CushionCatch: A Compliant Catching Mechanism for Mobile Manipulators via Combined Optimization and Learning

Bingjie Chen[†], Keyu Fan[†], Qi Yang, Yi Cheng, Houde Liu,
Kangkang Dong, Chongkun Xia, Liang Han, Bin Liang

Abstract— Catching flying objects with a cushioning process is a skill commonly performed by humans, yet it remains a significant challenge for robots. In this paper, we present a framework that combines optimization and learning to achieve compliant catching on mobile manipulators (CCMM). First, we propose a high-level capture planner for mobile manipulators (MM) that calculates the optimal capture point and joint configuration. Next, the pre-catching (PRC) planner ensures the robot reaches the target joint configuration as quickly as possible. To learn compliant catching strategies, we propose a network that leverages the strengths of LSTM for capturing temporal dependencies and positional encoding for spatial context (P-LSTM). This network is designed to effectively learn compliant strategies from human demonstrations. Following this, the post-catching (POC) planner tracks the compliant sequence output by the P-LSTM while avoiding potential collisions due to structural differences between humans and robots. We validate the CCMM framework through both simulated and real-world ball-catching scenarios, achieving a success rate of 98.70% in simulation, 92.59% in real-world tests, and a 28.7% reduction in impact torques. The open source code will be released for the reference of the community[1].

I. INTRODUCTION

Manipulating dynamic objects is challenging, requiring a sophisticated interplay of object detection, motion planning, and control [1]. In dynamic manipulation scenarios, tasks such as throwing [2], [3], juggling [4], and catching [5], [6] are essential. Catching dynamic objects requires quick and precise motion planning, followed by fast and accurate execution. For the motion planning of catching, the process can be divided into two phases, pre-catching (PRC) and post-catching (POC) [7]. In the POC phase, for a high-stiffness robot, the object may bounce away or generate significant impact forces upon contact. To address these problems,

Bingjie Chen, Keyu Fan, Qi Yang, Yi Cheng and Houde Liu are with the Center for Artificial Intelligence and Robotics, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China.

Kangkang Dong is with the Jianghuai Advanced Technology Center, Hefei, 230000, China; and with University of Science and Technology of China, Hefei, 23000, China.

Chongkun Xia is with the School of Advanced Manufacturing, Sun Yat-Sen University, Shenzhen 518055, China.

Liang Han is with the School of Electrical and Automation Engineering, Hefei University of Technology, Hefei 230009, China.

Bin Liang is with the Navigation and Control Research Center, Department of Automation, Tsinghua University, Beijing 100084, China

[1]Code:https://github.com/bingjiechen-THU/CushionCatch
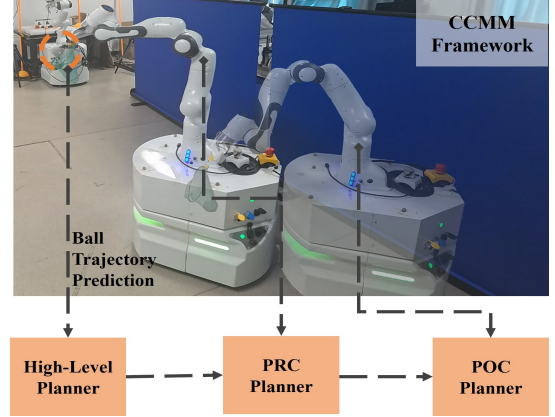

Fig. 1: The mobile manipulator's motions under CCMM framework for achieving compliant catching.

incorporating a cushioning mechanism during catching is beneficial, as it reduces impact forces [8], [9], prevents damage to both the robot and the object [10], and minimizes the risk of rebound, especially for high-speed or high-mass objects [11], [12]. Moreover, since humans always exhibit cushioning behavior when catching high-mass objects, an effective approach would be for the robot to learn this motion from humans [7].

Mobile manipulators, with their high degrees of freedom, are better suited to learning from human behavior, enabling them to outperform conventional stationary robots in terms of flexibility and efficiency [13]–[17]. Their mobility allows them to actively navigate and interact with objects [18], [19]. In catching tasks, most prior research focused on rigid control approaches, often overlooking the importance of compliant control [1], [6], [14], [20]. We focus on introducing a framework to achieve compliant catching for mobile manipulators (CCMM). Compared with previous works, the contributions of this letter are:

1. A novel CCMM framework effectively addresses all the planning and tracking challenges throughout the entire compliant catching process for mobile manipulators, while being designed for modularity.

2. Propose a P-LSTM network that combines Long Short-Term Memory (LSTM) with positional encoding (PE), designed to effectively learn compliant catching strategies from human demonstrations. Address the potential collisions arising from the structural differences between humans and robots when learning demonstrations.

3. Simulation and physical experiments were conducted

on a mobile manipulator to validate the effectiveness of our framework and the role of compliance. A catching success rates of 98.70% in simulation, 92.59% in physical experiments and a 28.7% reduction in impact torque were achieved. An open source code of the proposed method will be released[1].

II. RELATED WORK

Many studies have explored ball catching tasks using fixed-base manipulators [1], [6], [21], [22], with a particular focus on motion planning for the manipulator. Early research in this area relied on heuristic methods to determine the optimal grasping point and generated motion trajectories through interpolation [22]. However, these approaches often fell short in achieving the most effective capture motion for the robot. A more promising approach involves nonlinear optimization, which helps identify the optimal joint configuration for catching the ball, often in combination with parameterization techniques such as five-order polynomials [1]. The approach in [6] introduces a control law in dynamical systems, allowing the system to perform high-speed manipulations for catching fast-moving objects. However, these methods have typically been limited to fixed-base manipulators.

