A Novel Deep Learning-Based Spatio-Temporal Model for Prediction of Pose Residual Errors in Optical Processing Hybrid Robot

Li, Jun; Cheng, Gang; Pang, Yusong

DOI
10.1109/TII.2024.3369246

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Industrial Informatics

A Novel Deep Learning-Based Spatio-Temporal Model for Prediction of Pose Residual Errors in Optical Processing Hybrid Robot

Jun Li, Gang Cheng, and Yusong Pang

***Abstract*—The accuracy of high-precision optical processing robots is influenced by various factors, including static error factors and dynamic error factors. These factors pose significant challenges to the deterministic processing of precision optics. This article proposes a pose residual prediction model for optical processing hybrid robots based on deep spatio-temporal graph convolutional neural networks. In this study, we establish a geometric error model for hybrid robots and calibrate the geometric error parameters using an extended Kalman filter to obtain the pose residuals component. To address the complex spatio-temporal interactions between multiple sensor variables in joint space during robot motion, we introduce the non-Euclidean spatio-temporal graph convolutional neural network. This model effectively extracts advanced spatio-temporal interaction features based on a spatio-temporal attention mechanism. Finally, the performance of the proposed method in pose residual prediction was validated through real experiments, and the results demonstrated its advantages over other state-of-the-art methods.**

***Index Terms*—Optical processing hybrid robot, residual error prediction, spatio-temporal graph neural network.**

I. Introduction

ROBOT systems play a pivotal role in advancing machining efficiency and cost-effectiveness within manufacturing processes, particularly through the substitution of expensive and cumbersome computer numerical control (CNC) machines. As sophisticated spatial mechanisms characterized by multiple degrees of freedom, hybrid robots seamlessly integrate the advantages inherent in both serial and parallel mechanisms. This integration endows these robots with flexible degrees of freedom in the workspace, excelling in critical aspects such as load-bearing capacity, stiffness, and precision. The potential of hybrid robots in the domain of high-precision machining, especially for complex surfaces such as large optical freeform surfaces, is substantial [1], [2], [3]. However, challenges arise from assembly errors, linkage deformation, and temporal variations in robot behavior, significantly constraining the absolute positioning accuracy of robot end effectors. This limitation poses a considerable obstacle to achieving the requisite quality and stability in optical processing, particularly for traditional industrial robots faced with the demands of high-precision processing. The intricacies become particularly apparent when dealing with large optical primary mirrors, where pose errors may fluctuate due to the influence of polishing postures. The refinement of large optical surfaces typically entails multiple iterations. The presence of pose errors within the polishing trajectory not only introduces uncertainty in the dwell time but also amplifies the number of iterations and prolongs the overall processing time, as illustrated in Fig. 1. Consequently, these factors adversely impact processing efficiency [4], [5]. Enhancing the motion accuracy of polishing robots has become an urgent goal to improve the performance of optical components and pursue high-precision optical processing.

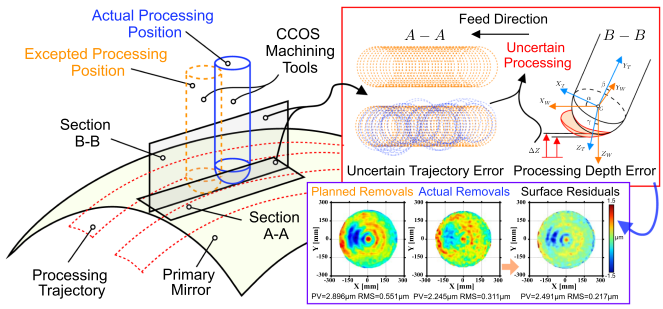


Fig. 1. Uncertain optical machining induced by robot pose residuals.

A. Literature Review

Two primary approaches are employed for addressing pose error and suppression compensation, namely, the model-based method and the zero-model method [6]. The model-based method involves the establishment of an error model, such as the spiral theory [7], vector discretization method, or kinematic

Manuscript received 20 June 2023; revised 27 November 2023 and 27 January 2024; accepted 12 February 2024. Date of publication 25 March 2024; date of current version 5 June 2024. This work was supported in part by National Nature Science Foundation of China (NSFC) under Grant 52275039 and in part by the Priorty Academic Program Development of Jiangsu Higher Education Institutions (PAPD). Paper no. TII-23-2259. *(Corresponding author: Gang Cheng.)*

Jun Li and Gang Cheng are with the School of Mechanical and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China (e-mail: tb20050006b2@cumt.edu.cn; chg@cumt.edu.cn).

Yusong Pang is with the Section of Transport Engineering and Logistics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: y.pang@tudelft.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3369246.

Digital Object Identifier 10.1109/TII.2024.3369246

equivalent limb [8], to delineate the mapping between geometric parameter errors and pose errors. Chen et al. [9] contributed to this field by devising a rigid-flexible coupled error model for the nonkinematic calibration of robots. In addition, an enhanced method for full pose measurement and identification optimization was proposed, resulting in the identification of 36 error parameters within the model, consequently enhancing accuracy and stability. Sun et al. [10], [11] introduced a calibration method grounded in finite and instantaneous spiral theory. This method involves transforming instantaneous spiral errors identified during calibration into joint actuation errors, which are then rectified by adjusting the inputs. Li et al. [12] and Luo et al. [13], utilizing the improved Levenberg–Marquardt algorithm and the unscented Kalman filter (UKF) algorithm for industrial robot calibration, demonstrated the commendable performance of the proposed calibration algorithm. While these methods contribute to enhancing robot pose accuracy, it is noteworthy that model-based robot pose error calibration methods often neglect nongeometric errors, such as transmission errors associated with gear backlash. Furthermore, in practical applications, the intricate structure of hybrid robots poses challenges in comprehensively addressing error sources. Developing a comprehensive mathematical error model for hybrid robots is a time-consuming and labor-intensive undertaking, thereby constraining their practical applications.

In recent years, researchers have shown significant interest in zero model-the data-driven prediction of robot pose errors. These approaches eliminate the need for establishing a complex mathematical model, opting instead to directly measure robot pose errors using external pose measuring instruments. Subsequently, they establish the mapping relationship between pose errors and robot pose through mathematical tools such as Feedforward Neural Network [14], [15], [16], [17], Deep Belief Network [18], and Residual Neural Network [19]. Although these works utilize deep learning methods to construct a variety of mapping relationships for the pose and the corresponding residual errors, it is clear that these methods remain ambiguous in their principles for the distribution of the pose residuals, leaving the prediction accuracy of the pose residuals in a bottleneck.

B. Research Challenges and Motivation

The methods described above take into consideration the error distribution in the overall state and implement compensation for corresponding pose errors. However, they partially overlook dynamic influences on pose errors arising from time-varying nongeometric factors like motion pair wear and gear backlash. The presence of these time-varying factors can lead to dynamically unstable pose errors for the same pose [20], posing challenges to the accuracy and generalization capability of the prediction model, as the pose error for the same pose can be influenced by time-varying factors. Since the trajectory error distribution is impacted not only by the current frame input but also by historical state accumulation, the influence of each joint variable on the error distribution introduces a high coupling in both spatio-temporal relationships and spatio-temporal convolution characteristics.

Meanwhile, the aforementioned data-driven robot pose error prediction models overlook the non-Euclidean nature of potential spatial coupling relationships between process variables, failing to offer an advanced representation for these spatio-temporal coupling patterns. Consequently, there is a need to consider the spatio-temporal coupling characteristics between joint space input and pose errors. Through the effective utilization of intricate nonlinear spatio-temporal data and the revelation of complex spatio-temporal characteristics in residuals, a notable enhancement in the accuracy and reliability of robot pose error prediction is anticipated.

Nowadays, graph neural networks have garnered significant attention across various domains, particularly in the realm of industrial information. Scholars have delved into research on deep learning methods grounded in graph convolutional neural networks, with a focus on articulating the intricate spatio-temporal coupling relationships between complex variables. Liu et al. [21] addressed the challenge of predicting complex ship motion trajectories by employing spatio-temporal multigraph convolutional layers. This approach effectively models the dynamic interaction between adjacent ships, seamlessly integrating ship trajectory features into the prediction framework. Similarly, Jiang et al. [22] introduced an electrical spatio-temporal graph convolutional network (GCN) for intelligent prediction and maintenance. This model concurrently considers attribute interaction and temporal correlation, with its effectiveness validated through practical examples. In addition, the temporal GCN model utilizes GCNs to capture road network topology and spatial correlations [23]. These advancements showcase the utility of graph neural networks in tackling complex spatio-temporal problems across diverse industrial applications. Traditional deep learning methods struggle with such complex problems, making graph neural network-based deep learning methods an innovative solution. In summary, the key to addressing the challenges is accurately describing the correlations between variables. Deep learning methods, particularly those grounded in spatio-temporal graph learning techniques, prove effective in capturing complex correlations and evolutions. This capability enables them to surpass the limitations of traditional methodologies, serving as powerful tools for addressing intricate spatio-temporal problems.

C. Contributions and Organization

To tackle the aforementioned challenge, this study introduces an innovative deep learning model explicitly crafted for predicting residual errors in machining trajectory poses. To the best of our knowledge, this marks the inaugural attempt to harness deep spatio-temporal graph neural networks for predicting robot pose residual errors. The principal contributions of this article are outlined as follows.

1) To address the complex spatio-temporal interaction between multiple sensor variables during robot motion, and drawing inspiration from non-Euclidean learning and soft sensors, we introduced a groundbreaking method—the non-Euclidean spatio-temporal graph convolutional neural network for robot residuals prediction.

