IJRCS
International Journal of Robotics and Control Systems
Vol. 6, No. 2, 2026, pp. 980-998
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs
ASCEE

MLP-Based Inverse Kinematics for Efficient Pick and Place Motion Generation of 6-DOF Robotic Manipulators

Andara Dwi Hermawan [a,1], Aufaclav Zatu Kusuma Frisky [a,2,*], Andi Dharmawan [a,3]

[a] Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia
[1] andaradwihermawan@mail.ugm.ac.id; [2] aufaclav@ugm.ac.id; [3] andi_dharmawan@ugm.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received December 24, 2025
Revised February 04, 2026
Accepted April 03, 2026

Keywords
Inverse Kinematics;
Multilayer Perceptron;
Joint-Limit Constraints;
Pick and Place;
Pybullet

ABSTRACT

Control systems for six degree of freedom (6-DOF) robotic manipulators in pick and place tasks commonly rely on classical inverse kinematics (IK) solvers that are iterative, sensitive to initialization, and can become unreliable near workspace boundaries. These limitations may reduce motion stability in repetitive execution and increase computation overhead due to repeated numerical refinement. This paper proposes a Multi-Layer Perceptron (MLP)-based inverse kinematics approach that provides deterministic feed-forward prediction while enforcing feasibility through joint-limit constraints. Although the robot is described as a 6-DOF manipulator, the IK mapping in this study targets five actuated revolute joints, while the sixth joint corresponds to the gripper and is excluded from the learning target. An end-to-end pipeline is developed covering URDF-based dataset generation, constrained learning with a bounded output layer, and deployment in both simulation and physical experiments. The proposed model is evaluated using end-effector point testing and pick and place motion execution in PyBullet, compared with a Jacobian-based numerical IK baseline implemented in IKPy (warm-start), the proposed method improves the success rate from 40% to 65% ($\tau = 10mm$), reduces the mean Cartesian position error from 18.38 mm to 4.28 mm, and decreases pick/place end points error from 44.2/53.86 mm to 13.8/25.6 mm, respectively. The average runtime per target is reduced from 20 ms to 7.46 ms. These results indicate that the proposed MLP-based IK offers a practical accuracy efficiency trade off for real time pick and place applications.

1. Introduction

Industrial automation in the industry 4.0 era increasingly demands robotic systems that are accurate, repeatable, and adaptive for high-throughput manipulation tasks such as pick and place [1]–[5]. Among various robotic platforms, six-degree-of-freedom (6-DOF) manipulators are widely adopted because they provide sufficient dexterity to reach target poses across a workspace while maintaining operational flexibility in constrained environments [6]–[10]. In practical deployments, however, manipulation accuracy and stability are strongly influenced by how reliably the controller convert an end-effector

Cartesian target into feasible joint commands under hardware limits, nonlinearities, and repetitive motion requirements [11]–[15].

A key computational component in manipulator control is inverse kinematics (IK), which maps desired Cartesian targets to joint angles [16]–[21]. Classical IK approaches include analytical formulations for specific structures or numerical/iterative solvers such as Jacobian-based methods [22]–[27]. Despite their generality and ease of integration, iterative IK solvers can be sensitive to initialization, may require iterative refinement with non-uniform computation time, and can yield variable solutions in repetitive task [28]–[31]. In pick-and-place operations, such variability may increase positioning error near workspace boundaries, lengthen cycle time, and raise the risk of collision in cluttered workspace [31]–[35]. These challenges motivate IK solutions that are computationally efficient, stable for repeated execution, and consistent with kinematic constraints.

Recently, learning-based IK has been explored as an alternative to classical solvers by directly approximating the inverse mapping from end-effector positions to joint configurations [36]–[40]. While promising, many approaches still face practical limitations, including limited enforcement of joint limits during inference, difficulty handling periodicity in revolute joints, and insufficient validation under sequential pick and place motion [41]–[44]. Moreover, reported speed accuracy trade-offs are sometimes confounded by system level execution settings (e.g., trajectory discretization and controller update rate) rather than changes in the underlying model itself [45]–[47].

Learning based IK can be implemented with various function approximators (e.g., MLPs graph based models, or sequence models) [48]. In this work, we choose an MLP backbone because the problem is a low dimensional continuous regression from Cartesian position (x, y, z) to small set of actuated joint angles $(q_1, q_2, q_3, q_4, q_5)$ making a fully connected regressor a natural and lightweight choice. Compared to more complex architecture, an MLP offers fast and deterministic inference that is well-suited for real-time control loops, while remaining expressive enough to approximate the nonlinear kinematic mapping over the reachable workspace [49], [50]. Prior learning-based IK studies have reported that MLP-style regressors can reduce inference latency relative to iterative solvers, and that feature encoding and explicit constraint handling can improve accuracy and feasible [51], [52].

Motivated by these gaps, this paper proposes an end-to-end MLP based inverse kinematics pipeline for a 6-DOF robotic manipulator model described in URDF [53]–[55]. In our implementation, IK learning focuses on the five actuated positioning joints, while the remaining joint corresponds to the gripper mechanism and is excluded from the IK mapping [56]–[58]. The proposed method integrates (i) URDF-based dataset generation using a numerical IK solver to obtain paired Cartesian targets and joint-angle labels over a bounded workspace [59], (ii) data preprocessing with global normalization and Fourier feature encoding of Cartesian inputs to improve representational capacity [60], and (iii) a constraint aware network design (ResMLPBound) that bounds predicted joint angles within URDF-defined limits [61]. Comprehensive validation is conducted through end effector point testing, trajectory-based evaluation under different execution presets that vary controller level settings (without modifying network weights), and comparison against a Jacobian-based numerical IK baseline implemented in IKPy [62], followed by validation on a physical manipulator.

Building on these observations, we design a practical learning-based IK pipeline that combines scalable dataset generation, constraint aware learning, and deployment-oriented evaluation. We demonstrate its effectiveness through simulation and physical experiments, and provide a baseline comparison against a standard numerical IK solver. Our contributions are summarized as follows:

- We present an end-to-end MLP-based IK pipeline for URDF-defined manipulators covering dataset generation, training, and deployment in simulation and on a physical robot.
- We introduce a constraint-aware bounded output formulation that enforces URDF joint limits by design, improving feasibility at inference time.

- We provide comprehensive evaluation on point -wise targets and continuous motion (trajectory and pick and place), including system level accuracy throughput trade offs under controller level execution presets.
- We benchmark against a Jacobian-based numerical IK baseline (IKPy, warm-start) under identical workspace and joint limit constraints using success rate, error metrics, and runtime.

2. Method

2.1. System Overview

This study proposes an end-to-end inverse kinematics framework based on a Multi-Layer Perceptron (MLP) for a six-degree-of-freedom (6-DOF) robotic manipulator, as illustrated in Fig. 1. The objective is to map Cartesian end-effector target positions to physically feasible joint configurations in a stable and computationally efficient manner for pick and place applications. The framework integrates dataset generation, data preprocessing, MLP-based learning, and deployment within a unified pipeline.

The dataset is generated using the robot URDF model and a numerical universal inverse kinematics solver to obtain paired end-effector positions an joint-angle configurations across the reachable workspace. Although the robot has six joints, the IK mapping in this works predicts the five actuated positioning joints, while the sixth joint corresponds to the gripper and is excluded from the leraning target. The cartesian inputs are normalized and augmented with Fourier features before being fed into a constraint-aware MLP model with bounded output (ResMLPBound) that enforces URDF-defined joint limits. The trained model is evaluated through end-effector point testing and pick and place trajectory execution in the PyBullet simulation environment, followed by validation on a physical robotic manipulator.

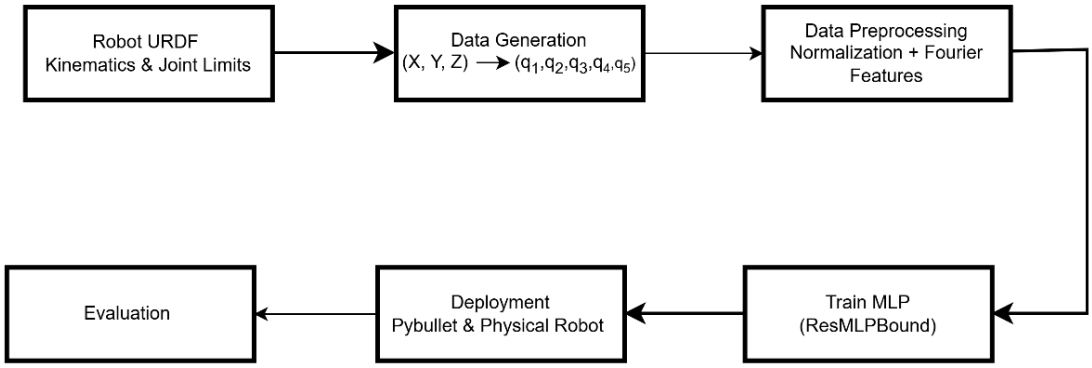

Fig. 1. Overall pipeline of the proposed MLP-based inverse kinematics framework: URDF-based dataset generation, preprocessing (normalization and Fourier features), constraint-aware MLP training (ResMLPBound), and deployment and evaluation in Pybullet and on a physical manipulator

As summarized in Fig. 1, the proposed pipeline starts from URDF-based dataset generation, followed by normalization and fourier feature encoding of cartesian inputs. The constraints- aware MLP (ResMLPBound) then predicts joint angles within URDF-defined limits, and the predicted commands are executed sequentially to produce smooth pick and place motion in Pybullet and on the physical robot.

