DOI: 10.1111/mice.13129

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Intention-aware robot motion planning for safe worker–robot collaboration

Yizhi Liu[1] | **Houtan Jebelli**[2]

[1]Department of Civil and Environmental Engineering, Syracuse University, Syracuse, New York, USA

[2]Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, Champaign, Illinois, USA

Correspondence
Houtan Jebelli, Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, Champaign, IL, USA.
Email: hjebelli@illinois.edu

Funding information
National Science Foundation Awards, Grant/Award Numbers: 2222654, 2221167

Abstract

Recent advances in robotics have enabled robots to collaborate with workers in shared, fenceless workplaces in construction and civil engineering, which can improve productivity and address labor shortages. However, this collaboration may lead to collisions between workers and robots. Targeting safe collaboration, this study proposes an intention-aware motion planning method for robots to avoid collisions. This method involves two novel deep networks that allow robots to anticipate the motions of workers based on inferences about workers' motion intentions. Then, a probabilistic collision-checking mechanism is developed that enables robots to estimate the collision probability with the motions of workers and generate collision-free adjustments. The results verify that the method enables robots to predict workers' intended motions 1 s in advance and generate adjustments with a collision probability of less than 5.0% during collaborative masonry tasks. This study facilitates the safe implementation of collaborative robots in construction and civil engineering.

1 | INTRODUCTION

With the recent advancement of mechatronics and sensing technology, the research and practice of robotics in industrial settings is undergoing a profound paradigm shift. Traditionally, robots were confined to tasks within cages or behind fences, isolated from workers (Haddadin & Croft, 2016; Villani et al., 2018). However, recent developments empower robots to be capable of engaging in physical collaboration with workers within shared, fenceless workspaces (Ajoudani et al., 2018; Bauer et al., 2008). The shift enables workers and robots to collaborate closely, aiming to integrate the tirelessness and repeatability of machines with the adaptability and decision-making ability of workers, so as to enhance the efficiency and precision in executing physically demanding tasks that necessitate robotic efficiency and human dexterity (Thomaz et al., 2016; Villani et al., 2018).

In the domains of construction and civil engineering, the ongoing shift toward human–robot collaboration (HRC) persists as a significant trend (Bock & Linner, 2016; You et al., 2018). Researchers have been endeavoring to transition from purely robotic solutions to HRC solutions, aiming to help mechanical machines (robots) perform dynamic and dexterous construction tasks by leveraging the flexibility and versatility of workers. Representatively, in masonry tasks, the initial prototypes of standalone robotic solutions such as Flexible Manufacturing System–integrated bricklayer (Altobelli et al., 1993) and Experimental Robotic Masonry System (ERMaS) (Rihani & Bernold, 1994) have evolved into HRC solutions represented by the Semiautomated Mason (SAM) robot and Material Unit Lift Enhancer (MULE) robot. In these setups, both workers and robots operate within a shared environment. Workers are responsible for dexterous tasks like positioning bricks, while collaborative robots are responsible for physically demanding tasks such as lifting and placing bricks (Bock & Linner, 2016; Mule 135 Owner's Manual, 2018; Petters & Belden, 2014). Furthermore, the newly developed construction robot, Hilti Jaibot (Xu et al., 2022), is designed to work alongside human workers, permitting them to closely supervise operations, especially during the mechanical, electrical, and plumbing (MEP) construction process.

While HRC solutions hold promise for enhancing work quality and efficiency, they face challenges distinct from traditional human–human collaboration in construction. In the latter, human intelligence fosters mutual understanding among workers, ensuring safety and dependability (Goody, 1995). In contrast, HRC solutions struggle to maintain the same level of safety and dependability and may even introduce risks to worker well-being. This discrepancy arises from robots' limitations in understanding human collaborators' positions, movement direction, and motion intentions due to their lack of human-like perception and reasoning capabilities (Lee & Adams, 2004; Robla-Gomez et al., 2017; Shim et al., 2023). Consequently, within the dynamic context of HRC, when workers move in close proximity to the robot, the robot that lacks the comprehension of the worker's position and motion may lead to collisions with workers (Vasic & Billard, 2013), resulting in severe injuries (Haddadin et al., 2012; Kulić & Croft, 2005).

To prevent collisions between collaborative robots and workers, researchers adhere to the "Safety-rated Monitored Stop" (SMS) and "Speed and Separation Monitoring" (SSM) collaborative modes regulated by the International Organization for Standardization (ISO) 10218-1/2 (Vasic & Billard, 2013). Employing proximity sensing technology (radar, GPS, and camera), researchers have developed several methodologies that enable robots to avoid collisions by gauging separation distances from workers (Fragkiadaki et al., 2015; Hu et al., 2016; Liu et al., 2021). In these methods, if the separation distance falls beneath the safety threshold, the robot will be controlled to pause or halt its operation, thereby avoiding potential collisions with workers (Figure 1a). Such distance-based collaborative modes have demonstrated their efficiency, particularly in manufacturing contexts. The structured layouts and clearly defined workspaces for both workers and robots in manufacturing environments ensure that these distance-based methods can effectively mitigate collision risks (Balan & Bone, 2006; Luo & Mai, 2019; Wang et al., 2018).

In transitioning distance-based collaborative modes directly to construction environments, these methods remain effective in preventing collisions. However, they introduce a trade-off: Their tendency to cause halts or pauses can affect the smoothness and efficiency of worker–robot collaboration in construction (Liu & Wang, 2021). Construction sites, being dynamic environments, frequently have workers moving in close proximity to robots. When workers approach within the safety boundary (threshold) set by distance-based collaborative modes, the robot needs to temporarily halt or pause its operation, reducing the overall efficiency of the HRC in practice. To enhance this smoothness and improve the efficiency of HRC, there is potential in refining the existing distance-based methods by integrating two new functions—the first is to enable the robot to perceive the motion intentions of the worker, and the second is to let the robot estimate the collision probability with the worker.

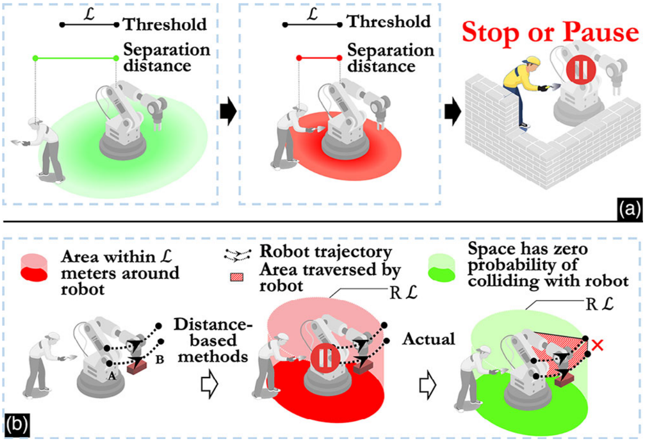


FIGURE 1 (a) Distance-based human–robot collaborative mode regulated by ISO; (b) example of distance-based collaborative modes making collaborative robots too conservative in regulating their performance.

The first function seeks to empower robots with the capability to anticipate the upcoming movements of workers. With current methods, when the robot detects the distance from the worker lower than the safety threshold, the absence of information about the worker's upcoming actions will prevent the robot from adjusting its work in advance. Then, the robot is constrained to a choice of pausing or halting the collaboration. However, the new function of motion intention perception can enable the robot to anticipate the worker's subsequent actions based on their motion intentions, allowing it to adjust its movement in advance and maintain the safe threshold without pausing its operation.

The second function, collision probability estimation, equips robots with a precise and quantitative tool to assess potential collision risks with workers. Current distance-based methods fail to make robots have quantitative and precise estimates of collision risks, resulting in the robot being overly conservative in regulating (halting or pausing) its actions to avoid collisions (Vicentini et al., 2014). Illustrated in Figure 1b, consider a scenario wherein a robot is assigned the task of delivering concrete bricks from location A to B during HRC. In this setup, if the safety threshold of the separation distance is set to $\mathscr{L}$, the robot would promptly stop working when a worker enters the area within $\mathscr{L}$ around the robot. However, it is important to recognize that a significant portion of this area holds zero probability of collision with the worker (i.e., space not traversed by the robot trajectory; green area in Figure 1b). In contrast, with the advanced collision probability estimation function, the robot can accurately discern the area free from collisions, enabling it to more precisely adjust its behavior for safer interactions in construction.

Taken together, relying on these two new functions, the robot can anticipate the worker's subsequent actions based on their motion intentions (Wang et al., 2013). Simultaneously, the robot is able to more quantitatively and accurately estimate the space where it may collide with the worker's next move. Consequently, the robot can proactively devise alternative motion plans that ensure zero collision probability with the worker's actions. As the worker moves to the anticipated next position, the robot will continue its work with the replanned motions without causing collisions or pausing its operation. Therefore, the incorporation of human motion intention perception and collision probability calculation through the enhanced method empowers the collaborative robot to execute motion adjustments with a range of technical advantages, including collision prevention, efficient space utilization, and a significant enhancement in the smoothness of collaboration with workers. All these technical merits can facilitate the deployment of collaborative robots in cluttered and dynamic work environments, particularly in civil and infrastructure engineering. Unfortunately, no research in the field has developed a comparable method for collaborative robots.

To this end, this study introduces a novel method, denoted as the intention-aware probabilistic motion planner (IAPMP). This method enables robots to perceive workers' motion intentions, measure the collision probability with workers, and accordingly generate seamless motion adjustments with the desired features of collision-free and space-saving. In it, the authors develop two novel deep neural networks (DNNs), 3D-MotNet and IntentionNet, to enable the robot to extract and estimate the 3D human poses and consequently predict the motion intentions (next move) of workers based on workers' 3D poses. Additionally, a probabilistic collision-checking mechanism is developed to allow the robot to quantitatively estimate the collision probability with the future position of workers. By integrating the proposed DNNs and the collision-checking mechanism, the IAPMP enables the robot to generate adjustments with the desired features to avoid collisions with workers. The proposed IAPMP is expected to enhance the safety and efficiency of human–robot collaborative tasks in construction. Furthermore, this research is expected to facilitate the widespread implementation of collaborative robots in construction and civil engineering.

2 | RELATED WORK

2.1 | State-of-the-art techniques for developing IAPMP

To develop the intended IAPMP, two fundamental questions need to be answered: First, how to enable a robot to infer the motion intention of its human collaborators; second, how to allow a robot, based on the inferred motion intentions (intended motions of workers), to estimate the collision probability with its human collaborators during the collaboration.

To answer the first question, one can draw insights from research conducted in industries such as manufacturing. Existing literature reveals efforts in the manufacturing domain to predict human motion intentions, ensuring safer human–robot interaction. Moreover, one of the core objectives of these studies mirrors the goal to further enhance the efficiency of the conventional distance-based collaborative mode (Lasota & Shah, 2015; Wang et al., 2017; Zheng et al., 2022). The existing studies focused on integrating sensing technology with artificial intelligence (AI) to infer motion intentions from human poses and motion dynamics (Casas et al., 2021; Zhao et al., 2019). These studies employ sensing technologies—motion capture systems and depth cameras—to capture three-dimensional (3D) human poses as they move in collaborative space. Subsequently, motion features are extracted from the captured human poses, based on human kinematics. AI models are then applied to predict the motion intentions from the extracted features. For instance, Hu et al. (2016) employed the RGB-D camera-based Open Natural Interaction (OpenNI) system to track the human skeleton positions. Then, they applied the max-margin learning framework and a hidden Markov model classifier to extract final-level features (bending angle of the torso, rotation of body joints relative to the head, and human kinematic constraints) from the dynamic changes in human skeleton positions and accordingly recognize human intentions. Unhelkar et al. (2018) employed the RGB-D camera-based OpenNI system with the multiple-predictor system (MPS) to predict human motions based on motion features like motion velocity and changes of postures. Similarly, Luo and Mai (2019) employed the depth camera to capture the motion data and employed the Probabilistic Dynamic Movement Primitive (PDMP) model to predict human motions (hand motions) from the motion trajectory features. Kratzer et al. (2020), on the other hand, applied an Optitrack motion capture system to detect human postures. They utilized the Position-Velocity Recurrent Encoder-Decoder (PVRED) model to predict human motion intentions according to human motion features like body joint motion velocity and acceleration. In a similar vein, Luo et al (2018) employed the Vicon motion capture system to capture human arm movements. They then introduced a two-layer Gaussian mixture model (GMM) approach to predict human motion in the context of human–robot collaborative tasks based on features like palm position and arm joint center positions.