TABLE I
DEFINITION OF GEOMETRIC ERROR SOURCES

Geometric error source	Definition	Error / reference coordinate system
$\boldsymbol{\Delta r}_A$	Position error vector of dynamic platform geometric center A	$\{A-x'y'z'\}/\{O-XYZ\}$
$\widehat{\boldsymbol{\theta}}_A$	Attitude error vector of dynamic platform geometric center A	$\{A-x'y'z'\}/\{O-XYZ\}$
$\boldsymbol{\Delta b}_i$	Position error vector of UPS branch chain Hooker joint B_i	$\{B_i-x_iy_iz_i\}/\{O-XYZ\}$
$\widehat{\boldsymbol{\theta}}_{B_i}$	Attitude error vector of UPS branch chain Hooker joint B_i	$\{B_i-x_iy_iz_i\}/\{O-XYZ\}$
$\boldsymbol{\Delta n}_i$	Position error vector of the UPS branch chains link origin N_i	$\{N_i-u_iv_iw_i\}/\{B_i-x_iy_iz_i\}$
$\widehat{\boldsymbol{\theta}}_{N_i}$	Attitude error vector of the UPS branched chains link origin N_i	$\{N_i-u_iv_iw_i\}/\{B_i-x_iy_iz_i\}$
$\boldsymbol{\Delta m}_i$	Position error vector of the origin M_i of joint shafts in composite ball joints	$\{M_i-u_i'v_i'w_i'\}/\{A_i-x_i'y_i'z_i'\}$
$\widehat{\boldsymbol{\theta}}_{M_i}$	Attitude error vector at the origin M_i of joint shafts in composite ball joints	$\{M_i-u_i'v_i'w_i'\}/\{A_i-x_i'y_i'z_i'\}$
$\boldsymbol{a}_{0_i}$	Position error vector of Hooke joint origin A_i in composite ball joints	$\{A_i-x_i'y_i'z_i'\}/\{A-x'y'z'\}$
$\widehat{\boldsymbol{\theta}}_{A_i}$	Attitude error vector of Hooke joint origin A_i in composite ball joints	$\{A_i-x_i'y_i'z_i'\}/\{A-x'y'z'\}$
$\boldsymbol{\Delta S}_i$	Position error vector of the center in composite ball hinges	$\{T_i-x_{T_i}y_{T_i}z_{T_i}\}/\{S_i-x_{S_i}y_{S_i}z_{S_i}\}$
δ_i	Scalar length errors of composite ball hinge connection shafts	$\{M_i-u'v'w'\}/\{S_i-x_{S_i}y_{S_i}z_{S_i}\}$
Δl_i	Scalar Rod length errors of the drive chains	$\{T_i-x_{T_i}y_{T_i}z_{T_i}\}/\{N_i-u_iv_iw_i\}$

2) To ensure the enduring prediction capability of pose residuals, a model refinement strategy was implemented, integrating sample ensemble and model warm start. This strategic approach effectively guarantees the long-term predictive performance of the model.
3) The effectiveness of the proposed method was validated through comparisons with various state-of-the art spatio-temporal models across multiple scenarios, whose performance was further affirmed through real-world experiments.

The rest of this article is organized as follows. In Section II, the extended Kalman filter algorithm is utilized to calibrate the geometric parameter errors of the hybrid optical machining robot, leading to the determination of residual error components. Section III presents the proposed theoretical framework for trajectory residual error prediction based on deep learning. The training of the model and experimental testing carried out in a real environment are expounded upon in Section IV. Finally, Section V concludes this article.

II. DYNAMIC ERROR DECOMPOSITION AND TRACING

The optical polishing robot comprises a 3-UPS/UP parallel mechanism, a 2R serial rotating head, and a computer controlled optical surface (CCOS) polishing system, as illustrated in Fig. 2(a). Here, U, P, R, and S symbols denote the universal joint, prismatic joint, revolute joint, and spherical joint, respectively. The underlined symbol indicates an active joint, while the absence of underlining denotes a passive joint. The three-degrees-of-freedom (3-DoF) parallel mechanism consists of a fixed platform, three identical UPS limbs, and one UP limb. The UPS limbs are actuated by an integrated ball screw, and the passive prismatic joint of the UP limb is achieved through a pair of linear sliding rails. To enhance the load-bearing capacity and operational accuracy of the hybrid mechanism and simplify the implementation of spherical joints, each passive spherical joint at the end of the driving limbs is optimized as a composite spherical joint, comprising a 2-DoF universal joint and a pair of angular contact roller bearings. The end effector, the CCOS optical grinding system, primarily consists of a grinding tool, a revolution servo motor, a rotation servo motor, and a pneumatic loading system. Two servo motors drive the grinding tool,

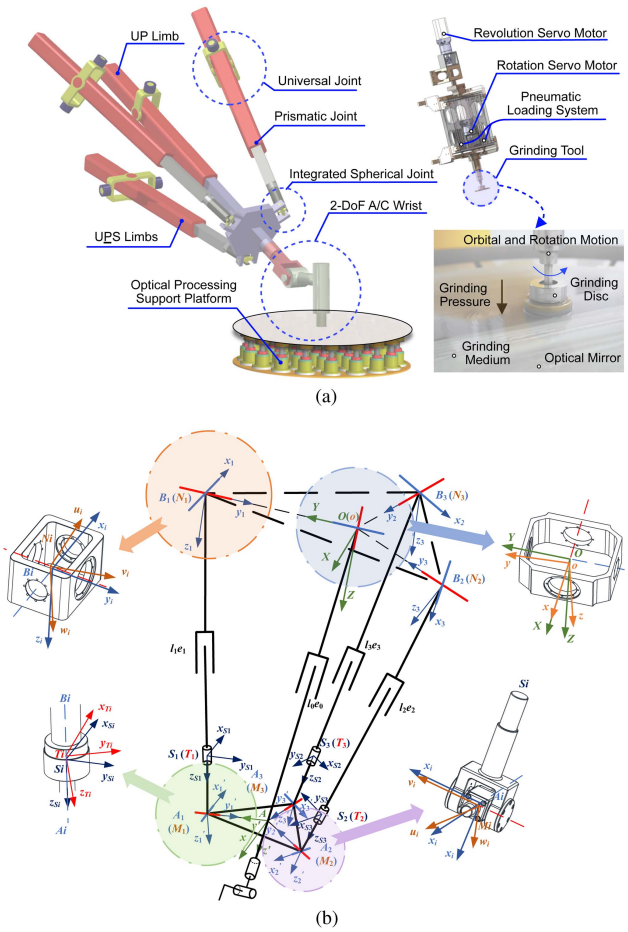


Fig. 2. Architecture of the hybrid robot. (a) Configuration of the hybrid robot for optical processing. (b) Representation of geometric error source in coordinate system.

enabling pressure grinding and polishing of optical mirror surfaces through a pneumatic loading system.

A. Geometric Error Model

As shown in Fig. 2(b), according to the above coordinate system and the related geometric error definition, the closed-chain $O-B_i-N_i-S_i-T_i-M_i-A_i-A-o$ vector equation

in the ideal condition can be expressed as

$$\begin{aligned} &\boldsymbol{r}_A + {}^{O}_{A}\boldsymbol{R}\,\boldsymbol{a}_{0i} - \boldsymbol{\delta}_i\,{}^{O}_{A}\boldsymbol{R}\,{}^{A}_{A_i}\boldsymbol{R}\,\boldsymbol{R}_{A_i}\,\boldsymbol{R}_{M_i}\,\boldsymbol{e} \\ &= \boldsymbol{b}_i + l_i\,{}^{O}_{B_i}\boldsymbol{R}\,\boldsymbol{R}_{B_i}\,\boldsymbol{R}_{N_i}\,\boldsymbol{e}. \end{aligned} \tag{1}$$

Considering the geometric parameter errors in Table I, the closed chain vector equation can be expressed as

$$\begin{aligned} &\boldsymbol{r}_A + \boldsymbol{\Delta r}_A + \left[\boldsymbol{E} + \widehat{\boldsymbol{\theta}}_A\right]\,{}^{O}_{A}\boldsymbol{R}\,(\boldsymbol{a}_{0i} + \boldsymbol{\Delta a}_{0i}) \\ &+ \left[\boldsymbol{E} + \widehat{\boldsymbol{\theta}}_A\right]\,{}^{O}_{A}\boldsymbol{R}\,{}^{A}_{A_i}\boldsymbol{R}\left[\boldsymbol{E} + \widehat{\boldsymbol{\theta}}_{A_i}\right]\boldsymbol{R}_{A_i} \\ &\left\{\boldsymbol{\Delta m}_i + \left[\boldsymbol{E} + \widehat{\boldsymbol{\theta}}_{M_i}\right]\boldsymbol{R}_{M_i}\left[-\left(\delta_i + \Delta\delta_i\right)\,\boldsymbol{e} + \boldsymbol{\Delta S}_i\right]\right\} \\ &= \boldsymbol{b}_i + \boldsymbol{\Delta b}_i + {}^{O}_{B}\boldsymbol{R}\left[\boldsymbol{E} + \widehat{\boldsymbol{\theta}}_{B_i}\right]\boldsymbol{R}_{B_i} \\ &\left\{\boldsymbol{\Delta n}_i + \left[\boldsymbol{E} + \widehat{\boldsymbol{\theta}}_{N_i}\right]\boldsymbol{R}_{N_i}\left(l_i + \Delta l_i\right)\,\boldsymbol{e}\right\} \end{aligned} \tag{2}$$

where $\boldsymbol{e}$ is the unit direction vector, denoted as $[0\ 0\ 1]^{\mathrm{T}}$ in the global coordinate system $O-XYZ$; $\boldsymbol{E}$ is the third-order unit matrix; $\hat{\boldsymbol{\theta}}$ is the antisymmetric matrix of attitude error vector $\boldsymbol{\theta}$, and ${}^{j}_{i}\boldsymbol{R}$ is the attitude matrix in coordinate system i with respect to reference coordinate system j. Neglecting the higher order terms of the error parameters yields

$$\begin{aligned} &\Delta\boldsymbol{r}_A + {}^{O}_{A}\boldsymbol{R}\,\boldsymbol{\Delta a}_{0_i} + \widehat{\boldsymbol{\theta}}_A\,{}^{O}_{A}\boldsymbol{R}\,\boldsymbol{a}_{0_i} + {}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{\Delta m}_i \\ &- {}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\Delta\delta_i\boldsymbol{e} + {}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\boldsymbol{\Delta S}_i \\ &- {}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\widehat{\boldsymbol{\theta}}_{M_i}\boldsymbol{R}_{M_i}\delta_i\boldsymbol{e} - {}^{O}_{A_i}\boldsymbol{R}\widehat{\boldsymbol{\theta}}_{A_i}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\delta_i\boldsymbol{e} \\ &- \widehat{\boldsymbol{\theta}}_A\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\delta_i\boldsymbol{e} \\ &= \boldsymbol{\Delta b}_i + {}^{O}_{B}\boldsymbol{R}\boldsymbol{R}_{B_i}\boldsymbol{\Delta n}_i + {}^{O}_{B}\boldsymbol{R}\boldsymbol{R}_{B_i}\boldsymbol{R}_{N_i}\Delta l_i\boldsymbol{e} \\ &+ {}^{O}_{B}\boldsymbol{R}\boldsymbol{R}_{B_i}\widehat{\boldsymbol{\theta}}_{N_i}\boldsymbol{R}_{N_i}l_i\boldsymbol{e} + {}^{O}_{B}\boldsymbol{R}\widehat{\boldsymbol{\theta}}_{B_i}\boldsymbol{R}_{B_i}\boldsymbol{R}_{N_i}l_i\boldsymbol{e}. \end{aligned} \tag{3}$$

Left multiply unit vector $\boldsymbol{e}$ at both ends, and approximate

$$\begin{aligned} &\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\Delta r}_A + \left(\boldsymbol{a_i}\times\boldsymbol{e}\right)^{\mathrm{T}}\boldsymbol{\theta}_A - \left[\delta_i\left({}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\boldsymbol{e}\times\boldsymbol{e}\right)\right]^{\mathrm{T}}\boldsymbol{\theta}_A \\ &= \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\Delta b}_i + \boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{B}\boldsymbol{R}\boldsymbol{R}_{B_i}\boldsymbol{\Delta n}_i + \left(\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{B}\boldsymbol{R}\boldsymbol{R}_{B_i}\boldsymbol{R}_{N_i}\boldsymbol{e}\right)\Delta l_i \\ &- \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\Delta a}_i - \left(\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\right)\boldsymbol{\Delta m}_i + \left(\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\boldsymbol{e}\right)\Delta\delta_i \\ &- \left(\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\right)\boldsymbol{\Delta S}_i \\ &+ \left[\delta_i\left(\boldsymbol{R}_{M_i}\boldsymbol{e}\right)\times\left(\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\right)^{\mathrm{T}}\right]^{\mathrm{T}}\boldsymbol{\theta}_{M_i} \\ &+ \left[\delta_i\left(\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\boldsymbol{e}\right)\times\left(\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A_i}\boldsymbol{R}\right)^{\mathrm{T}}\right]^{\mathrm{T}}\boldsymbol{\theta}_{A_i}. \end{aligned} \tag{4}$$