2.2. Dataset Generation

The dataset used in this study was generated to represent the inverse kinematics mapping of a six degree of freedom (6-DOF) robotic manipulators across its reachable workspace. A robot description model in the Unified Robot Description Format (URDF) was employed to define the manipulator's kinematic structure, joint limits, and link parameters. Based on this URDF model, a numerical universal

inverse kinematics solver was utilized to compute valid joint configurations corresponding to target end effector positions. End effector positions were systematically sampled within the robot workspace using a cartesian grid with a spatial resolution of 1 mm step along each axis, ensuring dense and uniform coverage. The sampling bounds were defined as $x \in [-0.20, 0.35]\, m$, $y \in [-0.20, 0.20]\, m$, $z \in [-0.20, 0.35]\, m$, covering the reachable region while avoiding physically infeasible targets.

Although the robot is described as a 6-DOF manipulator in the URDF, this study models the positioning kinematics using five actuated revolute joints. The remaining joint corresponds to the end-effector/ gripper mechanism and is excluded from the inverse kinematics mapping. Therefore, the learning target for the proposed model is a five-dimensional joint-angle vector $(q_1, q_2, q_3, q_4, q_5)$, consistent with the active joint set used during dataset generation and evaluation. The joint indexing and the excluded gripper joint are illustrated in Fig. 2.

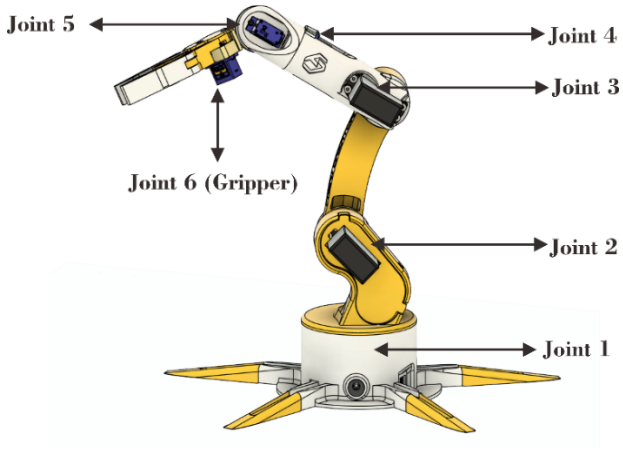

Fig. 2. Joint indexing used in this study. The IK mapping predicts the five positioning joints (joint 1 – joint 5), while Joint 6 corresponds to the gripper mechanism and is excluded from the IK mapping

For each sampled Cartesian position (x, y, z) the inverse kinematics solver was executed to obtain a feasible joint-angle vector $(q_1, q_2, q_3, q_4, q_5)$ that satisfied the kinematic constraints defined in the URDF. Only solutions that respected joint-limit constraints were retained, guaranteeing that all samples corresponded to physically realizable robot configurations. In total, the sampling process produced 60.471.201 target positions, from which 34.431.743 valid joint configurations were retained after applying feasibility and joint-limit filtering.

The resulting dataset consists of paired input-output samples mapping Cartesian end-effector positions to joint angles. The data were split into training, validation, and test subsets using a ratio of 80%: 10%: 10% to support model selection and unbiased evaluation.

2.3. Data Preprocessing and Feature Engineering

Before training the MLP-based inverse kinematics model, the generated dataset underwent a preprocessing stage to improve numerical stability and learning efficiency. Cartesian end effector positions (x, y, z) were normalized using global mean (μ) and standard deviation (σ) computed from the training dataset. The normalized input $\tilde{\boldsymbol{x}}$ is defined in (1) as:

$$\tilde{x} = \frac{x - \mu}{\sigma};\ x = [x, y, z]^T \tag{1}$$

This global normalization ensures consistent scaling across training, validation, and test subsets, and prevent data leakage during evaluation and deployment.

To enhance the model's capability in representing nonlinear and periodic relationships in joint space, Fourier feature augmentation was applied to the normalized Cartesian inputs. The Fourier mapping projects $\tilde{\boldsymbol{x}}$ into a higher-dimensional feature space using sinusoidal functions across multiple frequency bands, as defined in (2):

$$\emptyset(\tilde{x}) = \left[\tilde{x}, \left\{sin\left(2\pi \cdot 2^k \tilde{x}\right), cos\left(2\pi \cdot 2^k \tilde{x}\right)\right\}_{k=0}^{K-1}\right] \quad (2)$$

In our implementation, the mapping in (2) is applied element wise to each coordinate ot the three-dimensional input $\tilde{x} = [\tilde{x}, \tilde{y}, \tilde{z}]^T$. The frequency bands follow $f_k = 2^k$ with angular frequency $\omega_k = 2\pi f_k$, where $k = 0, \ldots, k-1$ and K denotes the number of frequency bands. Since the cartesian input is three-dimensional ($\tilde{x} \in R^3$), the resulting feature dimension becomes $D = 3 + 2 \cdot 3 \cdot K$(the factor 2 accounts for sine and cosine terms). In this work, we use K = 12, resulting in D =75 input features. After training, the normalization parameters (μ, σ) and the Fourier mapping configuration (K and the frequency bands) are fixed and consistently applied during inference to ensure stable and repeatable joint-angle predictions.

2.4. MLP Architecture and Constraint Aware Learning

The inverse kinematics mapping from Cartesian end-effector positions to joint configurations is modeled using a Multi-Layer Perceptron (MLP) architecture, as illustrated in Fig. 3. Although the robot is described as a 6-DOF manipulator in the URDF. This study focused on the five actuated positioning joints for IK learning, while the gripper joint is excluded. Accordingly, the network predicts a five-dimensional joint-angel vector $q = (q_1, q_2, q_3, q_{4,} q_5)$ from a 3D Cartesian target position.

The network input is the normalized Cartesian position $\tilde{x}$ augmented with Fourier features (Section 2.3), which improves the representation of nonlinear and periodic relationships commonly observed in joint-space mappings. These features are processed by a sequence of fully connected residual MLP blocks (ResMLP) with nonlinear activations, enabling the model to approximate the inverse kinematics function over the reachable workspace.

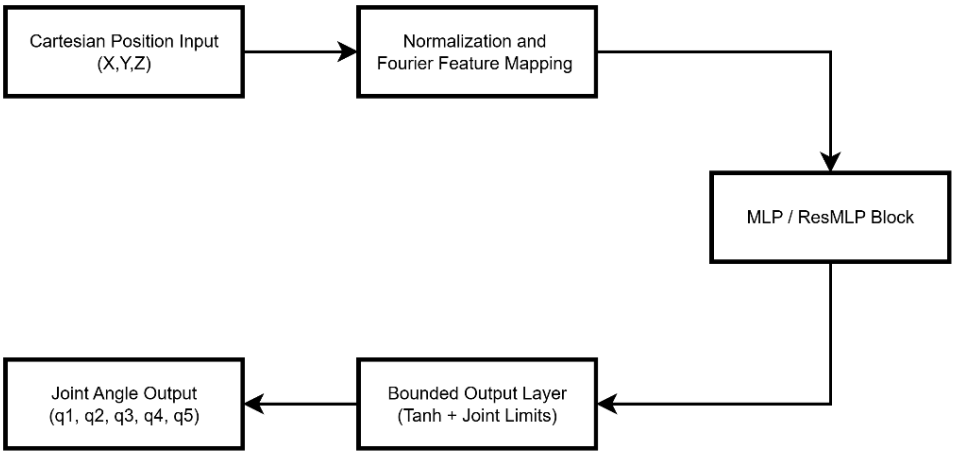

Fig. 3. Architecture of the proposed MLP-based inverse kinematics model, illustrating Cartesian input preprocessing, ResMLP blocks, and bounded joint-angle output ($q_1 - q_5$)

To ensure physically feasible predictions, a constraint-aware output layer is incorporated. Instead of directly outputting unconstrained joint angles, the network produces raw outputs z_i, which are bounded using a tanh nonlinearity and then scaled to the URDF-defined joint limits:

$$q_i = \frac{q_i^{max} - q_i^{min}}{2} \tanh(z_i) + \frac{q_i^{max} + q_i^{min}}{2}; \quad i = 1, \ldots, 5 \quad (3)$$

This design guarantees $q_i \in [q_i^{min}, q_i^{max}]$ by construction, preventing joint limit violations while maintaining smooth outputs. Overall, the proposed architecture provides a lightweight, real-time inverse kinematics predictor that supports continuous pick and place execution with joint limit guarantees.