The work in [20], [23] explores the use of mobile manipulators for catching, considering both the manipulator and the mobile base for coordinated control during capture. However, they overlook the importance of compliance. In reality, a ball thrown from a distance can easily impact a joint, exceeding the torque limit [24]. Current research on cushioning the impact force during capture has focused primarily on fixed-base manipulators [25], [26]. These studies rely on impedance controller based approaches, which are challenging to implement in mobile manipulators for generating unified, coordinated control.

Learning from human demonstrations is an effective method for teaching robots to execute fast and reactive motions. Humans can catch moving objects with nonzero velocity, creating smooth trajectories, even for unexpected objects [27]. In [5], the robot learns motion from human throwing demonstrations, but it halts immediately when the object makes contact with the end-effector. [7] develops nucleated motion primitives from human demonstration data to generate compliant trajectories for catching. However, this research focuses only on the capture of vertically falling objects and not account for the potential safety issues arising from structural differences between humans and robots. Transferring humans' compliant behavior to mobile manipulators for catching flying objects while ensuring safety, remains an unresolved challenge.

III. MODELING AND PROBLEM FORMULATION

A. System Model

The mobile manipulator we use includes a mobile base and a manipulator, with $n = n_b + n_m$ degrees of freedom. The state of the manipulator is described by joint positions $\boldsymbol{q}_m \in \mathbb{R}^{n_m}$. For a mobile base with a differential drive, we use the full location description (x_b, y_b, φ) and define the $\boldsymbol{q}_b \in \mathbb{R}^{n_b}$ as virtual joints.

For the container fixed at the end for catching the ball, we get the overall forward kinematics as follows:

$$ {}_c^w\boldsymbol{T} = {}_b^w\boldsymbol{T}(x_b, y_b, \varphi) \cdot {}_m^b\boldsymbol{T} \cdot {}_e^m\boldsymbol{T}(\boldsymbol{q}_m) \cdot {}_c^e\boldsymbol{T}, \qquad (1) $$

where w represents the world coordinate system; ${}_c^w\boldsymbol{T}$ represents the homogeneous transformation of the container to the world; ${}_b^w\boldsymbol{T}$ is the transformation of the mobile base frame relative to the world; ${}_m^b\boldsymbol{T}$ denotes a constant relative pose from the mobile base frame to the manipulator frame; ${}_e^m\boldsymbol{T}$ is the position forward kinematics of the manipulator where the end-effector (EE) frame is e and ${}_c^e\boldsymbol{T}$ refers to the transformation from the EE frame to the container frame.

Analogous to the Jacobian used in manipulator kinematics, we define the extended Jacobian matrix ${}_c^w\boldsymbol{J}$ for the mobile manipulator to compute the container's velocity in the world coordinate system.

$$ {}_c^w\boldsymbol{v} = {}_c^w\boldsymbol{J}(x_b, y_b, \varphi, \boldsymbol{q}_m, {}_c^e\boldsymbol{T})\dot{\boldsymbol{q}}, \qquad (2) $$

where $\dot{\boldsymbol{q}} = (\dot{\boldsymbol{q}}_b, \dot{\boldsymbol{q}}_m)^{\mathrm{T}}$ and ${}_c^w\boldsymbol{v} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^{\mathrm{T}}$. This maps the velocity of whole-body joints to the container's velocity.

The flying ball is treated as a point mass influenced by aerodynamic drag and gravity [28]. The equation governing the ball's motion can be described as:

$$ \ddot{\boldsymbol{p}} = -\boldsymbol{g} - k_{ad}\|\dot{\boldsymbol{p}}\|\dot{\boldsymbol{p}}, \qquad (3) $$

where $\boldsymbol{p} \in \mathbb{R}^3$ is the ball's position in the world frame, $\boldsymbol{g} = [0, 0, g]^{\mathrm{T}}$ is the gravitational acceleration, k_{ad} is the aerodynamic drag coefficient and $\|\cdot\|$ is the Euclidean norm.

IV. METHODOLOGY

A. Ball Estimation and Prediction

To predict the trajectory of a ball in flight, we should first estimate its current state and then use a dynamical model to forecast its future position over a given time horizon. We use a modified discrete-time Kalman filter that accounts for aerodynamic drag, as described in [28]. Given measurements at the discrete time index k, the state of the ball at the next time step $\boldsymbol{p}[k+1|k]$ can be predicted as follows:

$$ \ddot{\boldsymbol{p}}[k|k] = -\boldsymbol{g} - k_{ad}\|\dot{\boldsymbol{p}}[k|k]\|\dot{\boldsymbol{p}}[k|k], \qquad (4a) $$

$$ \dot{\boldsymbol{p}}[k+1|k] = \dot{\boldsymbol{p}}[k|k] + \delta_k\ddot{\boldsymbol{p}}[k|k], \qquad (4b) $$

$$ \boldsymbol{p}[k+1|k] = \boldsymbol{p}[k|k] + \delta_k\dot{\boldsymbol{p}}[k|k] + \frac{1}{2}\delta_k^2\ddot{\boldsymbol{p}}[k|k], \qquad (4c) $$