Separating the geometric error variables $\boldsymbol{\Delta r}_A$, $\boldsymbol{\theta}_A$ yields

$$\boldsymbol{\varepsilon} = \boldsymbol{J}_{\varepsilon}\boldsymbol{\delta} = \begin{bmatrix}\boldsymbol{\Delta r}_A \\ \boldsymbol{\theta}_A\end{bmatrix} \tag{5}$$

where

$$\boldsymbol{J}_{\varepsilon} = \boldsymbol{J}_{\varepsilon_1}\boldsymbol{J}_{\varepsilon_2} = \begin{bmatrix}\boldsymbol{e}^{\mathrm{T}} & \boldsymbol{\alpha}_1 \\ \boldsymbol{e}^{\mathrm{T}} & \boldsymbol{\alpha}_2 \\ \boldsymbol{e}^{\mathrm{T}} & \boldsymbol{\alpha}_3\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{\beta}_1 & & \\ & \boldsymbol{\beta}_2 & \\ & & \boldsymbol{\beta}_3\end{bmatrix}$$

$$\boldsymbol{\delta} = \begin{bmatrix}\boldsymbol{\Delta b}_i & \boldsymbol{\Delta n}_i & \Delta l_i & \boldsymbol{\Delta S}_i & \boldsymbol{\Delta\delta}_i & \cdots \\ & \boldsymbol{\Delta m}_i & \boldsymbol{\theta}_{m_i} & \boldsymbol{\theta}_{A_i} & \boldsymbol{\Delta a}_{0i} & \end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{\alpha_i} = \left({}^{O}_{A}\boldsymbol{R}\,\boldsymbol{a}_{0i}\times\boldsymbol{e}\right)^{T} - \left[\delta_i\left({}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\boldsymbol{e}\times\boldsymbol{e}\right)\right]^{\mathrm{T}}$$

$$\boldsymbol{\beta}_i = \begin{bmatrix} \boldsymbol{e}^{\mathrm{T}} \\ \boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{B_i}\boldsymbol{R}\boldsymbol{R}_{B_i} \\ \boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{B_i}\boldsymbol{R}\boldsymbol{R}_{B_i}R_{N_i}\boldsymbol{e} \\ -\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i} \\ \boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\boldsymbol{e} \\ -\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i} \\ \left[\delta\left(\boldsymbol{R}_{M_i}e\right)\times\left(\boldsymbol{e}^{T}\,{}^{O}_{A_i}\boldsymbol{R}\boldsymbol{R}_{A_i}\right)^{\mathrm{T}}\right]^{\mathrm{T}} \\ \left[\delta\left(\boldsymbol{R}_{A_i}\boldsymbol{R}_{M_i}\boldsymbol{e}\right)\times\left(\boldsymbol{e}^{T}\,{}^{O}_{A_i}\boldsymbol{R}\right)^{\mathrm{T}}\right]^{\mathrm{T}} \\ -\boldsymbol{e}^{\mathrm{T}}\,{}^{O}_{A}\boldsymbol{R} \end{bmatrix}^{\mathrm{T}}. \tag{6}$$

For serial modules, the modified Denavit–Hartenberg model is used. The homogeneous pose matrix of adjacent joints can be expressed as

$${}^{i-1}_{i}\boldsymbol{T} = R\left(x,\alpha_{i-1}\right)T\left(x,a_{i-1}\right)R\left(z,\theta_i\right)T\left(z,d_i\right) \tag{7}$$

where the link length, joint twist angle, link deflection, and rotation angle are kinematic parameters of the ith joint. The pose deviation matrix $\boldsymbol{\Delta T}$ represents the disparity between the actual transformation matrix $\boldsymbol{T}_{\text{real}}$ and the nominal pose matrix $\boldsymbol{T}_{\text{no}}$ relative to base coordinates

$$\boldsymbol{\Delta T} = \boldsymbol{T}_{\text{real}}\,\boldsymbol{T}_{\text{no}}^{-1} = \boldsymbol{T}\left(\boldsymbol{q}+\boldsymbol{\Delta q}\right)\boldsymbol{T}\left(\boldsymbol{q}\right)^{-1} \tag{8}$$

where $\boldsymbol{q}$ and $\boldsymbol{\Delta q}$ are the nominal kinematic parameters and the actual kinematic parameter errors, respectively.

Full differentiation of the pose errors leads to

$$\boldsymbol{\Delta T} = \frac{\partial\boldsymbol{T}}{\partial\alpha_i}\Delta\alpha_i + \frac{\partial\boldsymbol{T}}{\partial a_i}\Delta a_i + \frac{\partial\boldsymbol{T}}{\partial\theta_i}\Delta\theta_i + \frac{\partial\boldsymbol{T}}{\partial d_i}\Delta d_i = \boldsymbol{J}_s\boldsymbol{\Delta q} \tag{9}$$

where $\boldsymbol{J}_s$ is the serial module error Jacobi matrix.

B. EKF-Based Geometric Calibration

Considering the impact of measurement noise, the extended Kalman filter (EKF) is considered an effective method for parameter identification in the presence of noise in nonlinear systems. The geometric parameter error and the robot end pose error vector are defined as state and observed variables in the filtering algorithm. The linear differential equation governing the constant process can be expressed as

$$\boldsymbol{X}_k = \boldsymbol{X}_{k-1} + \boldsymbol{\mu}_{k-1} \tag{10}$$

where $\boldsymbol{X}_k$ is the vector of the robot geometric errors at measurement configuration k. $\boldsymbol{\mu}_{k-1}$ denotes a vector of a sequence of white noise with $\boldsymbol{E}(\boldsymbol{\mu}_{k-1}) = 0$ and $\boldsymbol{Q}_{k-1} = \mathrm{Cov}(\boldsymbol{\mu}_{k-1})$ representing the expection and covariance matrix of $\boldsymbol{\mu}_{k-1}$. The measurement equation for the kinematic parameter identification can then be expressed as

$$\boldsymbol{P}_k = \boldsymbol{J}_k\boldsymbol{X}_k + \boldsymbol{\xi}_k \tag{11}$$

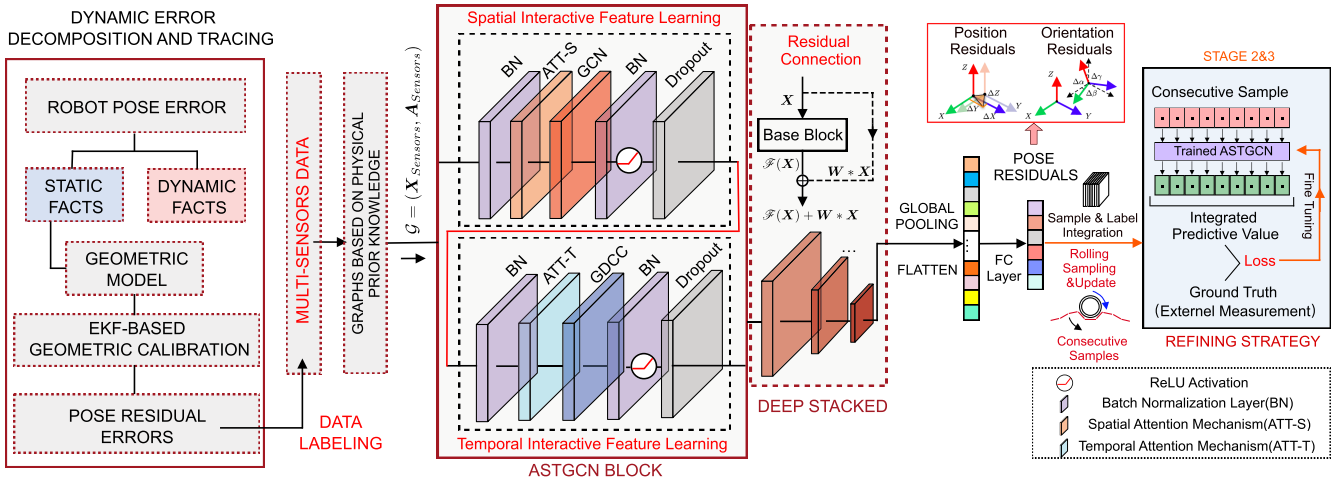


Fig. 3. Architecture of the proposed spatio-temporal framework for prediction.

where $\boldsymbol{\xi}_k$ is the observation noise satisfying the Gauss-Markov process. The covariance matrix of the vector $\boldsymbol{\xi}_k$ is $\boldsymbol{R}_k = \mathrm{Cov}(\boldsymbol{\xi}_k)$. The equations of the EKF process for geometric errors identification are formulated thus

$$\hat{\boldsymbol{X}}_{k|k-1} = \hat{\boldsymbol{X}}_{k-1|k-1} \tag{12}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_{k-1|k-1} + \boldsymbol{Q}_{k-1} \tag{13}$$

where the vector $\hat{\boldsymbol{X}}_{k|k-1}$ represents the robot geometric errors at the measurement configuration k without considering pose error, the vector $\hat{\boldsymbol{X}}_{k|k}$ is an update of the vector $\hat{\boldsymbol{X}}_{k|k-1}$ considering the pose error vector $\boldsymbol{P}_k$. Thus, the optimal Kalman gain $\boldsymbol{K}_k$ can be expressed as

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{J}_k^{\mathrm{T}}\left(\boldsymbol{J}_k\boldsymbol{P}_{k|k-1}\boldsymbol{J}_k^{\mathrm{T}} + \boldsymbol{R}_k\right)^{-1}. \tag{14}$$

The recursive identification equation is expressed by

$$\hat{\boldsymbol{X}}_{k|k} = \hat{\boldsymbol{X}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{P}_k - \boldsymbol{J}_k\hat{\boldsymbol{X}}_{k|k-1}\right). \tag{15}$$

The covariance estimate is update by the Riccati equation

$$\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{J}_k\right)\boldsymbol{P}_{k|k-1}. \tag{16}$$

Consequently, the criterion for the terminal of the identification process is expressed as

$$\boldsymbol{\Gamma} = \left\|\hat{\boldsymbol{X}}_{k|k} - \hat{\boldsymbol{X}}_{k-1|k-1}\right\| \tag{17}$$

where $\boldsymbol{\Gamma} \leq 10^{-4}$ is used in the paremeter calibration process. After obtaining the error of the optimal robot kinematic parameters, then the real kinematic parameters yield

$$\boldsymbol{X}_{\mathrm{Real}} = \boldsymbol{X}_{\mathrm{Nom}} + \boldsymbol{\delta X}. \tag{18}$$

Even though the robot has undergone calibration to address geometric errors, the accuracy of the robot can still be affected by nongeometric errors arising from other factors. The contribution of residual errors to the pose error of the robot can be determined as

$$\delta\boldsymbol{T}_{ng} = \boldsymbol{T}_M\,\boldsymbol{T}_{cg}^{-1} \tag{19}$$

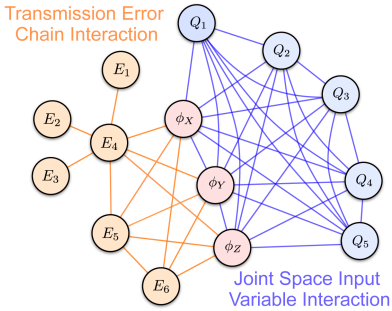


Fig. 4. Construction of spatio-temporal graphs.

where $\boldsymbol{T}_{ng}$, $\boldsymbol{T}_M$, $\boldsymbol{T}_{cg}$ denote the pose error factors caused by residual errors, the actual measured values of the end effector pose, and the calibrated kinematic positive pose, respectively.