2.5. Loss Function

To address the periodic nature of revolute joint angles, an angle-aware joint loss was employed instead of a conventional mean squared error. The angular difference between the predicted joint angle $\widehat{q_i}$ and the ground truth joint angle $q_i^{(gt)}$ is computed using a wrap function defined in (3):

$$Wrap(\Delta) = arctan2(sin\Delta, cos\Delta) \tag{4}$$

The joint-angle loss is computed as the mean absolute wrapped error over the five actuated joints:

$$L_{joint} = \frac{1}{5}\sum_{i=1}^{5}\left|wrap(\widehat{q_i} - q_i^{(gt)})\right| \tag{5}$$

This loss formulation prevents discontinuities at angular boundaries and improves training stability for inverse kinematics learning involving periodic joint variables. To enforce task-space consistency, a forward kinematics (FK) loss was introduced by comparing the end-effector position. the FK loss is defined using the L1-norm as shown in (5):

$$L_{FK} = \|\widehat{FK}\,(\hat{q}) - X_{target}\|_1 \tag{6}$$

The total loss function used during training is expressed as a weighted combination of the joint-angle loss and the FK loss, as defined in (6):

$$L_{total} = L_{joint} + \lambda L_{FK} \tag{7}$$

Where λ is a weighting coefficient that balances accuracy in joint space and task space. This composite loss encourages the model to generate joint-angle predictions that are both physically meaningful and effective for accurate end-effector positioning in pick-and-place tasks.

2.6. Training Configuration

The proposed MLP-based inverse kinematics model is trained in a supervised manner using the generated paired dataset. The dataset is split into training, validation, and test subsets as described in Section 2.2 to enable unbiased model selection and final evaluation. Model parameters are optimized using the Adam optimizer with a fixed learning rate 5×10^{-4} and training is performed using mini-batch updates to improve stability and computational efficiency.

To mitigate overfitting and select the best-performing checkpoint, early stopping based on the validation loss is applied. The model weights corresponding to the lowest validation loss are retained for deployment and all subsequent evaluations. After training, the normalization statistics (μ, σ) and the Fourier feature mapping configuration are fixed and consistently applied during inference to ensure stable and repeatable predictions.

Training is conducted on a standard computing platform with CPU/GPU acceleration. Fig. 4 presents the learning curves, showing the training and validation loss across epochs, including the training total loss and validation metrics.

As shown in Fig. 4, both the training objective and validation metrics decrease steadily and stabilize, indicating stable convergence without severe overfitting. In our experiments, the best FK-consistency checkpoint achieves a validation FK error of 0.00393 m (3.93 mm) at epoch 19, while the best join-angle checkpoint achieves a validation joint-angle MAE of 0.01095 rad (0.584°) at epoch 20. The deployed

model uses an MLP architecture with width = 512, depth = 8, and K = 12 Fourier feature bands, and is configured according to the robot URDF with five active joints.

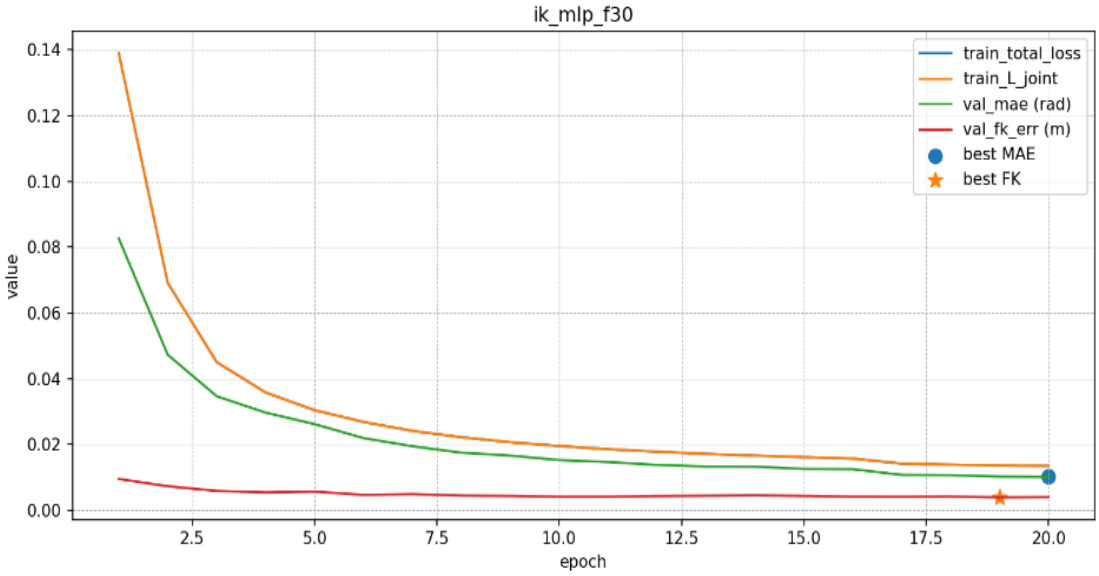

Fig. 4. Learning curves across epochs: training objective and validation metrics, including joint-angle MAE and FK-consistency error

2.7. Deployment and Evaluation Setup

After training, the proposed MLP-based inverse kinematics model is deployed and evaluated in both simulation and physical robot environments to assess its practical performance. Simulation-based evaluation is conducted using the PyBullet physics engine, where the trained model is integrated to generated joint-angle commands directly from Cartesian end-effector target positions under the same workspace and joint-limit constraints used during dataset generation.

Two evaluation scenarios are considered. First, end-effector point testing in performed by commanding the robot to reach a set of predefined Cartesian target positions sampled within the workspace bounds (BBOX). For each target, the MLP predict the joint configuration, which is executed in the simulator. The achieved end-effector position is then computed via forward kinematics and compared with the target position to obtain the Cartesian position error. Second, pick and place evaluation is conducted by executing smooth Cartesian trajectories between predefined pick and place proses to reflect sequential motion execution. During this scenario, the resulting end effector trajectory and joint commands are recorded across waypoints to assess trajectory deviation and motion feasibility under continuous execution.

Fig. 5 and Fig. 6 illustrate the PyBullet evaluation setup for the two scenarios: end effector point testing with forward kinematics validation and pick and place trajectory execution shown as a sequence of snapshots. To validate real-world applicability, the trained model was further deployed on a physical 6-DOF robotic manipulator. The same inference pipeline used in simulation was applied to the physical robot without modification. This evaluation aimed to verify that the model generalizes beyond simulation and produces physically feasible and stable motions under real hardware constraints. All experiments were conducted in an indoor environment with fixed target positions to ensure repeatability and fair comparison.

2.8. Evaluation Metrics

The proposed inverse kinematics model is evaluated using quantitative metrics that reflect task-space accuracy, joint-space consistency, and computational efficiency. All metrics are computed under

the same workspace bounds and joint-limit constraints, and timing results are reported under the same hardware/software configuration.

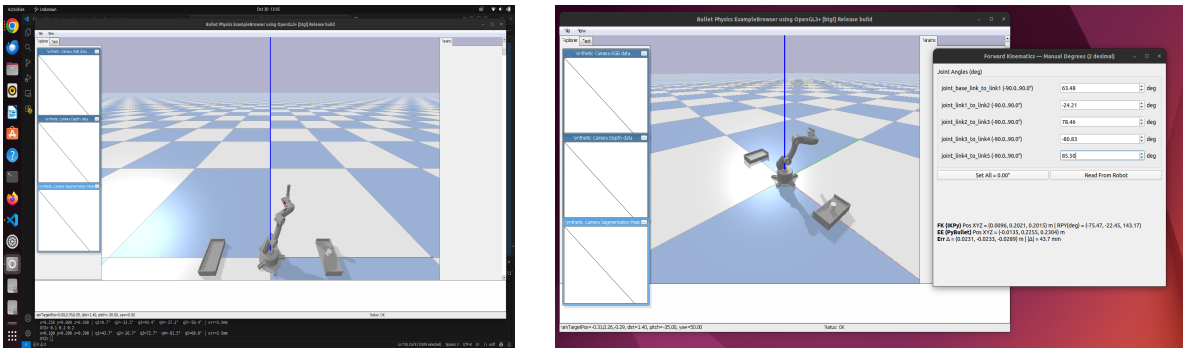

Fig. 5. End-effector point testing in PyBullet. The MLP-predicted joint. The MLP-predicted joint configurations are executed in simulation and validated using forward kinematics to compute the Cartesian position error

