, E, H), \tag{14}$$

where $H = \{\boldsymbol{h_i}\}_{i=1}^{N}$ denotes the set of node features containing both geometric coordinates and load vectors.

3.2. Graph neural network encoding and embedding

The geometric–load graph provides node features and connectivity information that describe the robot geometry and external forces. To extract a unified global representation from this graph, a GNN is employed to aggregate and transform node-level information. This unified representation captures the essential structural and functional relationships across the geometry, posture, and external loading condition, which is important for efficient learning and generalization, and enabling the model to effectively approximate the underlying deformation operator and predict continuous deformation fields across diverse configurations.

Let h_i^0 denote the initial feature of node i, consisting of its geometric coordinate and load vector. The connectivity encoded in the edge set E determines the neighborhood structure over which message passing is performed. For each node i, messages are exchanged only with nodes connected by edges $(i,j) \in E$. At the k-th layer of the GNN, the message aggregation and node feature update are given by,

$$\boldsymbol{m}_i^k = \sum_{j\,:\,(ij)\in E} \phi_m\left(\boldsymbol{h}_i^k, \boldsymbol{h}_j^k\right), \tag{15}$$

$$\boldsymbol{h}_i^{k+1} = \phi_h\left(\boldsymbol{h}_i^k, \boldsymbol{m}_i^k\right), \tag{16}$$

where ϕ_m and ϕ_h are learnable message and update functions implemented with multilayer perceptions.

Through this edge-based aggregation, each node iteratively incorporates information from its neighbors, enabling the network to capture the geometric relationships and local load transfer patterns induced by the structure.

After K rounds of message passing, the node features $\left\{\boldsymbol{h}_i^K\right\}$ encode both local and nonlocal structural information. To obtain a compact latent embedding $\boldsymbol{z}$ that summarizes the global characteristics of the geometry and loading conditions, an average pooling operator is applied across all nodes,

$$\boldsymbol{z} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{h}_i^K. \tag{17}$$

The resulting latent embedding vector $\boldsymbol{z} \in \mathbb{R}^d$ captures essential attributes of the robot structure, including, geometry, posture, and concentrated loads. This global representation is used as the structural input to the query-based neural operator decoder.

The graph neural encoding provides two key advantages for deformation field prediction. First, through iterative aggregation of node features, the GNN effectively extracts the structural patterns embedded in the geometric–load graph, including the relationships between geometry, posture, and applied forces. Second, the global latent vector produced by pooling condenses this information into a compact representation that summarizes the essential mechanical characteristics of the structure. This latent representation serves as the foundation for predicting continuous deformation fields in the subsequent neural operator decoder.

3.3. Query-based neural operator decoder

The global latent vector $\boldsymbol{z}$ extracted from the geometric–load graph encodes the structural characteristics of the robot under a specific geometry, posture, and loading condition. To obtain the deformation field at arbitrary spatial locations, the proposed framework employs a query-based decoder that maps any query coordinate to its corresponding displacement value. Let $\boldsymbol{x}_q \in \mathbb{R}^3$ denote a query point in the geometric domain. The decoder receives both the query coordinate and the global representation as inputs, forming the feature,$\eta_q = \left[x_q, \boldsymbol{z}\right]$.

A multilayer perception is then used to approximate the operator evaluation at this coordinate,

$$\widehat{u}\left(\boldsymbol{x}_q\right) = \phi_{\mathrm{dec}}\left(\eta_q\right), \tag{18}$$

where $\widehat{u}\left(\boldsymbol{x}_q\right) = \left[\widehat{u}_x, \widehat{u}_y, \widehat{u}_z\right] \in \mathbb{R}^3$ represents the predicted displacement components in the x, y, and z directions.

This formulation treats deformation prediction as a function evaluation problem defined over a continuous domain. Since the decoder operates on arbitrary query coordinates rather than pre-defined mesh nodes, it enables resolution-independent prediction, allowing the model to evaluate the deformation field at any spatial resolution or sampling density. This contrasts with traditional data-driven regression models that are tied to fixed discretization and can only output deformation values at predetermined nodes.

By conditioning the prediction on the global representation $\boldsymbol{z}$, the decoder effectively captures the influence of geometry, gravity orientation, and external loads on the deformation response. The query-based structure allows the network to approximate the underlying deformation operator,

$$\widehat{\mathscr{F}_\theta} : (\Omega,\ p) \mapsto u\left(\boldsymbol{x}_q\right), \tag{19}$$

where θ denotes the trainable parameters of the neural network. This formulation enables the model to learn a continuous mapping from spatial coordinates to displacement values and to generalize deformation prediction across different geometries and poses while producing full-field outputs suitable for design and optimization tasks.

The neural operator $\widehat{\mathscr{F}_\theta}$ is trained to approximate the true deformation operator by minimizing the discrepancy between predicted and reference displacement fields across all training samples. For each sample n, a set of query points $\left\{\boldsymbol{x}_q^n\right\}$ is selected from its geometric domain, and the predicted displacement $\widehat{u}\left(\boldsymbol{x}_q^n\right)$ is compared with the ground-truth displacement $u\left(\boldsymbol{x}_q^n\right)$. The overall training objective is the mean squared error computed across all samples and all query points,

$$\mathscr{L} = \frac{1}{\sum_{n=1}^{N} Q_n}\sum_{n=1}^{N}\sum_{q=1}^{Q_n} w\left(\widehat{u}\left(\boldsymbol{x}_q^n\right)\right)\left\|\widehat{u}\left(\boldsymbol{x}_q^n\right) - u\left(\boldsymbol{x}_q^n\right)\right\|^2, \tag{20}$$

where N is the total number of training samples and Q_n is the number of

query points for sample n. w is a weight function used to adaptively scale the squared error according to the magnitude of the ground-truth target, so that regions with larger physical response receive higher training emphasis. This objective enforces pointwise accuracy of the predicted deformation field over diverse geometries, postures, and loading conditions. By minimizing $\mathscr{L}$, the model learns a generalizable neural operator capable of predicting continuous deformation fields. The overall formulation allows the network to approximate the underlying static deformation operator across varying geometries, postures, and loading conditions.

4. Implementation details and experimental results

This section describes the data generation process and the implementation details of the proposed framework. The robot is parameterized at the joint level and assembled automatically, allowing the creation of diverse instances with varying geometries and postures. These configurations are used to generate deformation samples under different gravity orientations and external loads. The model setup, training procedure, and inference settings are then introduced. Finally, the prediction performance and comparisons with baseline methods are presented.

4.1. Dataset generation details

To evaluate the proposed method, a serial robotic configuration was selected as the test platform. The robot consists of seven structural components, including the base, link1, link21, link22, link3, link4, and the end effector. Each component is parameterized to allow variations in its geometric dimensions, as illustrated in Fig. 3. The specific parameter ranges for each joint and link are summarized in Table 1. By sampling these parameters, a large number of robot geometries can be generated.

The connections between link components are modeled using standardized shaft–hole interfaces, i.e. revolute joints. Similarly, link21 and link22 are connected to link1 via two parallel shaft–hole pairs. This consistent interface design allows the automatic construction of robot forward kinematic model by extracting D-H parameters. To obtain deformation samples under different robot configurations, each joint is associated with a local base frame (B_x, B_y, B_z) and a following frame (F_x, F_y, F_z), as shown in Fig. 3. The local base frame of the next link is aligned with the following frame of the previous link, enabling sequential assembly of a robot. Then, by randomly sampling the rotation angles of the five actuated joints within the limits shown in Table 2, robots with diverse spatial postures are generated. Representative examples of sampled configurations are shown in Fig. 4.

Table 1
Ranges of allowable geometry parameters (unit: mm).

Component	Parameter	Range (mm)	Component	Parameter	Range (mm)
Base	l_b	130 ~ 250	Link 1	l_1	100 ~ 140
	w_b	100 ~ 250		w_1	40 ~ 60
	h_b	40 ~ 60		h_1	40 ~ 60
	r_{1b}	20 ~ 30		l_{1r}	$20\sim\left(\frac{l_1}{2}-10\right)$
	h_{b1}	20 ~ 40		h_{1b}	20 ~ 40
				r_{12}	20 ~ 40
Link 21	l_2	200 ~ 300	Link 3	l_3	100 ~ 160
Link 22	w_2	20 ~ 30		w_3	40 ~ 60
	h_2	50 ~ 70		h_3	20 ~ 80
				r_3	20 ~ 35
				l_{34}	20 ~ 60
Link 4	l_4	70 ~ 100	End effector	l_e	(80 ~ 140)* (4/10 ~ 7/10)
	w_4	60 ~ 100		h_e	30 ~ 80
	h_4	50 ~ 80		w_e	$30\sim(w_4-16)$
	l_{43}	5 ~ 20		w_{4e}	$(w_4-w_e)/2$
	r_4	10 ~ 20		r_e	$\min(w_e, h_r)/2$
	l_{slot}	$50\sim(l_4-8)$		l_{er}	20 ~ 40
	l_{4r}	$l_{slot}*\frac{2}{3}$			

External forces are applied at the designated force application point on the end effector. The load is represented as a three-dimensional vector consisting of the force components in the X, Y, and Z directions. The range of each force component ranges from −20 N to 20 N.

The robot components are assumed to be made of aluminum alloy, with material properties defined by an elastic modulus of 71.7 GPa, a density of 2700 kg/m^3, a Poisson's ratio 0.33 and a gravitational

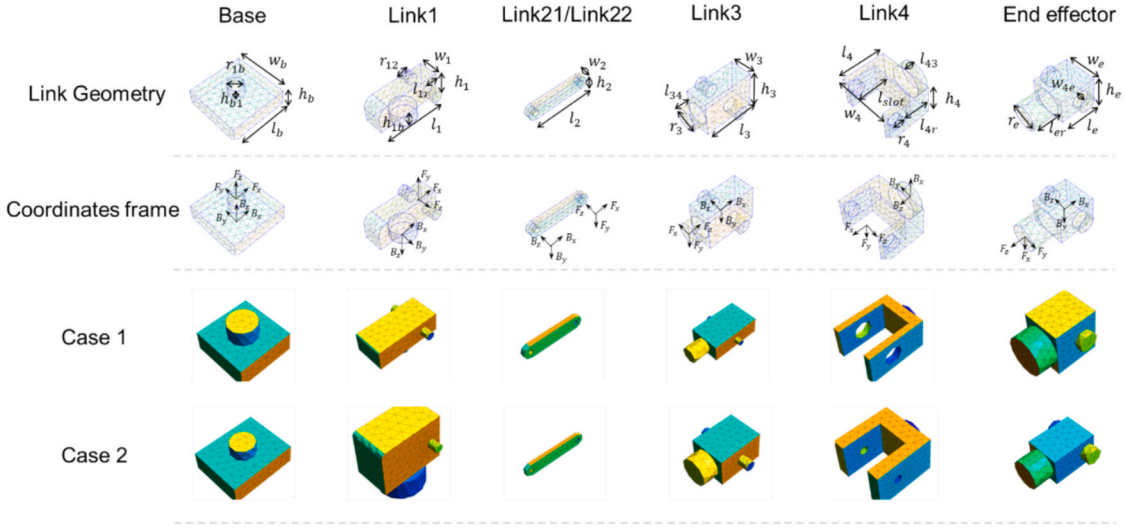


Fig. 3. Schematic representation of the parameterized robot joints.