III. Pose Residual Errors Prediction Framework

In this section, we propose an integrated attention mechanism spatio-temporal graph wavelet convolutional network (IAST-GCN) to capture the spatio-temporal dependencies in robot processing pose residual errors. The framework for trajectory residual prediction is illustrated in Fig. 3. Importantly, this study incorporates the robot's structural information to construct a non-Euclidean spatial graph for the sensor data. An evolving non-Euclidean structural graph is established to unveil the interplay between robot motion and trajectory residual errors.

A. Prior Knowledge-Based Graph Construction

We have constructed a representation of intricate interaction processes in joint space based on sensor variables through the established graph, as illustrated in Fig. 4 and Table II. In this graphical representation, sensors are represented as nodes, and the edges connecting them symbolize information exchange between sensors. It is noteworthy that while the error transmission process in traditional robot modeling is typically unidirectional, real-world scenarios involve complex joint coupling effects. Hence, errors will not propagate strictly in a

TABLE II
DEFINITION OF NODES IN GRAPHS

S/N	Node	Definition
1	E_1	Driving error of the UPS 1# in the parallel module.
2	E_2	Driving error of the UPS 2# in the parallel module.
3	E_3	Driving error of the UPS 3# in the parallel module.
4	E_4	Driven error of UP limb in the parallel module.
5	E_5	Angular error of the 1st joint in serial module.
6	E_6	Angular error of the 2nd joint in serial module.
7	Q_1	Actual inputs of the UPS 1# in the parallel module.
8	Q_2	Actual input of the UPS 2# in the parallel module.
9	Q_3	Actual input of the UPS 3# in the parallel module.
10	Q_4	Actual input of the 1st joint in the serial module.
11	Q_5	Actual input of the 2nd joint in the serial module.
12	ϕ_X	Attitude of CCOS machining system in X-axis.
13	ϕ_Y	Attitude of CCOS machining system in Y-axis.
14	ϕ_Z	Attitude of CCOS machining system in Z-axis.

one-way manner. Subsequent joint movements or close-range movements can affect each other, leading to bidirectional propagation in local areas. Consequently, we represent the relationship between sensors as an undirected graph. To capture the spatio-temporal interaction process of complex spatial error transmission within a specific time period, we characterized it through continuous spatio-temporal graphs.

B. Differential Entropy Feature Extraction

For discrete variables, let $r(n)$ denote the filtered time-series signal. To effectively capture the information during mutation of signals, the differential entropy is defined as

$$D(n) = r\left(n+1\right) - r(n) \tag{20}$$

$$p(n) = \frac{|D(n)|}{\sum_1^{n-1} |D(i)|} \tag{21}$$

$$H(n) = -\sum_1^{N-1} p(i)\log_2 p(i). \tag{22}$$

The change amplitude of each signal point is determined by the difference calculation of $D(n)$, while the change in weight of each point relative to the overall signal is computed using $p(n)$. The numerical quantification of the change degree for each discrete signal point is achieved by performing entropy summation of the change weights using $H(n)$, which facilitates the identification of signal mutation features.

C. Spatio-Temporal Interaction Modeling

To capture the spatio-temporal correlation between the joint space variables' spatio-temporal sequence and the residual error in the robot end trajectory poses, a deep spatio-temporal graph convolutional neural network is employed. This network comprises a graph wavelet convolutional neural network with a spatial attention mechanism and a 1-D temporal dilated causal convolutional neural network with a temporal attention mechanism. Spatial features and temporal features are extracted by aggregating information from nodes in adjacent and neighboring phases.

1) Wavelet Convolution of Spatial Graph: An undirected weighted graph is generally defined as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{A})$, where $\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{A}$ denote the set of nodes, the set of connected edges, and the adjacency matrix of the graph $\boldsymbol{G}$, respectively. The number of node elements and edge set elements in the graph $\boldsymbol{G}$ are n, m. The graph adjacency matrix $\boldsymbol{A}$ satisfies

$$A_{ij} = \begin{cases} 1, \text{ if } e_{ij} \in \boldsymbol{E} \\ 0, \text{ if } e_{ij} \notin \boldsymbol{E}. \end{cases}, \ \boldsymbol{A} \in \Re^{n\times n}. \tag{23}$$

Then the normalized Laplacian matrix of the undirected graph is represented as

$$\boldsymbol{L} = \boldsymbol{I}_n - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{\frac{1}{2}}, \ \boldsymbol{L} \in \Re^{n\times n} \tag{24}$$

where diagonal graph degree matrix $\boldsymbol{D} \in \Re^{n\times n}$ and $\boldsymbol{I}_n$ is a unit matrix of order n. Since $\boldsymbol{L}$ is real symmetric, there exists an eigenvector matrix $\boldsymbol{U}$ and a set of eigenvalues $\boldsymbol{\Lambda}$

$$\boldsymbol{U} = (u_1, u_2, \ldots, u_n) \in \Re^{n\times n} \tag{25}$$

$$\boldsymbol{\Lambda} = \text{diag}\left(\lambda_1, \lambda_2, \ldots, \lambda_n\right). \tag{26}$$

The graph method uses the graph Fourier transform and standard convolution to define the graph convolution operation, and the graph convolution layer is defined as

$$\boldsymbol{g}_\theta *_G \boldsymbol{X} = \boldsymbol{U}\left(\boldsymbol{U}^T\boldsymbol{g}_\theta\right) \odot \left(\boldsymbol{U}^T\boldsymbol{X}\right) = \boldsymbol{U}\boldsymbol{g}_\theta\boldsymbol{U}^T\boldsymbol{X} \tag{27}$$

where $\boldsymbol{g}_\theta$ is the convolution kernel, $\boldsymbol{g}_\theta = \text{diag}(\theta) \in \Re^{n\times n}$. θ is the graph Fourier coefficient vector, $\theta \in \Re^n$. $\odot$ denotes the matrix Hadamard product.

Similar to the graph Fourier transform, another method of transforming a signal from the nodal domain to the spectral domain is defined using the wavelet transform

$$\boldsymbol{g}_\theta *_G \boldsymbol{X} = \boldsymbol{\psi}_s\left(\boldsymbol{\psi}_s^{-1}\boldsymbol{g}_\theta\right) \odot \left(\boldsymbol{\psi}_s^{-1}\boldsymbol{X}\right) = \boldsymbol{\psi}_s\boldsymbol{g}_\theta\boldsymbol{\psi}_s^{-1}\boldsymbol{X} \tag{28}$$

where the wavelet transform spectral space base is $\boldsymbol{\psi}_s = \boldsymbol{U}\boldsymbol{G}_S\boldsymbol{U}^T = \{\psi_{s_1}, \psi_{s_2}, \ldots, \psi_{s_n}\}$ and the heat kernel function scale matrix is $\boldsymbol{G}_S = \text{diag}(e^{s\lambda_i}\ |_{i=1:n}) \in \Re^{n\times n}$. $\boldsymbol{\psi}_s$ and $\boldsymbol{\psi}_s^{-1}$ are solved by the spectral graph wavelet transform (SGWT) for fast approximation with a computational complexity of $O(mK)$. In contrast to the graph Fourier transform, the graph convolution operation defined on the basis of the graph wavelet transform is obviously more computationally efficient. Thus, the graph wavelet convolution is defined as

$$\boldsymbol{Z} = \boldsymbol{\psi}_s\boldsymbol{\Theta}\boldsymbol{\psi}_s^{-1}\boldsymbol{X}\boldsymbol{W} = \widetilde{\boldsymbol{\psi}}\boldsymbol{X}\boldsymbol{W} \tag{29}$$

where the parameter matrix to be learned is $\boldsymbol{W} \in \Re^{d\times q}$ and the diagonal matrix of the graph convolution kernel is $\boldsymbol{\Theta} \in \Re^{n\times n}$.

2) Temporal Convolution: In modeling the temporal correlation of spatio-temporal graphs of trajectory residual errors, dilated causal convolution (DCC) is employed as a temporal dimensional convolution layer for temporal information fusion. This approach effectively reduces the number of network layers required when processing long sequence inputs. In a DCC network, the receptive field expands exponentially with the depth of the network layers. This exponential growth allows for a larger receptive field to be achieved by stacking fewer network layers. Compared to RNN-based methods, DCC processes long time-series data in a nonrecursive manner, effectively avoiding

the issue of gradient explosion. Assuming a 1-D time series $x \in \Re^T$ and a filter kernel $k \in \Re^K$ at graph node v_i, the DCC is defined as

$$x *_{\mathrm{DCC}} k = \sum_{s=0}^{K-1} k[s]\, x[t - d * s]. \tag{30}$$

where $*_{\mathrm{DCC}}$ denotes the DCC operation symbol. K and d are the convolution kernel size and the extension factor size, respectively.

To effectively characterize correlations in the time dimension, a gating mechanism is introduced and a gated time convolution layer is defined

$$\mathscr{Z}^{i} = \delta\left(\theta_1 *_{\mathrm{DCC}} \mathscr{Z}^{i-1} + b\right) \odot \sigma\left(\theta_2 *_{DCC} \mathscr{Z}^{i-1} + c\right) \tag{31}$$

where δ and σ are the tanh and sigmod activation functions, respectively. θ_1 and θ_2 are the convolution parameters in the gate convolution and dilation convolution.

D. Spatio-Temporal Attention Module

In a non-Euclidean spatio-temporal graph data space involving multiple sensor variables, the transmission of node information within a single time step and its evolution across different time steps contribute to the intricate spatio-temporal interaction characteristics of robot motion processes. Diverse node inputs within a single time step exert varying degrees of influence on interaction features, and different time steps manifest distinct dynamic impacts on the outcomes. To elevate the representation of advanced features in the robot motion, we introduce a spatio-temporal attention mechanism. This enhancement facilitates a more effective modeling of complex spatio-temporal features at higher levels, adjusting spatial and temporal correlations between adjacent nodes based on input data. A detailed elucidation of the adaptive aggregation process of spatio-temporal evolution information follows.