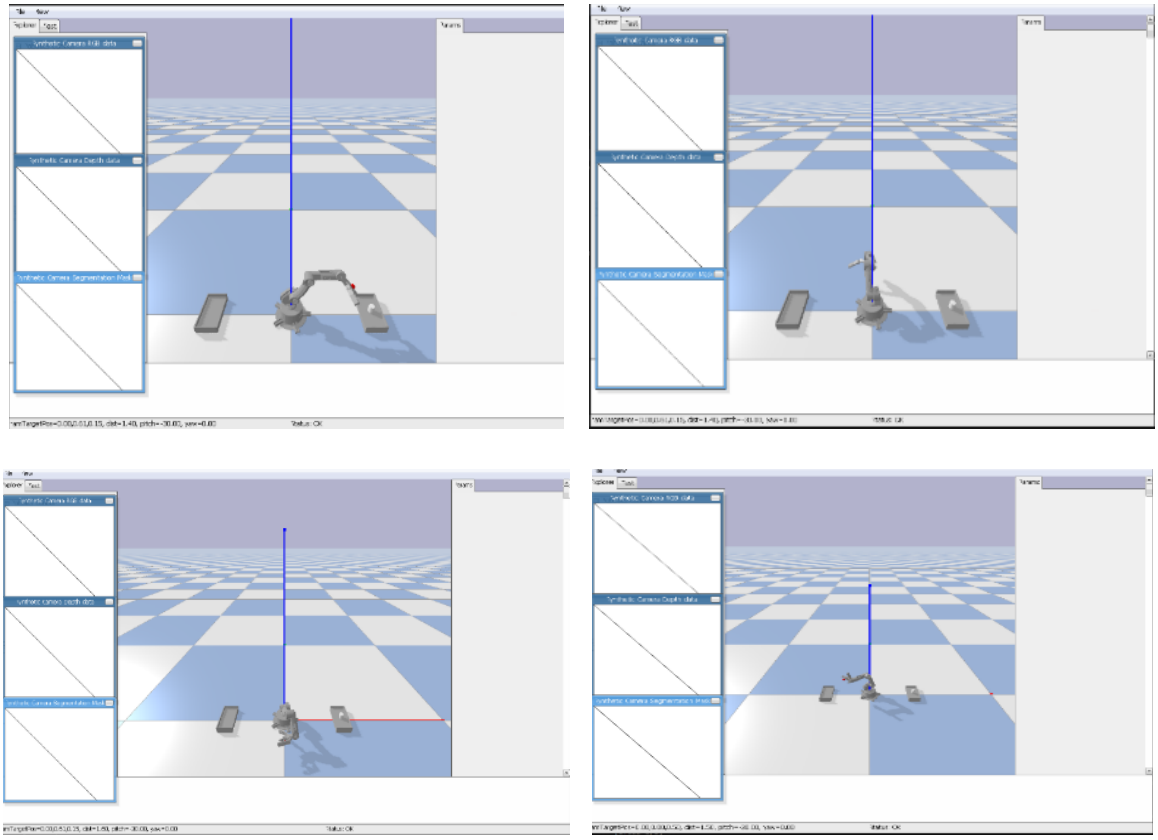

Fig. 6. Pick and place trajectory execution in PyBullet shown as sequence of snapshots between predefined pick and place poses

Task-space positioning error is measured as the Euclidean distance between the target Cartesian position $p = [x, y, z]^T$ and the achieved end-effector position $\tilde{p} = [\tilde{x}, \tilde{y}, \tilde{z}]^T$ obtained via forward kinematics:

$$e = \|p - \tilde{p}\|_2 \tag{8}$$

For point testing, we report the mean position error (mm) across all targets. For pick and place evaluation, we additionally report the endpoint errors at the pick and place poses, computed using the same definition of error (e) at the corresponding trajectory endpoints.

Success rare (%) is defined as the proportion of targets whose final Cartesian error satisfies $e \leq \tau$. For the numerical IK baseline (IKPy) a target is counted as successful only if the solver converges and the final end effector error satisfies $e \leq \tau$.

Joint angle consistency is evaluated using the mean absolute wrapped angular error (MAE) across the five actuated joints, using the wrap operator defined in (3):

$$MAE = \frac{1}{5}\sum_{i=1}^{5}\left|wrap(\hat{q}_i - q_i^{(gt)})\right| \tag{9}$$

Computational efficiency is measured using (i) IK inference time per target (ms), defined as the time required to compute $\hat{q}$ from a given Cartesian input, and (ii) the mean loop period (ms), defined as the end-to-end cycle time per pose including IK inference, trajectory interpolation, and controller execution.

3. Results and Discussion

The section presents the experimental results and discussion of the proposed MLP-based inverse kinematics framework. The evaluation focuses on end-effector positioning accuracy, pick-and-place trajectory performance, joint-angle stability and computational efficiency. The proposed method is compared against a classical inverse kinematics solver to demonstrate its effectiveness and practical advantages.

3.1. Dataset Characteristics and Training Outcome

The inverse kinematics dataset was generated using systematic Cartesian grid sampling with a spatial resolution of 1 mm over the reachable workspace defined by the robot URDF model. In total 60.471.201 Cartesian target positions were sampled, from which 34.431.743 valid IK solutions were retained after applying feasibility checks (URDF joint-limits satisfaction) and forward kinematics verification. This corresponds to a workspace validity ratio of 56.9 %, reflecting the kinematic constraints and reachability limitations of the manipulator. A summary of the dataset characteristics is provided in Table 1.

Table 1. Summary of the generated inverse kinematics dataset

Description	Value
Sampled Cartesian Points	60.471.201
Valid IK solutions	34.431.743
Valid ratio	56.9%
Workspace resolution	1 mm
Active joints	5

The large-scale dataset enables the model to observe diverse kinematic configurations, including regions close to joint limits and workspace boundaries. Training results show stable convergence within fewer than 25 epochs. On the test set, the model achieved a mean joint angle MAE of 0.68° and a mean forward kinematics (FK) error of 4 mm, with 95% of samples exhibiting FK errors below 6.83 mm.

These results indicate that the proposed MLP-based model effectively learn the inverse kinematics mapping over a wide workspace. The relatively small FK error demonstrates that the learned mapping generalizes beyond individual joint configurations and maintains task-space accuracy, which is essential for manipulation tasks.

3.2. End-Effector Point Testing and Workspace Error Distribution

To evaluate the positioning accuracy of the proposed MLP-based inverse kinematics model, point-wise end-effector tests were conducted at 20 target Cartesian poses distributed across the manipulator workspace. For each target position $P = [x, y, z]^T$, the network predicted a joint configuration $\hat{q}$, which was then mapped back to Cartesian position $\hat{p}$ through forward kinematics, The absolute positioning error was quantified using the Euclidean distance, defined in (7) as:

$$e = \|p - \hat{p}\| = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2 + (z - \hat{z})^2} \tag{10}$$

Beyond reporting individual point results, the evaluation summarizes accuracy using statistical aggregation to highlight trends across different workspace regions. Specifically, target point were categorized relative to the workspace bounding box (BBOX) into inside, boundary and outside regions in order to examine the effect of proximity to workspace limits on kinematic feasibility and prediction accuracy. The resulting mean positioning errors are summarized in Table 2. Overall, the model achieves the lowest error for points located well inside the workspace, while the error increases near the workspace boundary and becomes highest for targets outside the nominal workspace. This behavior is attributed to increased kinematic difficulty and constraint saturation near the limits of reachability.

Table 2. Mean end-effector positioning error grouped by workspace BBOX region (20-point test)

BBOX Region	Interpretation	Mean Error (mm)
Inside	Points well within feasible workspace	4.28
Boundary	Points near workspace limits	23.7
Outside	Points beyond nominal workspace	61.8

To further characterize the spatial behavior of positioning error, the workspace was stratified along the vertical axis z into three height layers; low (5-15mm), mid (16-25mm), and high (26-36mm). The distribution of positioning error was then analyzed on the $x - y$ plane using spatial aggregation, as illustrated in Fig. 3. The visualization shows a consistent pattern across all height layers, where positioning errors tends to increase near the edges of the workspace particularly at extreme values of x dan y while the central workspace region exhibits consistently lower error. This observation indicates that reduced kinematics redundancy and increased sensitivity near workspace boundaries dominate error behavior, regardless of the end-effector height. Consequently, the spatial analysis visually reinforces the numerical results in Table 2, confirming that the proposed inverse kinematics model performs most reliably for targets located within the core workspace region.

In addition to the spatial analysis in Fig. 7, we further examine the overall distribution of positioning error on the test set. Fig. 8 reports a histogram of the Forward kinematic (FK) Cartesian position error, providing a global view of how errors are distributed across all evaluated targets.

