ManiCast: Collaborative Manipulation with Cost-Aware Human Forecasting

Anonymous Author(s)
Affiliation
Address
email

Abstract: Seamless human-robot manipulation in close proximity relies on accurate forecasts of human motion. While there has been significant progress in learning forecast models at scale, when applied to manipulation tasks, these models accrue high errors at critical transition points leading to degradation in downstream planning performance. Our key insight is that instead of predicting the most likely human motion, it is sufficient to produce forecasts that capture how future human motion would affect the cost of a robot's plan. We present MANICAST, a novel framework that learns cost-aware human forecasts and feeds them to a model predictive control planner to execute collaborative manipulation tasks. Our framework enables fluid, real-time interactions between a human and a 7-DoF robot arm across a number of real-world tasks such as reactive stirring, object handovers, and collaborative table setting. We evaluate both the motion forecasts and the end-to-end forecaster-planner system against a range of learned and heuristic baselines while additionally contributing new datasets.

Keywords: Collaborative Manipulation, Forecasting, Model Predictive Control

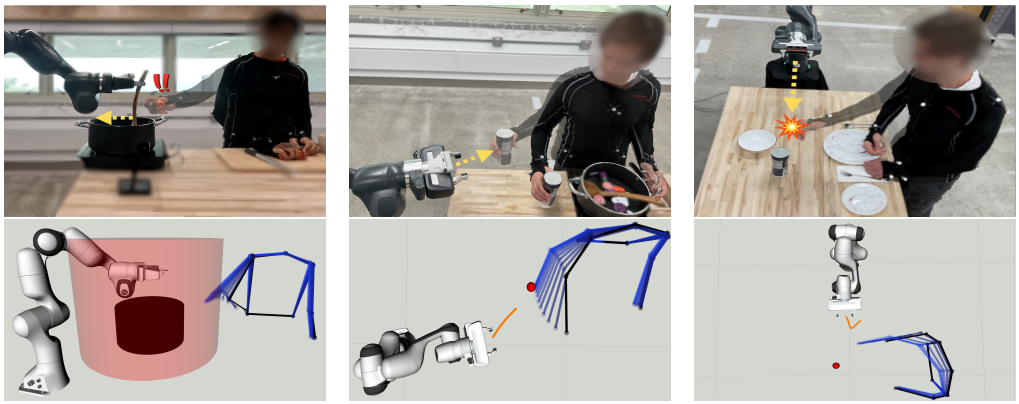
1 Introduction

Seamless collaboration between humans and robots requires the ability to accurately anticipate human actions. Consider a shared manipulation task where a human and a robot collaborate to cook a soup – as the robot stirs the pot, the human adds in vegetables. Such close proximity interactions require fluid adaptions to the human partner while staying safe. To do this, the robot must predict or forecast the human's arm movements, and plan with such forecasts. This paper addresses the problem of generating human motion forecasts that enable seamless collaborative manipulation.

Recent works have made considerable progress in training human motion forecast models by leveraging large-scale human activity datasets such as AMASS [1] and Human 3.6M [2]. However, directly applying such models for human-robot collaboration presents several challenges. First, the space of all possible human motions is very large and pre-trained forecast models typically average out their performance over the distribution of activities seen at train time. Second, although fine-tuning with task-specific data helps improve overall forecasting accuracy, it does not necessarily lead to better performance for downstream planning. This because these models are usually very accurate at predicting frequent and predictable events in the data, e.g. continuing to stir a ladle. However, they struggle with rare and more unpredictable transitions, e.g. pulling the ladle back as a human hand enters the workspace. Such transitions are critical for seamless collaboration and errors at such data points have a substantial impact on the overall performance of the robot.

Our key insight is that instead of predicting the most likely human motion, it is sufficient to produce forecasts that capture how future human motion would affect the cost of a robot's plan. For instance, in the cooking task, the robot's planned trajectory has high cost if it comes close to the human arm, and low cost otherwise. While trying to accurately predict how the human arm may

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.



Reactive Stirring **Object Handover** **Collaborative Table Setting**

Figure 1: Closed-loop, real-time collaborative human-robot manipulation across three different kitchen tasks by combining learned human pose forecasts with model predictive control.

move at any given moment is difficult, it is much easier to predict whether that movement results in a high cost. We achieve this by modifying the forecast training objective to match the cost of ground truth future motions rather than the exact motion itself. We propose a novel framework, MANICAST (Manipulation Forecast), that learns cost-aware human motion forecasts and plans with such forecasts for collaborative manipulation tasks. At train time, we fine-tune pre-trained human motion forecasting models on task specific datasets by upsampling transition points and upweighting joint dimensions that dominate the cost of the robot's planned trajectory. At inference time, we feed these forecasts into a model predictive control (MPC) planner to compute robot plans that are reactive and keep a safe distance from the human. To the best of our knowledge, this is the first paper to combine state-of-the-art pre-trained human forecast models like STS-GCN [3] with a real-time MPC planner [4] for collaborative human-robot manipulation tasks. Our key contributions are:

1. A framework to learn and plan with cost-aware human forecasts for collaborative human-robot manipulation.
2. A new dataset of human-human collaborative manipulation on 3 kitchen tasks.
3. Extensive real-world evaluation of a human-robot team on each of the 3 tasks and comparison against a range of learned and heuristic baselines.

2 Related Work

Navigation with Human Forecasts. Navigating around humans has been a long-standing challenge in areas such as self-driving [5–10] and social robotics [11–17]. In the context of self-driving, there's a rich history in trajectory forecasting of agents observed by the autonomous vehicle (AV) [18] ranging from physics-based methods (e.g. Constant Velocity/Acceleration [19–23], Kalman Filter [24–26]) to machine learning-based methods (e.g. Gaussian Processes [27, 28], Hidden Markov Models [29, 30]) to inverse reinforcement learning [31, 32]. In recent years, the rise of sequence models in deep learning has led to great advances in accurately predicting future states of traffic participants. Parallel to the developments in autonomous driving, social navigation in crowd environments has similarly converged on sequence models [33–37] to forecast trajectories within a crowd. In addition to making predictions about trajectories, many recent works have used forecasts as inputs to motion planners [38–40] or jointly forecasted and planned [41, 42]. The locations swept out by forecasts are considered to be "unsafe" collision regions that the planner avoids. Consequently, evaluation of trajectory forecasting has shifted from accuracy-based metrics [43] to more task-aware metrics that focus on the performance of a downstream planner. However, all of these works approach forecasting as a 2D problem, whereas we look to plan 7D robot manipulation trajectories by forecasting 21D human pose, which is higher dimensional and much more complex.