where δ_k is the time interval between k and $k+1$. The Kalman filter we used follows the standard approach described in [29]. The aerodynamic drag coefficient k_{ad} is estimated offline using a Recursive Least-Squares estimator [28]. Then the ball's predicted trajectory point $\hat{\boldsymbol{p}}$ can be obtained by (3) through numerically integrating over a finite time horizon. We use a cubic spline interpolation between two-time steps to query the $\hat{\boldsymbol{p}}(t)$ and $\hat{\dot{\boldsymbol{p}}}(t)$ at any time.
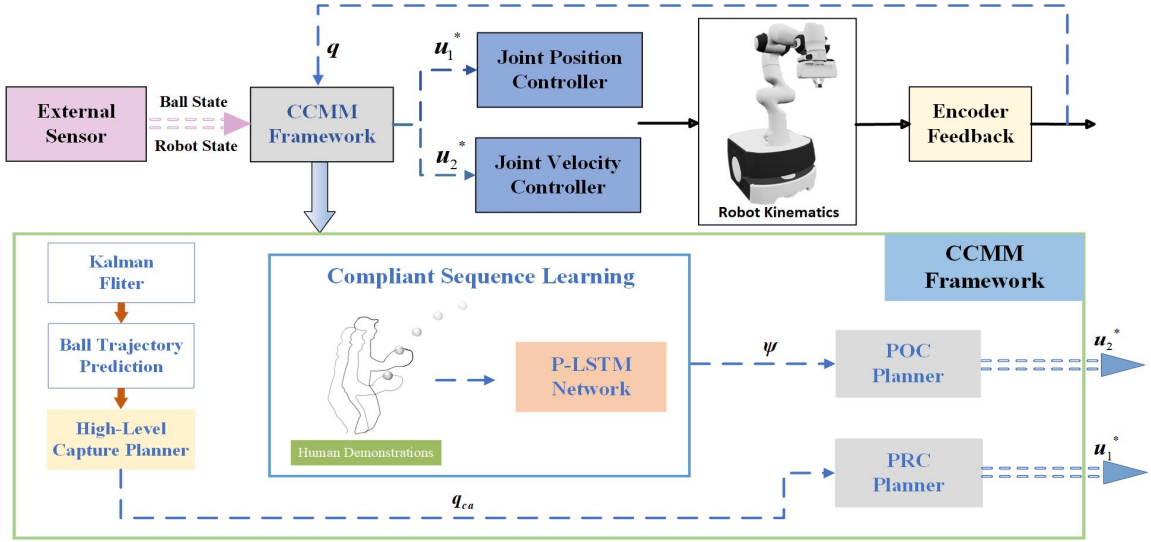
Fig. 2: Overview of the system framework: External sensors provide status for both the ball and the robot. The ball's state is refined using a Kalman filter, which is then used to predict its future trajectory over a time horizon. The high-level capture planner computes the optimal capture joint state, $\boldsymbol{q}_{ca}$. The PRC planner generates rapid motion trajectories to $\boldsymbol{q}_{ca}$ and sends them to the joint position controller. The PRC planner tracks the compliant velocity sequence from the P-LSTM network, which is designed to learn from human demonstrations.

B. High-Level Capture Planner

The high-level planner outputs an optimal catching configuration $\boldsymbol{q}_{ca} \in \mathbb{R}^n$ and catch time $t_{ca} \in \mathbb{R}$ that enables the container to intercept the ball's trajectory at time t_{ca}. From current state $\boldsymbol{q}_0$, catching a moving ball requires the container and ball to be aligned in position and parallel in the direction vector. For this purpose, the following QP problem is formulated:

$$\min_{\boldsymbol{q}_{ca},t_{ca}} \frac{1}{2}\left(\boldsymbol{\Lambda}\|\boldsymbol{q}_{ca}-\boldsymbol{q}_0\|^2 - \alpha\|t_{ca}\|^2\right) \tag{5a}$$

$$\text{s.t.} \quad {}_c^w\boldsymbol{P}(\boldsymbol{q}_{ca}) = \hat{\boldsymbol{p}}(t_{ca}), \tag{5b}$$

$$\frac{{}_c^w\boldsymbol{z}(\boldsymbol{q}_{ca})}{|{}_c^w\boldsymbol{z}(\boldsymbol{q}_{ca})|} = -\frac{\dot{\hat{\boldsymbol{p}}}(t_{ca})}{|\dot{\hat{\boldsymbol{p}}}(t_{ca})|}, \tag{5c}$$

$$ {}_c^w\boldsymbol{T}_z(\boldsymbol{q}_{ca}) \geq \beta, \tag{5d}$$

$$\boldsymbol{q}_{ca}^{min} \leq \boldsymbol{q}_{ca} \leq \boldsymbol{q}_{ca}^{max}. \tag{5e}$$

The objective function tends to minimize joint movement and extend capture time. $\boldsymbol{\Lambda}$ and α are the weights associated with each aspect. $\boldsymbol{\Lambda}$ can be defined as:

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{E}(\Lambda_b) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E}(\Lambda_m) \end{pmatrix}, \tag{6}$$

where $\boldsymbol{E}$ represents identity matrix, Λ_b, Λ_m denotes the cost values associated with the mobile base and the manipulator. Due to the motion control accuracy of the mobile base is generally lower than the manipulator, Λ_b tends to set higher than Λ_m to encourage more movement of the manipulator. ${}_c^w\boldsymbol{P} \in \mathbb{R}^3$ represents the position of the container in the world frame. Equation (5b) ensures that the position of the container is the same as the ball's position at $\boldsymbol{q}_{ca}$. ${}_c^w\boldsymbol{z}(\boldsymbol{q}_{ca}) \in \mathbb{R}^3$ indicates the container's z-axis vector expressed in the world frame. Through (5c), the direction of the container is opposite to the velocity of the ball at t_{ca}, that is, parallel.

${}_c^w\boldsymbol{T}_z(\boldsymbol{q}_{ca}) \in \mathbb{R}$ in (5d) denotes the Z-axis height of the container at the capture configuration $\boldsymbol{q}_{ca}$. Due to the cushioning motions in subsequent POC phase, (5d) is meaningful for the robot to maintain a specific height when catching the ball.