1) Spatial Attention Mechanism: To effectively extract the dynamic spatial correlation between nodes of different input variables, attention mechanisms are defined in the spatial dimension. As depicted in Fig. 5(a), given the local feature $\boldsymbol{\chi} \in \Re^{N\times C\times T}$, we perform elementwise multiplication with learnable weights $\boldsymbol{W}_1 \in \Re^T$ and $\boldsymbol{W}_3 \in \Re^C$. This series of weight adjustments enables the model to make more fine-grained adjustments to different channel and time step inputs in the spatio-temporal dimension, capturing dynamic correlations. We multiply the learnable weight $\boldsymbol{W}_2 \in \Re^{C\times T}$ by it and activate it as $\boldsymbol{S} \in \Re^{N\times N}$

$$\boldsymbol{S} = \boldsymbol{V}_s\, \sigma\left(\left(\boldsymbol{W}_1 \odot \boldsymbol{\chi}\right) \boldsymbol{W}_2 \left(\boldsymbol{W}_3 \odot \boldsymbol{\chi}\right)^{\mathrm{T}} + \boldsymbol{b}_s\right). \tag{32}$$

Here, $\boldsymbol{V}_s$ and $\boldsymbol{b}_s$ denote the weights to be learned. σ and $\odot$ represent the sigmoid function activation and the elementwise multiplication operation. Finally, the Softmax function is applied to normalize it, yielding the standard spatial dimension attention matrix $\boldsymbol{S}' \in \Re^{N\times N}$

$$\boldsymbol{S}'_{i,j} = \mathrm{softmax}\left(S_{i,j}\right) = \frac{\exp\left(S_{i,j}\right)}{\sum_{j=1}^{N} \exp\left(S_{i,j}\right)}. \tag{33}$$

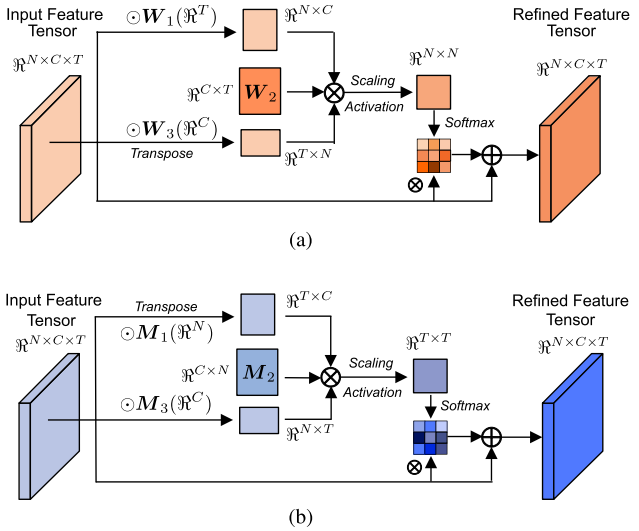


Fig. 5. Spatio-temporal attention module. (a) Spatial attention mechanism module. (b) Temporal attention mechanism module.

During graph convolution, the standard spatial attention matrix $\boldsymbol{S}'$ dynamically adjusts the importance of adjacent nodes when aggregating information in input-based graph convolution, which can be expressed mathematically as

$$\boldsymbol{\chi}^{(l)} = \left(\boldsymbol{S}' + \boldsymbol{E}\right) \boldsymbol{\chi}^{(l-1)}. \tag{34}$$

Here, $\boldsymbol{\chi}^{(l)}$ represents the node features at the lth layer of the graph convolution, and $\boldsymbol{\chi}^{(l-1)}$ represents the node features at the previous layer $(l-1)$. The operation $\boldsymbol{S}'\boldsymbol{\chi}^{(l-1)}$ involves multiplying the standard spatial attention matrix $\boldsymbol{S}'$ with the node features from the previous layer, adjusting the features based on the spatial attention information.

2) Temporal Attention Mechanism: Simultaneously, the non-Euclidean contextual information within each time step is interconnected. Leveraging the interdependence between contextual information allows us to emphasize the size of interdependent feature maps, improving feature representation. Consequently, we constructed a time dimension attention module to explicitly model the interaction between time dimensions.

Similarly, as depicted in Fig. 5(b), given a local feature $\boldsymbol{\chi} \in \Re^{N\times C\times T}$, we perform elementwise multiplication with learnable weights $\boldsymbol{M}_1 \in \Re^N$ and $\boldsymbol{M}_3 \in \Re^C$, enabling the model to make more fine-grained adjustments to different node and channel inputs in the spatio-temporal dimension, capturing dynamic correlations. We multiply the learnable weight $\boldsymbol{M}_2 \in \Re^{C\times N}$ by it and activate it as $\boldsymbol{Q} \in \Re^{T\times T}$

$$\boldsymbol{Q} = \boldsymbol{V}_t\, \sigma\left(\left(\left(\boldsymbol{M}_1 \odot \boldsymbol{\chi}\right)^{\mathrm{T}}\right) \boldsymbol{M}_2 \left(\boldsymbol{M}_3 \odot \boldsymbol{\chi}\right) + \boldsymbol{b}_t\right). \tag{35}$$

Here, $\boldsymbol{V}_t$ and $\boldsymbol{b}_t$ denote the weights to be learned. σ and $\odot$ represent the sigmoid function activation and the elementwise multiplication operation. Finally, the Softmax function is used to normalize it, resulting in the standard time dimension attention matrix $\boldsymbol{Q}' \in \Re^{T\times T}$

$$\boldsymbol{Q}'_{m,n} = \mathrm{softmax}\left(Q_{m,n}\right) = \frac{\exp\left(Q_{m,n}\right)}{\sum_{n=1}^{T} \exp\left(Q_{m,n}\right)}. \tag{36}$$

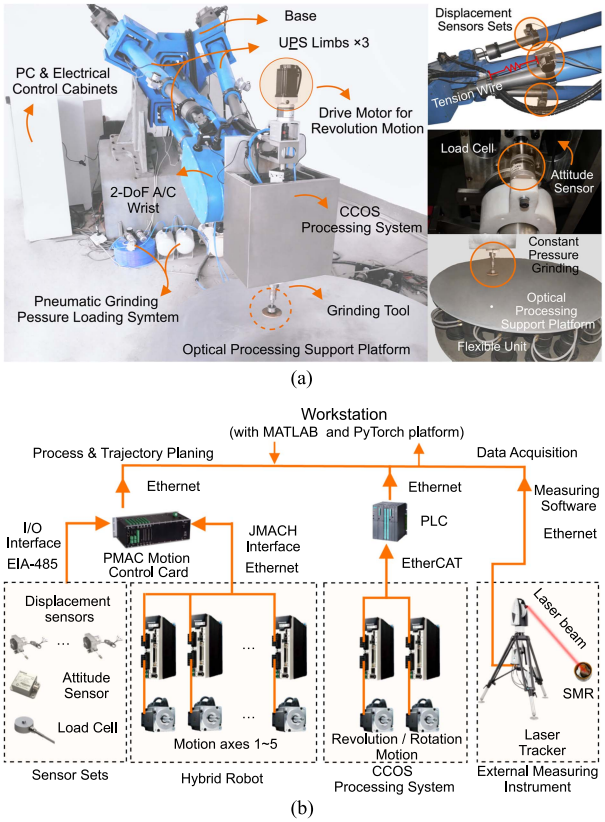


Fig. 6. Details of the experimental setup. (a) Experimental setup based on the CCOS hybrid robot. (b) Hardware structure of the experimental data acquisition process.

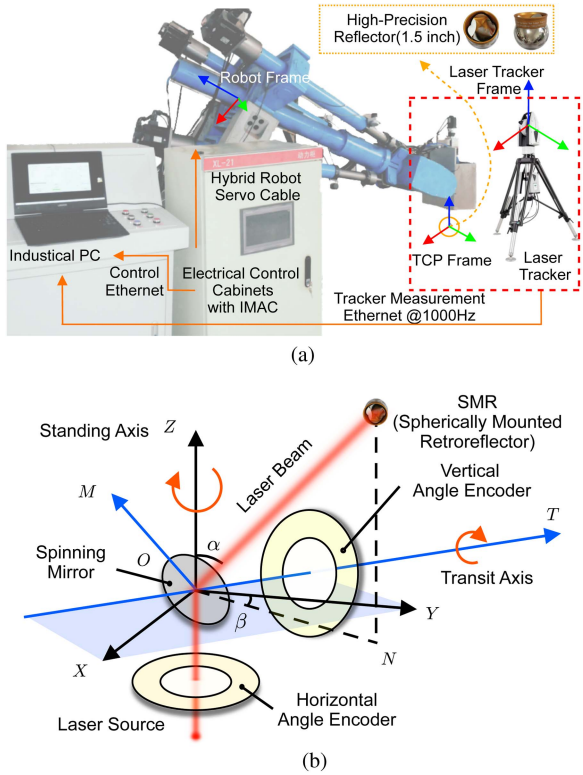


Fig. 7. Data acquisition process. (a) Layout of geometric parameters calibration process. (b) Laser tracker-based measurement process of pose residuals.

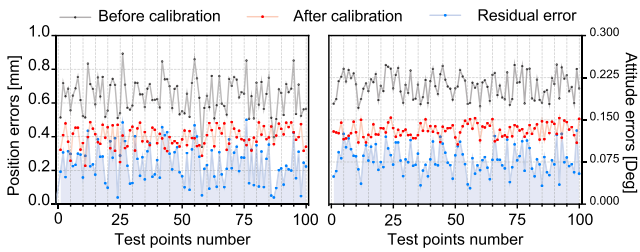


Fig. 8. Characterization of pose residual errors after calibration.

We directly apply the normalized temporal attention matrix to the input in order to dynamically adjusts the contribution level of each time step when aggregating information

$$\boldsymbol{\chi}^{(l)} = \boldsymbol{\chi}^{(l-1)} \left(\boldsymbol{Q}' + \boldsymbol{E} \right). \tag{37}$$

IV. Experiments and Discussion

To evaluate the performance of the proposed framework for residual error prediction in optical machining hybrid robots, a parametric calibration of the robot geometric errors is conducted followed by residual error prediction. The experimental setup, as depicted in Fig. 6, includes a hybrid robot equipped with a CCOS grinding system, an IMAC motion control system, a pneumatic grinding pressure loading platform, a hydraulic flexible supporting platform, and the external sensors. The calibration and prediction processes were executed on a server with an i7-12700F CPU and a RTX 3060 GPU, utilizing the MATLAB and PyTorch libraries, respectively.