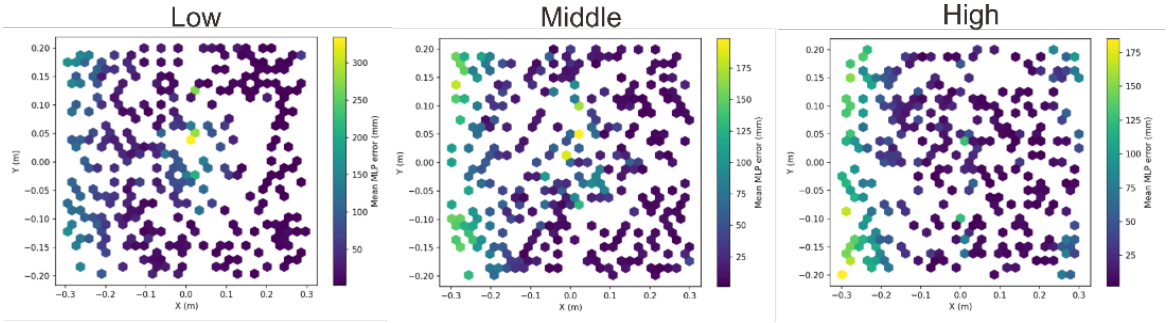

Fig. 7. Spatial distribution of mean end-effector positioning error on the $x - y$ plane at different workspace height layers (low, mid, and high)

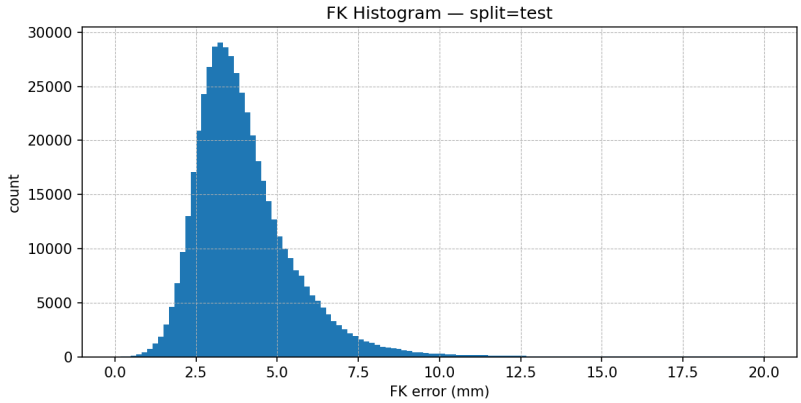

Fig. 8. Histogram of FK cartesian position error (mm) on the test set for the proposed MLP model

As shown in Fig. 8, most samples concentrate within a few millimeters, while the distribution exhibits a long tail that indicates occasional larger errors. This long-tail behavior is consistent with the workspace-boundary effects observed in Fig. 3, where kinematics difficulty increases near the limits of reachability. Consequently, the reported success rate under strict thresholds (e.g., $\tau = 3$ mm) is strongly influenced by the proportion of samples around the threshold and the presence of outliers.

3.3. End-Effector Trajectory Accuracy Evaluation

To evaluate the proposed inverse kinematics model under continuous motion, end-effector trajectory tests were conducted using a sequence of predefined Cartesian waypoints representing two target poses (denoted as Pose A and Pose B) connected by a smooth Cartesian path. For each waypoint $P_t = [x_t, y_t, z_t]^T$, the MLP model predicts joint angles $\hat{q}_t$, which are then executed in the control loop. The realized end-effector position $\hat{p}_t$ is obtained through forward kinematics, and the instantaneous trajectory error is computed as $e_t = \|p_t - \hat{p}_t\|_2$, consistent with the point testing metric in Section 3.2.

The results indicate that the model maintains stable trajectory tracking performance, with the largest deviations typically occurring during transitions near workspace boundaries or kinematically constrained regions. When both trajectory endpoint are located well within the feasible workspace, the model produces low endpoint errors and smooth end-effector motion, demonstrating that the learned inverse mapping generalizes beyond isolated point targets. This behavior is particularly relevant for practical deployment scenarios, where inverse kinematics solutions are required sequentially along a trajectory rather than at a single static pose.

To analyze the trade-off between positioning accuracy and execution speed at the system level, we evaluate the same MLP-based IK model under three execution presets: accuracy focused, balanced, and speed focused. Importantly, these presets do not modify the MLP architecture or weights; instead, they adjust motion-execution parameters in the motion controller (e.g., waypoint/interpolation density, control update rate, and smoothing/filtering). As a result, differences in the measured Cartesian endpoint deviation primarily reflect end-to-end execution effects (IK prediction, trajectory discretization, and tracking control), rather than changes in the IK model itself. As summarized in Table 3, the accuracy focused preset follows the reference trajectory more closely, whereas faster presets increase throughput through more aggressive execution settings, which can lead to larger endpoint deviations especially near curved segments and workspace boundaries. Throughout execution, the joint limit bounding mechanism ensures all predicted joint angles remain within the URDF-defined limits, preventing infeasible configurations. In the section, we summarize trajectory execution accuracy using the endpoint deviation at the final waypoint of each target pose (Pose A and Pose B), as reported in Table 3.

Fig. 9 qualitatively illustrates the impact of the execution presets by comparing the reference Cartesian path and the realized end effector trajectory obtained via forward kinematics. Consistent with

Table 3, the accuracy focused preset exhibits the smallest deviations, while the balanced and speed focused presets show increased deviation, particularly in curved segments and near workspace boundaries, notably, all presets generate continuous and physically feasible trajectories without abrupt discontinuities, indicating that the proposed MLP provides stable IK commands under continuous motion; the remaining differences are primarily attributed to the trajectory execution settings.

Table 3. Endpoint positioning error under different trajectory execution presets (same MLP IK model; presets adjust controller-level execution parameters).

Preset mode	Endpoint error at Pose A (mm)	Endpoint error at Pose B (mm)	Qualitative Speed
Accuracy Focused	13.8	25.6	Low
Balanced	66.6	24	Medium
Speed Focused	76.1	47.1	High

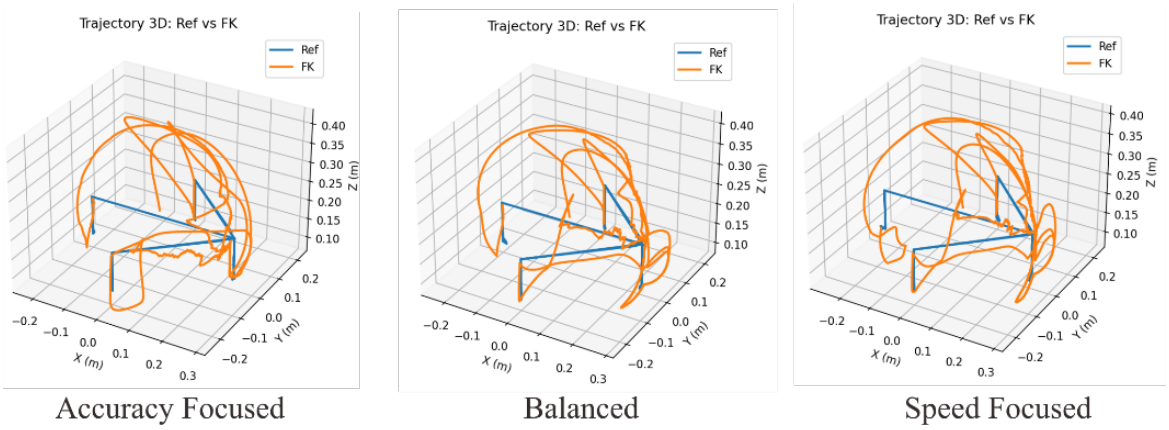

Fig. 9. Comparison of reference and realized end-effector trajectories under different execution presets (accuracy focused, balanced, and speed focused)

Overall, the trajectory results complement the point-wise accuracy analysis in Section 3.2 by demonstrating that the proposed model can generate consistent joint commands for sequential Cartesian targets. While the motion remains continuous and physically feasible under all presets, the observed trajectory deviations increase under faster execution settings due to more aggressive trajectory discretization and controller parameters.

3.4. Joint-Angle Consistency and Stability

While Section 3.3 evaluates trajectory tracking performance in Cartesian space, this section analyzes the behavior of the predicted joint angles in joint space to assess motion stability and physical feasibility. For each trajectory waypoint, the MLP-based inverse kinematics model outputs a joint configuration $\hat{q} = [\hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4, \hat{q}_5]$, which is examined across successive waypoints to evaluate consistency, smoothness, and compliance with joint constraints.

The results show that the predicted joint-angle trajectories remain smooth and continuous throughout execution without abrupt discontinuities or oscillatory behavior. Transitions between successive waypoint produce gradual joint variations, indicating that the learned inverse mapping is locally consistent and does not introduce sudden changes in joint commands, even under continuous trajectory execution. This behavior is essential for practical robotic systems as abrupt joint changes can induce mechanical stress and degrade motion reliability.