C. PRC Planner

In the high-level capture planner, the robot's joint configuration $\boldsymbol{q}_{ca}$ for catching the ball is determined. The robot then needs to reach this joint configuration as quickly as possible. Considering the maximum acceleration and speed limits for each joint, we determine the minimum trajectory planning time for each joint, denoted as t_{prc}:

$$t_{prc} = \begin{cases} 2\sqrt{\frac{\triangle q}{\ddot{q}_{max}}}, & \triangle q < 2q_{acc} \\ 2\frac{\dot{q}_{max}}{\ddot{q}_{max}} + \frac{\triangle q - \ddot{q}_{max}\left(\frac{\dot{q}_{max}}{\ddot{q}_{max}}\right)^2}{\dot{q}_{max}}, & \triangle q \geq 2q_{acc} \end{cases} \tag{7}$$

where $\triangle q = |q_{ca} - q_0|$, $q_{acc} = \frac{1}{2}\ddot{q}_{max}\left(\frac{\dot{q}_{max}}{\ddot{q}_{max}}\right)^2$. This section determines the shortest theoretical time for the robot to reach the target configuration q_{ca} if following the limits on maximum acceleration and speed. While such an extreme motion is not used in practice, this simplified method helps us quickly estimate the duration of the joint movement. we can define the time parameter T_{prc} of PRC as follow:

$$T_{prc} = \lambda \cdot max(t_{prc}^i) \quad i \in [1,n], \tag{8}$$

where $\lambda > 1$ is a proportion parameter to enlarge planning time appropriately. Then, we can get the PRC planner ϕ_{prc} in joint space:

$$\Phi_{prc} = \text{Poly5}(t; T_{prc}, \boldsymbol{q}_0, \boldsymbol{q}_{ca})), \tag{9}$$

where Poly5() represents a Quintic Polynomial Planning that maps the time t(ranging from 0 to T_{prc}) to the joint position based on the inputs $T_{prc}, \boldsymbol{q}_0, \boldsymbol{q}_{ca}$.
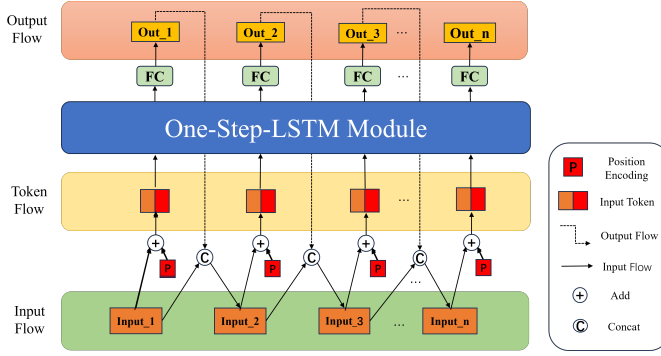
Fig. 3: P-LSTM network for compliant sequence learning from human demonstrations

Fig. 4: The collision occurs when there are no safety constraints in the POC planner.

D. Compliant Sequence Learning

Given the velocity and position states of the object before catching, we employ the P-LSTM network to output the compliant sequences required for POC phase.

Specifically, we concatenate the initial six-dimensional input sequence (comprising position and velocity along the x,y,z-axes) into a variable denoted as $input_1$. This serves as the first input at time step 1. This concatenated vector is then input into a one-step LSTM module, which consists of an LSTM cell followed by a fully connected layer that projects the hidden state to the output dimension. Then, the first output out_1 is produced at time step 1 by the one-step LSTM. Subsequently, we concatenate $input_1$ and out_1 as the second input $input_2$ at time step 2 and again put it into the same one-step-LSTM module to get the second output flow out_2 at time step 2. This process repeats until the maximum sequence length is reached.

To incorporate the positional order of the input sequences, similar to the approach in [30], we apply position encoding in our LSTM model. Specifically, we encode the sine and cosine of the input position sequence and combine these encodings with the input data. The resulting position-encoded input tokens are then passed into the LSTM model, preserving the order information between sequences for improved learning performance. As shown in Fig. 3, the overall output flow can be described as:

$$input_i = input_{i-1} \oplus out_{i-1} + PE(i-1, 6), \quad (10a)$$
$$out_i = \mathcal{L}(input_i), \quad (10b)$$

where $PE(\cdot)$ means positional encoding and i is the i-th time step, $\oplus$ represents concatenation vertically and $\mathcal{L}(\cdot)$ is the one-step-LSTM module.

For each position l in the sequence, the positional encodings are calculated as follows:

$$pe(l, 2i, k) = \sin\left(\frac{l}{10000^{2i/k}}\right), \quad (11a)$$
$$pe(l, 2i+1, k) = \cos\left(\frac{l}{10000^{2i/k}}\right), \quad (11b)$$
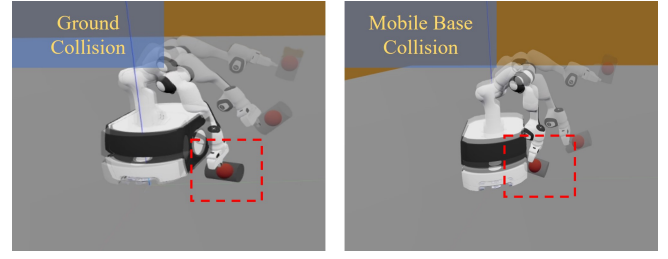$$PE(l, k) = pe(l, 0, k)|pe(l, 1, k)|...|pe(l, k, k), \quad (11c)$$

where $|$ means concatenation horizontally and k is the embedding dimension.

The P-LSTM network is trained on data obtained from human demonstrations. During the application phase, the predicted ball position and velocity at q_{ca}, denoted as $\hat{p}(q_{ca})$ and $\hat{p}(q_{ca})$, is computed and subsequently used as input to the network. Based on this input, the network generates the compliant velocity sequence $\psi \in \mathbb{R}^{6 \times s}$, where s denotes the length of the sequence. We focus on tracking the compliant velocity sequence because position is heavily dependent on the world coordinate system. In contrast, velocity is relative and more general, independent of the position of world coordinate system, making it more universal.