A. Identification Process

The parameter calibration is conducted in a graded manner based on the robot configuration. The error parameters of the serial mechanism are calibrated first, followed by the calibration of the error parameters of the parallel mechanism. This approach enhances the efficiency and robustness of the entire identification process. The total positional error twist, denoted as $\boldsymbol{\xi}$, can be expressed as the superposition of the error twists of the two parts of the mechanism, $\boldsymbol{\xi}p$ and $\boldsymbol{\xi}s$

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{I}_3 & \widehat{\boldsymbol{T}}_o \\ \boldsymbol{0} & \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{\xi}_p + \begin{bmatrix} {}^T\boldsymbol{R}_o & \boldsymbol{0} \\ \boldsymbol{0} & {}^T\boldsymbol{R}_o \end{bmatrix} \boldsymbol{\xi}_s. \tag{38}$$

The experimental setup for the identification process is illustrated in Fig. 7. Consequently, the kinematic of the mechanism were revised to accurately represent the real conditions. To assess the effects of calibration, a comparison was made between motion accuracy before and after calibration, employing metrics such as absolute position error and absolute attitude error. The results demonstrate that solely compensating for geometric parameter errors in the robot leads to residual errors in the robot end poses, primarily attributed to dynamic nongeometric parameter errors, as illustrated in Fig. 8.

B. Data Acquisition and Description

The servo encoder values for three active chains and two rotating joints in the hybrid mechanism were obtained. The joint variables of the three active chains were acquired utilizing tension wire displacement sensors. Furthermore, force sensors and attitude sensor were employed to measure the pneumatic polishing force and the three-axis tilt angle of the CCOS grinding system, respectively. The collected sensor data necessitate preprocessing to address inherent noise and inconsistent sampling frequencies. Subsequently, noise reduction techniques were applied to sensor signals, and the raw data were normalized to facilitate feature extraction.

To predict the pose error changes within specific time intervals, the sensor sequence obtained from the laser tracker and sensors are partitioned using a fixed-length sliding time window. Each sliding window consists of 100 time steps and a step size of 20 time steps between consecutive windows, resulting in approximately 60 000 training examples. The input data for the model consisted of multidimensional time series sensor data, where each time step encompassed 20 sensor measurements. Each time window is assigned a prediction label based on the residual error calculated using (19). By utilizing the sensor data within each window, the magnitude of the pose error variation can be labeled.

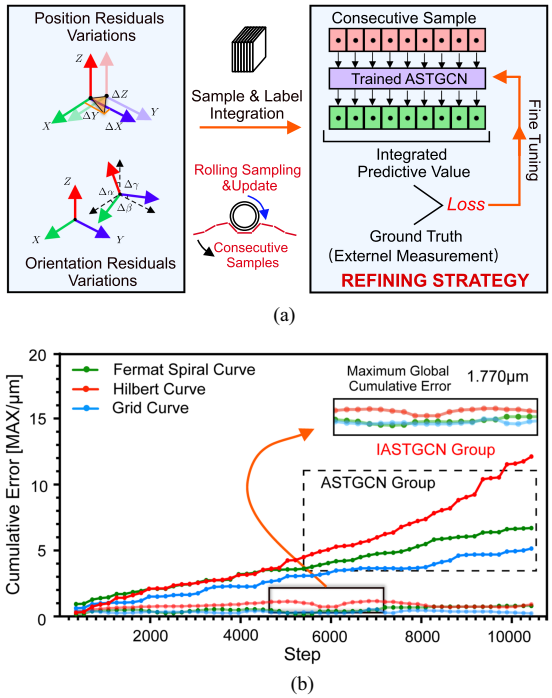


Fig. 9. Model refining strategy. (a) Warm start-based model refining strategy. (b) Cumulative errors with and without refinement.

C. Implementation Details

1) Model Training: Training samples were extracted from each time window using a sliding window approach, and a random sampling method was utilized to select these time windows. In the training process, 70% of the original samples were allocated to the training set, while 20% and 10% were assigned to the validation and test sets, respectively. The ASTGCN model employed a stacking block size of 5 and a temporal kernel size of 3 in the temporal convolutional layer. An inverse time decay variable learning rate strategy with an initial value of 1E-3 was implemented and a mini-batch size of 64 is used. The model implements early stopping according to the validation loss and the maximum training epoch is configured to be 150. In addition, dropout regularization with a rate of 0.2 was employed to mitigate the risk of overfitting. The loss function of the model is defined as

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix}\sum_{m=1}^{3}\left(\hat{Y}_{i,m}-Y_{i,m}\right)^2 \\ +\mathcal{P}\sum_{m=4}^{6}\left(\hat{Y}_{i,m}-Y_{i,m}\right)^2\end{bmatrix} + \lambda\|\mathcal{W}\|_2 \quad (39)$$

where N, Y, and $\hat{Y}$ represent the number of samples, the actual label, and the predicted value of the pose residual error variation, respectively. $\mathcal{P}$ and $\lambda\|\mathcal{W}\|_2$ denote the attitude error penalty factor for resolving label imbalance and L_2 regularization loss to the model parameters.

2) Refining Strategy: Despite the extensive training of the local model on multisensor sequences with an ample sample size, various confounding factors, such as truncation errors and measurement errors in the sliding time windows, continue to persist. These factors, in conjunction with data sample characteristics and biases, contribute to the unsatisfactory performance of the trained local model in accurately predicting long-term pose residual errors. To enhance the overall performance of the global model, a further refinement of the parameters of the finalized local model is proposed by incorporating pose errors accumulated over a specific time interval as labels. Subsequently, after training on samples obtained from individual time windows, multiple time windows are combined to generate new sample data for secondary training of the model. The initial algorithm, referred to as the single-sample algorithm, is outlined in Algorithm 1. It serves as the foundation for the subsequent fine-tuning procedure, depicted in Algorithm 2, which leverages the integrated samples to enhance the predictive capabilities of the model. By employing this refined approach, it is anticipated that the global model will exhibit improved performance, thereby addressing the limitations of long-term pose residual error predictions in the local model.

The schematic representation of the refinement strategy for the IASTGCN model, based on sample ensemble reconstruction, is depicted in Fig. 9(a). In addition, Fig. 9(b) offers a visualization of accumulated long-term pose residual prediction errors. After the warm start training of the local model, IASTGCN exhibits significant advantages in the long-term prediction of pose residuals compared to the ASTGCN model prior to the warm start. Following the second stage of training, the model effectively enhances the predictive accuracy of long-term pose residuals, rendering it suitable for extended-term prediction and aiding the model in better adapting to the global features of the target task. This approach simultaneously alleviates training

Algorithm 1: Training Procedure of Model: Core Stage.

Input: The defined graph structure, preprocessed training set with supervision labels, hyperparameters of model, e.g., mini-batch size, maximum epoch, number of ASTGCN stack blocks, learning rate schedule, size of kernel, the initialization strategy of weights, the size of sliding window.
Output: Trained ASTGCN model for pose residuals prediction
procedure ASTGCN
  Construct spatio-temporal graph $\mathcal{G}$ using sensor data $\mathcal{X}$
  Initialize layers, e.g., $\mathcal{L}$, $\mathcal{A}$ using He Uniform initialization [24].
  while not converged **do**
    Propagate data through graph convolutional layers
    $\mathcal{H} \leftarrow \mathcal{G}$
    for l in $\mathcal{L}$ **do**
      Applying spatial & temporal attention mechanism
      Applying stacked spatio-temporal graph convolution blocks
    end for
    Applying the FC Layer
    Compute loss function using (39)
    Update model parameters using backpropagation
  end while
end procedure

Algorithm 2: Refining Strategy: Warm Start Stage.

Input: Trained ASTGCN networks, newly integrated training set with new labels, hyperparameters of model, e.g., batch size, maximum epoch.
Output: Improved IASTGCN model
procedure IASTGCN
  Conduct training set sample reconstruction (Magnification = 10)
  while 2 rd model not converged **do**
    Compute loss function
    Update model parameters using backpropagation
  end while
  Update the training samples (Magnification = 5)
  while 3 rd model not converged **do**
    Compute loss function
    Update model parameters using backpropagation
  end while
  return Predicted the pose residual error $\hat{\mathbf{Y}}$
end procedure

complexities and assists the model in overcoming initial local minima, thereby bolstering overall robustness.

D. Comparison With State-of-the-Art ST-Model

An extensive comparison between the proposed method and state-of-the-art spatio-temporal models has been undertaken. In this comparative experiment, the evaluation encompasses the following representative models.

1) *Literature [25]:* ConvLSTM model, the architecture of which is built on 2-D Euclidean convolution and long short-term memory (LSTM) networks.
2) *Literature [23]:* T-GCN, the temporal GCN model, which is combined with the GCN and the gated recurrent unit (GRU).
3) *Literature [26]:* EAT-GCN, using Chebyshev GCN to extract spatial features and introducing evolutionary attention to improve the GRU to pay varying degrees of attention to window features across multiple time steps.
4) *Literature [27]:* ST-GAT, a hybrid spatio-temporal graph attention module, which consists of a multihead graph attention network (GAT) for extracting time-varying spatial features and a gated dilated convolutional network for temporal features.

1) Accuracy Performance Index: For performance evaluation, three widely utilized optical processing trajectories were employed, namely, grid trajectories, Fermat spiral trajectories, and Hilbert trajectories. A comprehensive evaluation of the models was conducted, and accuracy metrics, including mean absolute error (MAE), root-mean-square error (RMSE), and mean absolute percentage error (MAPE) for position residual bias as well as attitude residual bias, were reported across various datasets. Furthermore, within each dataset generated using the same trajectory, a horizontal comparison of the prediction accuracy for each model was conducted, as illustrated in Tables III and IV. The results indicate that IASTGCN demonstrated significantly higher prediction accuracy across all three datasets, comparable to the performance of ST-GAT. This superiority can be attributed to the simultaneous consideration of spatial and temporal correlations in these two models, whereas traditional 2-D convolution-based spatial feature extraction methods exhibited unimpressive accuracy. These findings underscore the viability and effectiveness of the proposed IASTGCN model, which adeptly captures the spatio-temporal coupling interaction characteristics among sensor data variables while considering their coupling relationships. The outcomes affirm the IASTGCN model's superiority in extracting advanced spatial interaction features using non-Euclidean graphs, validating its capacity to leverage spatio-temporal information and interactions between variables in sensor data.

2) Ablation Experiments: To evaluate the impact of individual components within IASTGCN on enhancing model performance, three modified versions were generated by removing or substituting the attention mechanism module, the gated dilation convolution module, and the spatial convolution module in IASTGCN. These modified versions were designated as IASTGCN-A, IASTGCN-T, and IASTGCN-S. The prediction results for IASTGCN and its three variants are presented in Table III. The results consistently demonstrate that IASTGCN outperforms the other models, highlighting the effectiveness of the excluded modules in capturing spatio-temporal correlations.