Table 2
Ranges of allowable joint rotation angles.

Joint	Joint 1	Joint 21	Joint 22	Joint 3	Joint 4	Joint 5
Range	$0^{\circ} \sim 360^{\circ}$	$30^{\circ} \sim 150^{\circ}$	$30^{\circ} \sim 150^{\circ}$	$-90^{\circ} \sim 90^{\circ}$	$0^{\circ} \sim 120^{\circ}$	$-90^{\circ} \sim 90^{\circ}$

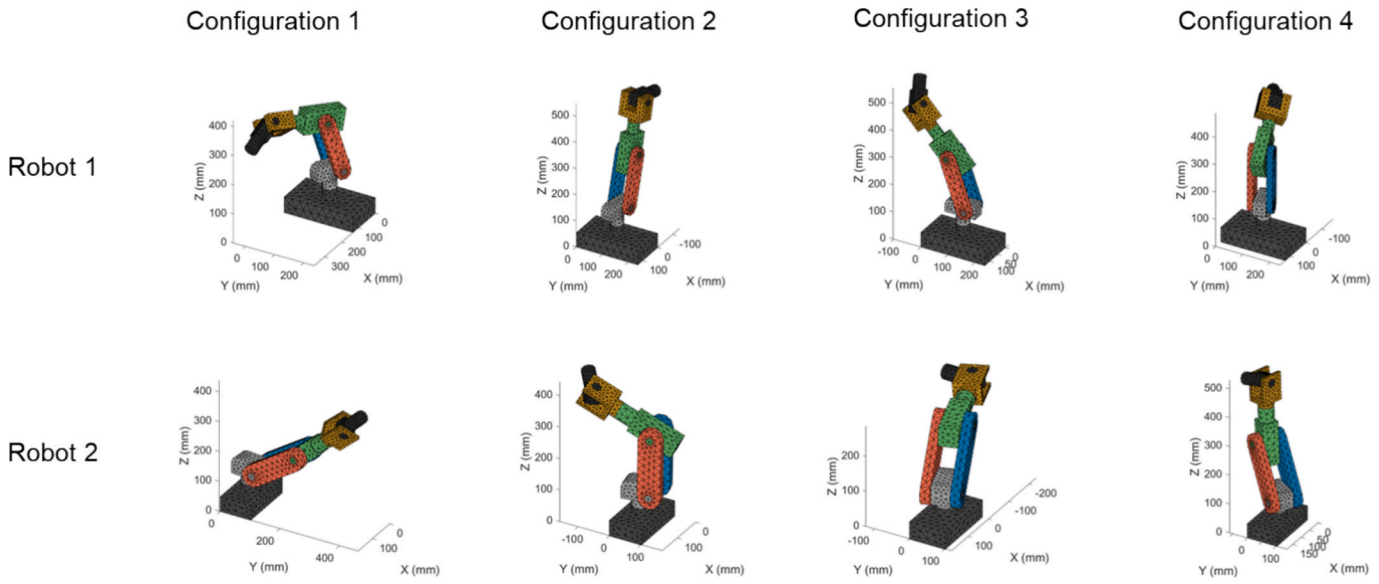


Fig. 4. Representative robot configurations generated by sampling joint rotation angles.

acceleration of 9.8 m/s^2. All robot assemblies and static deformation simulations were performed in Abaqus for generating data for training and benchmarking. An automatic meshing procedure with tetrahedral elements of maximum size of 10 mm is used for robot geometry discretization, resulting in varying numbers of nodes and elements across the dataset. Combining geometric variations, random joint angles, and variable external loads, a total of 9000 robot instances were generated, each corresponding to a unique combination of geometry, posture, and loading condition.

The geometric–load graph used as the operator input is constructed from the surface mesh extracted from the FEM software, where the mesh nodes and their connectivity provide the graph nodes and edges. While FEA relies on elements for volumetric integration and solving equilibrium equations, Robot-NO approximates the solution operator by learning the mapping between inputs (geometry-load) and output (deformation) field. The node-edge graph is adopted because it provides a simple representation and precise characterization of geometric connectivity, while leveraging the inherent ability of GNNs to process complex structural topologies of meshes. For each robot instance, all mesh nodes are used as query points for supervising the continuous deformation field.

4.2. Implementation details of Robot-NO

The proposed Robot-NO framework is implemented using PyTorch and trained on an NVIDIA TITAN RTX GPU. The model architecture comprises a GNN-based encoder and a multi-head query decoder, totaling 310,023 trainable parameters. Specifically, the GNN-based encoder utilizes four Graph Convolutional Network (GCN) layers with successive output dimensions of 64, 128, and 256 to aggregate geometric and loading features from the mesh topology. A global pooling layer then condenses these features into a latent structural embedding. This embedding is processed through a linear layer to form a 128-dimensional unified global representation.

For the decoding stage, a query-based module maps this global representation along with arbitrary spatial coordinates to the predicted deformation field. The decoder architecture employs three parallel output heads for the three displacement components, each consisting of two linear layers with a hidden dimension of 128 and Parametric ReLU (PReLU) activation functions to capture complex nonlinear structural responses.

The training process utilizes the Adam optimizer with a weight decay of 0.00001 to minimize the mean squared error (MSE) loss between the predicted and ground-truth displacement fields. The model is trained with an initial learning rate of 0.0003 and a batch size of 128. All linear layers are initialized using the Xavier normal distribution to facilitate stable convergence.

4.3. Experimental results

4.3.1. Case Studies: Full-field deformation prediction

To evaluate the prediction performance of the proposed operator, we first examine two representative robot cases with different geometries, postures, and external loads. For each case, the deformation fields along the X, Y, and Z directions are visualized and compared with the reference fields obtained from Abaqus simulation. The corresponding error distributions and slice-based internal views are also provided to illustrate the accuracy of the predicted full-field deformation.

For Case 1, the predicted deformation fields closely match the ground-truth distributions in all three directions, as illustrated in Fig. 5. The spatial patterns, deformation gradients, and relative magnitudes are well reproduced, indicating that the model effectively captures both the global deformation trend and the fine-scale local variations. In particular, the predicted fields exhibit smooth and continuous deformation profiles across the entire structure, consistent with the physical characteristics of elastic deformation. The model also accurately captures localized high-gradient regions, such as the joint interfaces and connection areas where deformation changes rapidly due to geometric discontinuities or concentrated stiffness variations. The absolute error remains small throughout the structure, and the slice-based

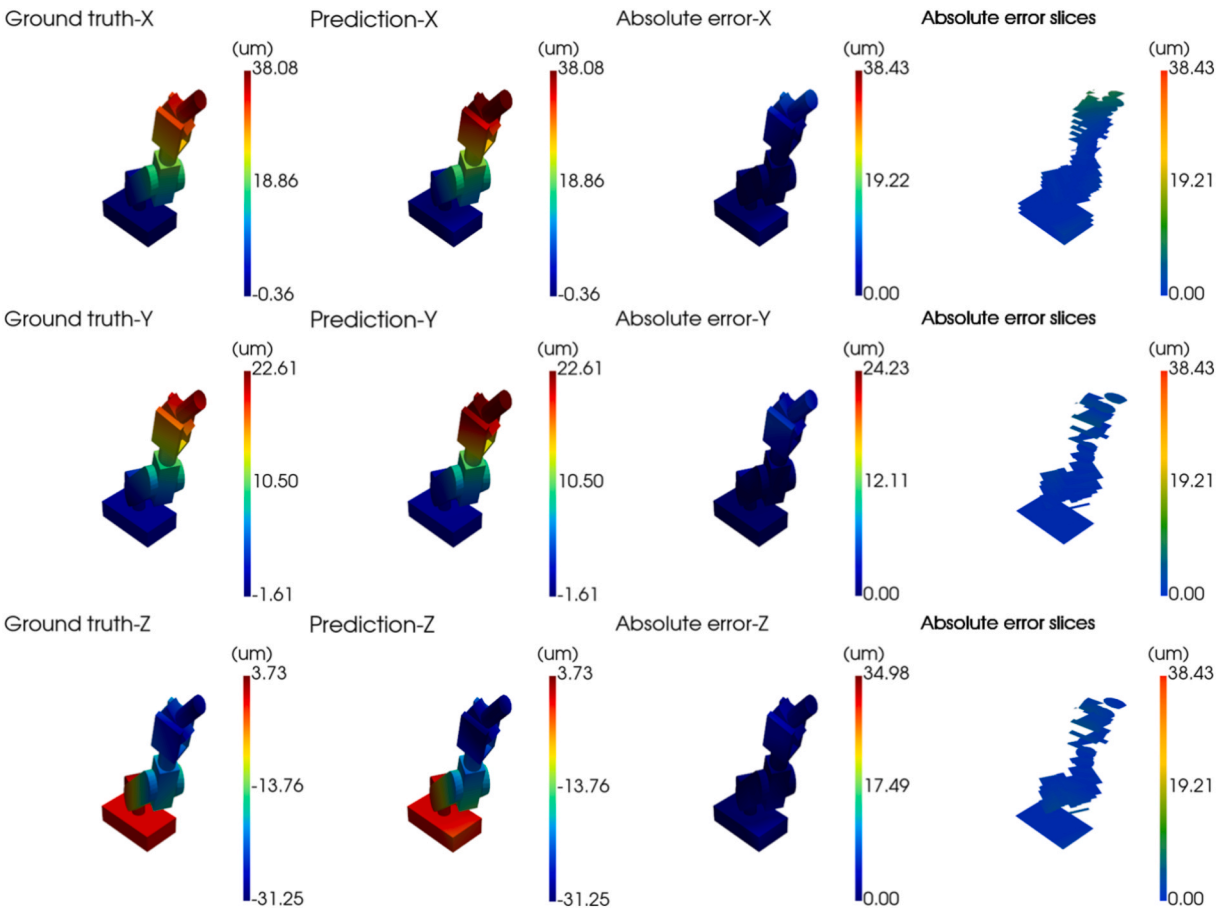


Fig. 5. Full-field deformation prediction for Case 1.

visualizations further confirm that the prediction accuracy extends from the surface mesh to the interior of the geometry. These results demonstrate that the learned operator not only preserves field continuity but also delivers reliable predictions across both smooth regions and mechanically sensitive areas with sharp deformation gradients.