Even though the studies mentioned above show significant potential in predicting human motion intentions, they have several limitations when using them to develop the perspective IAPMP for collaborative robots in construction and civil engineering. Specifically, from a practical point of view, the sensing technologies (e.g., motion capture systems and depth cameras) used to capture 3D human poses for motion intention inference are difficult to be widely implemented at actual construction sites (Fang et al., 2020). Motion capture systems need to be intrusively attached to workers, potentially interfering with workers' tasks (Yan et al., 2017). Furthermore, depth cameras are limited in working range and are not well suited for outdoor environments (Liu et al., 2016; Tadic et al., 2022). For instance, traditional depth cameras like the Microsoft Kinect sensor, primarily designed for home/indoor use, experience a significant reduction in capturing visual and depth information when used outdoors, due to infrared ray interference from sunlight (Liu et al., 2016; Tadic et al., 2022). While recent advancements in depth cameras, such as ZED 2i and D435 models, have made strides in outdoor applications including object detection and robotic navigation (Tadic et al., 2022), they still face several challenges. These include a limited effective range of about 1 m and reduced image quality in varying light conditions (Tadic et al., 2022; Vit & Shani, 2018). Consequently, the motion intention inference methods based on these sensing technologies encounter difficulties in being applied to practical HRC tasks in construction and civil engineering.

On the computational front, most methods (as mentioned above) for predicting human motion intention rely heavily on the manual feature engineering process. Typically, in the domains of machine learning and AI, manual feature engineering is a phase wherein domain experts utilize their expertise to extract and select informative features from raw input data. These features are then employed to train the model with an enhanced predictive performance (Adamczyk & Malawski, 2021; Xu et al., 2023). To train the models for efficient motion intention, this manual process requires researchers to use domain-specific knowledge in human kinematics and human dynamics to extract features such as the rotation and displacement of individual body joints as well as the kinematic features of human motions. These features are then used to train predictive AI models for human intentions. However, this process is time consuming and challenging for researchers to extract features from the high-dimensional feature space (Hu et al., 2016; Park et al., 2019; Wang et al., 2013). This process will become even more difficult for researchers in the fields of construction and civil engineering, who may not possess in-depth proficiency in human kinematics and dynamics. As a result, AI models trained through the manual feature engineering process may fall short in efficiently allowing robots to predict human motion intentions. Notably, in light of the challenges presented by manual feature engineering, several deep learning methods have been pioneered for automatic feature extraction and subsequent motion intention prediction. Martinez, Black, et al. (2017) integrated the sequence-to-sequence (seq2seq) architecture into a deep recurrent neural network (RNN) for direct, unsupervised feature extraction. Similarly, Xu et al. (2021) developed a Bayesian neural network (BNN) based on the RNN structure for automated feature extraction and predictive modeling. Deep learning models developed based on the convolutional network (CN) and generative adversarial network (GAN) structures have also demonstrated their capabilities for automatic feature extraction (Mao et al., 2020; Martinez, Black, et al., 2017; Xu et al., 2021). However, these deep learning models remain dependent on data sourced from the motion capture systems and the depth cameras. In sum, while motion intention estimation techniques have reached an advanced stage, there is a clear need to develop a new motion intention estimation function for IAPMP, specifically addressing the identified challenges within the civil and construction industry.

The second question—calculating the collision probability between the collaborative robot and its human collaborator—remains an open-ended question (Park & Manocha, 2020). Fortunately, in a broader context, solutions for computing the collision probability between objects have been proposed in other domains. In the field of aerospace engineering, for instance, studies aimed to estimate collision probabilities between satellites and space debris by employing a heuristic process. Geometric representations of the corresponding objects were modeled, using a spherical function to represent the shape of the debris and a cylindrical function to represent the shape of the spacecraft (Patera, 2001). According to the mathematical expression of each object, the corresponding primitive was derived. Next, these primitive functions were employed to convert the calculation of collision probability between two objects into the computation of potential overlapping space between the objects through a domain-specific collision probability density function (PDF), so as to obtain the collision probability (Chan, 2008; Patera, 2001). Likewise, in the fields of autonomous driving (Du Toit & Burdick, 2011; Haghighat & Sharma, 2023; Yoo & Langari, 2019), researchers leveraged geometric representations—such as rectangular (e.g., axis-aligned bounding box), prism, polyhedron, and sphere—to model the shape of obstacles and vehicles. Then, by using the above procedure for collision probability calculation, vehicles were able to generate collision-free paths to avoid collisions with obstacles. Although the studies in the fields of aerospace engineering (Patera, 2001) and autonomous driving (Lu & Dai, 2023; Yoo & Langari, 2019) provided an insightful procedure for quantitatively measuring the collision probability between two objects, a major limitation arises when applying this procedure to collaborative robots and workers. Unlike debris, obstacles, and vehicles—whose shapes are often assumed to be rigid and approximated by simple geometric representation—worker poses and the configurations of collaborative robots (e.g., robotic arm) are subject to constant change during collaboration, and thus cannot be efficiently approximated by an exact, simple geometric representation. Hence, the current procedure cannot be directly applied to accurately measure the collision probability between workers and robots. All in all, a new function that can evaluate the collision probability between robots and workers should be developed for IAPMP to ensure safe and efficient HRC in civil and construction engineering.

2.2 | Contributions

Addressing the limitations discussed in Section 2.1, this study proposes a novel methodology, IAPMP. This approach empowers the robot to generate collision-free and space-saving adjustments, as per the perception of human motion intention and the calculation of the collision probability with workers. The novelty and significance of the proposed IAPMP can be summarized as follows:

First, to achieve the motion intention function in IAPMP, the authors design two deep learning networks, 3D-MotNet and IntentionNet. 3D-MotNet can enable the robot to extract 3D human motion postures from 2D images captured by a single RGB camera—a practical tool widely applied on job sites. Building upon this, IntentionNet works synergistically with 3D-MotNet to enable the robot to predict the motion intentions of workers based on estimated workers' 3D poses. In real-world settings, this duo empowers robots to predict motion intentions using solely a 2D camera input. Computationally, both networks are designed to automatically extract features, eliminating the need for the manual feature engineering process.

Second, to achieve the collision probability estimation function in IAPMP, the study develops a novel probabilistic collision-checking mechanism. This mechanism enables the robot to quantitatively estimate the collision probability with workers and accordingly generate collision-free adjustments, thereby avoiding collision with workers' intended motion positions.

Third, two new data sets—one for training a DNN to estimate human poses during civil and construction tasks, as well as another for training the network to infer human motion intentions while performing civil and construction tasks—were developed. These two data sets can contribute to training and validating models and algorithms in future related studies (pose estimation, motion intention inference), especially in the fields of civil and infrastructure engineering.

In addition, the IAPMP is the first paradigm in construction and civil engineering to enable the collaborative robot to infer the motion intentions of workers, calculate the collision probability, and proactively generate collision-free adjustments during HRC. Details of the IAPMP will be introduced in the next section.

3 | METHODOLOGY

The structure of the IAPMP is shown in Figure 2, which is factorized into three modules: (1) a 2D-to-3D pose/motion estimation deep network, named 3D-MotNet; (2) a motion intention inference deep network, IntentionNet; and (3) a collision-checking robotic motion planning mechanism. When robots collaborated with workers, the first module, 3D-MotNet, was developed to equip robots with the ability to evaluate the poses of workers in 3D space through image frames captured by the onsite 2D camera. To achieve this, a multistage convolutional neural network (CNN) was initially designed based on the authors' previous work (Liu & Jebelli, 2022) to extract 2D body skeletons of workers from 2D images. Next, a 3D pose transformation network was constructed to convert the extracted 2D pose representations into 3D human poses (Figure 2.1). The second module, IntentionNet, was constructed to make robots understand workers' motion intentions based on the worker poses estimated by Module 1. In this study, the worker's "motion intention" was defined as a set of intended motion sequences represented by poses (Park et al., 2019; Wang et al., 2013). To enable robots to infer the motion intention of workers, the IntentionNet leveraged the variational autoencoder (VAE) model to predict the worker's intended motion goal based on the estimated worker poses. Then, along the trajectory from the worker's

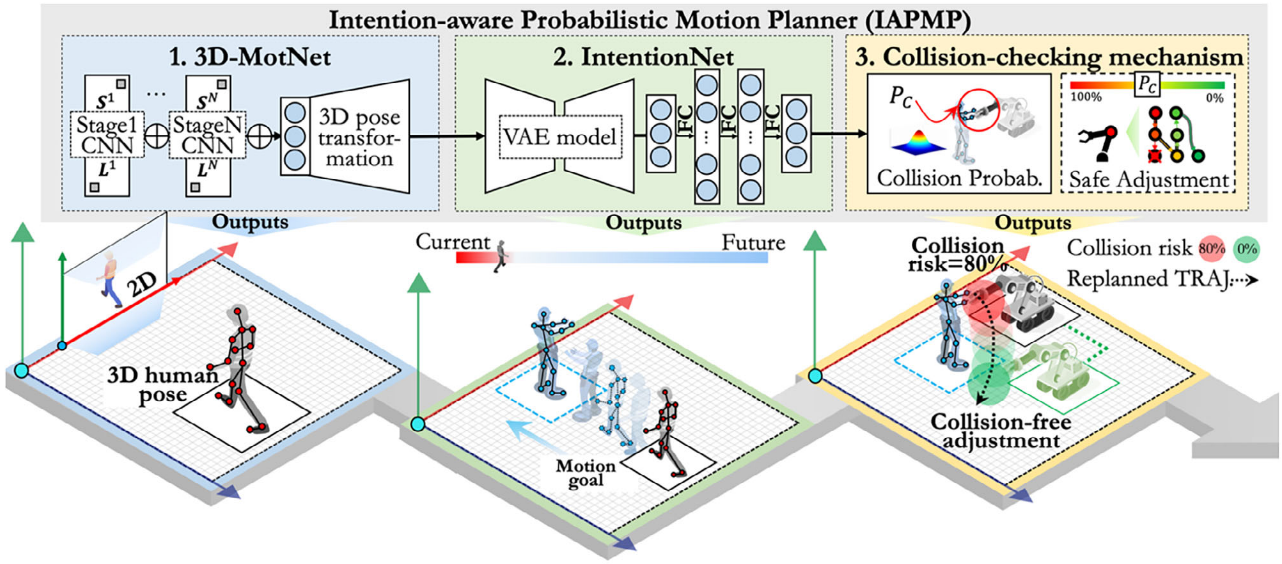


FIGURE 2 Overview of the intention-aware probabilistic motion planner (IAPMP).

current location to the predicted motion goal, IntentionNet generated the sequence of 3D human poses using the fully connected (FC) neural network structure (Figure 2.2). Finally, according to the inferred motion intentions, the third module empowered robots to generate collision-free trajectories for the effective completion of construction tasks. In this module, a probabilistic collision-checking mechanism was formulated to enable the collaborative robot to estimate the collision probability between its planned trajectory and the anticipated motion sequences of the worker. As per the estimated collision probability, the robot could adjust its trajectory to ensure zero risk of collision with the worker (Figure 2.3), thereby ensuring physical safety during HRC. Details of each module will be presented in the subsequent sections.

3.1 | 3D-MotNet: Pose/motion estimation deep network

Figure 3a illustrates the architecture of the 3D-MotNet utilized for worker pose estimation. This proposed network used 2D images ($\boldsymbol{I}_{\mathrm{2D}}$) as input and finally generated 3D human poses connected by 3D anatomical key points of the human body as output. Notably, to ensure a uniform input, all 2D images underwent preprocessing: The aspect ratio scaling approach was employed to resize images to a consistent dimension of $w \times h \times 3$ (with $w = 640$ pixels and $h = 480$ pixels used in this study); in addition, the pixel values of each image were normalized to fall within the range 0 to 1. Following these preprocessing steps, the images were input to the 3D-MotNet network. 3D-MotNet consisted of several components. Initially, an H-stage convolutional neural network (H-stage CNN; subnetwork 1 in Figure 3a) was employed to extract 2D human skeletons from the input images. The H-stage CNN represents a specific architecture of the CN structured to process input images in a hierarchical manner through "H" sequential stages (Cao et al., 2017). The first stage of the H-stage CNN was designed to extract features from the input images, providing an initial estimation of the human posture. Then, each succeeding stage refined the features generated from the previous stage while also extracting new features from inputs, thereby progressively enhancing the pose estimation precision of the network.