E. POC Planner

After the robot catches the ball, it needs to apply compliant control to have a cushioning mechanism. Given the velocity-compliant sequence ψ produced by the network, we need to track this sequence. Additionally, applying the sequence directly to the robot could result in a series of collisions due to the structural differences between the robot and the human, as shown in Fig. (4). To ensure safety, we must account for potential collisions between the mobile base and the manipulator, as well as between the manipulator and the ground. Therefore, we look at how Control Barrier functions(CBF) [31] can encode safety constraints. The POC planner can be expressed as follows:

$$\Phi_{poc} = \min_{\dot{q}} \frac{1}{2}\left(\|\dot{q}\|^2 + \|\delta\|^2\right) \quad (12a)$$
$$\text{s.t.} \quad {}_c^w J \dot{q} = \psi_i + \delta, \quad (12b)$$
$$\dot{f}(z_c) \cdot {}_c^w J_z \cdot \dot{q} \geq -\gamma_z f(z_c), \quad (12c)$$
$$\dot{g}(x_c, y_c) \cdot {}_c^w J_{xy} \cdot \dot{q} \geq -\gamma_{xy} g(x_c, y_c), \quad (12d)$$

where ψ_i represents the i-th sequence of ψ and δ is the slack vector to reduce the rigor of the equation. The components of the Jacobian in z-axis and the x, y-axis are denoted by ${}_c^w J_z$ and ${}_c^w J_{xy}$. The function $f(z_c)$ represents the safe distances between the container and the ground. And $g(x_c, y_c)$ is the safe horizontal distance between the container and the mobile base. Safety is maintained when $f(z_c) > 0$ and $g(x_c, y_c) > 0$. In the optimization problem above, equality constraints ensure the container keeps track of expectations ψ. To ensure safety, CBFs are used as inequality constraints, which help maintain safe forward invariance [32], [33], that is $f(z_c) > 0$ and $g(x_c, y_c) > 0$. By imposing limits on
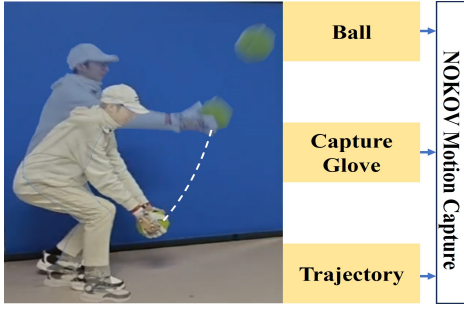
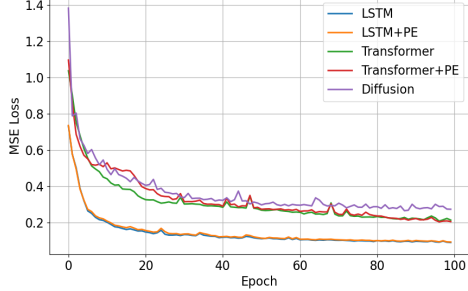Fig. 5: Collecting compliant data from human demonstration in the real world

Fig. 6: Training loss curves for compliant sequence learning across different networks.

TABLE I: Test Experiments

Method	Test MSE	Inference time (s)
Transformer	0.168	0.201
Transformer+PE	0.152	0.223
LSTM	0.096	**0.061**
LSTM+PE	**0.091**	0.062
Diffusion	0.161	0.307

TABLE II: Simulation results over 6000 random catching tasks

Approach	Success Rate	Ground Crash	Base Crash	Not Catch
General	98.70%	0.00%	1.05%	0.25%
Without (12c)	82.80%	13.70%	3.20%	0.30%
Without (12d)	89.40%	0.00%	10.40%	0.20%

the container's height along the z-axis and regulating its horizontal distance from the base, the optimization process filters out potential collisions while minimizing deviations from the desired trajectory.

V. EXPERIMENTS

We evaluate our framework through experiments conducted both in simulation and on a physical mobile manipulator. We use the following value for the parameters of our framework: $k_{ad} = 0.0295$ in (3); $\alpha = 2$ in (5a); $\beta = 0.5$ in (5d); $\Lambda_b = 5, \Lambda_m = 1$ in (6); $\lambda = 1.5$ in (8); $\gamma_z = 0.1$ in (12c); $\gamma_{xy} = 0.1$ in (12d). We use "SLSQP" method to solve nonlinear optimization problems (5a) and (12a). In P-LSTM, we set the hidden size to 64 and a fully connected layer block from 64 to 6. During training, we use mean squared error (MSE) between the predicted output and the ground truth. The learning rate is set to 0.001, and the max output sequence length is capped at 16.

A. Data Collection and Processing

We capture the velocity and position sequences of the ball using a NOKOV motion-capture device during human-catching demonstrations as Fig. (5) shows. Reflective balls are placed on both the glove and the ball. We use the distance between them to determine when to start recording data. In total, 302 trajectories were collected. To distinguish between inputs and outputs of the data for the network, we identified the point with the highest average velocity variation across all data points. This point is then used to split the sequence into input data and the corresponding output labels.

B. Network Comparison Experiment

To evaluate our P-LSTM method, we compare it with four other methods: LSTM without PE, transformer without PE, transformer with PE, and diffusion policy. All methods use 2 hidden layers with a hidden size of 32, and the transformer's number of heads is set to 2 (the input size is 6, with 3 position components and the remaining 3 being velocities, so 2 heads are sufficient to capture the relationships between them). The training and testing metrics are the MSE between the six-dimensional true sequences and the predicted sequences. We randomly divide the data into a training set (4/5) and a test set (1/5).

As shown in TABLE I and Fig. 6, transformer-based methods struggle to converge and validate effectively due to their high parameter demands and the short input sequences. The diffusion policy also fails to train effectively due to the limited data size. Additionally, adding PE to both LSTM and transformer models reduces the error, with almost no impact on inference speed in test experiments."