Case 2 shows similar consistency, as shown in Fig. 6. The predicted fields in the X, Y, and Z directions again exhibit agreement with the reference results. Internal slice views also show low error levels, confirming the reliability of the model across different loading and configuration settings.

Quantitative results for the two cases are summarized in Table 3. Case 1 achieves mean absolute errors (MAE) of 0.95 μm, 0.93 μm and 1.43 μm in the X, Y, and Z directions, respectively. Case 2 yields errors of 0.76 μm, 1.19 μm and 0.82 μm in the X, Y, and Z directions, respectively. These examples demonstrate that the proposed Robot-NO model provides highly accurate full-field deformation predictions across different robot geometries and postures.

4.3.2. *Effect of configuration variations*

To evaluate the ability of the proposed method to generalize across different joint configurations, a set of robot postures was generated for a fixed geometric design with 5 N force in each direction. Each configuration corresponds to a unique combination of joint angles, resulting in distinct spatial poses and gravity orientations. The distribution of end-effector positions together with the associated prediction MAE is shown in Figs. 7–9, covering a broad region of the workspace. In these figures, each point represents the end-effector location of a particular configuration, with its color indicating the MAE for that configuration. Four representative configurations with substantial differences in posture are further visualized to illustrate the detailed prediction results for the X, Y, and Z displacement components.

Across all three directions, the predicted deformation fields closely match the reference solutions, and the spatial deformation patterns remain consistent over the full range of sampled configurations. The MAE stays within the micrometer range for all cases, demonstrating that the model accurately captures configuration-dependent deformation behaviors. Overall, the results indicate that Robot-NO maintains high prediction accuracy under significant variations in joint angles and workspace locations. The method generalizes effectively to a wide variety of configurations, confirming that the learned neural operator successfully models the intrinsic relationship between posture and deformation.

In addition to examining representative configurations, the prediction errors were further analyzed across the full range of the six joint angles. Fig. 10 shows the MAE distribution for the X, Y, and Z deformation along the joint angles. Each point corresponds to a single robot configuration, and the color represents its prediction MAE. The error distributions remain smooth and uniformly low across all angle intervals. This indicates that the model does not exhibit sensitivity to any specific joint rotation and that local changes in a single joint angle do not lead to systematic error variations and further validates the robustness of Robot-NO across the entire configuration domain.

Table 4 shows the overall MAE and Root Mean Square Error (RMSE) for the X, Y, and Z displacement components. The MAE are 0.49 μm, 0.38 μm, and 0.51 μm, respectively, while the corresponding RMSE values remain below 0.60 μm in all directions. To further quantify the prediction stability, a violin plot of the MAE across all test samples is presented in Fig. 10(d). The distribution reveals that for all three displacement components, the errors are highly concentrated in the sub-micrometer range, with the median values consistently below 0.6 μm.

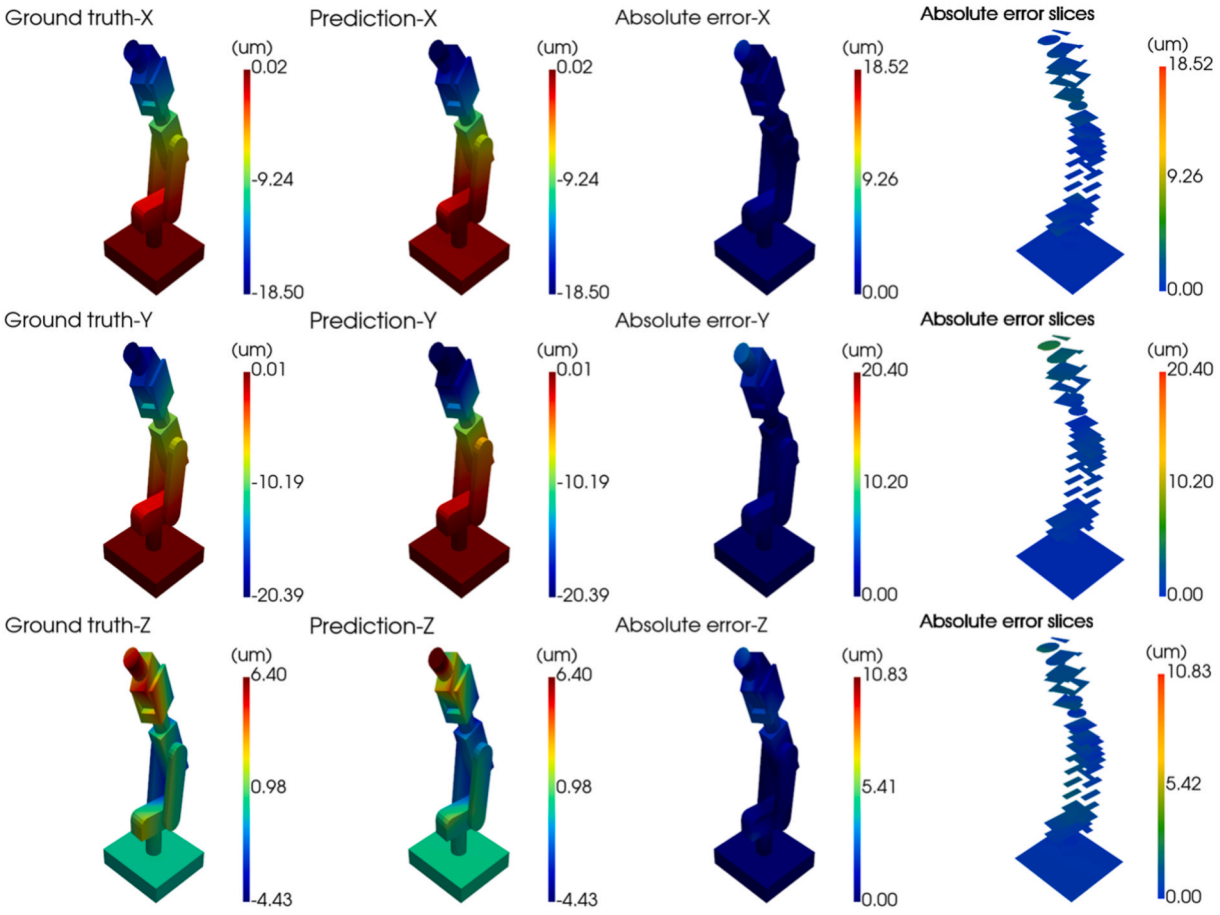


Fig. 6. Full-field deformation prediction for Case 2.

Table 3
MAE for Case 1 and Case 2 in the *X*, *Y*, and *Z* deformation components (units: μm).

Case	*X* MAE (μm)	*Y* MAE (μm)	*Z* MAE (μm)
1	0.95	0.93	1.40
2	0.76	1.19	0.82

These error levels across thousands of distinct joint configurations confirm that Robot-NO provides stable and configuration-independent deformation prediction performance.

4.3.3. *Effect of force variations*

To validate the robustness of the proposed method under diverse loading conditions, the deformation prediction accuracy across a wide range of external forces applied at the end-effector was further evalu-

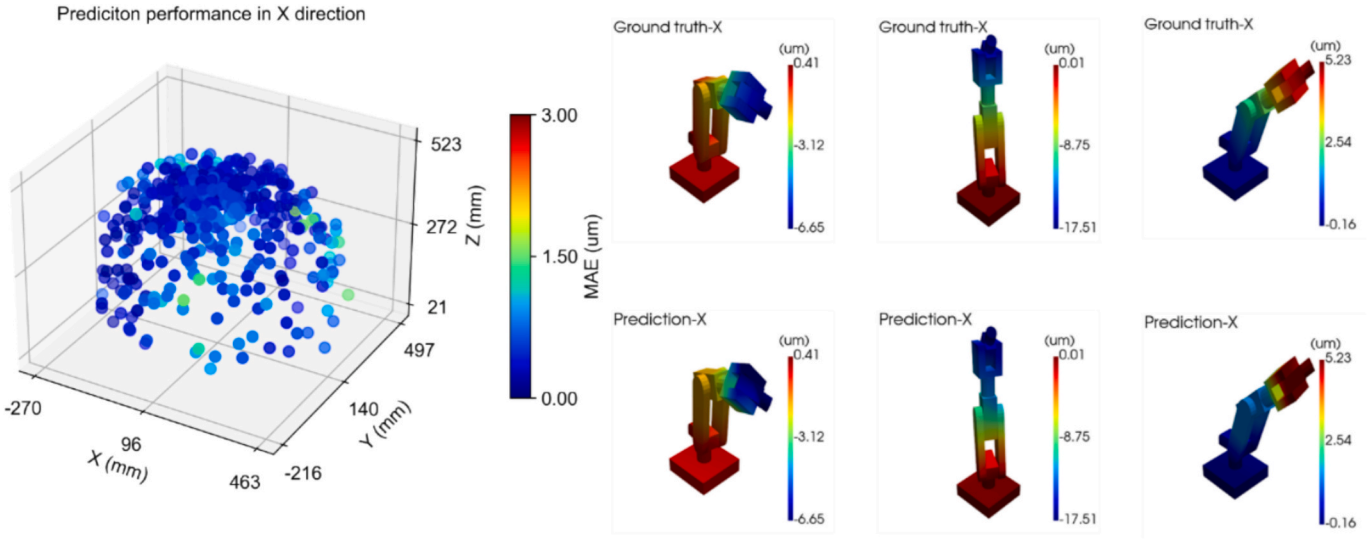


Fig. 7. MAE across different robot configurations for the predicted deformation in *X* direction.