In this study, each stage of the H-stage CNN was trained to generate two sets of outputs: (1) confidence maps $\boldsymbol{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_{\mathrm{q}}\}$ for detecting q anatomical key points of the worker, where $\boldsymbol{S}_{\mathrm{q}}$ indicates the confidence map associated with the qth key point; (2) 2D vectors $\boldsymbol{L} = \{\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_{\mathrm{p}}\}$ for associating the body joints of workers, where $\boldsymbol{L}_{\mathrm{p}}$ represents the pth vector tasked with linking the relevant body joints. For all H stages, the outputs ($\boldsymbol{S}$ and $\boldsymbol{L}$) of each stage were used as the inputs of its next stage; this iterative structure could enhance the performances of the body joint detection ($\boldsymbol{S}$) and body joint association ($\boldsymbol{L}$) for each successive stage (Cao et al., 2017). Upon completing the training process, subnetwork 1, for each input image, generated a set of optimized confidence maps $\boldsymbol{S}_{\mathrm{opt}}$ to detect the body joints of workers and a set of optimized 2D vectors $\boldsymbol{L}_{\mathrm{opt}}$ to associate the detected joints. Relying on these optimized sets, subnetwork 1 associated detected joints with the corresponding 2D vectors to generate the 2D skeleton representation of workers, $\boldsymbol{pose}_{\mathrm{2D}}$. The applied H-stage CNN was adopted from the authors' prior work (Liu & Jebelli, 2022), retaining the same architecture of each stage

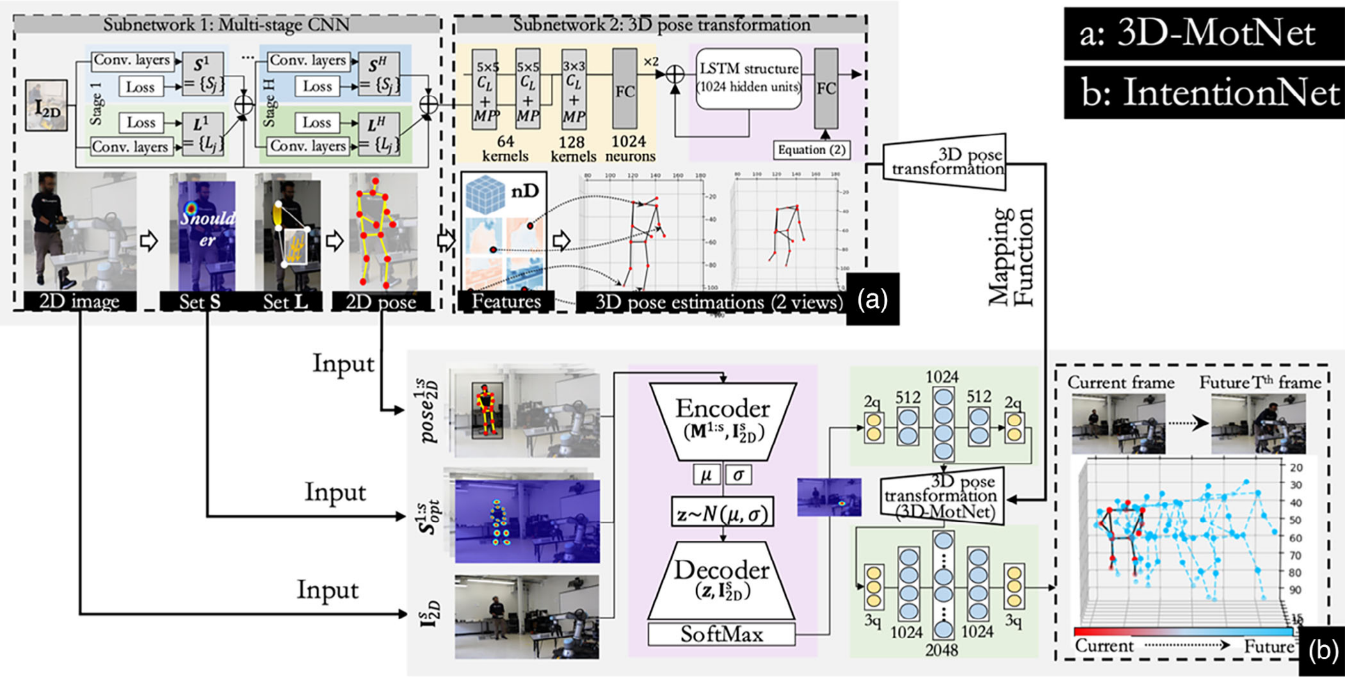


FIGURE 3 (a) Architecture of the 3D-MotNet in the proposed intention-aware probabilistic motion planner (IAPMP); (b) architecture of the IntentionNet in the proposed IAPMP.

and the same settings of the training. In addition, the choice of the "H" value for the H-stage CNN was influenced by the training data set. Detailed information about this data set will be provided in Section 4, while the optimal H value will be reported in Section 5.

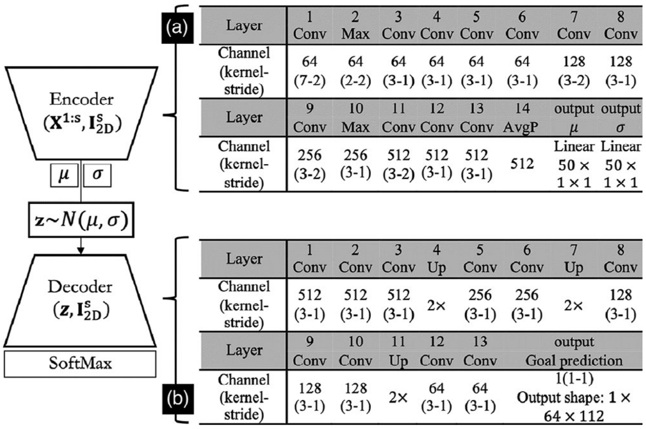
Next, using the extracted 2D worker skeletons ($\boldsymbol{pose}_{2\text{D}}$) along with the optimized set of confidence maps $\boldsymbol{S}_{\text{opt}}$ as the inputs, subnetwork 2 of the 3D-MotNet was developed to estimate the worker's pose in 3D space. As shown in Figure 3a, subnetwork 2 of the 3D-MotNet encompassed three convolutional layers (C_L). Each of the first two layers featured 64 kernels of size 5 × 5, and the third layer contained 128 kernels of size 3 × 3. Furthermore, each convolutional layer was followed by a max-pooling layer (MP) with a kernel size of 2 × 2. Following these three pairs of convolution–max-pooling layers, two FC layers with 1024 neurons were constructed. By leveraging this architecture, the subnetwork decoded the $\boldsymbol{pose}_{2\text{D}}$ and $\boldsymbol{S}_{\text{opt}}$ extracted on each image frame (frame s) into a high-dimensional (1024-dimensional) pose-aware feature vector, denoted as $\boldsymbol{f}_s$. As per this decoded $\boldsymbol{f}_s$, a long–short term memory (LSTM)-based structure was developed to reconstruct the 3D human pose. LSTM is an RNN architecture capable of processing the temporal sequences of data (Yu et al., 2019). In this study, the employment of the LSTM empowered subnetwork 2 to capture the temporal correlation between pose-aware feature $\boldsymbol{f}_s$ decoded from the image frame s and such features decoded from preceding $(s-1)$ image frames, ensuring that subnetwork 2 could maintain the geometric consistency of poses when projecting high-dimensional pose-aware features to 3D human poses (Wang et al., 2019). Specifically, for the pose-aware feature $\boldsymbol{f}_s$ extracted from frame s, the LSTM structure, equipped with 1024 hidden units, generated its hidden state vector $\boldsymbol{H}_{s-1}$. This vector contained the hidden states of pose features learned from the previous $(s-1)$ image frames— $\boldsymbol{H}_{s-1} = \text{L}STM(\boldsymbol{f}_{s-1}, \ldots, \boldsymbol{f}_1)$. Then, $\boldsymbol{H}_{s-1}$ was concatenated with $\boldsymbol{f}_s$ as the input for the feedback connection of the LSTM, allowing it to learn the temporal dependency between the feature vector $\boldsymbol{f}_s$ and the hidden state vector $\boldsymbol{H}_{s-1}$. Leveraging the LSTM outputs, a 3q–neuron FC output layer was followed to predict the 3D locations of q body joints, resulting in a 3D human pose estimation, $\boldsymbol{pose}_{3\text{D}}^s$. $\boldsymbol{pose}_{3\text{D}}^s$ can be expressed by Equation (1):

$$\boldsymbol{pose}_{3\text{D}}^s = \Upsilon\left(\boldsymbol{f}_s, \boldsymbol{W}_\Upsilon, \boldsymbol{H}_{s-1}\right) \tag{1}$$

where ϒ indicates the architecture of the subnetwork 2 shown in the purple area in Figure 3a, and $\boldsymbol{W}_\Upsilon$ indicates the weight matrix of ϒ. To enable subnetwork 2 to generate the 3D human pose for each image frame s, the training objective function for subnetwork 2 was formulated in Equation (2):

$$\min \sum_{s=1}^{N} \left\| \Upsilon\left(\boldsymbol{f}_s, \boldsymbol{W}_\Upsilon, \boldsymbol{H}_{s-1}\right) - \boldsymbol{pose}_{3\text{D}}^{s*} \right\|_2^2 \tag{2}$$

$\boldsymbol{pose}_{3\mathrm{D}}^{s*}$ represents the ground-truth 3D coordinates of q body joints corresponding to frame s. The parameter N indicates the total number of images used for training the 3D-MotNet. $\boldsymbol{pose}_{3\mathrm{D}}^{s*}$ and N will be explained in Section 4. Notably, this 3D-MotNet offered an end-to-end solution for pose estimation, transforming 2D image data directly into 3D pose estimation. As introduced, during the training phase, the H-stage CNN component within the 3D-MotNet produced the 2D pose estimations, $\boldsymbol{pose}_{2\mathrm{D}}$. The accuracy of these estimations was assessed by computing the Euclidean distance between the intermediate ground-truth data ($\boldsymbol{pose}_{2\mathrm{D}}^{s*}$) and the generated pose estimations, $\boldsymbol{pose}_{2\mathrm{D}}$. Details on $\boldsymbol{pose}_{2\mathrm{D}}^{s*}$ will also be provided in Section 4.


(a)
Encoder
$(\mathbf{X}^{1:s}, \mathbf{I}_{2\mathrm{D}}^{s})$
μ
σ
$z \sim N(\mu, \sigma)$
Decoder
$(\boldsymbol{z}, \mathbf{I}_{2\mathrm{D}}^{s})$
SoftMax
(b)

Layer	1 Conv	2 Max	3 Conv	4 Conv	5 Conv	6 Conv	7 Conv	8 Conv
Channel (kernel-stride)	64 (7-2)	64 (2-2)	64 (3-1)	64 (3-1)	64 (3-1)	64 (3-1)	128 (3-2)	128 (3-1)
Layer	9 Conv	10 Max	11 Conv	12 Conv	13 Conv	14 AvgP	output μ	output σ
Channel (kernel-stride)	256 (3-2)	256 (3-1)	512 (3-2)	512 (3-1)	512 (3-1)	512	Linear 50 × 1 × 1	Linear 50 × 1 × 1

Layer	1 Conv	2 Conv	3 Conv	4 Up	5 Conv	6 Conv	7 Up	8 Conv
Channel (kernel-stride)	512 (3-1)	512 (3-1)	512 (3-1)	2×	256 (3-1)	256 (3-1)	2×	128 (3-1)
Layer	9 Conv	10 Conv	11 Up	12 Conv	13 Conv	output Goal prediction		
Channel (kernel-stride)	128 (3-1)	128 (3-1)	2×	64 (3-1)	64 (3-1)	1(1-1) Output shape: 1 × 64 × 112		

FIGURE 4 (a) Architecture of the encoder in the variational autoencoder (VAE) model; (b) architecture of the decoder in the VAE model.

Note: In the figure, Conv denotes the convolutional layer, Max represents the max-pooling layer; AvgP stands for the global average pooling layer; $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the outputs of the encoder, their dimensions depend on the dimension of latent space (set as 50 × 1 × 1 in this study); Up stands for the upsampling layer; and Goal prediction is the final output of the VAE model.