C. Simulation Scenarios

We use the Swift and Rtb Python libraries [34] as our simulated environments. Initially, we generated 6,000 ball samples $[p_x, p_y, p_z, \dot{p}_x, \dot{p}_y, \dot{p}_z]$, restricting their positions within a cylindrical space of a specified height. We ensured that their trajectories, for $t > 0.5$, would pass through the robot's capture range. Then ball parameters are generated randomly by procedure. Additionally, we performed ablation studies on the collision avoidance constraints in (12c) and (12d) during the POC phase (2,000 pieces of data for each test). The statistical results are presented in TABLE II.

Fig. 7 shows the robot motions of some experiments where the ball comes from different directions. TABLE II demonstrates the results of 6000 random catching tasks, showing that the CCMM framework achieves a 98.70% success rate for catching. In practice, constraints (12c) and (12d) do not affect the PRC planner, resulting in similar "Not Catch" rates. However, incorporating collision avoidance constraints notably reduces the incidence of collisions. Without (12c)
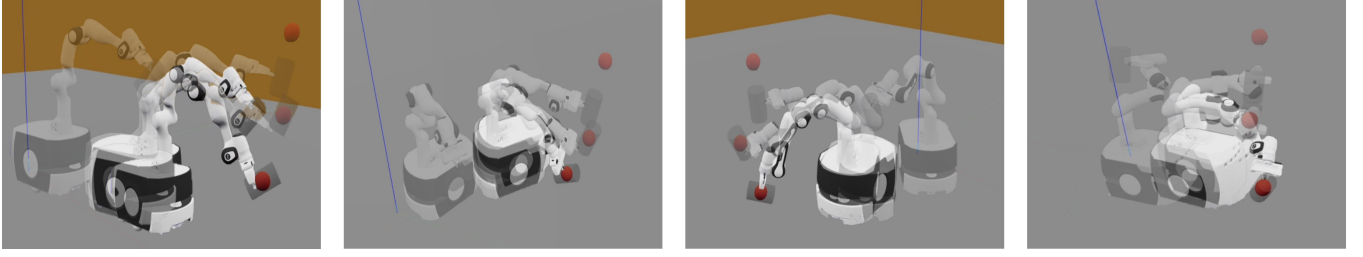
Fig. 7: Snapshots of some catching experiments in the simulation where robot opacity increased with time. It shows the ball approaching the robot from the front, right, left, and back, respectively.
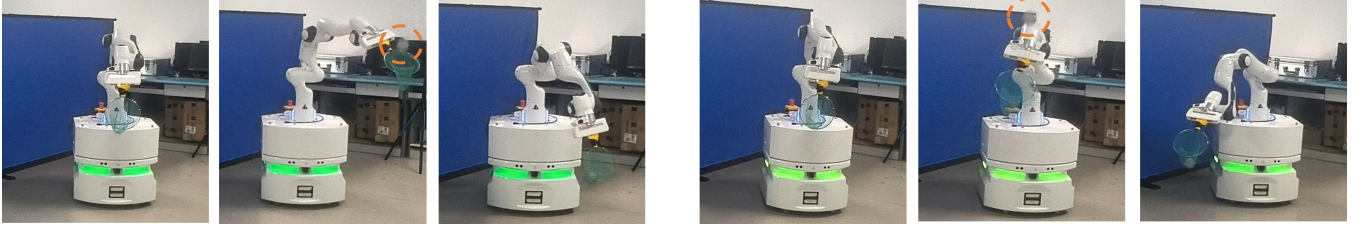

Fig. 8: The robot motions under CCMM framework to catch a ball with compliant control. They represent the starting state of motion, the end state of PRC planner, and the end state of POC planner.

TABLE III: Process statistics

cap. Error(cm)	cap. Time(s)	cal. Time(s)
0.98 ± 0.44	0.73 ± 0.12	0.10 ± 0.03

TABLE IV: Collision avoidance statistics

	Min Groud Dis.(m).	Min Base Dis.(m)
Compliance	0.46 ± 0.20	0.29 ± 0.12

TABLE V: Similarity statistics of trajectories

	Intra-group Dis.(m)	Inter-group Dis.(m)	P Value
DTW Distance	8.39 ± 4.35	9.12 ± 4.72	0.61

TABLE VI: Comparison of peak torque

	Primeval	Compliance	P Value
Max Peak Torque	0.87 ± 0.11	0.62 ± 0.08	$1.17e - 09$

or (12d), the probability of collisions in their respective dimensions increases substantially.

D. Real-World Scenarios

As shown in Fig. 8, a real-world experiment is conducted on a mobile manipulator. We use a C-100 differential mobile base (n_b = 2) and a franka panda manipulator (n_m = 7), interfacing with the robot through ROS Noetic. NOKOV motion capture provides position and velocity measurements, which are used for both ball state estimation and mobile base state tracking. We use a standard tennis ball with a reflective strip (diameter: 6.8 cm, mass: 60.3 g) for tracking purposes, along with an inelastic mesh bag as the container which would not weaken the impact. Due to the coupling effects of differential wheel steering, our high-level planner considers only the combined motion q_1 and restricts the differential motion q_0 to 0 to achieve more accurate tracking in the PRC. When a human hand throws the ball, a Kalman filter models the ball's dynamics using the initial frames and updates the model in the subsequent few frames. This model is then used to predict the ball's trajectory over the next 1.5s. After that, the High-Level planner, PRE planner, and network inference will be executed sequentially, and each will only be executed once. It is worth noting that we tested the delay in ROS master-slave communication and the robot's velocity control

startup time. This delay was accounted for by sending the POC planning command in advance, allowing the robot to start the POC planning right around the time of the catch.