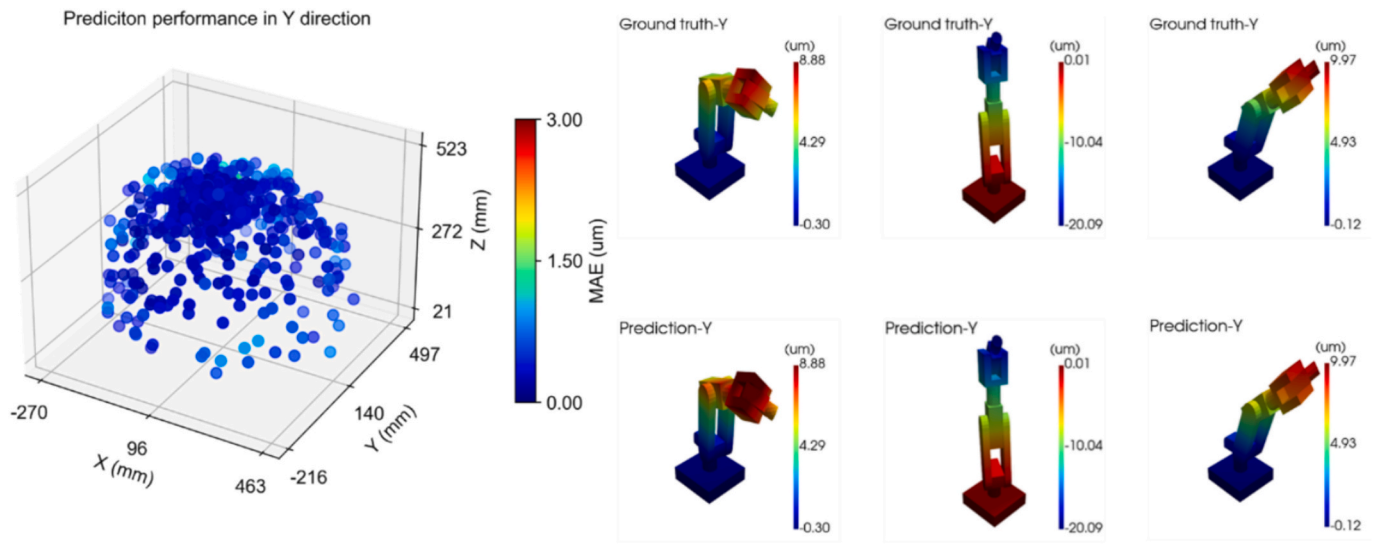


Fig. 8. MAE across different robot configurations for the predicted deformation in Y direction.

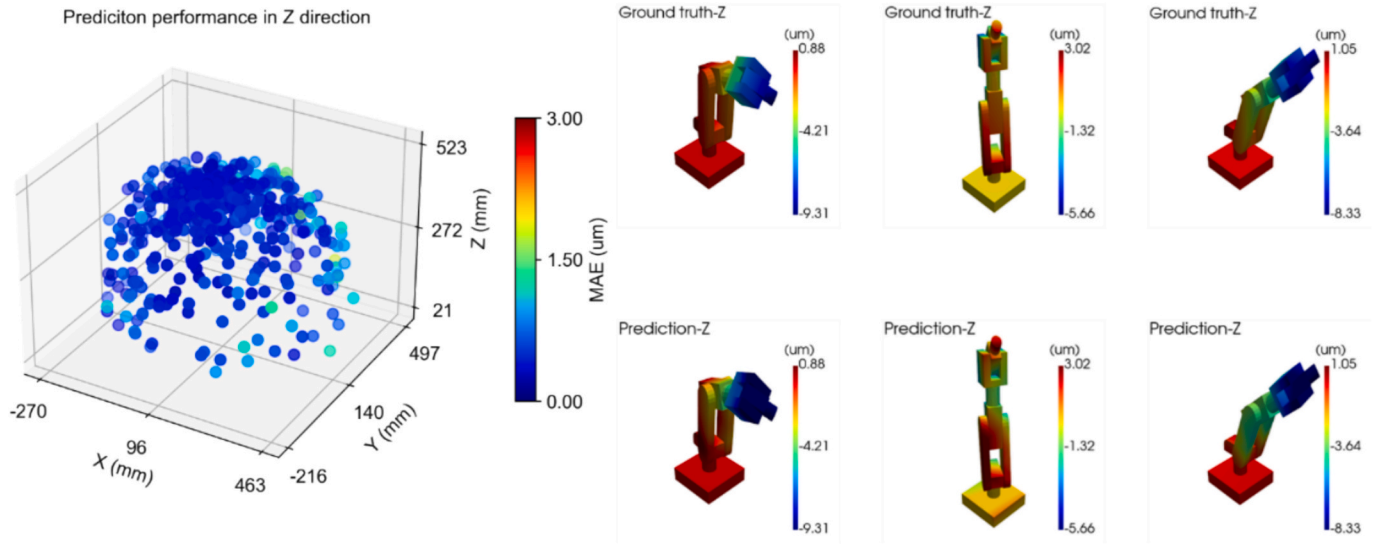


Fig. 9. MAE across different robot configurations for the predicted deformation in Z direction.

ated. Two robot configurations were selected, and for each configuration, 500 load vectors were generated by independently sampling the three force components within the predefined ranges. The prediction MAE for the X, Y, and Z deformation components under each load vector is visualized in Fig. 11, where the three axes correspond to the force components (F_x, F_y, F_z), and the color of each point denotes the resulting MAE.

It can be seen from the figure, across both robot configurations, the error distributions remain consistently low throughout the sampled force space. However, several load-dependent patterns can also be observed. For Pose 1, the MAE in the X direction shows a slight increase when the applied F_x component becomes large. A similar trend appears in Pose 2 for the Y direction, where larger F_y values correspond to mildly higher prediction errors. These phenomena are consistent with the fact that different robot postures exhibit different stiffness characteristics, and stronger directional loading may lead to larger or more nonlinear deformation responses, making the prediction task more challenging. Despite these localized increases, the errors remain within the micrometer range across all load combinations, indicating that the proposed operator maintains reliable accuracy even under force amplitudes that significantly vary in direction and magnitude. These results confirm that Robot-NO maintains stable and load-independent performance, enabling accurate deformation prediction under diverse and unpredictable external loading conditions.

The error statistics for the two sampled robot poses are summarized in Table 5. For Pose 1, the MAE values for the X, Y, and Z deformation components are 0.96 μm, 0.62 μm, and 0.56 μm, respectively, with corresponding RMSE values remaining below 1.10 μm. Pose 2 exhibits slightly lower errors overall, with MAE values of 0.67 μm, 0.62 μm, and 0.51 μm. These results indicate that, although loading-direction effects cause modest variations in prediction accuracy between poses, the model consistently maintains sub-micrometer to low-micrometer error levels in all directions.

4.3.4. *Effect of meshing size*

To evaluate the resolution independence and generalizability of Robot-NO across different discretization scales, an ablation study was conducted using varying mesh densities. While the model was originally trained on a dataset discretized at a 10 mm mesh size, its performance was tested across 30 distinct robot configurations using three different

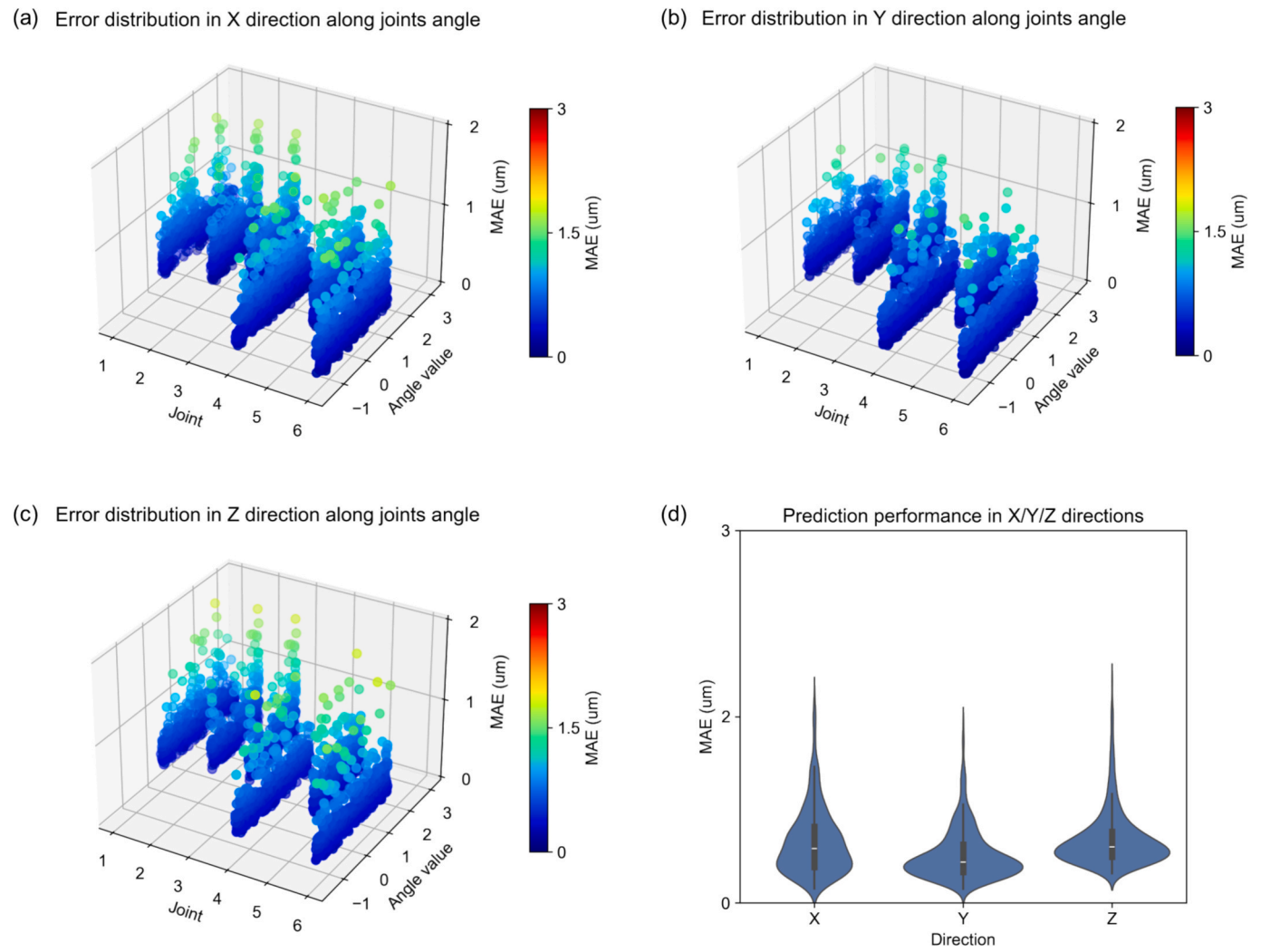


Fig. 10. (a–c) MAE distribution of predicted deformation in X, Y, and Z across full predefined ranges (in radians) of the six joint angles. (d) MAE distribution in X, Y, and Z directions.