3.2 | IntentionNet: Motion intention inference deep network

Module 2 was employed to enable collaborative robots to anticipate workers' motion intentions, utilizing the outputs of Module 1. As mentioned, the term "motion intention" refers to the set of intended motion sequences represented by worker poses (Park et al., 2019). To this end, the authors designed an intention inference deep network. This network was structured for seamless end-to-end prediction, anticipating the 3D pose of a worker in consecutive T image frames in the future. This prediction is based on the 2D skeletons and 3D poses of the worker extracted from both the current and past frames. Figure 3b shows the architecture of the proposed IntentionNet.

For each image frame s ($s \in \{1, \ldots, N\}$), the first part of IntentionNet aimed to predict the intended motion goal of the worker for the subsequent T image frames. This design concept was inspired based on the assumption that human motion generally follows a goal-oriented policy (Wang et al., 2013). As depicted in the purple area of Figure 3b, the authors leveraged the 2D image frame ($\boldsymbol{I}_{2\mathrm{D}}^{s}$), 2D worker skeletons extracted from all previous s frames ($\boldsymbol{pose}_{2\mathrm{D}}^{1:s}$), and the corresponding confidence maps of q body joints $\boldsymbol{S}_{\mathrm{opt}}^{1:s} = \{\boldsymbol{S}_{1}^{1:s}, \boldsymbol{S}_{2}^{1:s}, \ldots, \boldsymbol{S}_{\mathrm{q}}^{1:s}\}$ extracted from previous s frames as the inputs of the IntentionNet. Upon the inputs, the authors applied a VAE model (Dittadi et al., 2021) to predict the worker's motion destination in the Tth image frame in the future. According to the literature, VAE has been shown to have efficient performance in solving stochastic inference problems, particularly in human motion prediction (Cao et al., 2020; Dittadi et al., 2021). VAE shares a similar architecture to traditional autoencoders (Jang et al., 2021; Liu et al., 2022), where the encoder structure extracts the features from the inputs regarding worker motion sequences, and the decoder reconstructs these features to estimate the motion goal of the worker. The encoder in the applied VAE model processed the inputs ($\boldsymbol{I}_{2\mathrm{D}}^{s}$; $\boldsymbol{pose}_{2\mathrm{D}}^{1:s}$; $\boldsymbol{S}_{\mathrm{opt}}^{1:s} = \{\boldsymbol{S}_{1}^{1:s}, \boldsymbol{S}_{2}^{1:s}, \ldots, \boldsymbol{S}_{\mathrm{q}}^{1:s}\}$) to generate a feature represented as $\boldsymbol{z}$. For the inputs ($\boldsymbol{pose}_{2\mathrm{D}}^{1:s}$; $\boldsymbol{S}_{\mathrm{opt}}^{1:s} = \{\boldsymbol{S}_{1}^{1:s}, \boldsymbol{S}_{2}^{1:s}, \ldots, \boldsymbol{S}_{\mathrm{q}}^{1:s}\}$), the VAE model processed them sequentially, integrating the reference input $\boldsymbol{I}_{2\mathrm{D}}^{s}$ at each frame. Such frame-by-frame processing ensured temporal continuity, essential for the VAE model to consistently capture and analyze human motion patterns over time. This analysis would result in the accurate estimation of the feature $\boldsymbol{z}$. $\boldsymbol{z}$ was then decoded to pinpoint the motion destination of the worker in the $(s+T)$th frame accurately. The architecture of the encoder is shown in Figure 4a. Feature $\boldsymbol{z}$ was expressed in Equation (3):

$$\boldsymbol{z} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma}), \text{ where } \boldsymbol{\mu}, \boldsymbol{\sigma} = Encoder(\boldsymbol{M}^{1:s}, \boldsymbol{I}_{2\mathrm{D}}^{s})$$

$$\boldsymbol{M}^{1:s} = \mathrm{f}\left(\boldsymbol{pose}_{2\mathrm{D}}^{1:s}; \boldsymbol{S}_{\mathrm{opt}}^{1:s}\right) \tag{3}$$

N(·) represents the Gaussian distribution, which approximates the value of $\boldsymbol{z}$ by applying the reparameterization trick in VAE (Kingma et al., 2015). The mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ of N(·) were the outputs of the encoder. $\boldsymbol{M}^{1:s}$ refers to the stacked matrix of $\boldsymbol{pose}_{2\mathrm{D}}^{1:s}$ and $\boldsymbol{S}_{\mathrm{opt}}^{1:s}$, and f(·) indicates the stacked operation.

Following the encoder's outputs, the decoder in the VAE model was constructed to predict the anticipated motion destination ($\boldsymbol{G}^{s+T}$) of workers' motion in the $(s+T)$th image frame based on the feature z and the input sth image $\boldsymbol{I}_{2\mathrm{D}}^{s}$. The structure of the decoder of VAE was detailed in Figure 4b. Moreover, as expressed in Equation (4), the generation of $\boldsymbol{G}^{s+T}$ involved the reconstruction of the inputs into a confidence map representing the motion destination in the $(s+T)$th frame—Decoder($\boldsymbol{z}$, $\boldsymbol{I}_{2\mathrm{D}}^{s}$). Then, the SoftMax function (Alam et al., 2020) was applied to Decoder($\boldsymbol{z}$, $\boldsymbol{I}_{2\mathrm{D}}^{s}$) to estimate the destination $\boldsymbol{G}^{s+T}$.

$$\boldsymbol{G}^{s+T} = softmax\left(\mathrm{Decoder}\left(\boldsymbol{z}, \boldsymbol{I}_{2\mathrm{D}}^{s}\right)\right) \tag{4}$$

To efficiently predict the destination $\boldsymbol{G}^{s+T}$, the VAE model was trained using the loss function formulated in Equation (5):

$$\mathrm{L}_{\mathrm{des}} = min(\|\boldsymbol{G}^{s+T} - \boldsymbol{G}^{*(s+T)}\| \\ + \mathrm{D}_{\mathrm{KL}}[\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \,||\, \mathrm{N}(0,1)]) \tag{5}$$

where $\boldsymbol{G}^{*(s+T)}$ is the actual destination of workers in the $(s+T)$th frame, which is defined by the 2D location of the worker's center body joint (Cao et al., 2020). $\mathrm{D}_{\mathrm{KL}}(\cdot)$ measures the Kullback–Leibler divergence between the distribution of the latent vector $\boldsymbol{z} \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ and the normal distribution. The theoretical basis of this function can be found in Kingma et al. (2015). For the image frame s where the worker was located, after predicting the motion destination of the worker in the $(s+T)$th image frame, the second part of IntentionNet (green area of Figure 3b) was proposed to predict the corresponding human poses from the current position to $\boldsymbol{G}^{s+T}$. For image frame s, the authors first connected the worker's current position ($\boldsymbol{G}^{s}$) to the future destination $\boldsymbol{G}^{s+T}$. Notably, the worker's current position $\boldsymbol{G}^{s}$ was identified as the 2D location of the worker's center body joint (middle of the left and right hip joints; obtained from the 2D skeleton of workers generated in Module 1; Cao et al., 2020). Along the path between $\boldsymbol{G}^{s}$ and $\boldsymbol{G}^{s+T}$, the authors implemented a skeleton translation, shifting the worker's 2D skeleton centered at $\boldsymbol{G}^{s}$ to $\boldsymbol{G}^{s+T}$. The translated skeleton centered at the G^{s+T} served as the initial prediction of the worker's poses in the $(s+T)$th frame. Correspondingly, the initial prediction $\boldsymbol{pose}_{2\mathrm{D}}^{s+T}$ was fed into the trained subnetwork 2 of the 3D-MotNet to generate the initial 3D human pose prediction $\boldsymbol{pose}_{3\mathrm{D}}^{s+T}$.

To fine-tune the initial pose predictions, they were fed into an FC neural network consisting of two sequentially connected parts, as illustrated in the green area of Figure 3b. The upper part of the FC network was composed of five layers with $2q$ $(2 \times q)$, 512, 1024, 512, and $2q$ $(2 \times q)$ neurons, which was used to calibrate the initial pose prediction in 2D space, $\boldsymbol{pose}_{2\mathrm{D}}^{s+T}$. For clarity, the parameter q denotes the total number of identified body joints in $\boldsymbol{pose}_{2\mathrm{D}}^{s+T}$. Multiply by 2 to represent the two coordinate values (X and Y) associated with each joint. To enhance the accuracy of pose predictions with q body joints, the FC neural network was employed an architecture of $2 \times q$ neurons at both its input and output layers. This design enabled the network to capture and fine-tune the spatial position of each joint in a 2D space, generating accurate 2D pose predictions. Simultaneously, the lower part of the FC network consisted of five layers with $3q$, 1024, 2048,1024, and $3q$ neurons was trained to correct the initial human pose prediction in 3D space ($\boldsymbol{pose}_{3\mathrm{D}}^{s+T}$) converted from $\boldsymbol{pose}_{2\mathrm{D}}^{s+T}$. In addition, the activation function of each layer was Rectifier (ReLU). The loss functions of each part of the developed FC neural network were formulated in Equations (6) and (7). Specifically, Equation (6) determined the loss associated with 2D pose predictions for future T frames, while Equation (7) calculated the loss for 3D pose predictions for those same frames. During each iteration of training, these two functions operated in a hierarchical manner to refine pose predictions. Equation (6) initially fine-tuned the 2D predictions of the upper part of the employed two-part FC network. These refined 2D pose predictions then served as a superior starting point for the lower part of the network, offering enhanced initial 3D pose predictions. Following this, Equation (7) corrected these 3D pose predictions to yield accurate 3D pose predictions for the next T frames (motion intention). In addition, the overall loss function of the two-part FC neural network was the summation of these two equations.

$$\mathrm{L}_{\mathrm{upper}} = min \sum_{T=1}^{N} \sum_{s=1}^{N-T} \left\| \boldsymbol{pose}_{2\mathrm{D}}^{s+T} - \boldsymbol{pose}_{2\mathrm{D}}^{*(s+T)} \right\|_2^2 \tag{6}$$

$$\mathrm{L}_{\mathrm{lower}} = min \sum_{T=1}^{N} \sum_{s=1}^{N-T} \left\| \boldsymbol{pose}_{3\mathrm{D}}^{s+T} - \boldsymbol{pose}_{3\mathrm{D}}^{*(s+T)} \right\|_2^2 \tag{7}$$

where $\boldsymbol{pose}_{2\mathrm{D}}^{*(s+T)}$ and $\boldsymbol{pose}_{3\mathrm{D}}^{*(s+T)}$ indicate the actual poses of workers in 2D and 3D space, respectively. N indicates the total count of human poses available to train the network. The double summation regarding parameter T means that the network can be trained to predict the 3D human pose in future T frames. In this study, the optimal T was 2, which was determined based on the performance (overall loss) of the IntentionNet. The applied $\boldsymbol{pose}_{2\mathrm{D}}^{*(s+T)}$ and $\boldsymbol{pose}_{3\mathrm{D}}^{*(s+T)}$ were sourced from the same data set as the ground-truth employed to assess the performance of the 3D-MotNet model. This ensures that if the outputs from 3D-MotNet introduced errors into the inputs of IntentionNet, the latter would utilize these consistent benchmarks to correct these errors. As such, errors originating from 3D-MotNet were not accumulated unchecked in the outputs of IntentionNet. $\boldsymbol{pose}_{2\mathrm{D}}^{*(s+T)}$, $\boldsymbol{pose}_{3\mathrm{D}}^{*(s+T)}$, and N will be explained in Section 4.

3.3 | Collision-checking robotic motion planning mechanism

Relying on predicted 3D poses of workers in future T frames (motion intentions), this section first introduced the proposed probabilistic collision-checking mechanism, by which the robot could estimate the collision probability with workers according to their 3D poses in future T frames. The authors then illustrated the paired motion planning mechanism that allowed the robot to generate collision-free trajectories.