Human Pose Forecasting. Human pose forecasting involves predicting how a human pose evolves in both space and time. Several works have leveraged Recurrent Neural Networks (RNN) [44, 45] and Transformer Networks [46] to model time, while graph-based models such as Graph Convolu-

tional Networks (GCN) [3, 47] have been used to model the interaction of joints in space. Mao et al. [47] note that human motion over a time horizon is smooth and periodic. They exploit this observation by encoding the context of an agent with a Discrete Cosine Transform (DCT) and then apply an attention module to capture the similarities between current and historical motion sub-sequences. STS-GCN [3] learns the adjacency matrix between different joints and across different timesteps separately to bottleneck the cross-talk interactions between joints across time and use a Temporal Convolutional Network (TCN) for decoding the predictions over a fixed time horizon. Recent works have extended forecasting from single person to multi-person forecasting [48, 49]. Although there have been many advances in pose forecasting, to the best of our knowledge, our work is the first to integrate an entire upper body forecast into a robot manipulation planner.

Close-Proximity Human Robot Collaboration For close proximity human-robot tasks where fluidity is important, forecasting human pose is essential. While there are various works that forecast human pose in the context of robotics [50–53], they don't integrate such predictions into robot planners. Research focused on robot manipulation planning [54–56] around humans has modeled humans in some form at planning time, whether that be assuming static poses [56, 57], tracking the current pose [58, 59], or making predictions only about the motion of specific joints such as the wrist or head [60–63]. Scheele et al. [60] use an unsupervised online learning algorithm to build a model that predicts the remainder of a human reaching motion given the start of it. Ling et al. [61] use the human's head pose, wrist position, and wrist speed as inputs to their forecasting Long Short-Term Memory (LSTM) model to predict future wrist positions which are used as an input to their robot motion planner. In contrast to these methods which focus on specific motions such as reaching and consider certain parts of the body, our work uses a history of human poses to predict future poses of the entire upper body in real-time for safe planning around the human body. To do so, we leverage large databases of human motion [1] for pretraining and finetune forecasters on task specific data.

3 Problem Formulation

Notation. The human's state at timestep t, $s_t^H \in \mathbb{R}^{J\times 3}$, are the 3-$D$ coordinates of J upper-body joints. Let the context[1], $\phi = \{s_{-k+1}^H, \ldots, s_0^H\}$, be the history of human states over the past k timesteps. Let future human and robot trajectories over a horizon T be $\xi_H = \{s_1^H, s_2^H, \ldots, s_T^H\}$ and $\xi_R = \{s_1^R, s_2^R, \ldots, s_T^R\}$ respectively. Let $C(\xi_R|\xi_H)$ denote the cost of the robot trajectory given the human trajectory. Let $P_\theta(\xi_H|\phi) \propto \exp(-\|\xi_H - \mu_\theta(\phi)\|)$ be the learnt forecast model, modelled as a Gaussian, where $\mu_\theta(\phi)$ is the predicted mean trajectory, and θ are the learned parameters.

Planning. The goal of the planner is to compute the optimal robot trajectory ξ_R given human trajectory ξ_H. At inference time, ξ_H is unknown, and the planner must rely on a *forecast* $\hat{\xi}_H \sim P_\theta(.|\phi)$ of the human trajectory generated by the model. The planner then solves for the lowest cost robot trajectory given the forecast — $\arg\min_{\xi_R} \mathop{\mathbb{E}}_{\hat{\xi}_H \sim P_\theta(.|\phi)} C(\xi_R|\hat{\xi}_H)$.

Cost-Aware Human-Pose Forecasting. Typically, the forecast model is trained by maximizing log-likelihood (MLE) of the ground truth human trajectory ξ_H, i.e., $\max_\theta \log P_\theta(\xi_H|\phi)$, which for a Guassian model corresponds to a L2 loss. However, low log-loss can still result in a high mismatch between the costs $C(.|\xi_H)$ and $C(.|\hat{\xi}_H)$, which can result in poor downstream planner performance. For instance, in Fig. 3, the "base model" minimizing log-loss has low errors everywhere except at critical transition points, which has a significant impact on the cost function. Instead, we choose a loss function $\ell(\theta)$ such that the costs calculated with the forecasts match the cost calculated with ground truth for any robot plan ξ_R:

$$\ell(\theta) = \mathbb{E}_\phi \left[\left| \mathbb{E}_{\xi_H \sim P(.|\phi)} C(\xi_R|\xi_H) - \mathbb{E}_{\hat{\xi}_H \sim P_\theta(.|\phi)} C(\xi_R|\hat{\xi}_H) \right| \right] \tag{1}$$

This loss is equivalent to matching moments of a cost function [64, 65]. However, directly implementing this as a loss function is challenging in practice for a number of reasons. First, the cost function has non-differentiable components like collision check modules. Second, the cost function

[1] The context can have other factors like the history of the robot's past states, which we exclude for simplicity