3) Computational Costs: In this study, we systematically quantified the computational costs of all models, including state-of-the-art spatio-temporal models and the variants. These

TABLE III
ACCURACY PERFORMANCE COMPARISON WITH OTHER STATE-OF-THE-ART BASELINE AND ABLATION MODELS

Trajectoriest	Indicators	ConvLSTM [25]	T-GCN [23]	EAT-GCN [26]	ST-GAT [27]	IASTGCN-A	IASTGCN-T	IASTGCN-S	**IASTGCN**
Grid Curve	MAE [μm/mDeg]	0.397/5.831	0.257/5.031	0.235/4.975	**0.219**/4.625	0.429/5.174	0.248/5.089	0.252/4.825	0.226/**4.551**
	RMSE [μm/mDeg]	0.509/6.731	0.319/5.924	0.321/5.815	0.301/**5.513**	0.598/6.067	0.324/5.847	0.307/5.920	**0.293**/5.781
	MAPE [%/%]	0.936/0.906	0.645/0.819	0.612/0.804	0.581/**0.732**	1.051/0.925	0.752/0.797	0.742/0.758	**0.566**/0.746
Fermat Spiral Curve	MAE [μm/mDeg]	0.387/5.721	0.190/4.599	0.187/4.564	0.141/4.087	0.287/5.464	0.161/4.258	0.209/4.150	**0.135/3.921**
	RMSE [μm/mDeg]	0.448/5.243	0.214/4.879	**0.153**/4.971	0.168/4.411	0.391/6.012	0.261/4.689	0.282/4.527	0.157/**4.345**
	MAPE [%/%]	0.692/0.894	0.416/0.714	0.398/0.721	0.374/0.667	0.611/0.782	0.421/0.714	0.500/0.675	**0.361/0.659**
Hilbert Curve	MAE [μm/mDeg]	0.610/6.317	0.517/5.874	0.495/5.716	**0.464/5.252**	0.685/6.389	0.621/5.914	0.599/5.686	0.487/5.260
	RMSE [μm/mDeg]	0.974/7.121	0.821/6.904	0.717/6.135	0.692/5.890	0.840/7.101	0.779/6.898	0.712/6.019	**0.681/5.800**
	MAPE [%/%]	1.104/1.124	0.912/0.984	0.843/0.939	0.781/**0.877**	1.256/1.066	1.011/0.992	0.914/0.926	**0.762**/0.892

[1] The bold result signifies the optimal model selection for this scenario.

TABLE IV
EVALUATION METRICS AND FORMULA DEFINITION

Evaluation metrics	Formula definition
MAPE	$1/z \sum_{i=1}^{z} \lvert\hat{y}_i - y_i\ /y_i\rvert$
RMSE	$\sqrt{1/z \sum_{i=1}^{z} (\hat{y}_i - y_i)^2}$
MAE	$1/z \sum_{i=1}^{z} \lvert\hat{y}_i - y_i\rvert$

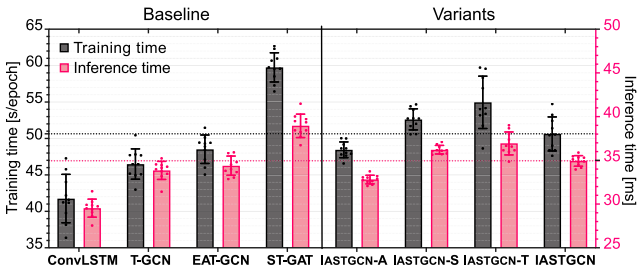


Fig. 10. Comparison of training time and inference time of the proposed method and variants with the baseline model.

TABLE V
ACCURACY PERFORMANCE COMPARISON WITH OTHER BASELINE ON RESIDUAL ERRORS PREDICTION

Accuracy metrics	Position error [mm]			Attitude error [Deg]		
	MEAN	S.D.	MAX.	MEAN	S.D.	MAX.
The initial	0.6267	0.0944	0.8915	0.1783	0.0182	0.2141
After calibration	0.1393	0.0626	0.2558	0.0811	0.0083	0.0904
BKLS-ANN [15]	0.1375	0.0513	0.2031	0.0231	0.0035	0.0270
ANN-BFPA [17]	0.1086	0.0254	0.1449	0.0249	0.0045	0.0327
ERC [19]	0.1073	0.0186	0.1377	0.0221	0.0029	0.0281
Proposed method	**0.0591**	**0.0085**	**0.0717**	**0.0183**	**0.0023**	**0.0212**

The bold values indicate the effective enhancement of the method proposed in this study within the comparative experiments.

computations were conducted within controlled hardware experimental environments on the PyTorch framework V2.1. Computational costs during training were taken into consideration by evaluating the average time required for a single epoch of model training. In addition, information on the inference time of the trained model is provided, as depicted in Fig. 10. In the training time costs for baseline models, it is evident that removing the spatio-temporal attention mechanism is a crucial factor in reducing training time. The ConvLSTM model, based on 2-D conventional Euclidean graph convolution, exhibits the shortest training time. However, it is worth noting that the spatio-temporal attention mechanism stands out as the component that most effectively enhances model accuracy, as indicated in Table III. Conversely, the ST-GAT model incurs the longest training time, surpassing the proposed IASTGCN model by 18.07%, because of its complex multihead attention mechanism. Nevertheless, the spatial convolution/temporal convolution ablation model of IASTGCN experiences an extension of more than 8.57% and 3.92%, respectively, due to the substitution of conventional Chebyshev graph convolution and spatial 1-D convolution. Examining the accuracy of multiple ablation models, the graph wavelet convolution and gated DCC components consistently outperform conventional graph convolution and time convolution in terms of both model prediction accuracy and computational cost. Concerning the inference time of the baseline model, it is noteworthy that IASTGCN has struck a balance between the baseline model with no or shallow attention mechanism and the ST-GAT model with a complex multihead attention mechanism. This balance has been achieved without compromising accuracy, resulting in a satisfactory performance for IASTGCN.

E. Exploration Study

1) Comparison With Offline Prediction Strategies: As mentioned earlier, the distribution of trajectory errors is influenced not only by the current frame input but also by the accumulation of historical states, indicating a spatio-temporal relationship and high coupling in the spatio-temporal convolution characteristics of each joint variable's impact on the end pose residual. Consequently, it is essential to compare the proposed online prediction method for robot pose residuals with other offline methods. We selected three other offline prediction methods for pose residuals as baseline models and evaluated tracking metrics on three optical processing trajectories. In optical mirror processing, the processing effect is particularly sensitive to the residual pose of the robot in the directions X and Y, given the pneumatic floating and constancy of the CCOS processing system at the robot's end and the main mirror surface in the direction Z. To illustrate, we conducted projection visualization of the pose residual in the spatial plane for the Hilbert processing trajectory, which exhibited the most significant spatial pose changes, as depicted in Fig. 11. Simultaneously, we compared the compensated pose accuracy of the baseline model, as detailed in Table V. The results indicate that, in comparison with other trajectory residual prediction models, the online pose residual prediction method

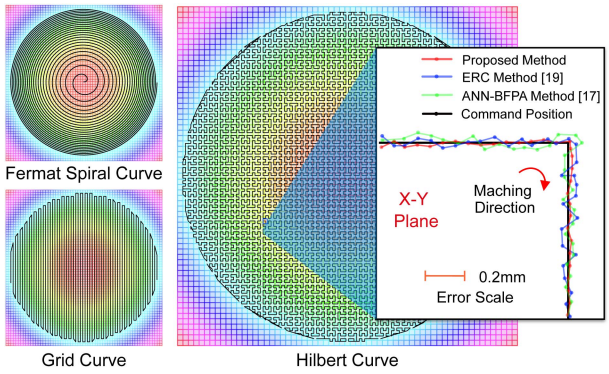


Fig. 11. Three typical CCOS optical machining trajectories and the error projection view under Hilbert machining trajectory real experiment.

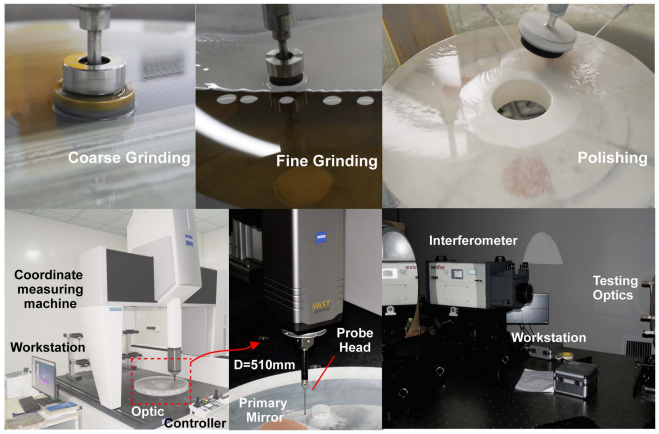


Fig. 12. Processing and testing of optical primary mirrors.

based on multivariate spatio-temporal interaction in joint space exhibits substantial improvements in both position error and attitude error. In contrast to the optimal value in the baseline model, the average position error is reduced by 44.92%, and the attitude error is reduced by 17.19%.

2) Real Experimental Verification: The processing of large aperture optical mirrors not only imposes stringent demands on the workspace of the processing robot but also establishes exceptionally strict criteria for processing accuracy. A comparative analysis of the optical processing performance was performed after we compensated for the pose residuals using the robotic compensation method in [28]. Subsequently, experimental mirrors were ground under two distinct working conditions following pose residual compensation, with meticulous control over the processing conditions, as illustrated in Fig. 12. In the grinding stage, a 510 mm optical primary mirror was utilized, and Fig. 13 illustrates the convergence curves of the machining surface under the aforementioned working conditions. The surface shape accuracy indicators at different stages are presented in Table VI. In the above grinding and polishing processes, we use a ZEISS three coordinate optical detector to detect the machined mirror surface during the grinding and rough polishing stages, and a Zygo interferometer to detect the machined mirror surface during fine polishing stage. Eventually, the peak to valley (PV)

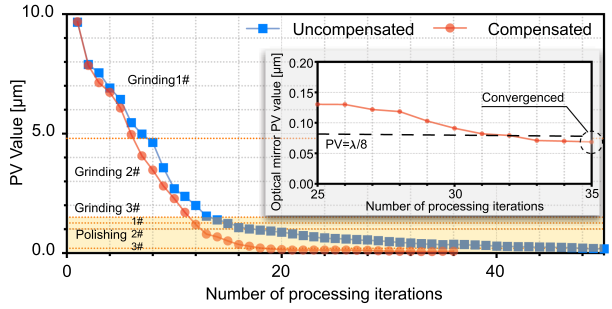
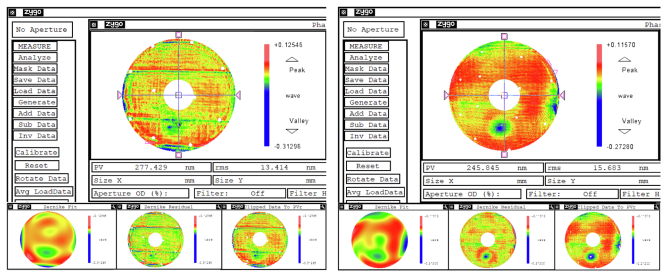
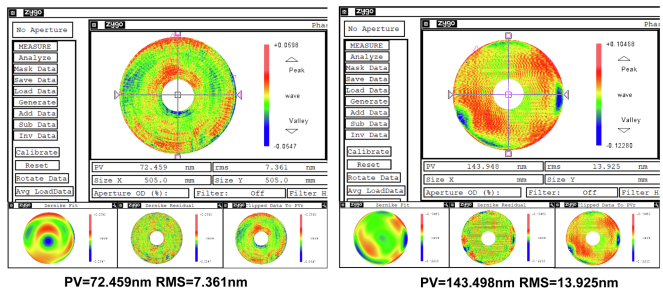


Fig. 13. Comparison of the convergence ability of the primary mirror manufacturing error between the two conditions.