3.3.1 | Probabilistic collision-checking mechanism between workers and robots

To quantify the collision probability between workers and robots, the authors defined the collision condition between these two entities. Let $\mathbb{S}$ denote the 3D workspace where workers and robots collaborate. All objects in $\mathbb{S}$ use the global coordinate system {G}. $\boldsymbol{X}_{\mathrm{R}}^{s} \in \mathbb{S}$ represents space occupied by the robot measured in $\mathbb{S}$ at frame s, and $\boldsymbol{x}_r$ indicates the random variable in $\boldsymbol{X}_{\mathrm{R}}^{s}$. In other words, $\boldsymbol{x}_r \in \boldsymbol{X}_{\mathrm{R}}^{s}$ indicates the location of each point that the robot occupies in $\mathbb{S}$ at frame s. Similarly, $\boldsymbol{X}_{\mathrm{H}}^{s} \in \mathbb{S}$ is the space occupied by workers at frame s, with $\boldsymbol{x}_h \in \boldsymbol{X}_{\mathrm{H}}^{s}$ indicating the location of each point that workers occupy in $\mathbb{S}$ at frame s. Using the definitions of $\boldsymbol{x}_h$, $\boldsymbol{X}_{\mathrm{H}}^{s}$, $\boldsymbol{x}_r$, and $\boldsymbol{X}_{\mathrm{R}}^{s}$, the collision between workers and robots occurs at frame s if and only if there is a spatial overlap between data points within $\boldsymbol{X}_{\mathrm{R}}^{s}$ and $\boldsymbol{X}_{\mathrm{H}}^{s}$, a condition that can be formulated as $\boldsymbol{X}_{\mathrm{R}}^{s}(\boldsymbol{x}_r) \cap \boldsymbol{X}_{\mathrm{H}}^{s}(\boldsymbol{x}_h) \neq \{\emptyset\}$. Accordingly, the collision probability between robots and workers, P_{C}^{s}, can be expressed as:

$$P_{\mathrm{C}}^{s} = \int\limits_{\boldsymbol{x}_r \in \boldsymbol{X}_{\mathrm{R}}^{s}} \int\limits_{\boldsymbol{x}_h \in \boldsymbol{X}_{\mathrm{H}}^{s}} \mathrm{I}_{\mathrm{C}}^{s} \times \mathrm{p}\left(\boldsymbol{x}_r, \boldsymbol{x}_h\right) d\boldsymbol{x}_r d\boldsymbol{x}_h \tag{8}$$

$$\text{where } \mathrm{I}_{\mathrm{C}}^{s} = \begin{cases} 1 & if\ \boldsymbol{X}_{\mathrm{R}}^{s}\left(\boldsymbol{x}_r\right) \cap \boldsymbol{X}_{\mathrm{H}}^{s}\left(\boldsymbol{x}_h\right) \neq \{\emptyset\} \\ 0 & if\ \boldsymbol{X}_{\mathrm{R}}^{s}\left(\boldsymbol{x}_r\right) \cap \boldsymbol{X}_{\mathrm{H}}^{s}\left(\boldsymbol{x}_h\right) = \{\emptyset\} \end{cases}$$

In Equation (8), the symbol $\mathrm{I}_{\mathrm{C}}^{s}$ represents the indicator function, where a value of 1 means that the points occupied by robots overlap the points occupied by workers at frame s, and a value of 0 indicates that the overlap does not occur at frame s. $\mathrm{p}(\boldsymbol{x}_r, \boldsymbol{x}_h)$ is the joint PDF of the random variable $\boldsymbol{x}_r \in \boldsymbol{X}_{\mathrm{R}}^{s}$ and $\boldsymbol{x}_h \in \boldsymbol{X}_{\mathrm{H}}^{s}$. As per Equation (8), to obtain the collision probability, P_{C}^{s}, the integral of $\mathrm{I}_{\mathrm{C}}^{s} \times \mathrm{p}(\boldsymbol{x}_r, \boldsymbol{x}_h)$ over space $\boldsymbol{X}_{\mathrm{R}}^{s}$ and $\boldsymbol{X}_{\mathrm{H}}^{s}$ needs to be calculated. To this end, the first step is to establish the mathematical expression of $\boldsymbol{X}_{\mathrm{H}}^{s}$ and $\boldsymbol{X}_{\mathrm{R}}^{s}$.

To represent $\boldsymbol{X}_{\mathrm{H}}^{s}$—the spatial area occupied by workers at frame s−the authors used 3D poses ($\boldsymbol{pose}_{3\mathrm{D}}^{s}$) of workers obtained from Module 2. As depicted in Figure 5a, for each body part $j \in \{1, 2, \ldots, \mathrm{p}\}$ in $\boldsymbol{pose}_{3\mathrm{D}}^{s}$, its occupied space is approximated by an ellipsoid area (regarded as E_{j}^{s}) between the connected body joints. The center of each ellipsoid is the midpoint of the line connecting the linked body joints, denoted as $\boldsymbol{e}_{j}^{s}$. The lengths of the semiaxes of the ellipsoid are set as $\ell_{e,j}$, $\ell_{e,j}/2$, and $\ell_{e,j}/2$, where $\ell_{e,j}$ is the distance from the center ($\boldsymbol{e}_{j}^{s}$) to the corresponding body joints. As such, for each sample, $\boldsymbol{x}_{h}^{j}$, in the ellipsoid E_{j}^{s}, its distribution can be assumed to follow a Gaussian distribution (Du Toit & Burdick, 2011) $\boldsymbol{x}_{h}^{j} \sim \mathrm{N}(\boldsymbol{e}_{j}^{s}, \boldsymbol{\Sigma}_{h}^{j})$, where the center $\boldsymbol{e}_{j}^{s}$ stands the mean of the distribution, and the correlation matrix $\boldsymbol{\Sigma}_{h}^{j}$ can be represented by $\mathrm{Diagonal}(\ell_{e,j}^{2}, \ell_{e,j}^{2}/4, \ell_{e,j}^{2}/4)$. By adopting the Gaussian distribution assumption, all samples were normally distributed within a Gaussian ellipsoid, corresponding to the ellipsoid E_{j}^{s}. After approximating the space occupied by each body part (i.e., $\boldsymbol{x}_{h}^{j} \in \mathrm{E}_{j}^{s}$), $\boldsymbol{X}_{\mathrm{H}}^{s}$ is obtained using the union operation (∪) between the ellipsoids corresponding to all body parts (Figure 5a), which is expressed as:

$$\boldsymbol{X}_{\mathrm{H}}^{s} = \mathrm{E}_{1}^{s}\ \left(\boldsymbol{x}_{h}^{1}\right) \cup \mathrm{E}_{2}^{s}\ \left(\boldsymbol{x}_{h}^{2}\right) \ldots \cup \mathrm{E}_{j}^{s}\ \left(\boldsymbol{x}_{h}^{j}\right);\ j \in \{1, 2, \ldots,\ p\} \tag{9}$$

Similarly, $\boldsymbol{X}_{\mathrm{R}}^{s}$—the space occupied by the robot—could be obtained using the waypoints and the corresponding configuration of the robot (Maeda et al., 2017; Park et al., 2019). $\boldsymbol{X}_{\mathrm{R}}^{s}$ is estimated with all ellipsoids (regarded as C_{a}^{s}; $\mathrm{a} \in \{1, 2, \ldots, \mathrm{l}\}$) between adjacent joints of the robotic arm (Figure 5a). For this study, the location of each robotic arm joint is acquired from the operating system of the robot during collaboration. The center of each ellipsoid $\boldsymbol{c}_{a}^{s}$ is the midpoint between adjacent joints. The lengths of the semiaxes of the ellipsoid are set as $\ell_{c,a}$, $\ell_{c,a}/2$, and $\ell_{c,a}/2$;$\ell_{c,a}$ is the distance from the center ($\boldsymbol{c}_{\mathrm{a}}^{s}$) to the corresponding robotic arm joints. In addition, every sample, $\boldsymbol{x}_{r}^{a}$, within the ellipsoid C_{a}^{s} follows a Gaussian distribution $\boldsymbol{x}_{r}^{a} \sim \mathrm{N}(\boldsymbol{c}_{a}^{s}, \boldsymbol{\Sigma}_{r}^{a}), \boldsymbol{\Sigma}_{r}^{a} = Diagonal(\ell_{c,a}^{2}, \ell_{c,a}^{2}/4, \ell_{c,a}^{2}/4). \boldsymbol{X}_{\mathrm{R}}^{s}$ is expressed as:

$$\boldsymbol{X}_{\mathrm{R}}^{s} = \mathrm{C}_{1}^{s}\ \left(\boldsymbol{x}_{r}^{1}\right) \cup \mathrm{C}_{2}^{s}\left(\boldsymbol{x}_{r}^{2}\right) \ldots \cup \mathrm{C}_{a}^{s}\left(\boldsymbol{x}_{r}^{a}\right);\ a \in \{1, 2, \ldots,\ l\} \tag{10}$$

As introduced above, for each segment of the human body and the robot, an ellipsoid was selected as the geometric representation of the corresponding occupied space. As demonstrated by multiple studies (Du Toit & Burdick, 2011; Park et al., 2019), the ellipsoid offers a high

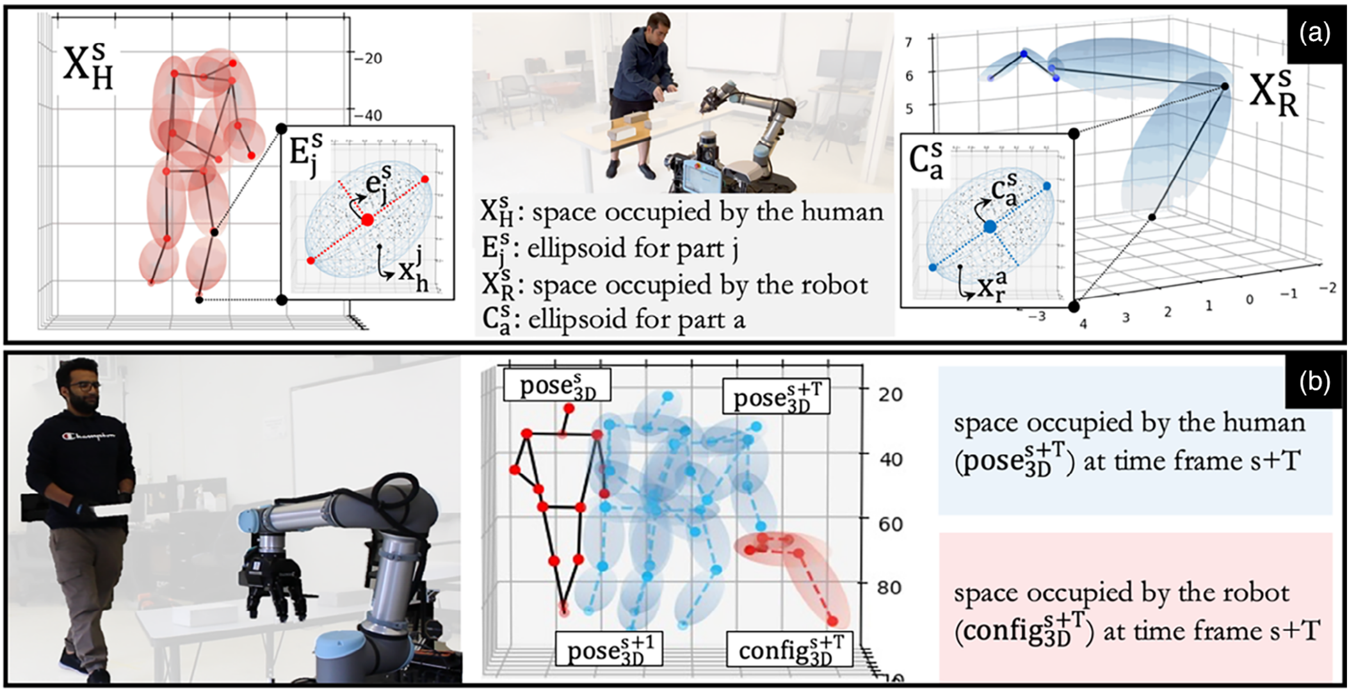

(a)
X_H^s
−20
−40
E_j^s
e_j^s
x_h^j
X_H^s: space occupied by the human
E_j^s: ellipsoid for part j
X_R^s: space occupied by the robot
C_a^s: ellipsoid for part a
7
6
5
X_R^s
C_a^s
c_a^s
x_r^a
−3
4
3
2
1
0
−1
−2
(b)
$pose_{3D}^s$
$pose_{3D}^{s+T}$
20
40
60
80
$pose_{3D}^{s+1}$
$config_{3D}^{s+T}$
space occupied by the human ($pose_{3D}^{s+T}$) at time frame s+T
space occupied by the robot ($config_{3D}^{s+T}$) at time frame s+T

FIGURE 5 (a) Illustration of key parameters in Algorithm 1; (b) illustration of key parameters in Algorithm 2.

degree of adaptability, enabling a close approximation to the occupied spaces of the segments of the human body. From a computational perspective, ellipsoids facilitated an efficient and continuous calculation of the collision probability, using the principles of multivariate Gaussian distribution (Du Toit & Burdick, 2011; Rafiei & Adeli, 2017)—details of which will be introduced in the following content. Moreover, the broader central region of the ellipsoid offered an added spatial buffer (compared to other shapes like cylinders) to the occupied space of robotic links. This extra space may serve as a safety buffer, enhancing the safety of human–robot interactions. Notably, this ellipsoid-based space approximation has the potential to be adaptable to diverse robotic structures: A robotic arm with l links could employ l ellipsoids to estimate its occupied space. Meanwhile, robots with simpler structures might only require a single ellipsoid for efficient spatial approximation of collision estimation, as evidenced by the study in Du Toit and Burdick (2011).