Constraint satisfaction is ensured by the bounded output formulation embedded in the network architecture. Specifically, the network output is passed through tanh-based bounding and then scaled to the URDF joint limits, yielding predicted joint angles that lie within the physically feasible range:

$$q_i{}^{min} \leq \hat{q}_i \leq q_i{}^{max}, i = 1, \ldots, 5 \quad (11)$$

Therefore, joint-limit violations are prevented by design, and none were observed in our experiments, confirming that the constraint-aware output layer enforces kinematic feasibility at inference time.

Overall, the joint-space analysis demonstrates that the proposed MLP-based inverse kinematics model achieves Cartesian accuracy, as shown ins Sections 3.2 and Section 3.3, without compromising joint-angle smoothness or safety. The combination of smooth joint evaluation and strict constraint adherence supports reliable deployment under continuous motion conditions.

3.5. Computational Efficiency

This section evaluates the computational performance of the proposed MLP-based inverse kinematics model under different execution presets to characterize the trade-off between accuracy and execution speed. We evaluate three execution presets (accuracy-focused, balanced, and speed-focused) that adjust controller-level execution parameters while keeping the same network architecture and weights. All measurements are conducted on the same hardware and software configuration for consistency.

The results shown in Table 4 indicate that the MLP inference time remains low and approximately constant across all presets, since joint angles are obtained via a single forward pass of the same network. Minor variations in the measured inference times (e.g., 7-8 ms) are attributed to measurement noise and system-level jitter (e.g., OS scheduling), rather than differences in network complexity. In contrast, the perceived speed differences among presets primarily arise from execution-level settings in the motion controller (e.g., waypoint density, control update rate, and smoothing), which affect the end-to-end loop period rather than the IK computation. The loop period is measured end-to-end per pose and includes IK inference, trajectory interpolation, and controller execution.

Table 4. Computational performance under different execution presets

Preset Mode	Mean Loop Period (ms)	IK Inference Time (ms)	Qualitative Execution Speed
Accuracy focused	19.2	8	Low
Balanced	17	7	Medium
Speed focused	15	7	High

Overall, these results confirm that the proposed MLP-based IK provides deterministic per-target computation without iterative refinement. Therefore, trade offs between accuracy and speed should be interpreted mainly at the execution level (controller and trajectory settings), while the IK inference latency remains stable across presets.

3.6. Comparison with Numerical IK Baseline (IKPy)

To asses performance against conventional inverse kinematics, we evaluate the proposed MLP model against a Jacobian based numerical IK baseline implemented in IKPy. All methods are evaluated on the same target set under identical workspace and joint limit constraints. Since numerical IK is sensitive to initialization IKPy is evaluated using a warm start initial guess from the previous valid joint configuration, which better reflects sequential pick and place execution. A target is considered successful if the solver converges and the final end effector Cartesian position error is below a fixed threshold of τ = 10 mm.

Table 5 summarizes the comparison using success rate, mean cartesian position error, pick and place endpoint errors, and per-target IK runtime. Improvement is reported as relative change with respect to IKPy. The IKPy solver settings (maximum iterations, tolerance, damping, and joint-limit handling) are reported together with the hardware/software configuration to ensure reproducibility.

Overall, the proposed MLP better performance than the IKPy numerical IK baseline under the same evaluation constraints. In addition to reducing mean Cartesian position error and pick/place endpoints errors, the MLP also attains a higher success rate at $\tau = 10$ mm, indicating more reliable target attainment. From a computational perspective, the MLP reduces per-target IK runtime, suggesting that the learned feed-forward mapping can provide both improved accuracy and faster inference for sequential pick and place execution.ru

Table 5. Performance comparison between the proposed MLP-based IK and the IKPy numerical IK baseline under identical workspace and joint-limit constraints (success criterion: $\tau = 10$ mm)

Metric	MLP-Based IK	IKPy (warm-start)	Relative Improvement (%)
Success rate (%)	65	40	62.5
Mean position error (mm)	4.28	18.38	76.7
Pick endpoint error (mm)	13.8	44.2	68.7
Place endpoint error (mm)	25.6	53.86	52.5
Runtime per target (ms)	7.46	20	62.7

3.7. Overall Discussion

This section synthesizes the experimental results to highlight the overall effectiveness and practical implications of the proposed MLP-based inverse kinematics framework. The point wise evaluation in Section 3.2 shows that the model achieves low cartesian positioning error for targets located within the core workspace region, while errors increase near workspace boundaries due to kinematics constraints. These indicate a clear relationship between workspace feasibility and prediction accuracy.

The trajectory evaluation in Section 3.3 further shows that the learned inverse mapping generalizes beyond isolated points target, enabling stable and continuous end effector motion along predefined Cartesian paths. Across the evaluated execution presets, trajectory deviation increases under faster controller settings, particularly near curved segments and workspace boundaries, while all presets produce continuous and physically feasible trajectories without abrupt discontinuities.

Joint space analysis in Section 3.4 confirms that the achieved Cartesian accuracy does not come at the expense of motion stability or mechanical safety. The predicted joint angle evolve smoothly across successive waypoints and strictly comply with URDF defined joint limits, validating the effectiveness of the constraint-aware output formulation. This property is particularly important for deployment on physical robotic platforms, where joint-limit violations and abrupt changes can cause mechanical stress or system failure.

From a computational perspective, Section 3.5 and Table 5 show that the proposed approach provides efficient and deterministic inference. Compared to the IKPy numerical IK baseline, the proposed MLP reduces the mean position error from 18.38 mm to 4.28 mm and decreases per-target IK computation time from 20 ms to 7.46 ms, while also improving the success rate from 40% to 65% under of $\tau = 10$ mm. Taken together, these findings demonstrate that the proposed method offers a practical balance between accuracy, stability, and efficiency. Making it suitable for real-time pick and place execution in robotic manipulation systems.

4. Conclusion

This paper presented an MLP-based inverse kinematics framework for robotic manipulation, covering an end-to-end pipeline from URDF-based dataset generation to constraint-aware learning and deployment in both simulation and physical experiments. The proposed approach learns a direct mapping from Cartesian end-effector targets to joint angle commands while enforcing URDF-defined joint limits through a bounded output layer, thereby promoting physically feasible predictions.

Experimental results show that the proposed model achieved low positioning error for targets located within the core workspace, while errors increase near workspace boundaries due to inherent kinematic constraints. In trajectory-based evaluation, the learned mapping generalizes beyond isolated points targets and produces smooth, continuous joint commands for sequential motion execution. Different execution presets exhibit the expected system level trade off: faster controller settings increase throughput but tend to increase trajectory deviation, particularly on curved segments and near workspace boundaries, without changing the underlying network architecture or weights.

Compared to the Jacobian-based numerical IK baseline in IKPy (warm-start), the proposed method provides better overall performance under identical workspace and joint-limit constraints and the same success criterion (at $\tau = 10$ mm), improving success rate (65% vs 40%), reducing mean position error (4.28 mm vs 18.38 mm), and lowering per-target runtime (7.46 ms vs 20 ms). These results indicate that the proposed MLP-based IK offers a practical balance between accuracy, stability, and computational efficiency for real-time pick and place applications

Despite these advantages, the current study focuses on open-loop execution and does not incorporate sensor feedback or physical interaction effects during manipulation. Future work will extend the framework toward closed-loop control with real-time sensing, robustness under external disturbances, and integration with grasp planning and contact-aware manipulation to improve reliability in more complex real-world scenarios.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This work is supported by the Department of Computer Science and Electronics, Universitas Gadjah Mada under the Publication Funding Year 2026.

Conflicts of Interest: The authors declare no conflict of interest

References

[1] C. Urrea, "Adaptive multi-objective reinforcement learning for real-time manufacturing robot control," *Machines*, vol. 13, no. 12, p. 1148, 2025, https://doi.org/10.3390/machines13121148.

[2] A. Iqdymat and G. Stamatescu, "Reinforcement learning of a six-DOF industrial manipulator for pick-and-place application using efficient control in warehouse management," *Sustainability*, vol. 17, no. 2, p. 432, 2025, https://doi.org/10.3390/su17020432.

[3] N. Kaur and A. Sharma, "Robotics and automation in manufacturing processes," in *Intelligent Manufacturing*, CRC Press, 2025, pp. 97-109, https://doi.org/10.1201/9781032655758-7.

[4] C. Urrea, "Hybrid deep learning-reinforcement learning for adaptive human-robot task allocation in Industry 5.0," *Systems*, vol. 13, no. 8, p. 631, 2025, https://doi.org/10.3390/systems13080631.

[5] K. Thurow, O. Peter, P. Courtney, K. Széll, and Á. Wolf, "Editorial: Robotics in laboratory automation," *SLAS Technology*, vol. 36, p. 100373, 2026, https://doi.org/10.1016/j.slast.2025.100373.