Table 4
Overall MAE and RMSE of the predicted displacement components across all sampled robot configurations.

Directions	X	Y	Z
MAE (μm)	0.49	0.38	0.51
RMSE (μm)	0.56	0.43	0.56

mesh sizes: 10 mm, 8 mm, and 5 mm.

As shown in Table 6, the prediction accuracy remains stable across all tested resolutions. For instance, the MAE in the X direction only fluctuates slightly from 0.85 μm (at 10 mm) to 0.9 μm (at 8 mm) and 1.06 μm (at 5 mm). Similar trends are observed in the Y and Z components, where the error metrics remain within a narrow micrometer-level range (0.24 μm) despite the significant increase in node density at mesh scales.

This stability is primarily attributed to the query-based decoder formulation. Unlike traditional GNNs that are tied to specific nodal indices, Robot-NO treats deformation prediction as a continuous function evaluation problem. By mapping global structural embeddings to arbitrary query coordinates, the model effectively approximates the underlying physical operator rather than just the discrete nodal displacements. These results demonstrate that Robot-NO is highly robust to variations in mesh discretization, providing consistent and accurate full-field predictions regardless of the input mesh resolution

4.3.5. *Effect of geometries variations*

Robot structural deformation is strongly influenced by its geometric design, and a key requirement for operator-based prediction models is the ability to generalize across different robot geometries. To evaluate this capability, a dataset of 250 robot instances by randomly sampling geometric parameters together with corresponding joint configurations and external forces was generated. This setting provides a comprehensive evaluation scenario in which geometry, posture, and load jointly vary, allowing us to isolate and examine the influence of geometry on prediction accuracy.

Fig. 12 shows the MAE distributions for the X, Y, and Z deformation components with respect to each geometric parameter. The sampled points span the full predefined ranges of all parameters, indicating adequate geometric coverage. Across nearly all parameters, the prediction errors remain within the micrometer range, demonstrating robust performance under geometric variability.

Nevertheless, several parameters exhibit slightly larger spreads. For instance, increases in link lengths (e.g., l_2, l_3, l_4) or radii (e.g., r_{1b}) that lead to larger lever arms can introduce larger deformation behavior, which results in broader error distributions. These parameters directly affect the global stiffness distribution, leading to larger deformation magnitudes and stronger coupling between geometry, posture, and loading. As a result, small variations in these parameters can induce more pronounced changes in the deformation field, making accurate

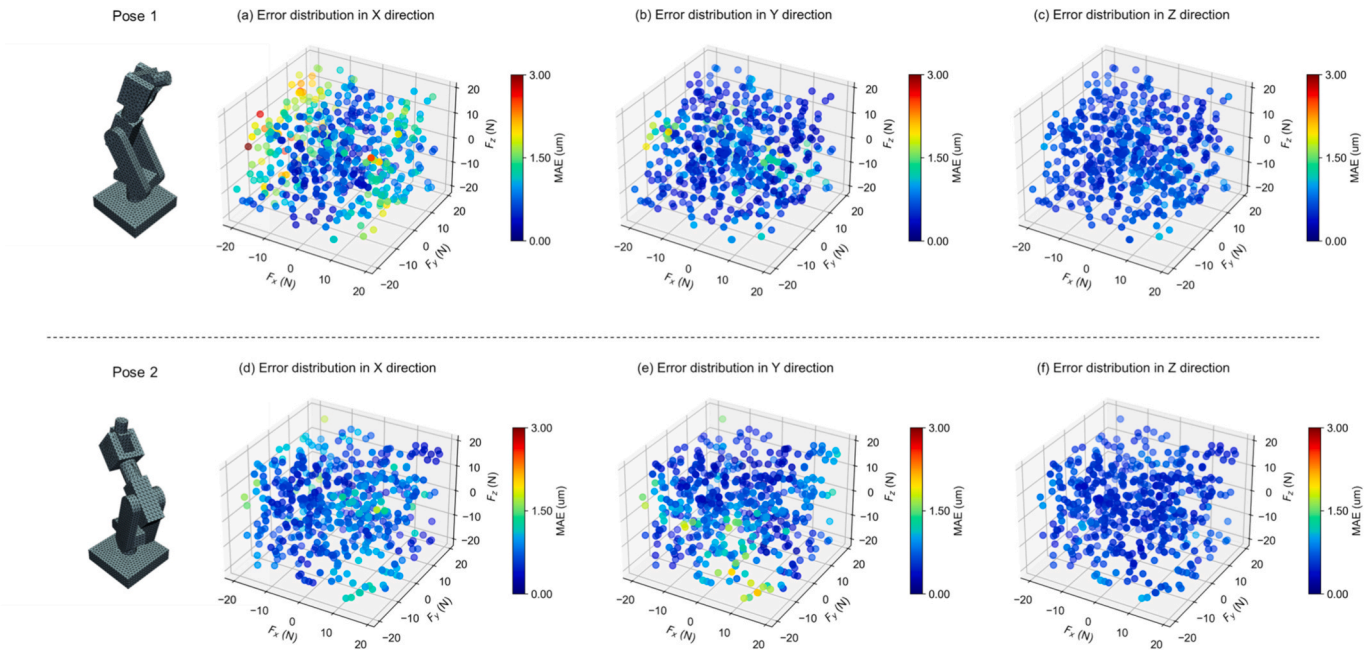


Fig. 11. Prediction MAE of the X, Y, and Z deformation components under randomly sampled external load combinations for two robot configurations.

Table 5
MAE and RMSE of the predicted displacement components under 500 randomly sampled external load combinations for two representative robot configurations.

Pose	Pose 1		Pose 2	
	MAE (μm)	RMSE (μm)	MAE (μm)	RMSE (μm)
X	0.96	1.10	0.67	0.73
Y	0.62	0.70	0.62	0.70
Z	0.56	0.57	0.51	0.53

Table 6
Prediction results under different mesh discretization sizes (10 mm, 8 mm, and 5 mm).

Direction	Metric	5 mm	8 mm	10 mm
X	MAE (μm)	1.06	0.90	0.85
	RMSE (μm)	1.65	1.44	1.29
Y	MAE (μm)	1.08	0.95	0.84
	RMSE (μm)	1.75	1.61	1.37
Z	MAE (μm)	0.92	0.83	0.73
	RMSE (μm)	1.45	1.34	1.13

prediction inherently more challenging.

In contrast, some geometric parameters (e.g., l_b, w_b, h_b) with more localized structural influence show consistently small errors with narrow variance. These parameters have limited impact on the global load transfer paths and overall structural compliance, resulting in deformation patterns that are easier for the model to capture. Importantly, despite these parameter-dependent variations, the overall error levels remain low across the entire geometric space.

To further quantify the geometric influence, Fig. 13 shows the Pearson correlation coefficients between each geometric parameter and the prediction errors. For all three deformation components, the correlations lie mostly within ± 0.1, indicating weak linear relationships and suggesting that no single parameter dominates the prediction error. A few parameters, however, show moderate correlations above 0.1. Notably, r_{1b}, which corresponds to a key structural radius in the first link connection, exhibits negative correlations in all three directions, reflecting its strong influence on local stiffness and the deformation pattern of the entire mechanism. Other parameters with slightly elevated correlations include those associated with major load-bearing cross-sections and link lengths, consistent with their mechanical significance. Overall, the low correlation magnitudes confirm that the model is not sensitive to specific geometric features and remains reliable across diverse robot designs.

To evaluate this capability and systematically isolate the contributions of the proposed architectural components, an ablation study on a dataset of 250 robot instances with randomly sampled geometric parameters, postures, and loading conditions was conducted. Several model variants were established to verify the necessity of the geometric–load graph, the GNN encoder, and the query module.

Specifically, to test the importance of geometry-load (GL) input, Robot-NO-GL was designed to process geometry and load (GL) separately, concatenating the load vector only after extracting the geometric latent embedding. To evaluate the graph-based topological representation against spatial-based methods, the GNN encoder and the query decoder were replaced with Point Transformer framework, resulting in PT-GE-GL (Geometry Encoder) and PT-FD-GL (Field Decoder), respectively, both utilizing the full geometry-load data. Table 7 summarizes the MAE and RMSE of the proposed method (Robot-NO) and other ablation models.

It can be seen from Table 7 that the models utilizing Point Transformers as encoders (PT-GE-GL) exhibit significantly inferior performance compared to the proposed GNN-based framework. This is primarily because Point Transformers rely on spatial neighborhood aggregation, which prioritizes proximal points while potentially ignoring the actual mechanical connectivity of the robot. In robotic structures, mechanical stiffness is governed by structural joints and interfaces; a point-cloud-based approach may misinterpret non-connected proximal regions as coupled, thereby failing to capture the intrinsic relationship between geometry, load, and the resulting deformation field. Consequently, when Point Transformers is applied to the field decoding stage (PT-FD-GL), the results further deteriorate, especially for full-field outputs. The neglect of physical topology causes the model to struggle between maintaining field continuity and interpreting spatial distances, which underscores the necessity of using Graph Neural Networks to extract geometric features in problems where deformation is tightly