(a)



(b)

Fig. 14. Surface error of the two experimental primary mirrors after same iterations as measured by the Zygo interferometer. (a) Initial surface error before machining. (b) Final surface error after machining.

TABLE VI
MACHINING PROCESS AND PRODUCTION DETAILS OF THE 510 MM DIAMETER OPTICAL PRIMARY MIRROR

Process	Tool	Abrasive particle size	Accuracy indicators [PV]
Grinding	1# Copper alloy grinding disc	W20	10.5μm
	2# Copper alloy grinding disc	W10	4.8μm
	3# Hard aluminum grinding disc	W7	1.5μm
Polishing	1# Polyurethane polishing pad	W5	2.0λ [1]
	2# Polyurethane polishing pad	W2.5	1.6λ
	3# Polyurethane polishing pad	W0.25	0.8λ

[1] The accuracy measurement wavelength λ is standardized at 632.8nm.

ratio and root mean square (RMS) of the mirror were obtained

$$\mathrm{PV} = \mathrm{Max}(\Delta W) - \mathrm{Min}(\Delta W) \tag{40}$$

$$\mathrm{RMS} = \sqrt{\frac{1}{\pi}\int_0^{2\pi}\int_0^1 [\Delta W(\rho,\theta) - \overline{\Delta W}]\rho d\rho d\theta} \tag{41}$$

where ΔW, ρ, θ represent the discrete surface error, the radius of the circular primary mirror, and the angle, respectively.

Fig. 14 depicts the individual processing of surface error under two conditions of pose residual during the continuous polishing stage. Prior to processing, the mirror detection results using our proposed method were PV = 277.429 nm, RMS = 13.414 nm, and for the uncompensated case, PV = 245.845 nm, RMS = 15.683 nm. However, after the same iterations for processing, the mirror detection results for our proposed method were PV = 72.459 nm, RMS = 7.361 nm, and for the uncompensated case, PV = 143.498 nm, RMS = 13.925 nm. This represented a decrease in PV values by 73.88% and 41.63%, and a decrease in RMS values by 45.12% and 11.21%, respectively, compared to preprocessing values. A comparison of interferometer surface error detection results before and after processing revealed that our proposed method improved the convergence performance of the surface by approximately 30% and further confirmed the effective improvement of optical processing certainty under accurate compensation of pose residuals.

V. Conclusion

This research presented a deep learning-based approach for predicting residual errors in the machining trajectory of an optical processing hybrid robot. Following the establishment of a geometric error model and the calibration of kinematics using the EKF algorithm, the IASTGCN model was employed to predict residual pose errors. Complex dynamic spatio-temporal interactions between sensor variables and pose residuals were addressed by the method. The efficacy of the proposed method was validated through real experiments, demonstrating its advanced performance. We validate the efficacy of the proposed method through real experiments, which demonstrated its advanced performance. In future investigations, we plan to tackle the following issues: 1) conduct an in-depth exploration of the dynamic accuracy of this optical processing robot when integrating multiple physical fields; and 2) enhance prediction stability by incorporating data fusion and reinforcement learning into the pose residual deep learning modeling method inspired by soft sensors, thus improving its performance.

References

[1] B. Lin, X. M. Jiang, Z. C. Cao, and T. Huang, "Development and theoretical analysis of novel center-inlet computer-controlled polishing process for high-efficiency polishing of optical surfaces," *Robot. Comput.- Integr. Manuf.*, vol. 59, pp. 1–12, 2019.

[2] J. Zhang and H. Wang, "Generic model of time-variant tool influence function and dwell-time algorithm for deterministic polishing," *Int. J. Mech. Sci.*, vol. 211, 2021, Art. no. 106795.

[3] X. Ke et al., "Review on robot-assisted polishing: Status and future trends," *Robot. Comput. Integr. Manuf.*, vol. 80, 2023, Art. no. 102482.

[4] C. Wang, Y. Han, H. Zhang, C. Liu, L. Jiang, and L. Qian, "Suppression of mid-spatial-frequency waviness by a universal random tree-shaped path in robotic bonnet polishing," *Opt. Exp.*, vol. 30, no. 16, pp. 29216–29233, 2022.

[5] T. Wang et al., "Universal dwell time optimization for deterministic optics fabrication," *Opt. Exp.*, vol. 29, no. 23, pp. 38737–38757, 2021.

[6] N. Shen et al., "Efficient model-free calibration of a 5-degree of freedom hybrid robot," *J. Mechanisms Robot.*, vol. 14, no. 5, 2022, Art. no. 051011.

[7] Z. He, Y. Song, B. Lian, and T. Sun, "Kinematic calibration of a 6-DoF parallel manipulator with random and less measurements," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 7500912.

[8] J. He, Q. Ding, F. Gao, and H. Zhang, "Kinematic calibration methodology of hybrid manipulator containing parallel topology with main limb," *Measurement*, vol. 152, 2020, Art. no. 107334.

[9] X. Chen, Q. Zhang, and Y. Sun, "Non-kinematic calibration of industrial robots using a rigid–flexible coupling error model and a full pose measurement method," *Robot. Comput. Integr. Manuf.*, vol. 57, pp. 46–58, 2019.

[10] T. Sun, B. Lian, S. Yang, and Y. Song, "Kinematic calibration of serial and parallel robots based on finite and instantaneous screw theory," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 816–834, Jun. 2020.

[11] T. Sun, C. Liu, B. Lian, P. Wang, and Y. Song, "Calibration for precision kinematic control of an articulated serial robot," *IEEE Trans. Ind. Electron.*, vol. 68, no. 7, pp. 6000–6009, Jul. 2021.

[12] Z. Li, S. Li, and X. Luo, "Efficient industrial robot calibration via a novel unscented Kalman filter-incorporated variable step-size Levenberg-Marquardt algorithm," *IEEE Trans. Instrum. Meas.*, vol. 72, Apr. 2023, Art. no. 2510012, doi: 10.1109/TIM.2023.3265744.

[13] J. Luo et al., "Kinematic calibration of a 4PPa-2PaR parallel mechanism with subchains on limbs," *IEEE Trans. Instrum. Meas.*, vol. 71, Mar. 2022, Art. no. 7502011, doi: 10.1109/TIM.2022.3156998.

[14] H. Liu, Z. Yan, and J. Xiao, "Pose error prediction and real-time compensation of a 5-DoF hybrid robot," *Mechanism Mach. Theory*, vol. 170, 2022, Art. no. 104737.

[15] S. Liao, Q. Zeng, K. F. Ehmann, and J. Cao, "Parameter identification and nonparametric calibration of the tri-pyramid robot," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 5, pp. 2309–2317, Oct. 2020.

[16] D. Yu, "A new pose accuracy compensation method for parallel manipulators based on hybrid artificial neural network," *Neural Comput. Appl.*, vol. 33, pp. 909–923, 2021.

[17] H. Q. Cao, H. X. Nguyen, T. N. C. Tran, H. N. Tran, and J. W. Jeon, "A robot calibration method using a neural network based on a butterfly and flower pollination algorithm," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3865–3875, Apr. 2022.

[18] W. Wang, W. Tian, W. Liao, B. Li, and J. Hu, "Error compensation of industrial robot based on deep belief network and error similarity," *Robot. Comput. Integr. Manuf.*, vol. 73, 2022, Art. no. 102220.

[19] W. Yang, S. Li, Z. Li, and X. Luo, "Highly accurate manipulator calibration via extended Kalman filter-incorporated residual neural network," *IEEE Trans. Ind. Inform.*, vol. 19, no. 11, pp. 10831–10841, Nov. 2023.

[20] J. Zhou, L. Zheng, W. Fan, X. Zhang, and Y. Cao, "Adaptive hierarchical positioning error compensation for long-term service of industrial robots based on incremental learning with fixed-length memory window and incremental model reconstruction," *Robot. Comput. Integr. Manuf.*, vol. 84, 2023, Art. no. 102590.

[21] R. W. Liu et al., "STMGCN: Mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multigraph convolutional network," *IEEE Trans. Ind. Inform.*, vol. 18, no. 11, pp. 7977–7987, Nov. 2022.

[22] Y. Jiang, P. Dai, P. Fang, R. Y. Zhong, and X. Cao, "Electrical-STGCN: An electrical spatio-temporal graph convolutional network for intelligent predictive maintenance," *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 8509–8518, Dec. 2022.

[23] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[25] Z. Zhang and Y. Wang, "A spatio-temporal model for global earthquake prediction based on convolutional LSTM," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5915712.

[26] L. Cheng, L. Li, S. Li, S. Ran, Z. Zhang, and Y. Zhang, "Prediction of gas concentration evolution with evolutionary attention-based temporal graph convolutional network," *Expert Syst. Appl.*, vol. 200, 2022, Art. no. 116944.

[27] Y. Wang, C. Jing, S. Xu, and T. Guo, "Attention based spatio-temporal graph attention networks for traffic flow forecasting," *Inf. Sci.*, vol. 607, pp. 869–883, 2022.

[28] R. Li, N. Ding, Y. Zhao, and H. Liu, "Real-time trajectory position error compensation technology of industrial robot," *Measurement*, vol. 208, 2023, Art. no. 112418.

Jun Li received the M.Sc. degree in mechanical design and theory in 2019 from the China University of Mining and Technology, Xuzhou, China, where he is currently pursuing the Ph.D. degree in mechanical engineering.

His current research interests include deep learning-based optical processing robotics and modern control theory.

Gang Cheng received the B.Sc. degree in mechanical design and theory from Wuhan University, Wuhan, China, in 2000, the M.Sc. degree in optoelectronic engineering from the Institute of Optoelectronic Technology, Chinese Academy of Sciences, Chengdu, China, in 2003, and the Ph.D. degree in mechanical design and theory from the China University of Mining and Technology, Xuzhou, China.

He is currently a Professor with the China University of Mining and Technology. His research interests include mechanics and fault diagnosis technology of large electromechanical equipment.

Yusong Pang received the M.Sc. degree in electrical engineering from Taiyuan University of Technology, Taiyuan, China, in 1996, and the Ph.D. degree in Intelligent Belt Conveyor Monitoring and control from the Delft University of Technology, Delft, The Netherlands, in 2007.

In 2000, he started working at Practic B.V. and Seaview B.V., Ijmuiden, The Netherlands, for industrial production life cycle management. He was at the Advisory Group Industrial Installations of Royal Haskoning, Utrecht, The Netherlands, as an Expert Material Handling. In 2010, he was appointed as an Assistant Professor with the Department of Transport Engineering and Logistics, Delft University of Technology. His research focuses on the intelligent control for large-scale material handling systems and logistics processes.