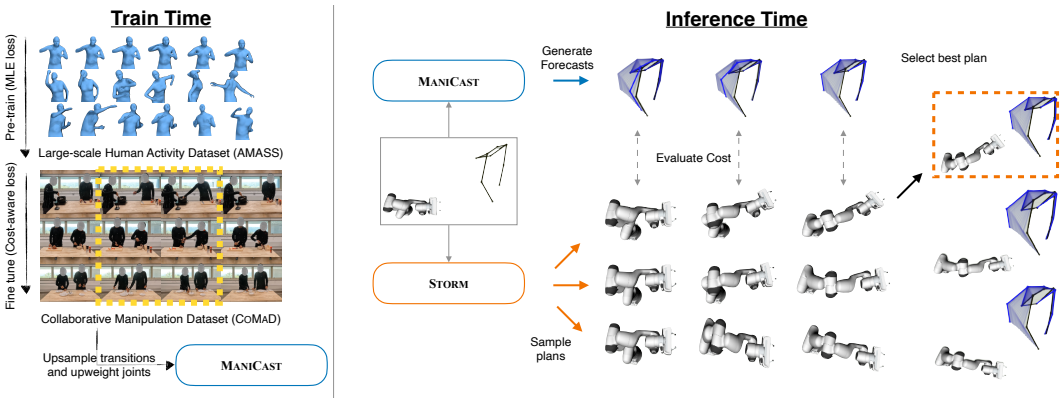


Figure 2: Overview of training and inference pipleine for ManiCast.

may not be convex and have poor convergence rates. Finally, the cost function may not be known upfront at train time, and likely to be tuned frequently. We address these challenges next.

4 Approach

We present ManiCast, a framework that learns cost-aware human forecasts and plans with such forecasts for collaborative manipulation tasks. Fig. 2 shows both the training time and inference time process. At train time, we pre-train a state-of-the-art forecast model (STS-GCN [3]) on large-scale human activity data (AMASS [1]) and fine-tune the forecasts on our Collaborative Manipulation Dataset (CoMaD), with a focus on optimizing downstream planning performance. At inference time, we detect the human pose at 120Hz using an Optitrack motion capture system, generate forecasts at 120Hz, feed the forecasts into a MPC planner (STORM [4]) to compute robot plans at 50 Hz.

4.1 Train Time: Fine-tune Forecasts with Cost-Aware Loss

We first pre-train the model using standard MLE loss on a large scale human activity dataset to get a baseline forecast model. We then fine-tune this model on our own Collaborative Manipulation Dataset (CoMaD), that has data of two humans collaboratively performing manipulation tasks. Since directly optimizing the cost-aware loss in (1) is challenging, we propose two strategies to approximately optimize the loss without significant changes to the model training pipeline.

Strategy 1: Importance Sampling. While the MLE loss measures errors uniformly on the distribution $P(\phi)$, the cost-aware loss $\ell(\theta)$ (1) is sensitive to errors at contexts where costs are generally higher. Notably, for many tasks, transition points where the human comes into the robot's workspace are typically high cost and hence dominate the loss. Since transition points are infrequent, the fine-tuned model has higher errors on them. Our strategy is to assign greater importance to transition points by importance sampling and then finetuning on a new distribution, which we formalize below.

Let $C_{\max}(\phi)$ be the maximum cost of a robot trajectory that a given context can induce, which we compute from collected data. We defined a *transition distribution* as $P_T(\phi) \propto P(\phi)\mathbb{I}(C_{\max}(\phi) \geq \delta)$, i.e., the distribution over contexts that induce a cost higher than a percentile threshold δ. Instead of minimizing the loss on $P(\phi)$, we choose a new distribution $Q(\phi) = 0.5P(\phi) + 0.5P_T(\phi)$ that mixes the original distribution with the transition distribution, effectively up-sampling the transitions. We prove the following performance bounds.

Lemma 1. *For a model θ with bounded loss of ε on $P(\phi)$, the final loss is bounded as $\ell(\theta) \leq C_{\max}\varepsilon$, where $C_{\max} = \max_\phi C_{\max}(\phi)$. In contrast, for a model with bounded loss of ε on the new distribution $Q(\phi)$, the final loss is bounded by $\ell(\theta) \leq 2\max(\delta, C_{\max}\mathbb{E}_{P(\phi)}[\mathbb{I}(C_{\max}(\phi) \geq \delta)])\varepsilon$*

We refer the reader to the Appendix for the proof and interpretation of the bound. For our tasks, we choose a small δ (10%) that works well in practice.