After obtaining the expressions of $\boldsymbol{X}_{\mathrm{H}}^{\mathrm{s}}$ and $\boldsymbol{X}_{\mathrm{R}}^{\mathrm{s}}$ approximated by ellipsoids, it is noteworthy that any intersection between segments of the human body and those of the robot will be regarded as a collision. To assess this collision risk, the authors performed pairwise calculations of the collision risk between any parts of the human body and the robot. The maximum value of the collision risk among all these pairs is the collision probability between the robot and the worker at frame s, $P_{\mathrm{C}}^{\mathrm{s}}$. Then, Equation (8) for calculating $P_{\mathrm{C}}^{\mathrm{s}}$ can be reformulated as follows:

$$P_{\mathrm{C}}^{\mathrm{s}} = \max_{\substack{a \in \{1,2,\dots,\mathrm{p}\} \\ j \in \{1,2,\dots,q-1\}}} \left\{ \int_{\boldsymbol{x}_r^a} \int_{\boldsymbol{x}_h^j} \mathrm{p}\left(\boldsymbol{x}_r^a \in C_a^s, \boldsymbol{x}_h^j \in \mathrm{E}_j^s, \mathrm{C}_a^s \cap \mathrm{E}_j^s \neq \{\emptyset\} \right) d\boldsymbol{x}_r^a d\boldsymbol{x}_h^j \right\} \tag{11}$$

According to the Kolmogorov definition, Equation (11) can be rewritten as follows:

$$P_{\mathrm{C}}^{\mathrm{s}} = max \quad \substack{a \in \{1,2,\dots,\mathrm{p}\} \\ j \in \{1,2,\dots,q-1\}} \left\{ \int_{\boldsymbol{x}_r^a \in \mathrm{C}_a^s} \left[\int_{\boldsymbol{x}_h^j \in \mathrm{C}_a^s \cap \mathrm{E}_j^s} \mathrm{p}\left(\boldsymbol{x}_h^j | \boldsymbol{x}_r^a \right) d\boldsymbol{x}_h^j \right] \times \mathrm{p}\left(\boldsymbol{x}_r^a \right) d\boldsymbol{x}_r^a \right\} \tag{12}$$

where $\mathrm{p}(\boldsymbol{x}_h^j | \boldsymbol{x}_r^a)$ is the PDF of $\boldsymbol{x}_h^j$ given $\boldsymbol{x}_r^a$, and $\mathrm{p}(\boldsymbol{x}_r^a)$ is the marginal density of $\boldsymbol{x}_r^a$. Additionally, based on the investigation reported by Du Toit and Burdick (2011), the inner integral can be estimated through the product between the volume of space occupied by the part of the robot ($\mathrm{V}(\mathrm{C}_a^s)$; ellipsoid) and the conditional distribution of worker over space occupied by the part of the robot, $\mathrm{p}(\boldsymbol{x}_h^j = \boldsymbol{x}_r^a \,|\boldsymbol{x}_r^a)$:

$$\int_{\boldsymbol{x}_h^j \in \mathrm{C}_a^t \cap \mathrm{E}_j^t} \mathrm{p}\left(\boldsymbol{x}_h^j | \boldsymbol{x}_r^a\right) d\boldsymbol{x}_h^j \approx \mathrm{V}\left(\mathrm{C}_a^t\right) \times \mathrm{p}\left(\boldsymbol{x}_h^j = \boldsymbol{x}_r^a \,|\boldsymbol{x}_r^a\right) \tag{13}$$

Using Equation (13), the probability of collision P_C^s can be simplified as the following:

$$P_\mathrm{C}^\mathrm{s} \approx max_{\substack{a \in \{1,2,\ldots,\mathrm{l}\} \\ j \in \{1,2,\ldots,\mathrm{p}\}}} \left\{ \mathrm{V}\left(\mathrm{C}_a^s\right) \times \int_{\boldsymbol{x}_r^a \in \mathrm{C}_a^s} \mathrm{p}(\boldsymbol{x}_h^j = \boldsymbol{x}_r^a \,|\boldsymbol{x}_r^a) \mathrm{p}\left(\boldsymbol{x}_r^a\right) d\boldsymbol{x}_r^a \right\} \tag{14}$$

Considering random variables $\boldsymbol{x}_h^j$ and $\boldsymbol{x}_r^a$ follow the Gaussian distribution—$\boldsymbol{x}_h^j \sim \mathrm{N}(\boldsymbol{e}_j^s, \boldsymbol{\Sigma}_h^j)$ and $\boldsymbol{x}_r^s \sim \mathrm{N}(\boldsymbol{c}_a^s, \boldsymbol{\Sigma}_r^a)$, P_C^s can be derived as follows (Du Toit & Burdick, 2011; Park et al., 2019):

$$P_\mathrm{C}^\mathrm{s} \approx max_{\substack{a \in \{1,2,\ldots,\mathrm{l}\} \\ j \in \{1,2,\ldots,p\}}} \left\{ \frac{\mathrm{V}\left(\mathrm{C}_a^s\right)}{\sqrt{\det\left(2\pi \boldsymbol{\Sigma}_C^{j,a}\right)}} \exp\left[-\frac{1}{2}\left(\boldsymbol{c}_a^s - \boldsymbol{e}_j^s\right)^\mathrm{T} \left(\boldsymbol{\Sigma}_C^{j,a}\right)^{-1} \left(\boldsymbol{c}_a^s - \boldsymbol{e}_j^s\right)\right] \right\} < 1 \tag{15}$$

where

$$\boldsymbol{\Sigma}_\mathrm{C}^{j,a} = \boldsymbol{\Sigma}_h^j + \boldsymbol{\Sigma}_r^a - \boldsymbol{\Sigma}_m^{j,a} - \left(\boldsymbol{\Sigma}_m^{j,a}\right)^\mathrm{T}; \; \boldsymbol{\Sigma}_m^{j,a} = \begin{bmatrix} \mathrm{cov}\left(x_{h,x}^j, x_{r,x}^a\right) & \mathrm{cov}\left(x_{h,x}^j, x_{r,y}^a\right) & \mathrm{cov}\left(x_{h,x}^j, x_{r,z}^a\right) \\ \mathrm{cov}\left(x_{h,y}^j, x_{r,x}^a\right) & \mathrm{cov}\left(x_{h,y}^j, x_{r,y}^a\right) & \mathrm{cov}\left(x_{h,y}^j, x_{r,z}^a\right) \\ \mathrm{cov}\left(x_{h,z}^j, x_{r,x}^a\right) & \mathrm{cov}\left(x_{h,z}^j, x_{r,y}^a\right) & \mathrm{cov}\left(x_{h,z}^j, x_{r,y}^a\right) \end{bmatrix} \tag{16}$$

In this equation, $\boldsymbol{\Sigma}_\mathrm{C}^{j,a}$ represents the combined position covariance matrix between workers and robots. $\boldsymbol{\Sigma}_m^{j,a}$ describes the matrix that computes the covariance values between random variables $\boldsymbol{x}_h^j$ and $\boldsymbol{x}_\mathrm{r}^\mathrm{a}$ with respect to x, y, and z coordinates, which can be derived using all samples from $\boldsymbol{x}_h^j \in \mathrm{E}_j^s$ and $\boldsymbol{x}_r^a \in \mathrm{C}_a^s$. At this point, all parameters in Equation (15) are known; thus, the probability of collision between workers and robots can be computed. Moreover, based on the derivation of the probabilistic collision-checking mechanism (Equations 8–15), the pseudo-code for computing P_C^s is shown in Algorithm 1.

ALGORITHM 1 Probabilistic Collision-Checking.

Inputs:
$\mathrm{pose}_{3\mathrm{D}}^\mathrm{s}$: worker's 3D pose estimated from timeframe s; including the 3D coordinates of all body joints ($\mathrm{joint}_1, \ldots, joint_\mathrm{q}$)
$\mathrm{config}_{3\mathrm{D}}^\mathrm{s}$: configuration (e.g., joint angles, joint coordinates, link lengths) of the robotic arm in timeframe s; obtained from robot operating system

```
1.  For j = 1, 2, ..., p
2.      $\boldsymbol{e}_\mathrm{j}^\mathrm{s} = \frac{1}{2}(\mathrm{joint_j} + \mathrm{joint_{j+1}}) \rightarrow$ obtain $\ell_\mathrm{e,j}$, $\frac{1}{2}\ell_\mathrm{e,j}$, and $\frac{1}{2}\ell_\mathrm{e,j}$ based on $\boldsymbol{e}_\mathrm{j}^\mathrm{s}$
3.      $\boldsymbol{\Sigma}_\mathrm{h}^\mathrm{j}$ = Diagonal($\ell_\mathrm{e,j}^2, \frac{1}{4}\ell_\mathrm{e,j}^2, \frac{1}{4}\ell_\mathrm{e,j}^2$)
4.      Generate 4,000 samples based on the Gaussian distribution $\boldsymbol{x}_h^j \sim \mathrm{N}(\boldsymbol{e}_j^s, \boldsymbol{\Sigma}_h^j)$
5.  End for
6.  For $a = 1, 2, \ldots, \mathrm{l}$
7.      Obtain $\boldsymbol{c}_\mathrm{a}^\mathrm{s}$ and $\boldsymbol{\Sigma}_\mathrm{r}^\mathrm{a}$
8.      Generate 2,000 samples based on the Gaussian distribution $\boldsymbol{x}_r^a \sim \mathrm{N}(\boldsymbol{c}_a^s, \boldsymbol{\Sigma}_r^a)$
9.  End for
10. For j = 1, 2, ..., p
11.     For $a = 1, 2, \ldots, \mathrm{l}$
12.         Calculate $\boldsymbol{\Sigma}_\mathrm{m}^\mathrm{j,a}$ and $\boldsymbol{\Sigma}_\mathrm{C}^\mathrm{j,a}$ using Equation (16)
13.     End for
14. End for
15. Calculate the $P_\mathrm{C}^\mathrm{s} = \max_{\substack{a \in \{1,2,\cdots,\mathrm{p}\} \\ j \in \{1,2,\cdots,q-1\}}} \left\{ \frac{\mathrm{V}(\mathrm{C}_a^s)}{\sqrt{\det(2\pi\boldsymbol{\Sigma}_C^{j,a})}} \exp\left[-\frac{1}{2}(\boldsymbol{c}_a^s - \boldsymbol{e}_j^s)^\mathrm{T}(\boldsymbol{\Sigma}_C^{j,a})^{-1}(\boldsymbol{c}_a^s - \boldsymbol{e}_j^s)\right] \right\}$
16. End
```

As outlined, the inputs for the probabilistic collision-checking mechanism are the worker 3D poses ($\boldsymbol{pose}_{3\mathrm{D}}^s$) predicted from Module 2 and the configuration of the robot containing all positions of joints ($\boldsymbol{config}_{3\mathrm{D}}^s$) obtained from the robot operating system. For the input $\boldsymbol{pose}_{3\mathrm{D}}^s$, the mechanism first computes the center point of the ellipsoid area, $\boldsymbol{e}_j^s$, between each two adjunct body joints, as shown in line 2. Then, the mechanism obtains the covariance matrix $\boldsymbol{\Sigma}_h^j$ of each ellipsoid, line 3. Next, the mechanism samples the data points within the space occupied by each body part of workers (E_j^s, $j \in \{1,2,\ldots,\mathrm{p}\}$) based on $\boldsymbol{x}_h^j \sim \mathrm{N}(\boldsymbol{e}_j^s, \boldsymbol{\Sigma}_h^j)$, employing a sample size of 4000 (line 4). Parallelly, for the input $\boldsymbol{config}_{3\mathrm{D}}^s$, the mechanism generates the center point of each pair of robotic arm joints ($\boldsymbol{c}_a^s$), obtains the covariance matrix $\boldsymbol{\Sigma}_r^a$, and estimates the space occupied by each part of the robot (C_a^s, $a \in \{1,2,\ldots,\mathrm{l}\}$) by generating

2000 samples based on $\boldsymbol{x}_r^a \sim \mathrm{N}(\boldsymbol{c}_a^s, \boldsymbol{\Sigma}_r^a)$. Then, based on all samples $\boldsymbol{x}_h^j$ in each body part of workers and $\boldsymbol{x}_r^a$ in each part of the robot, the $\boldsymbol{\Sigma}_m^{j,a}$ and $\boldsymbol{\Sigma}_\mathrm{C}^{j,a}$ can be obtained using Equation (16), line 10 to line 13. Accordingly, Equation (15) is applied to compute the collision risk between each pair of the body parts of workers and robots (line 15). The maximum value of the collision risk among all these pairs is the collision probability between the robot and the worker, P_C^s. Additionally, it is worth noting that the selection of sample size—4000 for each human body part and 2000 for each robot component—was determined through a rigorous iterative testing. During this process, the authors found that decreasing these sample sizes adversely affected the algorithm's performance in accurately representing the spaces occupied by both the human body and robot parts. On the other hand, expanding the sample size beyond these thresholds (4000 and 2000) failed to enhance the approximation accuracy but increased computational requirements. Hence, sample sizes of 4000 and 2000 were chosen to provide the algorithm with an optimal balance between computational efficiency and performance.