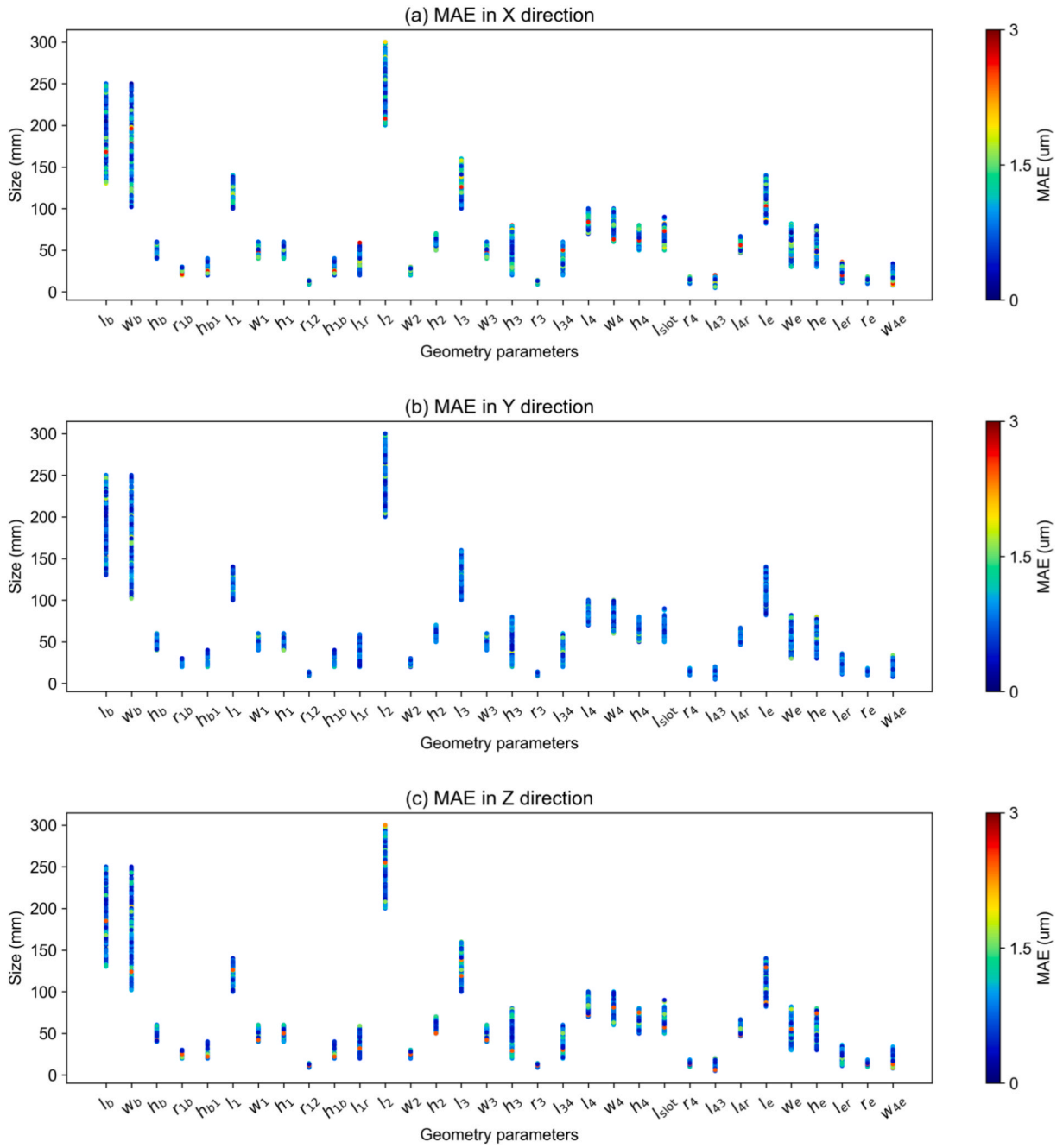


Fig. 12. MAE of the predicted *X*, *Y*, and *Z* deformation components across all sampled geometric parameters.

coupled with complex mechanical structures.

The predictive accuracy of Robot-NO is substantially higher than that of the separated formulation (Robot-NO-GL) and even lower than that of the PT-GE-GL variant. This demonstrates that a unified geometry-load representation allows the model to more effectively discover how external loads, when applied at different locations, propagate through the geometric topology to influence the deformation field.

Furthermore, the proposed method was compared with two versions of Transolver, a representative transformer-based neural operator [41]. Transolver was trained using the same dataset, loss function, and training hyperparameters as Robot-NO. The only difference was the batch size, which was set to 32 due to GPU memory limitations. The network architecture remained at its default settings, without modifications or task-specific tuning. The difference between the versions of Transolver is their parameter sizes, where the first version has 1,807,051 parameters (TS-Large-GL), and the second version has 297,071 parameters (TS-Small-GL), which is closer to the parameter count of Robot-NO.

Robot-NO, with 310,023 parameters, significantly outperforms TS-Large-GL (1,807,051 parameters), across all displacement components. Robot-NO achieves MAE of $0.92 \pm 0.51\,\mu$m, $0.88 \pm 0.50\,\mu$m, and $0.87 \pm 0.43\,\mu$m for the *X, Y,* and *Z* components, respectively, with similarly low RMSE values. Compared to the Transolver-Large (1,807,051 parameters), Robot-NO improves the MAE by an average of 20.06%, achieving a prediction RMSE of 0.91 μm.

While the smaller version of TS-Small-GL (297,071 parameters) shows a smaller average error in the Z components compared to Robot-NO, it exhibits larger error variance. Additionally, Robot-NO shows a high accuracy compare with Transolver in the *X* and *Y* components. This suggests that TS-Small-GL may be prone to local optima in multi-directional predictions, leading to less consistent performance across

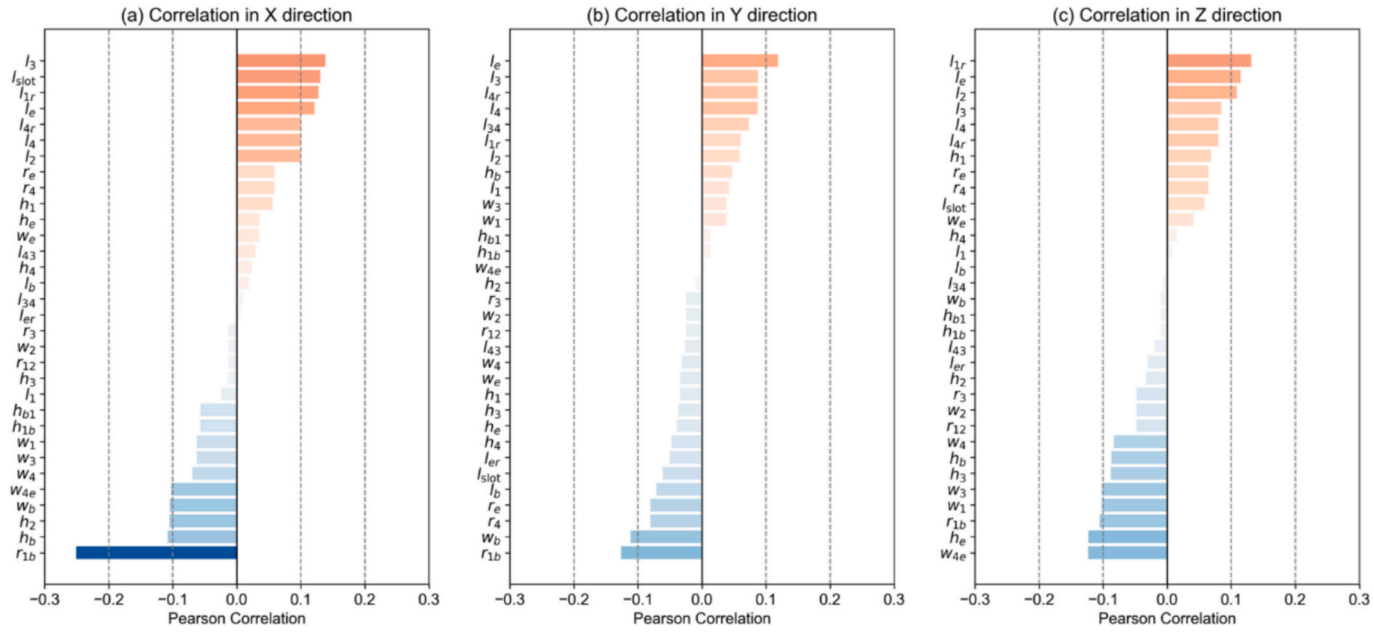


Fig. 13. Pearson correlation coefficients between geometric parameters and prediction errors in the *X*, *Y*, and *Z* deformation components.

Table 7
Overall MAE and RMSE of the predicted displacement components across 250 robot instances with randomly sampled geometric parameters.

Model	Index	X	Y	Z	Inference time
Robot-NO (Params: 310,023)	MAE (μm)	**0.92 ± 0.51**	**0.88 ± 0.50**	0.87 ± 0.43	0.0058 s
	RMSE (μm)	**0.95 ± 0.54**	**0.90 ± 0.52**	0.89 ± 0.45	
Robot-NO-GL (Params: 309,831)	MAE (μm)	2.46 ± 2.57	2.19 ± 2.22	2.95 ± 2.75	**0.0051 s**
	RMSE (μm)	2.54 ± 2.61	2.27 ± 2.27	3.09 ± 2.90	
PT-GE-GL (Params: 706,311)	MAE (μm)	1.47 ± 0.92	1.46 ± 0.89	1.22 ± 0.70	0.0121 s
	RMSE (μm)	1.50 ± 0.94	1.50 ± 0.92	1.25 ± 0.73	
PT-FD-GL (Params: 706,311)	MAE (μm)	5.58 ± 3.33	4.32 ± 2.93	3.71 ± 1.86	0.0113 s
	RMSE (μm)	5.68 ± 3.39	4.42 ± 3.05	3.78 ± 1.99	
TS-Large-GL (Params: 1,807,051)	MAE (μm)	1.22 ± 0.90	1.15 ± 0.88	0.97 ± 0.72	0.0095 s
	RMSE (μm)	1.26 ± 0.91	1.18 ± 0.90	1.00 ± 0.75	
TS-Small- GL (Params: 297,071)	MAE (μm)	1.13 ± 0.94	1.23 ± 1.12	**0.72** ± 0.67	0.0055 s
	RMSE (μm)	1.17 ± 0.96	1.27 ± 1.14	**0.75** ± 0.69	

all displacement components. In contrast, Robot-NO maintains smaller error variance in all three directions. Compared to the TS-Small-GL (297,071 parameters), Robot-NO improves the MAE by 13.31% on average.

To provide a more intuitive comparison of model reliability, a violin plot illustrating the MAE distributions of Robot-NO and the two Transolver baselines are presented in Fig. 14. As shown in Fig. 14, Robot-NO shows consistently more compact error distributions across all three directions, with interquartile ranges (IQRs) of 0.62 μm, 0.58 μm, and 0.53 μm in X, Y and Z directions, respectively. In contrast, both Transolver variants exhibit substantially broader distributions in all directions. For TS-Large-GL, the IQRs increase to 1.15 μm, 1.55 μm, and 0.79 μm in X, Y, and Z, respectively, comparing to 1.13 μm, 1.36 μm, and 0.80 μm from TS-Small-GL. These results quantitatively confirm that Robot-NO achieves consistently tighter error distributions and better prediction stability across all spatial directions.