	Models →	BASELINES		LEARNING-BASED					
	Metrics (mm) ↓	CUR	CVM	BASE	SCRATCH	FINETUNED	MANICAST-T	MANICAST	MANICAST-W
AMASS	All Joints ADE	102.5 (±0.2)	194.9 (±0.4)	63.0 (±0.1)	181.2 (±0.2)	99.0 (±0.1)	103.5 (±0.1)	103.5 (±0.1)	99.1 (±0.1)
	All Joints FDE	120.6 (±0.3)	333.6 (±0.6)	92.1 (±0.2)	196.7 (±0.2)	136.8 (±0.2)	141.6 (±0.2)	141.5 (±0.2)	135.3 (±0.2)
	Wrists ADE	182.9 (±0.3)	353.7 (±0.7)	103.9 (±0.2)	263.0 (±0.3)	157.5 (±0.3)	160.0 (±0.3)	157.2 (±0.3)	156.0 (±0.3)
	Wrists FDE	211.0 (±0.5)	602.5 (±1.3)	154.8 (±0.4)	275.4 (±0.4)	218.6 (±0.4)	219.7 (±0.4)	212.8 (±0.4)	214.4 (±0.4)
REACTSTIR	All Joints ADE	63.0 (±1.2)	84.3 (±1.3)	67.7 (±1.0)	67.4 (±0.9)	44.8 (±0.7)	50.6 (±0.6)	45.1 (±0.6)	45.2 (±0.6)
	All Joints FDE	103.2 (±2.2)	166.4 (±2.8)	110.8 (±2.0)	102.1 (±1.7)	81.0 (±1.6)	91.3 (±1.5)	80.5 (±1.5)	81.7 (±1.5)
	Wrists ADE	83.6 (±1.7)	116.2 (±1.8)	94.8 (±1.4)	93.6 (±1.3)	64.7 (±1.1)	75.9 (±1.0)	67.1 (±1.0)	62.9 (±1.0)
	Wrists FDE	136.9 (±3.0)	225.6 (±3.8)	154.2 (±2.8)	137.9 (±2.3)	113.2 (±2.3)	131.7 (±2.1)	115.0 (±2.2)	110.2 (±2.1)
	T-Wrists ADE	132.7 (±3.3)	166.9 (±3.7)	134.3 (±3.3)	129.2 (±2.7)	94.9 (±2.4)	91.1 (±2.1)	89.5 (±2.2)	85.3 (±2.1)
	T-Wrists FDE	226.1 (±5.8)	334.7 (±7.5)	233.8 (±6.0)	203.5 (±4.9)	179.5 (±4.9)	171.3 (±4.4)	168.9 (±4.6)	163.9 (±4.5)
HANDOVER	All Joints ADE	38.9 (±0.4)	57.1 (±0.5)	45.2 (±0.3)	49.9 (±0.3)	31.8 (±0.3)	34.1 (±0.3)	32.5 (±0.3)	32.4 (±0.3)
	All Joints FDE	62.3 (±0.7)	110.3 (±1.1)	70.4 (±0.7)	68.3 (±0.6)	54.6 (±0.6)	57.5 (±0.6)	54.7 (±0.6)	54.8 (±0.6)
	Wrists ADE	55.1 (±0.6)	84.3 (±0.8)	62.0 (±0.5)	68.4 (±0.5)	46.8 (±0.5)	50.7 (±0.4)	48.2 (±0.4)	45.8 (±0.4)
	Wrists FDE	88.1 (±1.1)	162.0 (±1.6)	100.1 (±1.0)	92.6 (±0.8)	79.0 (±0.9)	84.5 (±0.8)	79.4 (±0.8)	77.2 (±0.8)
	T-Wrists ADE	82.4 (±1.3)	114.0 (±1.6)	84.8 (±1.2)	91.2 (±1.0)	67.6 (±1.0)	64.8 (±0.9)	65.8 (±0.9)	62.6 (±0.9)
	T-Wrists FDE	140.0 (±2.2)	228.7 (±3.5)	146.7 (±2.1)	130.7 (±1.6)	124.5 (±1.9)	118.3 (±1.7)	118.7 (±1.7)	113.4 (±1.7)
TABLESET	All Joints ADE	76.5 (±1.1)	99.1 (±1.3)	80.4 (±1.3)	90.2 (±1.3)	58.6 (±1.0)	57.6 (±1.3)	53.6 (±0.7)	56.1 (±1.0)
	All Joints FDE	130.3 (±1.9)	199.9 (±2.9)	137.7 (±2.1)	134.3 (±1.9)	110.5 (±1.7)	106.7 (±2.0)	101.7 (±1.5)	105.1 (±1.6)
	Wrists ADE	103.6 (±1.5)	139.0 (±1.7)	107.3 (±1.5)	114.8 (±1.3)	82.5 (±1.2)	79.6 (±1.4)	76.7 (±1.0)	74.8 (±1.0)
	Wrists FDE	176.0 (±2.5)	276.1 (±3.7)	185.5 (±2.6)	165.0 (±2.2)	151.1 (±2.2)	143.1 (±2.3)	140.9 (±2.1)	138.8 (±2.0)

Table 1: Average forecast metrics (in mm) for all models across all datasets.

Strategy 2: Dimension Weighting. While the MLE or L2 loss sets equal weights to all joint dimensions J, the cost-aware loss $\ell(\theta)$ (1) is sensitive to errors along certain dimensions. For example for a handover task, a small error in predicting the wrist position can have a large impact on the cost. We empirically tune weights $w \in \mathbb{R}^J$ on the MLE loss to upweight joints that are explicitly used as terms in the cost functions we define.

We combine these two strategies to optimize the following proxy loss:

$$\hat{\ell}(\theta) = \mathbb{E}_{0.5P(\phi)+0.5P_T(\phi)} \left[\mathop{\mathbb{E}}_{\xi_H \sim P(.|\phi)} \sum_{j}^{J} w_j \log P_\theta(\xi_H^j | \phi) \right] \tag{2}$$

4.2 Inference Time: Sampling-Based Model Predictive Control with Learned Forecasts

At inference time, we invoke a sampling based model predictive control (STORM [4]) to compute plans for a 7-DoF robot arm. We designed a set of three cost functions, one for each collaborative manipulation task, that take as input a candidate robot plan and the forecasts generated by MANICAST and computes a cost. At every timestep, given a context ϕ, the model predictive control samples plans, evaluates the cost of each plan and updates the sampling distribution till convergence, returning the minimum cost plan. The robot takes a step along the plan and replans. We provide details on the cost function and the MPC planner in the Appendix.

5 Experiments

5.1 Experimental Setup

Collaborative Manipulation Dataset (CoMaD). We design 3 collaborative manipulation tasks shown in Fig 1 (1) *Reactive Stirring*: Robot stirs a pot while making way for human pouring in vegetables (2) *Object Handovers*: Robot moves to receive an object behind handed over by the human (3) *Collaborative Table Setting*: robot and the human manipulate objects on a table while not getting in each other's way. To train our forecasting model, we collect a dataset of two humans executing these tasks. It contains 19 episodes of reactive stirring, 27 episodes of handovers, and 15 episodes of collaborative table setting. We refer to Appendix for more details.

Large Human-Activity Databases. The AMASS (Archive of Motion Capture As Surface Shapes) [1] dataset is a large and diverse collection of human motion. It consists of over 40 hours of single-human motion spanning over 300 subjects. Hence we pre-train models on AMASS.