Algorithm 1. Pseudo-code for estimating the collision probability between robots and workers.

3.3.2 | Collision-free motion planning mechanism for the robot

After enabling the robot to evaluate collision probabilities with workers, the next step is to endow robots with the ability to replan and generate trajectories ensuring zero collision risk with the (anticipated sequences of) worker motion. To this end, the authors proposed a motion planning mechanism. The pseudo-code of the mechanism is shown in Algorithm 2. An illustration of the mechanism is shown in Figure 5b. As outlined by this mechanism, prior to the robot executing its upcoming action based on the preplanned trajectory, a preliminary check is conducted. The robot assesses the collision risk between its future configurations along with its preplanned trajectory ($\boldsymbol{config}_{3\mathrm{D}}^{s+1}, \ldots, \boldsymbol{config}_{3\mathrm{D}}^{s+T}$) and intended motion sequences of workers ($\boldsymbol{pose}_{3\mathrm{D}}^{s+1}, \ldots, \boldsymbol{pose}_{3\mathrm{D}}^{s+T}$) using the proposed probabilistic collision-checking mechanism (Figure 5b). For each pair of $\boldsymbol{config}_{3\mathrm{D}}^{s+i}$ and $\boldsymbol{pose}_{3\mathrm{D}}^{s+i}$ ($i \in \{1, \ldots, T\}$), if the collision probability P_C^{s+i} is lower than a user-defined threshold ($\delta = 5\%$; the determination of δ will be discussed in Section 6), the robot proceeds with executing the collaborative task using the preplanned trajectory (from line 1 to line 8). Conversely, when the collision risk $P_\mathrm{C}^{\mathrm{s+i}}$ is higher than δ, the robot engages in configuration adjustments (starts from line 9) by replanning its trajectory from the trajectory set $\boldsymbol{TRAJ} = \{\boldsymbol{traj}_1, \cdots, \boldsymbol{traj}_\mathrm{n}\}$. $\boldsymbol{TRAJ}$ contains all trajectories for the robot to perform the collaborative task, each determined by the target positions. Each element $\boldsymbol{traj}_k$ ($k \in \{1, \ldots, n\}$) in set $\boldsymbol{TRAJ}$ also consists of T configurations of the applied robotic arm, that is, $\boldsymbol{traj}_\mathrm{k} = \{\boldsymbol{config}_{3\mathrm{D},k}^{s+1}, \ldots, \boldsymbol{config}_{3\mathrm{D},k}^{s+T}\}$. To reselect the trajectory from $\boldsymbol{TRAJ}$, the robot will calculate $P_{\mathrm{C},k}^{s+i}$ between each new configuration ($\boldsymbol{config}_{3\mathrm{D},k}^{s+i}$) of the robot and each intended motion ($\boldsymbol{pose}_{3\mathrm{D}}^{s+i}$) of workers, as shown in line 10. For each possible trajectory in $\boldsymbol{TRAJ}$ set, $\boldsymbol{traj}_k$, if any of the collision risks $P_{\mathrm{C},k}^{s+i}$ ($i \in \{1, \ldots, T\}$) surpass the threshold, the robot dismisses this trajectory (line 11). Conversely, if all collision risks $P_{\mathrm{C},k}^{s+i}$ ($i \in \{1, \ldots, T\}$) fall

ALGORITHM 2 Collision-free Motion Planning.

```
Inputs:
$\boldsymbol{pose}_{3D}^{s+1}, \cdots, \boldsymbol{pose}_{3D}^{s+T}$: worker's 3D poses estimated from time
    frame s to future time frame $s+T$; each $\boldsymbol{pose}_{3D}^{s+i}$ includes the
    3D coordinates of all body joints
$\boldsymbol{config}_{3D}^{s+1}, \cdots, \boldsymbol{config}_{3D}^{s+T}$: configurations of the robotic arm
    spanning from time frame s to the subsequent time frame
    $s+T$; these configurations are acquired from the robot's
    pre-planned trajectory
$\boldsymbol{TRAJ} = \{\boldsymbol{traj}_1, \cdots, \boldsymbol{traj}_n\}$: all possible trajectories that enable
    the robot to perform the task by facilitating its movement from
    the current position A to the desired position B; Each
    $\boldsymbol{traj}_i$ contains the configurations of the robotic arm from time
    frame s to the subsequent time frame $s+T$
1.  Fori $= 1, 2, \ldots, T$
2.      Calculate $P_C^{s+i}$ between $\boldsymbol{pose}_{3D}^{s+i}$ and $\boldsymbol{config}_{3D}^{s+i}$
3.      If $P_C^{s+i} \le \delta = 5\%$:
4.          Continue the loop
5.      Else:
6.          Break the loop and go to line 9
7.  End for
8.  Return$\{\boldsymbol{config}_{3D}^{s+1}, \cdots, \boldsymbol{config}_{3D}^{s+T}\} \rightarrow$ End the
        algorithm
9.  For traj$_k$ in $\boldsymbol{TRAJ}$:
10.     Calculate $P_{C,k}^{s+i}$ between $\boldsymbol{pose}_{3D,k}^{s+i}$ and $\boldsymbol{config}_{3D,k}^{s+i}$
        for all i $= 1, 2, \ldots, T$
11.     If any $P_{C,k}^{s+i}$ (i $= 1, 2, \ldots, T$) $> \delta \rightarrow$ dismiss trajectory
        $\boldsymbol{traj}_k$
12.     Else save $\boldsymbol{traj}_k$ in set Collision$_{free}$
13. End for
14. For each element ($\boldsymbol{traj}_{free}^m$; m $= 1, \cdots,$ M) in
        Collision$_{free}$:
15.     Calculate the corresponding distance (D$^m$) to the
        current configuration of the robot
16. End for
17. Optimal $\boldsymbol{traj}^* = \{D^m\}$
18. Return $\boldsymbol{traj}^*$
19. End
```

below the threshold, $\boldsymbol{traj}_k$ will be selected as a feasible collision-free trajectory, and this feasible $\boldsymbol{traj}_k$ will be saved in $\text{Collision}_{\text{free}}$ set (line 12). After screening all feasible collision-free trajectories ($\boldsymbol{traj}^m_{\text{free}}; m = 1, \ldots, M$), the robot will further calculate the distance (D^{m}) between its current configuration and each feasible $\boldsymbol{traj}^m_{\text{free}}$ in $\text{Collision}_{\text{free}}$ set (from line 14 to line 16). The robot selects the $\boldsymbol{traj}^*$ with the shortest traversal distance as the optimal replanned trajectory to ensure the avoidance of collisions with the worker during HRC.

Algorithm 2. Pseudo-code for generating the collision-free adjustment for collaborative robots.

4 | CASE STUDY: COLLABORATIVE BRICKLAYING TASK

To assess the viability of the proposed IAPMP and verify its functionality, a human–robot collaborative bricklaying task was conducted, consisting of training and testing sessions. The training session was designed to generate two data sets to train the 3D-MotNet for pose estimation and to train the IntentionNet for motion intention inference. These trained deep networks were then integrated with the collision-checking motion planning mechanism to generate the trained IAPMP. Subsequently, the testing session was applied to evaluate the performance of the IAPMP in allowing the robot to generate collision-free trajectories based on the motion intention of workers and the collision probability with workers. Details of the training and testing sessions of the collaborative bricklaying task will be discussed below.

4.1 | Training session—Task description

During the training session of the task, 10 subjects ($\text{S}_1, \ldots, \text{S}_{10}$; mean age: 27.9; SD: 1.67) were recruited to collaborate with an Unmanned Ground Vehicle (UGV) robot equipped with a robotic arm (Universal Robot: UR5e) and a gripper (Robotiq 3-finger adaptive gripper). Their task was to perform a 15-round bricklaying task in a shared workplace. Figure 6a shows an example image captured during the task. For each round of the task, the subjects were asked to follow a sequence of actions. First, each subject needed to pick up a concrete brick from the material staging area. Second, each subject carried the brick from the material staging area to a designed collaboration area. Third, the subject was asked to place the brick into the material feeding area of the collaborative robot. As shown in Figure 6a,b, the material feeding area was positioned before the robot, featuring a row of three vacant spaces marked as B_1, B_2, and B_3. Each subject was asked to place bricks sequentially, starting from B_1 and continuing to B_3, one position per round. Once placed, the robot was controlled to pick up the brick to build a construction wall (height: 3 *bricks*; length: 5 *bricks*; width: 1 brick). By the completion of 15 rounds of this task, an equal number of bricks had been fed to each of the designed locations (B_1, B_2, B_3), and correspondingly, the robot could pick up the same number of bricks at each position. Notably, this "turn-taking" setting would ensure that the collaboration would not cause collisions between subjects and robots during the training session.

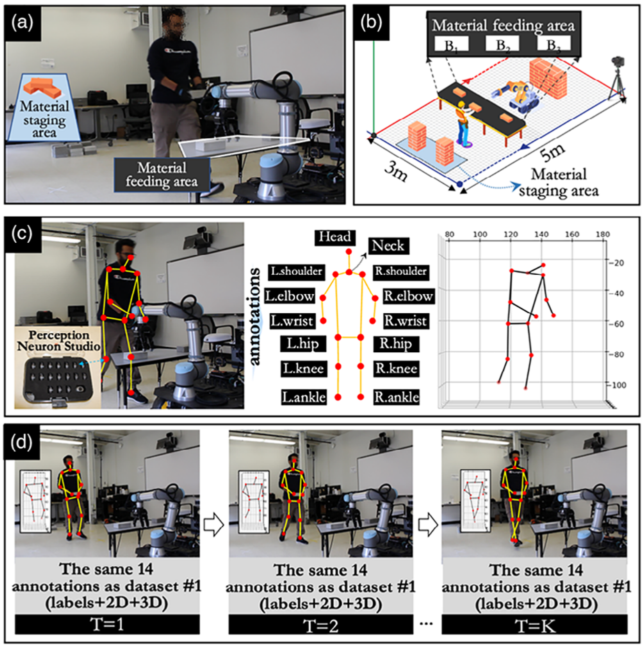


FIGURE 6 (a) Designed collaborative bricklaying task in the training session; (b) detailed setup of the bricklaying task; (c) example images in data set 1 and annotations (S_4); (d) example images in data set 2 (subject: S_4; group: 5).

The 15-round bricklaying task took around 20 min to complete for each subject. Besides, prior to the first round of the task, each subject engaged in a warm-up session consisting of a set of stretching exercises, following the guidelines presented in Holmström and Ahlborg (2005). Furthermore, before the task, the authors demonstrated a series of safe actions and postures for lifting, carrying, and placing materials by following the criteria regulated by the Occupational Safety and Health Administration (OSHA; Schneider, 1995). Subjects were instructed to follow the authors' demonstration throughout the bricklaying task. These two steps were implemented to mitigate the potential risk of back sprains, elbow injuries, and shoulder injuries for each subject.