We conducted a total of 100 hand-throwing experiments for compliant catching. We assess the effectiveness of the hand-thrown ball based on the closest distance between the container's initial position and the ball's trajectory. Due to inaccuracies in hand throwing, 19 attempts were deemed invalid. Out of the remaining 81 valid throws, the robot successfully captured the ball 75 times, resulting in a success rate of 92.59%, without any collision. The failure occurred because noise in the capture system caused a discrepancy between the filter's estimated state and the actual state, leading to deviations in the predicted trajectory. It is worth mentioning that, due to the limitation and some obstacles of the laboratory room (some areas wound do not capture the ball well), we did not test in all directions like the simulation, but we still tried to throw to different directions of the robot (left, right, top and middle). In all experiments, TABLE III presents the error between the container's capture point and the actual ball position "cap. Error", along with the capture time "cap. Time" and the total calculation time "cal. Time" for the Hive-Level, the PRC, and the Network inference phase. TABLE IV provides collision avoidance data, including the minimum distances between the container
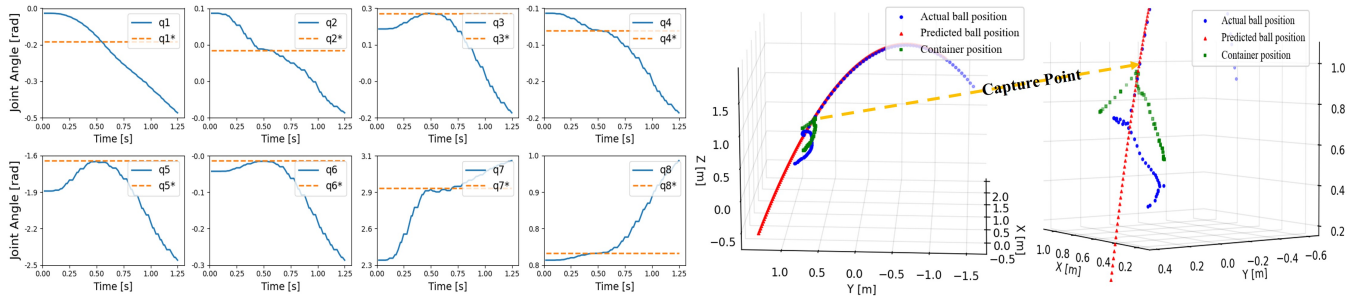
Fig. 9: The change of the joint angles of the robot during a catch, the expected joint angles calculated by the high-level planner, and the trajectory of the ball and the container.
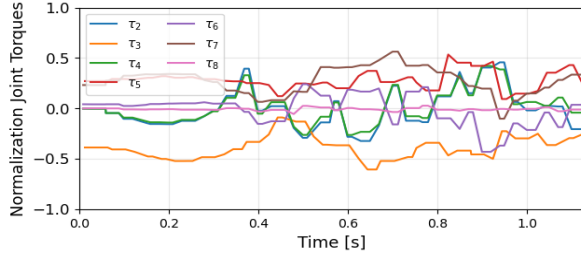

Fig. 10: The curves of normalized joint torques. The maximum peak torque of all joints is 0.606.

and the ground "Min Ground Dis.", as well as the container and the mobile base "Min Base Dis.", during the POC phase. We successfully achieved a high precision capture and collision avoidance.

We compare the effect of adding compliance to reduce torque impact. Since different ball trajectories can result in different torque impacts, ensuring trajectory similarity in the comparison is crucial. To control for variables, we ask the person to consistently face the same direction to maintain similarity and record data 30 times for both the compliant and non-compliant conditions. In each of the 30 trajectories, we quickly select the 20 most similar ones by calculating the vector from the container's initial position to the closest point on trajectory. We then use Dynamic Time Warping (DTW) to assess the similarity between the different group trajectories. The intra-group distance for the compliant group and the inter-group distance between the compliant and non-compliant groups have been calculated, as shown in TABLE V. The final p-value is much greater than the significance level of 0.05, indicating that the ball throwing trajectories of the two groups are similar. Therefore, the impact analysis can proceed to the next step.

Due to different hardware in each joint, every torque was normalized according to the maximum allowed torque at that joint, thus, $\tau'_i = \tau_i/\tau_i^{\max}, i \in [2, 8]$. We calculate the maximum peak torque $max(\tau'_i)$ in all joints during the experiment, as the "Max Peak Torque" in TABLE VI. The p-value for the "Max Peak Torque" between the two groups is significantly lower than the 0.05 significance level, indicating a notable difference between the groups and confirming the feasibility of our compliant strategy. Using the mean

value calculation, the quantitative attenuation ratio is $(0.87 - 0.62)/0.87 = 28.7\%$. Since there is currently no research specifically on compliant catching for mobile manipulators, we referred to the compliant ratio data for a fixed manipulator catching a vertically falling object [7], which is 28%. This result is similar to ours. However, our study is conducted in a more complex catching scenario and with a mobile manipulator. For one test in compliant catching, Figure 9(a) shows the expected output of the high-level capture planner, along with the changes in joint angles during the catching process. At the same time, Figure 9(b) presents the actual predicted trajectory of the ball and the container's trajectory during the compliant catching experiment. Additionally, Fig. 10 illustrates the normalized joint torques.

VI. CONCLUSIONS

This paper aims to address the challenge of compliant catching for mobile manipulators. We have developed a novel CCMM framework that combines optimization and learning to address all the planning and tracking challenges during the catching process. A high-level planner is used to calculate the optimal capture point and joint configuration, while a PRC planner ensures the manipulator reaches the target joint configuration as quickly as possible. The compliant strategies are learned from human demonstrations, with a P-LSTM network proposed to capture temporal dependencies and spatial context through positional encoding. Additionally, a POC planner is proposed to track the sequence predicted by the network while addressing safety concerns arising from structural differences between humans and robots. Experimental results show a 98.70% success rate in simulations, 92.59% in real-world tests, and a 28.7% reduction in impact torque. The limitation of our framework is its reliance on the accuracy of external sensing, as the data from the initial frames is crucial for subsequent predictions and planning. Moving forward, a key focus will be exploring how to complete above tasks using only the robot's camera. Finally, due to the modular design of our CCMM framework, we envision that it would be easily ported to other robots or fields.