Forecast Models. We run the planner with different forecasts. (1) *Ground Truth*: CUR assumes the current human pose remains constant over the planning horizon, and FUT uses the real futures obtained from playing back CoMaD's episodes. While the latter can not be used for real-world closed-loop planning, it's the gold standard for comparing other models. (2) *Baselines*: The Constant

	Model →	GroundTruth		Baselines		Learning-Based					
	Metric ↓	Cur	Fut	CVM	Worst	Base	Scratch	FineTuned	ManiCast-T	ManiCast	ManiCast-W
React	Stop Time (ms)	0 (±0)	1000 (±0)	367.8 (±50.9)	1000	-138.9 (±18.8)	-237.5 (±54.4)	203.3 (±22.3)	271.1 (±25.6)	246.7 (±25.8)	290.0 (±29.9)
React	Restart Time (ms)	0 (±0)	1000 (±0)	235.6 (±18.2)	0 (±0)	256.7 (±45.6)	235.0 (±16.8)	441.1 (±30.8)	454.4 (±36.7)	455.6 (±33.5)	496.7 (±27.9)
React	FDR	0%	0%	67%	100%	11%	11%	0%	11%	0%	11%
Hand.	Goal Detection (ms)	0 (±0)	1000 (±0)	246.0 (±146.0)	-	-127.0 (±16.6)	-61.6 (±23.6)	189.3 (±47.8)	473.3 (±123.7)	450.0 (±124.8)	488.4 (±133.4)
Hand.	Correct Goal Rate	100%	100%	20%	0%	90%	80%	100%	100%	100%	100%
Hand.	Path Length (mm)	459.0 (±37.0)	381.1 (±39.8)	485.0 (±85.0)	-	488.9 (±56.2)	466.2 (±33.2)	432.0 (±39.0)	428 (±42.0)	404.0 (±45.0)	436.0 (±49.4)
Hand.	Time to Goal (s)	4.59 (±0.37)	2.87 (±0.24)	3.77 (±0.86)	-	4.32 (±0.48)	4.60 (±0.68)	3.87 (±0.28)	3.91 (±0.32)	3.96(±0.54)	4.17 (±0.45)

Table 2: We integrate forecasts of different models into STORM for the reactive stirring and object handover tasks. The standard error for each planning metric is shown inside parentheses.

Velocity Model (CVM) calculates future human pose by integrating a constant velocity, calculated from the immediate 0.4s history of each joint. The Worst case model is a conservative model that constructs two large spheres (arm length radius) centered at each shoulder joint. (3) Learning-Based: We train our forecasting models using the STS-GCN [3] architecture. The Base model is trained only on AMASS data, whereas the Scratch model is trained only on CoMaD. The Fine-tuned model pre-trains on AMASS, but is finetuned on CoMaD only using the MLE objective. ManiCast is finetuned using the loss objective in Eq.2 with the weights $w_j = 1$. ManiCast-T is finetuned only on transition data in CoMaD. ManiCast-W upweights loss for wrist joints by 5x.

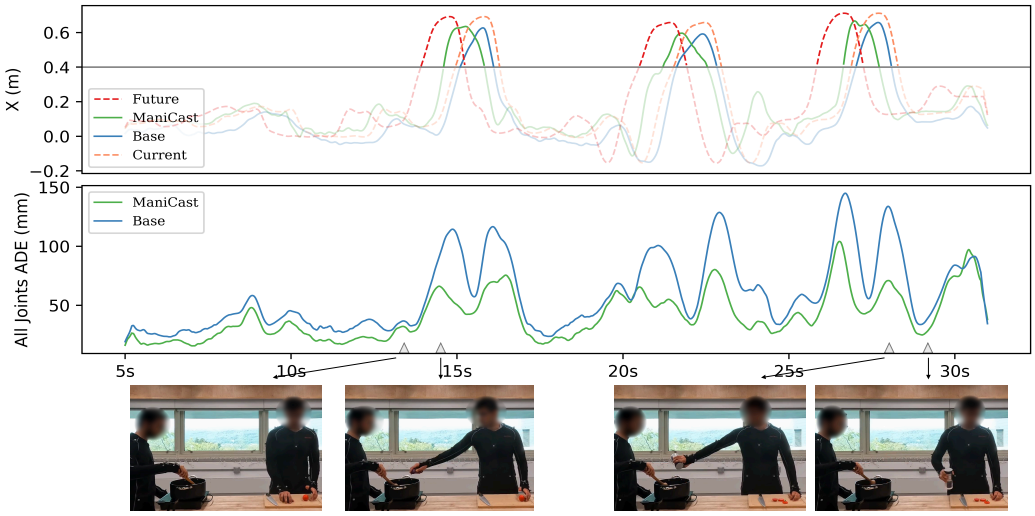


Figure 3: (Top) The x-position of the reaching human's wrist in a Reactive Stirring test set episode. $x \geq 0.4$ indicates the wrist is near the pot. (Bottom) Base model's forecasts have high errors during transitions and lag behind the current pose. ManiCast, trained on CoMaD by upsampling transitions, predicts the reaching human's pose faster than tracking the current pose.

Forecasting Metrics. We measure the Average Displacement Error (ADE) across all 25 timesteps of prediction and the Final Displacement Error (FDE) at the final timestep of prediction. Metrics are measured separately for 7 upper body joints and wrist joints. We also report the wrist errors inside the transition windows for the stirring and handover tasks.

Planning Metrics. For quantitative results, we ran a 7-DoF Franka Emika Research 3 in the real-world and played back a recording of a human partner from CoMaD. *Reactive Stirring*: Test set had 9 human movements into the pot. We report average time required by the robot to stop and restart motion compared to using the current position for planning. We also report the False Detection Rate (FDR) for the human coming into the robot's workspace. *Object Handover*: Test set had 10 handovers. We report the average gain in Goal Detection time compared to using the current pose and the percentage of times when the correct goal is detected by the forecaster. Among instances where the correct goal is detected, we report the average time the robot arm took to reach the handover location and the total path distance moved by its end-effector. For qualitative closed-loop evaluation (Fig 1, Fig 4, Fig 5), we run our entire pipeline with two new human subjects **not part of CoMaD**.