[6] S. Mustary, M. A. Kashem, J. M. Sony, N. Hossain, and M. A. Chowdhury, "Advancing stability in robot manipulators: A review of recent progress and parameters," *Engineering Reports*, vol. 7, no. 7, p. e70207, Jul. 2025, https://doi.org/10.1002/eng2.70207.

[7] S. Bai, W. Song, J. Chen, Y. Ji, Z. Zhong, J. Yang, H. Zhao, W. Zhou, W. Zhao, Z. Li, P. Ding, C. Chi, H. Li, C. Xu, X. Zheng, D. Wang, S. Zhang, and B. Chen, "Towards a unified understanding of robot manipulation: A comprehensive survey," *arXiv preprint,* arXiv:2510.10903, 2025, https://arxiv.org/abs/2510.10903.

[8] M. Cheng and S. Xu, "Research on trajectory planning of six-degree-of-freedom robotic arm based on improved genetic algorithm," *Journal of Intelligent & Robotic Systems*, vol. 111, no. 3, p. 105, Sep. 2025, https://doi.org/10.1007/s10846-025-02306-4.

[9] S. Waseem, M. Adnan, M. S. Iqbal, A. A. Amin, and A. Shah, "Expert and intelligent systems for robotic manipulators control: A comprehensive review," Feb. 2025, https://doi.org/10.22541/au.174077476.64719692/v1.

[10] S. M. Mahdi, A. I. Abdulkareem, A. J. Humaidi, A. K. Al Mhdawi, and H. Al-Raweshidy, "Comprehensive review of control techniques for various mechanisms of parallel robots," *IEEE Access*, vol. 13, pp. 63381-63416, 2025, https://doi.org/10.1109/ACCESS.2025.3557937.

[11] J. Jang, Z. Wang, Z. Zhou, F. Wu, and Y. Zhao, "SEEC: Stable end-effector control with model-enhanced residual learning for humanoid loco-manipulation," *arXiv preprint,* arXiv:2509.21231, 2025, https://arxiv.org/abs/2509.21231.

[12] M. Kaur, S. Sondhi, and V. K. Yanumula, "Design and optimization of controller-based approach for magnetic-field driven robotic arm joints and end-effector," *Journal of Field Robotics*, vol. 42, no. 7, pp. 3285-3307, Oct. 2025, https://doi.org/10.1002/rob.22564.

[13] V. Tinoco, M. F. Silva, F. N. Santos, R. Morais, S. A. Magalhães, and P. M. Oliveira, "A review of advanced controller methodologies for robotic manipulators," *International Journal of Dynamics and Control*, vol. 13, no. 1, p. 36, Jan. 2025, https://doi.org/10.1007/s40435-024-01533-1.

[14] A. P. O. Barros, R. C. S. Alegria, G. M. da Silva, P. V. D. S. Matias, C. A. O. de Freitas, V. F. D. L. Junior, and V. J. da Silva, "Neuromorphic control of the Serv-Arm robot using spiking neural networks," *Preprints*, p. 2025121045, 2025, https://doi.org/10.20944/preprints202512.1045.v1.

[15] J. Haviland and P. Corke, "Manipulator differential kinematics: Part I: Kinematics, velocity, and applications [Tutorial]," *IEEE Robotics & Automation Magazine*, vol. 31, no. 4, pp. 149-158, Dec. 2024, https://doi.org/10.1109/MRA.2023.3270228.

[16] A. Calzada-Garcia, J. G. Victores, F. J. Naranjo-Campos, and C. Balaguer, "Inverse kinematics for robotic manipulators via deep neural networks: Experiments and results," *Applied Sciences*, vol. 15, no. 13, p. 7226, 2025, https://doi.org/10.3390/app15137226.

[17] S. Patong and N. Suresh, "Robotic motion planning using the Jacobian matrix-based inverse kinematics algorithm," *Association Journal of Interdisciplinary Technics in Engineering Mechanics*, vol. 3, no. 3, pp. 14-21, 2025, https://ajitem.org/index.php/journal/article/view/EM33003/85.

[18] V. Lakshmi Narayanan, J. Narayan, H. Gritli, and S. K. Dwivedy, "A decade of inverse kinematics methods for serial manipulators: A systematic review," *Journal of Field Robotics*, vol. 43, no. 1, pp. 184-229, Jan. 2026, https://doi.org/10.1002/rob.70014.

[19] N. Manjegowda, Muralidhara, N. R. Jain, and S. K. Shivaswamy, "A comprehensive review of inverse kinematics techniques from analytical foundations to artificial intelligence integration," *International Journal of Advanced Mechatronic Systems*, vol. 12, no. 4, pp. 258-282, 2025, https://doi.org/10.1504/IJAMECHS.2025.149356.

[20] A. Calzada-Garcia, J. G. Victores, F. J. Naranjo-Campos, and C. Balaguer, "A review on inverse kinematics, control and planning for robotic manipulators with and without obstacles via deep neural networks," *Algorithms*, vol. 18, no. 1, p. 23, 2025, https://doi.org/10.3390/a18010023.

[21] W. Xie, Z. Wang, J. Ma, J. Chen, and X. Xie, "An inverse kinematics solution for mobile manipulators in textile workshops based on an improved particle swarm optimization," *Symmetry*, vol. 17, no. 11, p. 1980, 2025, https://doi.org/10.3390/sym17111980.

[22] S. Sulaiman, A. Harikumar, N. Marturi, and S. Bøgh, "Comparative analysis of Jacobian-based motion planning methods for redundant manipulators," in *2025 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*, 2025, pp. 259-264, https://doi.org/10.1109/ARSO64737.2025.11124951.

[23] Y. Zhai, J. Xu, H. Mo, C. Zhang, and D. Sun, "Model-based control of a continuum manipulator with online Jacobian error compensation using Kalman filtering," *Cyborg and Bionic Systems*, vol. 6, p. 0339, 2026, https://doi.org/10.34133/cbsystems.0339.

[24] M. Y. Alwardat and H. Alwan, "Geometric Jacobians derivation and kinematic singularity analysis for 6-DOF robotic manipulator," *International Journal of Advanced Research in Computer Science*, vol. 16, no. 1, pp. 6-20, Feb. 2025, https://doi.org/10.26483/ijarcs.v16i1.7178.

[25] I. Agustian, N. Daratha, R. Faurina, A. Suandi, and S. Sulistyaningsih, "Robot manipulator control with inverse kinematics PD-pseudoinverse Jacobian and forward kinematics Denavit Hartenberg," *Jurnal Elektronika dan Telekomunikasi*, vol. 21, no. 1, pp. 8-18, Aug. 2021, https://doi.org/10.14203/jet.v21.8-18.

[26] S. Li and C. C. Cheah, "Deep neural network-based Jacobian control of robot manipulators: Offline regression and online adaptation," *IEEE/ASME Transactions on Mechatronics*, vol. 30, no. 6, pp. 6608-6619, Dec. 2025, https://doi.org/10.1109/TMECH.2025.3526648.

[27] M. Przystupa, M. Dehghan, M. Jagersand, and A. R. Mahmood, "Analyzing neural Jacobian methods in applications of visual servoing and kinematic control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 14276-14283, https://doi.org/10.1109/ICRA48506.2021.9561373.

[28] B. Tipary, A. Kovács, and F. G. Erdős, "Planning and optimization of robotic pick-and-place operations in highly constrained industrial environments," *Assembly Automation*, vol. 41, no. 5, pp. 626-639, 2021, https://doi.org/10.1108/AA-07-2020-0099.

[29] S. Teck, B. Engelen, J. Peeters, K. Kellens, and P. Vansteenwegen, "The development of an efficient scheduling heuristic for multi-robot pick-and-place operations," *International Journal of Production Research*, vol. 63, no. 23, pp. 8923-8942, 2025, https://doi.org/10.1080/00207543.2025.2517835.

[30] C. Tiseo, Q. Rouxel, Z. Li, and M. Mistry, "Robust impedance control for dexterous interaction using fractal impedance controller with IK-optimisation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 840-846, https://doi.org/10.1109/ICRA46639.2022.9812013.

[31] Y. Wang, C. Sifferman, and M. Gleicher, "Iklink: End-effector trajectory tracking with minimal reconfigurations," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12165-12171, https://doi.org/10.1109/ICRA57147.2024.10611510.

[32] M. B. Çetinkaya, K. Yildirim, and Ş. Yildirim, "Trajectory analysis of 6-DOF industrial robot manipulators by using artificial neural networks," *Sensors*, vol. 24, no. 13, p. 4416, 2024, https://doi.org/10.3390/s24134416.