REFERENCES

[1] B. Bäuml, T. Wimböck, and G. Hirzinger, "Kinematically optimal catching a flying ball with a hand-arm-system," in *2010 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, 2010, pp. 2592–2599.

[2] Y. Liu, A. Nayak, and A. Billard, "A solution to adaptive mobile manipulator throwing," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 1625–1632.

[3] A. Takahashi, M. Sato, and A. Namiki, "Dynamic compensation in throwing motion with high-speed robot hand-arm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6287–6292.

[4] J. Z. Woodruff and K. M. Lynch, "Robotic contact juggling," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1964–1981, Jun. 2023.

[5] S. Kim, A. Shukla, and A. Billard, "Catching objects in flight," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1049–1065, 2014.

[6] S. S. M. Salehian, M. Khoramshahi, and A. Billard, "A dynamical system approach for softly catching a flying object: Theory and experiment," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 462–471, 2016.

[7] J. Zhao, G. J. G. Lahr, F. Tassi, A. Santopaolo, E. De Momi, and A. Ajoudani, "Impact-friendly object catching at non-zero velocity based on combined optimization and learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 4428–4435.

[8] Y. Wang, N. Dehio, and A. Kheddar, "On inverse inertia matrix and contact-force model for robotic manipulators at normal impacts," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3648–3655, 2022.

[9] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, Nov. 2009.

[10] N. Uchiyama, S. Sano, and K. Ryuman, "Control of a robotic manipulator for catching a falling raw egg to achieve human-robot soft physical interaction," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012, pp. 777–784.

[11] S. Gholami, F. Tassi, E. De Momi, and A. Ajoudani, "A reconfigurable interface for ergonomic and dynamic tele-locomanipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4260–4267.

[12] T. Stouraitis, L. Yan, J. Moura, M. Gienger, and S. Vijayakumar, "Multi-mode trajectory optimization for impact-aware manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9425–9432.

[13] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020.

[14] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, "Trajectory planning for optimal robot catching in real-time," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3719–3726.

[15] G. Wang, W. Wang, P. Ding, Y. Liu, H. Wang, Z. Fan, H. Bai, Z. Hongbiao, and Z. Du, "Development of a search and rescue robot system for the underground building environment," *Journal of Field Robotics*, vol. 40, no. 3, pp. 655–683, 2023.

[16] M. Mashali, L. Wu, R. Alqasemi, and R. Dubey, "Controlling a non-holonomic mobile manipulator in a constrained floor space," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 725–731.

[17] Z. Xie, L. Jin, X. Luo, M. Zhou, and Y. Zheng, "A biobjective scheme for kinematic control of mobile robotic arms with manipulability optimization," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 2, pp. 1534–1545, Apr. 2024.

[18] G. Rizzi, J. J. Chung, A. Gawel, L. Ott, M. Tognon, and R. Siegwart, "Robust sampling-based control of mobile manipulators for interaction with articulated objects," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1929–1946, 2023.

[19] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, 2020.

[20] K. Dong, K. Pereida, F. Shkurti, and A. P. Schoellig, "Catch the ball: Accurate high-speed motions for mobile manipulators via inverse dynamics learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 6718–6725.

[21] M. M. Schill, F. Gruber, and M. Buss, "Quasi-direct nonprehensile catching with uncertain object states," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2468–2474.

[22] U. Frese, B. Bauml, S. Haidacher, G. Schreiber, I. Schaefer, M. Hahnle, and G. Hirzinger, "Off-the-shelf vision for a robotic ball catcher," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 3, 2001, pp. 1623–1629 vol.3.

[23] B. Bäuml, O. Birbach, T. Wimböck, U. Frese, A. Dietrich, and G. Hirzinger, "Catching flying balls with a mobile humanoid: System overview and design considerations," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, Oct. 2011, pp. 513–520.

[24] L. Yan, T. Stouraitis, J. Moura, W. Xu, M. Gienger, and S. Vijayakumar, "Impact-aware bimanual catching of large-momentum objects," *IEEE Transactions on Robotics*, vol. 40, pp. 2543–2563, 2024.

[25] T. Stouraitis, L. Yan, J. Moura, M. Gienger, and S. Vijayakumar, "Multi-mode trajectory optimization for impact-aware manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 9425–9432.

[26] N. Uchiyama, S. Sano, and K. Ryuman, "Control of a robotic manipulator for catching a falling raw egg to achieve human-robot soft physical interaction," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, Sep. 2012, pp. 777–784.

[27] S. Kajikawa, M. Saito, K. Ohba, and H. Inooka, "Analysis of human arm movement for catching a moving object," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, vol. 2, Oct. 1999, pp. 698–703 vol.2.

[28] M. Müller, S. Lupashin, and R. D'Andrea, "Quadrocopter ball juggling," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 5113–5120.

[29] T. D. Barfoot, *State Estimation for Robotics: Second Edition*, 2nd ed. Cambridge: Cambridge University Press, 2024.

[30] H. Kim, Y. Ohmura, and Y. Kuniyoshi, "Memory-based gaze prediction in deep imitation learning for robot manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2427–2433.

[31] X. Xu, "Constrained control of input–output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, Jan. 2018.

[32] H. Zhang, Z. Li, and A. Clark, "Model-based reinforcement learning with provable safety guarantees via control barrier functions," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 792–798.

[33] R. Herguedas, M. Aranda, G. López-Nicolás, C. Sagüés, and Y. Mezouar, "Multirobot control with double-integrator dynamics and control barrier functions for deformable object transport," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1485–1491.

[34] P. Corke and J. Haviland, "Not your grandmother's toolbox–the robotics toolbox reinvented for python," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 357–11 363.