5.2 Results and Analysis

***O1.* ManiCast forecasts outperform current pose tracking.** Table 1 shows that ManiCast models have lower ADE and FDE in forecasting human pose with improved planning performance

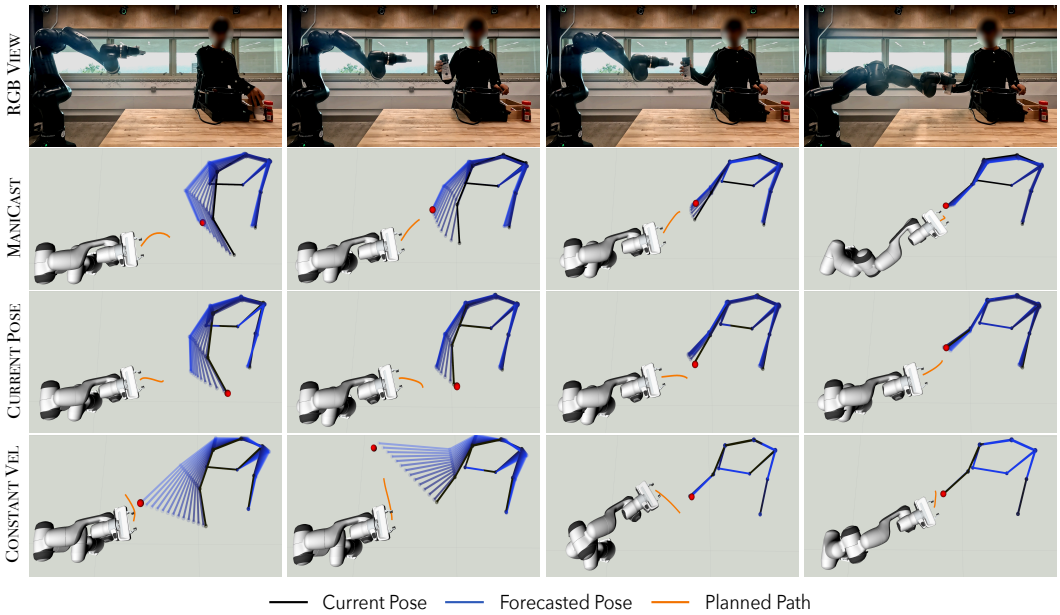
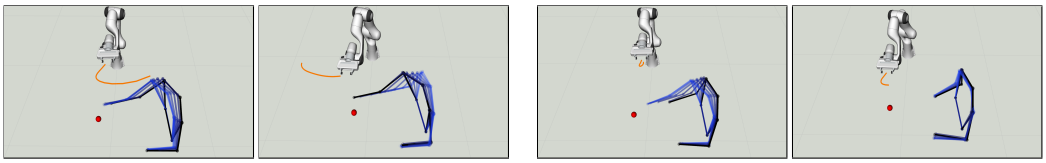


Figure 4: Integrating forecasting models with STORM for real-world human-robot handovers. ManiCast forecasts allow the robot to track a shorter path to the handover location in less time than following the current pose or the dynamically infeasible constant velocity predictions.



Planning with current pose tracking | Planning with ManiCast forecasts

Figure 5: Collision avoidance in Collaborative Table Setting. Following the current pose, the robot arm nearly collides with the human due to delayed prediction. ManiCast avoids this by slowing down when the human reaches in and speeding up when the human retracts their arm.

(Table 2). In the reactive stirring task, ManiCast models predict both the arrival and departure of humans before the current pose. Fig 3 showcases ManiCast predictions over an entire episode from the CoMaD's test set. In the object handover task, current pose tracking predicts the goal location slower than the ManiCast models. As Fig 4 demonstrates, the robot's end effector chases the current wrist pose towards its eventual final location, leading to a longer trajectory that takes more time to execute. Further, planning with forecasts leads to safe motion as collisions can be prevented with future human positions (Fig 5).

***O2.* Baseline models predict dynamically infeasible forecasts leading to suboptimal planning performance.** As seen in Fig 4, since the CVM's predictions are not dynamically constrained, its forecasts overpredict future human positions making the robot arm deviate from the optimal path during handovers. The forecasting metrics (Table 1) highlight the large errors accrued by the CVM. On the other hand, the robot arm remains retracted during the reactive stirring task and does not approach the human's wrist for handovers following the extremely conservative Worst case model.

***O3.* Training on AMASS or CoMaD alone is not sufficient to achieve optimal performance.** The Base and Scratch models produce erroneous forecasts on CoMaD (Table 1). In most cases, they have larger prediction errors than simply tracking the current pose. While the Base model produces the best forecasts on AMASS, the activities in CoMaD contains reaching arm motions and abrupt transitions that are not captured by the model. Fig 3 shows that its predictions lag behind current pose, when predicting the arm to retract. Whereas, Scratch is only trained on CoMaD. While this model is exposed to task-specific data, its training does not converge as forecasting 21D

human pose is a data-intensive complex task. Consequently, planning with both of these forecasters lead to worse performance than just tracking the current pose (Table 2).

***O4.* By upsampling transition points, MANICAST improves the accuracy and efficiency of task completion.** Observing Table 1, we note that MANICAST models have higher displacement errors on CoMaD's test set than the FINETUNED model trained on just the MLE objective. However, since the MANICAST models upsample transitions, they have lower prediction errors in transition windows relevant to planning performance. In the reactive stirring task, they detect the arrival and departure of the human in the robot's workspace quicker than FINETUNED. In the object handover task, this difference is pronounced as the MANICAST models predict the final handover location more than 2 times faster than FINETUNED (Table 2). While MANICAST-T, trained on only transitions, has lower forecasting errors during transitions, planning with it can lead to erratic performance in some cases. For example, in the reactive stirring task, it falsely predicts the human's arrival into the robot's workspace in a test set episode.

***O5.* Upweighting wrist joints in the loss function can improve performance in planning tasks.** For the collaborative manipulation tasks considered in this work, we note that predicting the location of the wrist joints has higher priority than the other upper body joints. MANICAST-W assigns 5 times more weight to the wrist joints in its loss function. This leads to lower forecasting errors for the wrist joint in CoMaD compared to all other models (Table 1). In most cases (Table 2), this model has the best planning performance with faster detection of human pose in the robot's workspace during both the reactive stirring and handover tasks. In some cases, we note that its predictions can be unstable leading to one false detection in the reactive stirring task and larger end-effector path movement in the handover task.