These results highlight that geometry–load graph representation and query–decoder formulation of the proposed method are better aligned with the structural mechanics characteristics of robotic components. Even under substantial geometric variability, Robot-NO maintains stable micrometer-level accuracy, demonstrating generalization capability across continuously changing geometric domains.

To quantitatively evaluate the geometric generalization of Robot-NO, the distributional relationship between the training and test sets in the parameter space is analyzed. First, the Mahalanobis Distance (MD) was calculated for each test instance relative to the training set distribution to measure its degree of novelty. As shown in Fig. 15 (a-c), while samples with smaller MD values generally exhibit lower errors, the Pearson correlation coefficients between MD and MAE across all three directions are close to zero (ranging from −0.012 to −0.026). The lack of strong correlation indicates that prediction accuracy does not systematically degrade as the geometry moves further from the training distribution, thereby confirming the stability of the model.

Furthermore, a Principal Component Analysis (PCA) was performed to visualize geometry parameter coverage. In Fig. 15(d–f), the test set (darker points) exhibits a more dispersed distribution compared to the training set (lighter points), effectively covering unobserved regions in the latent space. Notably, instances with larger prediction errors are uniformly distributed across the PCA domain rather than clustering in specific peripheral areas. This evidence demonstrates that Robot-NO has learned the underlying structural mechanics rather than merely interpolating between training samples, highlighting its robust generalization capability to out-of-distribution geometries.

In addition to accuracy, computational efficiency is a critical factor for deformation prediction in design and optimization workflows. A typical FEM for the evaluated robot models requires approximately 30 s to compute a full deformation field, whereas Robot-NO produces predictions in 0.0058 s on a single GPU (NVIDIA TITAN RTX), corresponding to a speedup of 6000 times. Although generating the training dataset involves the computation of approximately 8,000 FEM samples (covering 1,000 geometric variants, two configurations, and two force conditions), which amounts to roughly 8,000 × 30 s ≈ 67 h of simulation time, this cost is incurred only once during offline preparation. The

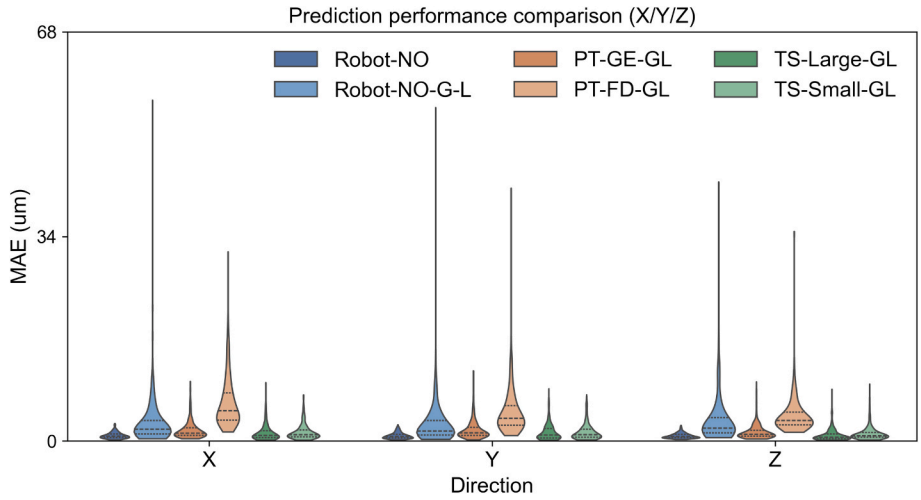


Fig. 14. Prediction performance of different models.

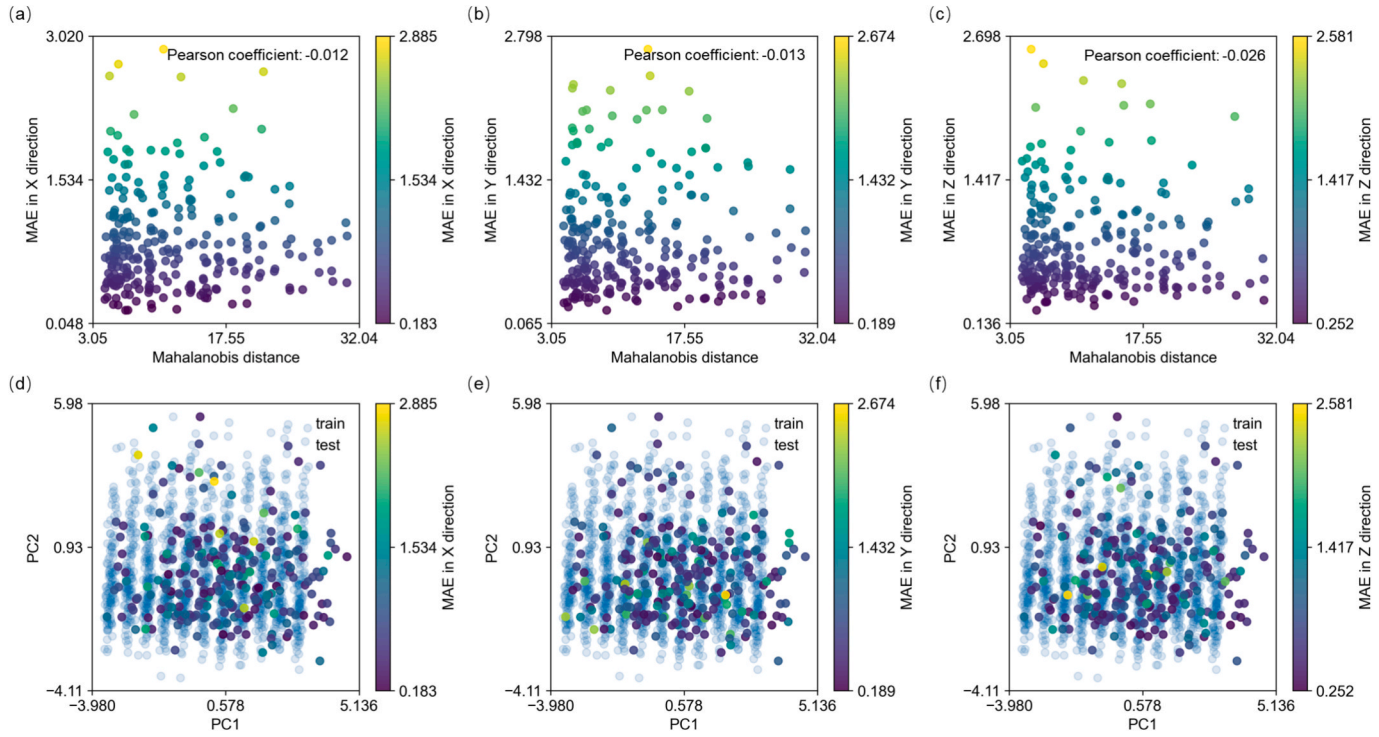


Fig. 15. Quantitative evaluation of geometric generalization: (a-c) Correlation between MD and MAE in X, Y, and Z directions respectively; (d-f) PCA visualization of the geometric parameter distribution for the training and test sets in *X*, *Y*, and *Z* directions, respectively.

subsequent model training requires about 12 h, after which Robot-NO can provide real-time predictions at negligible computational cost. Such substantial efficiency gains enable rapid exploration of large design spaces, support real-time deformation compensation, and significantly reduce the computational burden in iterative optimization pipelines. Combined with the consistent micrometer-level accuracy demonstrated across variations in geometry, posture, and loading conditions, these results highlight the practical value and scalability of the proposed neural operator framework for robotics applications.

Given the vast combinatorial space spanned by geometric parameters, joint configurations, and external loading conditions, the ability to replace repeated high-fidelity simulations with a single neural operator offers substantial practical benefits. Once trained, Robot-NO enables rapid exploration of large design domains, supports stiffness-aware optimization, and allows real-time deformation compensation without reliance on expensive numerical solvers. Combined with the consistent micrometer-level accuracy demonstrated across variations in geometry, posture, and loading, these results highlight the scalability and practical value of the proposed neural operator framework for advanced robot design and structural analysis.

5. Discussion

While point loads and gravity are considered in the current study, Robot-NO's node-level encoding is inherently flexible for handling complex load patterns such as distributed and multi-point forces. Due to the inherent non-linear learning capacity of the graph-based operator, the model can approximate non-linear elasticity and large-deformation

behaviors by training on datasets derived from non-linear FEM solvers. By augmenting the geometric–load graph with local orientation features, the model can explicitly account for various types of forces including follower loads, which provides a concrete foundation for design and analyses of complex robotic systems.

Beyond deformation fields, the architecture of Robot-NO is inherently compatible with the prediction of other critical physical quantities, such as stress distributions and strain fields. By extending the output dimension of the query-based decoder or employing a multi-head decoder structure, Robot-NO can also provide stress and strain outputs. This generalizable capability is vital for robot design, analysis, and optimization.

6. Conclusion

This work presents Robot-NO, a novel neural operator architecture for fast and accurate prediction of robot deformation fields. The main contributions of this study are as follows.

(1) A generalizable neural operator formulation is proposed to predict full-field elastic deformations under varying robot geometries, joint configurations, and external loading conditions, addressing a key limitation of existing data-driven deformation models that are restricted to fixed domains or partial outputs.
(2) The proposed method achieves micrometer-level full-field prediction accuracy, with mean absolute errors of $0.92 \pm 0.51\ \mu m$, $0.88 \pm 0.50\ \mu m$, and $0.87 \pm 0.43\ \mu m$ for the X, Y, and Z displacement components, respectively, across diverse geometric designs, postures, and force conditions.
(3) In terms of computational efficiency, Robot-NO provides a 6000 times speedup compared to conventional FEM simulations, reducing the inference time for a full deformation field from 30 s to 0.0058 s, thereby enabling large-scale design exploration and real-time deformation-aware optimization.

Because of its superior efficiency and strong generalization capabilities, Robot-NO enables rapid optimization of structural parameters, effective deformation compensation, and near real-time digital twin implementation. These capabilities are essential and increasingly indispensable for advancing robot design, planning, and real-world applications.

While the method demonstrates strong overall performance, certain extreme geometric or loading conditions can still lead to relatively high prediction errors. This suggests that further improvements may be possible through larger diverse training datasets, enhanced model capacity, and more advanced training strategies. Incorporating physical constraints into the operator formulation also represents a promising direction for increasing robustness, especially in regimes where deformation behavior becomes highly nonlinear or sensitive to structural variations. Generalization and transferability of the model across different robotic topologies will also be worth to explor further.