[33] S. S. Perumaal and N. Jawahar, "Automated trajectory planner of industrial robot for pick-and-place task," *International Journal of Advanced Robotic Systems*, vol. 10, no. 2, p. 100, Feb. 2013, https://doi.org/10.5772/53940.

[34] Y. Shen, Q. Jia, Z. Huang, R. Wang, J. Fei, and G. Chen, "Reinforcement learning-based reactive obstacle avoidance method for redundant manipulators," *Entropy*, vol. 24, no. 2, p. 279, 2022, https://doi.org/10.3390/e24020279.

[35] H. Li, D. Gong, and J. Yu, "An obstacles avoidance method for serial manipulator based on reinforcement learning and artificial potential field," *International Journal of Intelligent Robotics and Applications*, vol. 5, no. 2, pp. 186-202, Jun. 2021, https://doi.org/10.1007/s41315-021-00172-5.

[36] H. D. Trullo and O. A. V. Alban, "A systematic review of inverse kinematics methods for fixed-base serial manipulators: Analytical, numerical, and machine learning methods," *International Journal of Robotics and Control Systems*, vol. 5, no. 3, pp. 1808–1827, Jul. 2025, https://doi.org/10.31763/ijrcs.v5i3.1820.

[37] C. Zhao, Y. Wei, J. Xiao, Y. Sun, D. Zhang, Q. Guo, and J. Yang, "Inverse kinematics solution and control method of 6-degree-of-freedom manipulator based on deep reinforcement learning," *Scientific Reports*, vol. 14, no. 1, p. 12467, May 2024, https://doi.org/10.1038/s41598-024-62948-6.

[38] J. Lu, T. Zou, and X. Jiang, "A neural network based approach to inverse kinematics problem for general six-axis robots," *Sensors*, vol. 22, no. 22, p. 8909, 2022, https://doi.org/10.3390/s22228909.

[39] O. A. Al-Sharif, N. A. Abbass, A. M. Hanafi, and A. O. Elnady, "Enhancing robotic autonomy: A review and case study of traditional and deep learning approaches to inverse kinematics," *International Journal of Engineering and Applied Sciences-October 6 University*, vol. 1, no. 1, pp. 1-8, Jul. 2024, https://doi.org/10.21608/ijeasou.2024.374260.

[40] L. Marchesotti, "Machine learning techniques for solution of the inverse kinematic problem in space manipulators," Master's thesis, Politecnico di Milano, Milan, Italy, 2024, https://www.politesi.polimi.it/handle/10589/217772.

[41] J. Zhou, Q. Zhu, Y. Wang, M. Feng, C. Wu, X. Liu, J. Huang, and A. Mian, "PoseDiffusion: A coarse-to-fine framework for unseen object 6-DoF pose estimation," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 9, pp. 11127-11138, Sept. 2024, https://doi.org/10.1109/TII.2024.3399886.

[42] E. Govi, D. Sapienza, S. Toscani, I. Cotti, G. Franchini, and M. Bertogna, "Addressing challenges in industrial pick and place: A deep learning-based 6 degrees-of-freedom pose estimation solution," *Computers in Industry*, vol. 161, p. 104130, 2024, https://doi.org/10.1016/j.compind.2024.104130.

[43] A. Shrivastava, "Exploring optimal motion strategies: A comprehensive study of various trajectory planning schemes for trajectory selection of robotic manipulator," *Journal of The Institution of Engineers (India): Series C*, vol. 106, no. 2, pp. 691-710, 2025, https://doi.org/10.1007/s40032-025-01171-2.

[44] H. M. Le, G. K. Appuhamillage, and L. Nguyen, "High-speed motion planning with object acceleration constraint for industrial robots," *Journal of Control, Automation and Electrical Systems*, vol. 35, no. 4, pp. 614-624, Aug. 2024, https://doi.org/10.1007/s40313-024-01093-x.

[45] M. Baum, "Robustness in robotic and biological manipulation," Dissertation, Fakultät IV – Elektrotechnik und Informatik, Technische Universität Berlin, Berlin, 2024, https://depositonce.tu-berlin.de/bitstreams/282f80e5-688d-4cda-9811-c3222afb2a94/download.

[46] Y. Wang, M. Damani, P. Wang, Y. Cao, and G. Sartoretti, "Distributed reinforcement learning for robot teams: A review," *Current Robotics Reports*, vol. 3, no. 4, pp. 239-257, 2022, https://doi.org/10.1007/s43154-022-00091-8.

[47] D. Tanda, "Semantic obstacle avoidance of robotic arm for fruit harvesting," Master's thesis, Master of Science Program in Mechatronic Engineering, Politecnico di Torino, Turin, Italy, 2024, https://webthesis.biblio.polito.it/33194/?template=default.

[48] C.-K. Ho, L.-W. Chan, C.-T. King, and T.-Y. Yen, "A deep learning approach to navigating the joint solution space of redundant inverse kinematics and its applications to numerical IK computations," *IEEE Access*, vol. 11, pp. 2274-2290, 2023, https://doi.org/10.1109/ACCESS.2023.3234104.

[49] A. Keaveny, "Experimental evaluation of affordance detection applied to 6-DoF pose estimation for intelligent robotic grasping of household objects," Thesis, Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada, 2021, http://hdl.handle.net/10012/17716.

[50] B. Balasubramanian and K. Cetin, "Vision-based 6D pose analytics solution for high-precision industrial robot pick-and-place applications," *Sensors*, vol. 25, no. 15, p. 4824, 2025, https://doi.org/10.3390/s25154824.

[51] M. R. Diprasetya, J. Poeppelbaum, and A. Schwung, "KineNN: Kinematic neural network for inverse model policy based on homogeneous transformation matrix and dual quaternion," *Robotics and Computer-Integrated Manufacturing*, vol. 94, p. 102945, 2025, https://doi.org/10.1016/j.rcim.2024.102945.

[52] Z. Xu, Y. Li, X. Yang, Z. Zhao, J. Zhao, and H. Liu, "A combined inverse kinematics algorithm using FABRIK with optimization," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 4, p. 62, 2023, https://doi.org/10.1007/s10846-023-01895-2.

[53] X. Ma, S. Patidar, I. Haughton, and S. James, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18081-18090, https://doi.org/10.1109/CVPR52733.2024.01712.

[54] H. Lou, Y. Liu, Y. Pan, Y. Geng, J. Chen, W. Ma, C. Li, L. Wang, H. Feng, L. Shi, L. Luo, and Y. Shi, "Robo-GS: A physics consistent spatial-temporal model for robotic arm with hybrid representation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 15379-15386, https://doi.org/10.1109/ICRA55743.2025.11128786.

[55] F. Pretini, "Deep reinforcement learning for robotic manipulation on UR10e: From simulation to real deployment," Master's thesis, Master of Science Program in Computer Engineering, Politecnico di Torino, Turin, Italy, 2025, https://webthesis.biblio.polito.it/38675/?template=default.

[56] M. N. Vu, F. Beck, M. Schwegel, C. Hartl-Nesic, A. Nguyen, and A. Kugi, "Machine learning-based framework for optimally solving the analytical inverse kinematics for redundant manipulators," *Mechatronics*, vol. 91, p. 102970, 2023, https://doi.org/10.1016/j.mechatronics.2023.102970.

[57] M. Adel, S. M. Ahmed, and M. Fanni, "End-effector position estimation and control of a flexible interconnected industrial manipulator using machine learning," *IEEE Access*, vol. 10, pp. 30465-30483, 2022, https://doi.org/10.1109/ACCESS.2022.3157817.

[58] N. Wagaa, H. Kallel, and N. Mellouli, "Analytical and deep learning approaches for solving the inverse kinematic problem of a high degrees of freedom robotic arm," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106301, 2023, https://doi.org/10.1016/j.engappai.2023.106301.

[59] A. Ganapathi, P. Florence, J. Varley, K. Burns, K. Goldberg, and A. Zeng, "Implicit kinematic policies: Unifying joint and Cartesian action spaces in end-to-end robot learning," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2656-2662, https://doi.org/10.1109/ICRA46639.2022.9812165.

[60] K. Shi, X. Zhou, and S. Gu, "Improved implicit neural representation with Fourier reparameterized training," *arXiv preprint*, arXiv:2401.07402, 2024, https://arxiv.org/abs/2401.07402.

[61] L. Xia, S. Li, L. Yi, H. Ruan, and D. Zhang, "Measure for semantics and semantically constrained pose optimization: A review," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-14, 2024, https://doi.org/10.1109/TIM.2024.3428639.

[62] M. Peñacoba-Yagüe, J. E. Sierra-García, M. Santos-Peñas, and A. Ruano, "Enhancing robotic control efficiency with MLP-based inverse kinematics: First approach," in *CONTROLO 2024*, A. P. Aguiar et al., Eds., Cham: Springer Nature Switzerland, 2025, pp. 574-582, https://doi.org/10.1007/978-3-031-81724-3_51.