6 Discussion

This work considers the problem of collaborative robot manipulation in the presence of humans in the workspace. We present MANICAST, a novel framework for seamless human-robot collaboration that generates cost-aware forecasts and plans with them in real-time. Our approach was thoroughly tested on three collaborative tasks with a human partner in a real-world setting. MANICAST **leverages large databases of human activity**. Producing dynamically consistent 21D human pose is a complex and high-dimensional problem. However, by training on a large dataset, ManiCast is able to learn the statistical regularities of human motion. MANICAST **optimizes planning performance** by using an approximate cost-aware loss function. We finetune the forecasting model on a novel Collaborative Manipulation Dataset (CoMaD) by upsampling transition points and upweighting relevant joint dimensions. Our system can track a human user's pose, forecast their movements, and control a robotic arm in **real-time and at high speed**. Our system's forecasting module runs at 120Hz, used by an MPC at 50Hz. Fast replanning enables robots to safely collaborate with humans by allowing them to adjust their motion in response to abrupt changes in their environment. In future directions of work, we plan to extend our framework to produce conditional forecasts of human motion given robot trajectories. Further, we will attempt to directly optimize the cost function for a task instead of approximating the forecasting loss function.

7 Limitations

Several reasons limit our approach's deployment to real-world human users. Firstly, our method relies on a motion capture system consisting of 10 cameras to track the history of a human's pose. Further, the user is required to wear a motion capture suit which can be cumbersome. Future work will attempt to detect human skeletons using an ego-centric camera view. Secondly, CoMaD captures just two human subjects across its entire dataset. While we qualitatively show generalization to a new human subject, we do not extensively test it on a range of users. Additionally, the forecaster may not generalize to users with movement styles not represented in the data. Future datasets should encompass multiple subjects with diverse movement styles for widespread usage of human-pose forecasting in personal robotics. This must be accompanied by a comprehensive user study to test the overall pipeline. Finally, we only considered the upper body skeleton for forecasting motion as there was significant occlusion in detecting the lower body. Our forecaster can not be directly applied to robotics applications in which the human's lower body movement is of interest.

References

[1] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019.

[2] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[3] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso. Space-time-separable graph convolutional network for pose forecasting. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11189–11198, 2021.

[4] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. D. Ratliff, D. Fox, F. Ramos, and B. Boots. Fast joint space model-predictive control for reactive manipulation. In *Conference on Robot Learning*, 2021.

[5] Z. Huang, H. Liu, J. Wu, and C. Lv. Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving. *arXiv preprint arXiv:2207.10422*, 2022.

[6] A. Sadat, M. Ren, A. Pokrovsky, Y.-C. Lin, E. Yumer, and R. Urtasun. Jointly learnable behavior and trajectory planning for self-driving vehicles. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3949–3956. IEEE, 2019.

[7] Z. Huang, H. Liu, J. Wu, and C. Lv. Conditional predictive behavior planning with inverse reinforcement learning for human-like autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[8] K. Mangalam, E. Adeli, K.-H. Lee, A. Gaidon, and J. C. Niebles. Disentangling human dynamics for pedestrian locomotion forecasting with noisy supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2784–2793, 2020.

[9] A. Monti, A. Bertugli, S. Calderara, and R. Cucchiara. Dag-net: Double attentive graph neural network for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2551–2558. IEEE, 2021.

[10] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

[11] P. Kothari and A. Alahi. Safety-compliant generative adversarial networks for human trajectory forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[12] H. Nishimura, B. Ivanovic, A. Gaidon, M. Pavone, and M. Schwager. Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11205–11212. IEEE, 2020.

[13] C. Chen, Y. Liu, S. Kreiss, and A. Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *2019 international conference on robotics and automation (ICRA)*, pages 6015–6022. IEEE, 2019.

[14] S. Poddar, C. Mavrogiannis, and S. S. Srinivasa. From crowd motion prediction to robot navigation in crowds. *arXiv preprint arXiv:2303.01424*, 2023.

[15] S. Liu, P. Chang, Z. Huang, N. Chakraborty, W. Liang, J. Geng, and K. Driggs-Campbell. Intention aware robot crowd navigation with attention-based interaction graph. *arXiv preprint arXiv:2203.01821*, 2022.

[16] K. Li, M. Shan, K. Narula, S. Worrall, and E. Nebot. Socially aware crowd navigation with multimodal pedestrian trajectory prediction for autonomous vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[17] C. Mavrogiannis, K. Balasubramanian, S. Poddar, A. Gandra, and S. S. Srinivasa. Winding through: Crowd navigation via topological invariance. *IEEE Robotics and Automation Letters*, 8(1):121–128, 2022.

[18] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 7(3):652–674, 2022.

[19] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone. Sensor fusion for predicting vehicles' path for collision avoidance systems. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):549–562, 2007.

[20] R. Schubert, E. Richter, and G. Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. In *2008 11th international conference on information fusion*, pages 1–6. IEEE, 2008.

[21] N. Kaempchen, B. Schiele, and K. Dietmayer. Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 10(4):678–687, 2009.

[22] R. Pepy, A. Lambert, and H. Mounier. Reducing navigation errors by planning with realistic vehicle model. In *2006 IEEE Intelligent Vehicles Symposium*, pages 300–307. IEEE, 2006.

[23] C.-F. Lin, A. G. Ulsoy, and D. J. LeBlanc. Vehicle dynamics and external disturbance estimation for vehicle path prediction. *IEEE Transactions on Control Systems Technology*, 8(3): 508–518, 2000.

[24] V. Lefkopoulos, M. Menner, A. Domahidi, and M. N. Zeilinger. Interaction-aware motion prediction for autonomous driving: A multiple model kalman filtering scheme. *IEEE Robotics and Automation Letters*, 6(1):80–87, 2020.

[25] B. Jin, B. Jiu, T. Su, H. Liu, and G. Liu. Switched kalman filter-interacting multiple model algorithm based on optimal autoregressive model for manoeuvring target tracking. *IET Radar, Sonar & Navigation*, 9(2):199–209, 2015.

[26] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. Dietmayer. Imm object tracking for high dynamic driving maneuvers. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 825–830. IEEE, 2004.