CRediT authorship contribution statement

Zhiwei Zhao: Writing – review & editing, Writing – original draft, Validation, Methodology, Data curation, Conceptualization. **Yan Jin:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Caitlin Sands:** Writing – review & editing, Methodology, Conceptualization. **Paul Goodall:** Writing – review & editing, Methodology, Conceptualization. **Andrew West:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Mark Price:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by UK EPSRC Grant EP/V007335/1.

Data availability

Data will be made available on request.

References

[1] A. Keshvarparast, D. Battini, O. Battaia, A. Pirayesh, Collaborative robots in manufacturing and assembly systems: literature review and future research agenda, J. Intell. Manuf. 35 (2024) 2065–2118.
[2] Y. Shi, W.T. Chow, T.M. Kwok, Y. Wang, Y. Shi, W.T. Chow, T.M. Kwok, Y. Wang, Lightweight and low-cost cable-driven SCARA robotic arm with 9 DOF, Robotics 14 (2025).
[3] K. Wu, J. Li, H. Zhao, Y. Zhong, K. Wu, J. Li, H. Zhao, Y. Zhong, Review of industrial robot stiffness identification and modelling, Appl. Sci. 12 (2022).
[4] Y. Jian, Y. Jin, M. Price, J. Moore, A parallel robot with remote centre-of-motion for eye surgery: design, kinematics, prototype, and experiments, Int. J. Med. Robot. 20 (2024) e2665.
[5] T. Sun, B. Wang, X. Huo, Knowledge-driven automated design of industrial robots: a unified graph-based framework with multi-engine reasoning, Adv. Eng. Inform. 69 (2026) 103995.
[6] A.A. Malik, A. Bilberg, Collaborative robots in assembly: a practical approach for tasks distribution, Procedia CIRP 81 (2019) 665–670.
[7] Y. Lin, H. Zhao, H. Ding, Posture optimization methodology of 6R industrial robots for machining using performance evaluation indexes, Robot. Comput.-Integr. Manuf. 48 (2017) 59–72.
[8] L. Xu, D. Zhang, J. Xu, R. Wang, Y. Sun, A stiffness matching-based deformation errors control strategy for dual-robot collaborative machining of thin-walled parts, Robot. Comput.-Integr. Manuf. 88 (2024) 102726.
[9] S. Bandara, Y. Jin, M. Van, D. Sun, R. Fu, P. Curley, G. Rutherford, C. Higgins, Geometrical quality prediction of machining process by exechon X-mini PKM through deformation modelling and error compensation, Robot. Comput.-Integr. Manuf. 97 (2026) 103115.
[10] A. Alamdar, D.E. Usevitch, J. Wu, R.H. Taylor, P. Gehlbach, I. Iordachita, Steady-hand eye robot 3.0: optimization and benchtop evaluation for subretinal injection, IEEE Trans. Med. Robot. Bionics 6 (2024) 135–145.
[11] C. Chen, F. Peng, R. Yan, Y. Li, D. Wei, Z. Fan, X. Tang, Z. Zhu, Stiffness performance index based posture and feed orientation optimization in robotic milling process, Robot. Comput.-Integr. Manuf. 55 (2019) 29–40.
[12] J. Jia, X. Sun, J. Jia, X. Sun, Structural optimization design of a six-degrees-of-freedom serial robot with integrated topology and dimensional parameters, Sensors 23 (2023).
[13] S. Tonkens, J. Lorenzetti, M. Pavone, Soft robot optimal control via reduced order finite element models, in: 2021 IEEE Int. Conf. Robot. Autom. ICRA, 2021: pp. 12010–12016.
[14] Z. Wang, D. Gao, J. Liang, Y. Lu, F. Wang, S. Ma, Z. Jiang, K. Deng, Robot pose optimization based on spectral clustering guided MOSSA considering the flexibility of robot structure and mounting base, Adv. Eng. Inform. 68 (2025) 103645.
[15] C. Ye, J. Yang, H. Ding, High-accuracy prediction and compensation of industrial robot stiffness deformation, Int. J. Mech. Sci. 233 (2022) 107638.
[16] S.A. Kouritem, M.I. Abouheaf, N. Nahas, M. Hassan, A multi-objective optimization design of industrial robot arms, Alex. Eng. J. 61 (2022) 12847–12867.
[17] A. Kermanian, A.E. Kamali, A. Taghvaeipour, Dynamic analysis of flexible parallel robots via enhanced co-rotational and rigid finite element formulations, Mech. Mach. Theory 139 (2019) 144–173.
[18] A. Cammarata, Unified formulation for the stiffness analysis of spatial mechanisms, Mech. Mach. Theory 105 (2016) 272–284.
[19] J. López-Martínez, D. García-Vallejo, A. Giménez-Fernández, J.L. Torres-Moreno, A flexible multibody model of a safety robot arm for experimental validation and analysis of design parameters, J. Comput. Nonlinear Dyn. 9 (2013).
[20] C. Gosselin, Stiffness mapping for parallel manipulators, IEEE Trans. Robot. Autom. 6 (1990) 377–382.
[21] M. Johra, E. Courteille, D. Deblaise, S. Guégan, M. Johra, E. Courteille, D. Deblaise, S. Guégan, Elasto-geometrical model-based control of industrial manipulators using force feedback: application to incremental sheet forming, Robotics 11 (2022).
[22] Y.-F. Zhang, B.-G. Yao, F. Zhang, X.-F. Liang, G. Tao, Y.-X. Ge, T.-F. Niu, Y.-F. Zhang, B.-G. Yao, F. Zhang, X.-F. Liang, G. Tao, Y.-X. Ge, T.-F. Niu, Global stiffness modeling of robot drilling system incorporating end-effector and arm flexibility based on virtual joint method, Machines 13 (2025).

[23] W. Cao, H. Ding, A method for stiffness modeling of 3R2T overconstrained parallel robotic mechanisms based on screw theory and strain energy, Precis. Eng. 51 (2018) 10–29.
[24] I. Görgülü, M.I.C. Dede, G. Kiper, Stiffness modeling of a 2-DoF over-constrained planar parallel mechanism, Mech. Mach. Theory 185 (2023) 105343.
[25] Y. Zhao, Q. Chen, H. Li, H. Zhou, H.R. Attar, T. Pfaff, T. Wu, N. Li, Recurrent U-net-based graph neural network (RUGNN) for accurate deformation predictions in sheet material forming, Adv. Eng. Inform. 69 (2026) 104021.
[26] J.C. Hsiao, K. Shivam, C.L. Chou, T.Y. Kam, J.C. Hsiao, K. Shivam, C.L. Chou, T. Y. Kam, Shape design optimization of a robot arm using a surrogate-based evolutionary approach, Appl. Sci. 10 (2020).
[27] M. Leco, V. Kadirkamanathan, A perturbation signal based data-driven gaussian process regression model for in-process part quality prediction in robotic countersinking operations, Robot. Comput.-Integr. Manuf. 71 (2021) 102105.
[28] Z.-Y. Liao, Q.-H. Wang, Z.-H. Xu, H.-M. Wu, B. Li, X.-F. Zhou, Uncertainty-aware error modeling and hierarchical redundancy optimization for robotic surface machining, Robot. Comput.-Integr. Manuf. 87 (2024) 102713.
[29] K. Wu, Y. Zhang, D. Gao, S. Deng, W. Li, M. Wang, Neural network–based transfer learning to improve stiffness modeling of industrial robots with small experimental data sets, Int. J. Adv. Manuf. Technol. 135 (2024) 5253–5265.
[30] J. Chen, Z. Liu, C. Chen, B. Qi, J. Xu, L. Tao, Q. Yan, Robot stiffness modeling based on the rigid flexible coupling simulation and its application to trajectory planning, Precis. Eng. 92 (2025) 77–89.
[31] M. Li, H. Wu, H. Handroos, R. Skilton, A. Hekmatmanesh, A. Loving, Deformation modeling of manipulators for DEMO using artificial neural networks, Fusion Eng. Des. 146 (2019) 2401–2406.
[32] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces with applications to PDEs, J. Mach. Learn. Res. 24 (2024) 89:4061-89:4157.
[33] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (2021) 218–229.
[34] Z. Li, N.B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, Int. Conf. Learn. Represent. 2021 (2021) 1–16.
[35] W. Li, M.Z. Bazant, J. Zhu, Phase-field DeepONet: Physics-informed deep operator neural network for fast simulations of pattern formation governed by gradient flows of free-energy functionals, Comput. Methods Appl. Mech. Eng. 416 (2023) 116299.
[36] C. Kaewnuratchadasorn, J. Wang, C. Kim, X. Deng, Geometry physics neural operator solver for solid mechanics, Comput.-Aided Civ. Infrastruct. Eng. 40 (2025) 1388–1404.
[37] S. Cai, Z. Wang, L. Lu, T.A. Zaki, G.E. Karniadakis, DeepM&mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks, J. Comput. Phys. 436 (2021) 110296.
[38] K. Azizzadenesheli, N. Kovachki, Z. Li, M. Liu-Schiaffini, J. Kossaifi, A. Anandkumar, Neural operators for accelerating scientific simulations and design, Nat. Rev. Phys. 6 (2024) 320–328.
[39] C. Kaewnuratchadasorn, J. Wang, C.-W. Kim, Physics-informed neural operator solver and super-resolution for solid mechanics, Comput.-Aided Civ. Infrastruct. Eng. 39 (2024) 3435–3451.
[40] M. Yin, N. Charon, R. Brody, L. Lu, N. Trayanova, M. Maggioni, A scalable framework for learning the geometry-dependent solution operators of partial differential equations, Nat. Comput. Sci. (2024) 1–13.
[41] Z. Zhao, C. Liu, Y. Li, Z. Chen, X. Liu, Diffeomorphism neural operator for various domains and parameters of partial differential equations, Commun. Phys. 8 (2025) 15.
[42] Z. Hao, Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song, J. Zhu, GNOT: a general neural operator transformer for operator learning, in: Proc. 40th Int. Conf. Mach. Learn., 2023: pp. 12556–12569.
[43] A. Bryutkin, J. Huang, Z. Deng, G. Yang, C.-B. Schönlieb, A. Aviles-Rivero, HAMLET: graph transformer neural operator for partial differential equations, (2024).
[44] H. Wu, H. Luo, H. Wang, J. Wang, M. Long, Transolver: a fast transformer solver for PDEs on general geometries, (2024).