[27] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy. A bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, 31:383–400, 2011.

[28] Q. Tran and J. Firl. Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. In *2014 ieee intelligent vehicles symposium proceedings*, pages 918–923. IEEE, 2014.

[29] Q. Deng and D. Söffker. Improved driving behaviors prediction based on fuzzy logic-hidden markov model (fl-hmm). In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 2003–2008. IEEE, 2018.

[30] H. Berndt, J. Emmert, and K. Dietmayer. Continuous driver intention recognition with hidden markov models. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 1189–1194. IEEE, 2008.

[31] D. S. González, O. Erkent, V. Romero-Cano, J. Dibangoye, and C. Laugier. Modeling driver behavior from demonstrations in dynamic environments using spatiotemporal lattices. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3384–3390, 2018. doi:10.1109/ICRA.2018.8460208.

[32] N. Deo and M. M. Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans, 2021.

[33] P. Kothari, S. Kreiss, and A. Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7386–7400, 2021.

[34] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[35] D. Varshneya and G. Srinivasaraghavan. Human trajectory prediction using spatially aware deep attention models. *arXiv preprint arXiv:1705.09436*, 2017.

[36] F. Bartoli, G. Lisanti, L. Ballan, and A. Del Bimbo. Context-aware trajectory prediction. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1941–1946. IEEE, 2018.

[37] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural networks*, 108:466–478, 2018.

[38] L. Chen, L. Platinsky, S. Speichert, B. Osinski, O. Scheel, Y. Ye, H. Grimmett, L. D. Pero, and P. Ondruska. What data do we need for training an av motion planner? *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1066–1072, 2021.

[39] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J. M. Allen, V.-D. Lam, A. Bewley, and A. Shah. Learning to drive in a day. *2019 International Conference on Robotics and Automation (ICRA)*, pages 8248–8254, 2018.

[40] D. S. González, J. Pérez, V. M. Montero, and F. Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17:1135–1145, 2016.

[41] J. L. Vázquez, A. Liniger, W. Schwarting, D. Rus, and L. V. Gool. Deep interactive motion prediction and planning: Playing games with motion prediction models. *ArXiv*, abs/2204.02392, 2022.

[42] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. J. Weiss, B. Sapp, Z. Chen, and J. Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2022.

[43] B. Ivanovic and M. Pavone. Rethinking trajectory forecasting evaluation, 2021.

[44] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*, pages 4346–4354, 2015.

[45] A. Jain, A. Zamir, S. Savarese, and A. Saxena. Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA*, pages 27–30, 2016.

[46] Y. Cai, L. Huang, Y. Wang, T.-J. Cham, J. Cai, J. Yuan, J. Liu, X. Yang, Y. Zhu, X. Shen, et al. Learning progressive joint propagation for human motion prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 226–242. Springer, 2020.

[47] W. Mao, M. Liu, and M. Salzmann. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*, 2020.

[48] E. Vendrow, S. Kumar, E. Adeli, and H. Rezatofighi. Somoformer: Multi-person pose forecasting with transformers. *arXiv preprint arXiv:2208.14023*, 2022.

[49] J. Wang, H. Xu, M. G. Narasimhan, and X. Wang. Multi-person 3d motion prediction with multi-range transformers. In *Neural Information Processing Systems*, 2021.

[50] L. Gui, K. Zhang, Y.-X. Wang, X. Liang, J. M. F. Moura, and M. M. Veloso. Teaching robots to predict human motion. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 562–567, 2018.

[51] J. Laplaza, A. Pumarola, F. Moreno-Noguer, and A. Sanfeliu. Attention deep learning based model for predicting the 3d human body pose using the robot human handover phases. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pages 161–166. IEEE, 2021.

[52] J. Zhang, H. Liu, Q. Chang, L. Wang, and R. X. Gao. Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. *CIRP annals*, 69(1):9–12, 2020.

[53] J. Laplaza, F. Moreno-Noguer, and A. Sanfeliu. Context attention: Human motion prediction using context information and deep learning attention models. In *ROBOT2022: Fifth Iberian Robotics Conference: Advances in Robotics, Volume 1*, pages 102–112. Springer, 2022.

[54] M. Faroni, M. Beschi, and N. Pedrocchi. An mpc framework for online motion planning in human-robot collaborative tasks. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1555–1558. IEEE, 2019.

[55] M. V. Minniti, R. Grandia, K. Fäh, F. Farshidian, and M. Hutter. Model predictive robot-environment interaction control for mobile manipulation tasks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1651–1657. IEEE, 2021.

[56] W. Yang, B. Sundaralingam, C. Paxton, I. Akinola, Y.-W. Chao, M. Cakmak, and D. Fox. Model predictive control for fluid human-to-robot handovers. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6956–6962. IEEE, 2022.

[57] E. A. Sisbot and R. Alami. A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(5):1045–1057, 2012.

[58] P. A. Lasota, G. F. Rossano, and J. A. Shah. Toward safe close-proximity human-robot interaction with standard industrial robots. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 339–344. IEEE, 2014.

[59] H. Liu and L. Wang. Collision-free human-robot collaboration based on context awareness. *Robotics and Computer-Integrated Manufacturing*, 67:101997, 2021.

[60] S. Scheele, P. Howell, and H. Ravichandar. Fast anticipatory motion planning for close-proximity human-robot interaction. *arXiv preprint arXiv:2305.11978*, 2023.

[61] H. Ling, G. Liu, L. Zhu, B. Huang, F. Lu, H. Wu, G. Tian, and Z. Ji. Motion planning combines human motion prediction for human-robot cooperation. In *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 672–677. IEEE, 2022.

[62] V. Unhelkar, P. A. Lasota, Q. Tyroller, R.-D. Buhai, L. Marceau, B. Deml, and J. A. Shah. Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. *IEEE Robotics and Automation Letters*, 3:2394–2401, 2018.

[63] J. Mainprice, R. Hayne, and D. Berenson. Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 885–892, 2015.

[64] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[65] G. Swamy, S. Choudhury, J. A. Bagnell, and S. Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pages 10022–10032. PMLR, 2021.