4.2 | Training session—Data set preparation

Figure 6b shows the detailed setup of the above collaborative task. The designated area for the task was approximately 3m × 5 m. One digital video (DV) camera was placed in one corner of the space. The resolution of the DV camera was 1920 pixels by 1080 pixels, with a frame rate locked at 24 frames per second (fps). During the training session, the camera was used to capture the images of the HRC. These images were collected to train the 3D-MotNet and IntentionNet within the IAPMP method. To reduce the computational resources in training these networks, a strategy was used: For every 12 consecutive frames captured by the camera, one frame was selected for IAPMP training. As mentioned, each subject performed around 20 min of the task. Therefore, around 2400 images [= (24 images/s ÷ 12 × 60 s/min × 20 min)] were collected per subject. In this study, 24,056 images containing human poses were captured from all 10 subjects.

In addition to using the camera for capturing 2D human pose images, a motion capture system (Perception Neuron Studio; Wei et al., 2017) was used to collect the 3D coordinates of human poses to train the IAPMP. Each subject was equipped with a set of motion capture sensors. During the task, these sensors recorded the 3D coordinates of corresponding body joints. The recording of the motion capture system was synchronized with the recording of the DV camera. The frame rate of the recording was set as 24 fps as well. Likewise, for every 12 frames, the authors generated one set of 3D coordinates for each chosen 2D human pose image. As such, every human pose image was paired both with its 2D representation, as well as the corresponding 3D coordinates of the human pose. Notably, after obtaining these 3D coordinates from the motion capture system, the coordinates were aligned with the coordinate system of the robot. This alignment was achieved through the calculation of a homogeneous transformation matrix between these two coordinate systems. For this purpose, the 3D coordinates of the four corners of the material feeding area were measured in both the motion capture's and the robot's coordinate systems. According to the 3D coordinates of the identical object in both coordinate systems, the homogeneous transformation matrix could be obtained using the process introduced in Khajwal et al. (2023), Mihelj et al. (2019), and Pan and Yang (2023). Upon aligning the two sets of 3D coordinates using the homogeneous matrix, the 3D coordinates of both the UR5e robot and the captured human poses were harmonized into a unified coordinate system (global frame). Then, these aligned coordinates of the 3D human poses were used in subsequent steps to train the proposed networks for pose estimation and motion intention inference.

After obtaining the human pose images and their corresponding (aligned) 3D coordinates, the authors generated two data sets. The first data set (data set 1) was used to train the 3D-MotNet, allowing it to extract 3D human poses from the images captured by the camera. The second data set (data set 2) was applied to train the IntentionNet to predict the motion intentions of the subjects. To generate data set 1, the authors first annotated the captured images by following widely accepted annotation protocols. Specifically, for every image in data set 1, the authors annotated 14 human body joints (Lin et al., 2014), including the head, neck, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, and right ankle joints, one annotation per body joint. Each annotation consisted of three key elements—(1) the label of the corresponding body joint (e.g., head, left shoulder), (2) the 2D coordinates of the body joint, and (3) the 3D coordinates of the joint. The 2D coordinates of each joint were manually labeled using the Clickworker platform (Cao et al., 2021). The 3D coordinates of the joint were obtained from the motion capture system. An exemplar image in data set 1 and its associated annotation are shown in Figure 6c. During the stages of training and testing, the 2D coordinates of the body joint in every image were served as the ground-truth $\boldsymbol{pose}_{2D}^{S*}$ for assessing the intermediate loss of 3D pose estimation. The 3D coordinates of the body joint in every image were served as the ground-truth $\boldsymbol{pose}_{3D}^{S*}$ for calculating the loss of Equation (2). $\boldsymbol{pose}_{3D}^{S*}$ was a 14×3 matrix, where 14 indicates the annotated 14 joints and 3 indicates the coordinates of each joint were 3D. Data set 1 contained 24,056 images with the corresponding annotations. Eighty percent of the images in data set 1 were used to train the 3D-MotNet, with the remaining 20% being used for performance validation. The results will be reported in the Section 5.

Data set 2 was generated based on the modification of data set 1. The authors first divided all images in data set 1 into 10 sub–data sets, with each sub–data set dedicated to a particular human subject. Then, for each sub–data set, the authors further divide it into 15 groups according to the 15 rounds of the collaborative bricklaying tasks. Every group corresponded to a single task round. In each group, every image had 14 annotations; each annotation consisted of three key elements, the same as in data set 1. The sequence of images in each group was arranged based on the chronological order of their capture. In this study, the network was trained with the group of images (and the corresponding annotations) in each sub–data set of data set 2. Throughout the training and validation phases, the 2D and 3D coordinates for each body joint in every image were regarded as ground-truth $\boldsymbol{pose}_{2D}^{*(s+T)}$ and $\boldsymbol{pose}_{3D}^{*(s+T)}$ to measure the loss of Equations (6) and (7). Examples of the images in one group in data set 2 are shown in Figure 6d. In addition, the authors randomly selected eight sub–data sets to train the IntentionNet, and the remaining two sub–data sets to validate the IntentionNet. The corresponding results will be reported in the Section 5.

As outlined, data sets 1 and 2 provided annotated image data to train the proposed 3D-MotNet and IntentionNet for 2D/3D pose estimation and motion intention estimation. The authors would like to note that while recent advancements have introduced several annotation-free methods for pose estimation (Zheng et al., 2023), such as unsupervised, semisupervised, and geometry-aware representation-based methods, the significance of annotated data sets remains evident. Specifically, models trained on annotated data sets tend to achieve enhanced accuracy in pose estimation, an advantage that might not be matched by annotation-free models (Zhang et al., 2019). Within the context of this study, the precision of pose estimation directly impacted the subsequent motion intention and collision probability assessments. This highlighted the importance of precise pose estimation and the significance of annotated data sets in model development. Thus, despite the effort required to prepare annotated data sets, the important role of these data sets in ensuring accuracy proved their use in this study.

Furthermore, before utilizing data set 1 to train and validate the 3D-MotNet, the authors performed an additional step to pretrain the network using two publicly available data sets—the Microsoft Common Objects in Context (COCO) data set (Lin et al., 2014) was used to pretrain the subnetwork 1 of the 3D-MotNet; and the Human3.6 M data set (Ionescu et al., 2014) was used to train the subnetwork 2 of the 3D-MotNet. The COCO data set was a large-scale data set (328,000 images with annotations of 2.5 million objects) widely used to train and calibrate deep networks for 2D human pose estimations across various domains (Lin et al., 2014). Likewise, the Human3.6 M data set was a widely accepted data set containing 3.6 million images of human poses and the corresponding 3D coordinates of each body joint (Ionescu et al., 2014), used for training deep networks to extract 3D human poses. In theory, this pretraining step could enable the 3D-MotNet to estimate the 3D human poses from 2D images. However, in the context of domain-specific actions, such as construction tasks performed by subjects, the authors' empirical observations (Liu & Jebelli, 2022) revealed instances where the pretrained network may occasionally fail to accurately detect human poses. To solve this problem, the step of using data set 1 to train (fine-tune) the 3D-MotNet was conducted, aiming to allow the network to have an enhanced performance in extracting the 3D human poses from the input images. In this study, the model pretraining step was conducted by following the widely adopted training protocols (Wang et al., 2019). Eighty percent of the COCO and Human3.6 M data were used to pretrain the 3D-MotNet, and the remaining 20% were used to validate its performance. For simplicity, this study will not introduce the pretraining step for the 3D-MotNet. Details, including descriptions of the COCO and Human3.6 M data sets, as well as the pretraining process, can be found in the authors' previous study and related investigations (Liu & Jebelli, 2022; Wang et al., 2019).

4.3 | Testing session: Performance evaluation of the IAPMP

Once obtaining the trained 3D-MotNet and IntentionNet, these two networks were integrated with the collision-checking motion planning mechanism to generate the IAPMP. Subsequently, the testing session was conducted to assess the performance of the IAPMP. For this testing session, six subjects were randomly selected to engage in 20 rounds of a modified collaborative bricklaying task. The settings of the modified task were similar to that of the task outlined in Section 4.1. The only difference was that two of the positions within B_1, B_2, B_3 had bricks prepositioned before the start of each round. At the beginning of each round in this modified task, each subject picked up and delivered a concrete brick from the material staging area to the material feeding area for the collaborative robot. The subject was required to place the brick in an empty slot among B_1, B_2, B_3, which was randomized by the authors prior to each round. Notably, in this modified task, the robot did not wait for the subject to complete brick placement before proceeding to pick up the corresponding brick. Instead, the robot was programmed to randomly choose one of the positions—B_1, B_2 or B_3—to pick up the brick at a time close to when the subject placed the brick. If there was no brick in the selected position, the robot moved to the adjacent locations to pick up the brick and build the wall. Given this dynamic scenario, where the collaborative turn-taking plan (in Section 4.1) was absent, the robot may collide with subjects, particularly when the motion trajectory of the robotic arm spatially overlaps the motion of the subject. To avoid collisions, the proposed IAPMP was activated in the operating system of the applied Universal Robot (UR5e). If the proposed IAPMP worked properly, the robot could anticipate the motion intention of the subject, understand the future position of the subject, calculate the collision probability accordingly, and generate the collision-free trajectories to pick up the brick. Specifically, as elaborated in Section 3.3.2 (Algorithm 2), the collision-free trajectories were generated by IAPMP from a predefined trajectory set, $\boldsymbol{TRAJ}$. Prior to the testing session, the authors utilized the forward and inverse kinematic functions to define all plausible robotic trajectories within $\boldsymbol{TRAJ}$ set. These trajectories, informed by the initial configuration and goal position (B1, B2, or B3) of the UR5e robot for each round, were represented by a series of angles associated with every robotic joint. Each trajectory enabled the UR5e robot to pick up the brick from one of the goal positions (B1, B2, or B3). In each round, if the collision probability was higher than the threshold in Algorithm 2, the IAPMP directed the robot to select an alternative trajectory from $\boldsymbol{TRAJ}$, ensuring collision avoidance with the subject.

In cases where the proposed IAPMP failed to generate appropriate adjustments, to prioritize the safety of all subjects, the authors implemented two safety measures. First, throughout the testing session, two trained members from the research team were appointed as “on-site supervisors,” each with specific safety-monitoring responsibilities. One individual was responsible for monitoring the performance of the proposed IAPMP and the real-time robot operations (visualized on an onsite workstation). If the UR5e robot generated incorrect adjustments due to predictive errors of the IAPMP or its own unexpected operational errors, this supervisor immediately halted the experiment by activating the robot’s emergency stop feature. Concurrently, the other individual focused on the direct physical interactions between the subject and the UR5e robot. This role was designed to supplement the first supervisor’s duties: In situations where the first supervisor’s reaction to activate the emergency stop might be delayed, or in cases where a correct robotic adjustment might lead to a collision due to collision-free thresholds (δ) (though this scenario was not realized during the study), the second supervisor had the authority to stop the robot. Moreover, both supervisors documented any failure instances, noting causes for reasons for subsequent analysis. Second, during the case study, the robotic arm was programmed to operate at a controlled speed by setting the safety configuration of the UR5e robot (Kirschner et al., 2021). This measure was used to mitigate the potential impact force from any unexpected collisions. Through these precautions, the safety of all subjects throughout the testing session could be ensured. The performance of the IAPMP will be reported in Section 5.

5 | RESULTS

5.1 | Performance of the 3D-MotNet in 3D pose estimation

In this section, the training performance of the 3D-MotNet in extracting the 3D poses of workers from 2D images was presented. As introduced, the 3D-MotNet was first pretrained with two public data sets and then trained (fine-tuned) with data set 1. During pretraining, the Xavier initialization method (Kumar, 2017) was used to initialize all weights of the 3D-MotNet. Stochastic gradient descent (SGD) with momentum optimizer was applied to optimize the network with a learning rate of 1.5×10^{-5}. Subsequently, data set 1 was employed to fine-tune the efficiency of the pretrained network in extracting the 3D poses of workers engaged in construction tasks. The initial weights of the fine-tuned network were obtained from the pretrained network’s weights after training on the public data sets. The optimizer and the learning rate were the same as those used in the pretraining process. Furthermore, to determine the optimal value of H for subnetwork 1 in the 3D-MotNet (as mentioned in Section 3.2), the entire 3D-MotNet was pretrained and trained with varying values of H. H was chosen within the range 3 to 7, based on investigations reported in Cao et al. (2021). For each H value, Table 1 reported the validation performance of the fine-tuned 3D-MotNet on data set 1 in extracting subjects’ 3D postures. The table presented the Euclidean distance between each body joint of the 3D human poses extracted by the fine-tuned 3D-MotNet and its corresponding ground-truth value. When $H = 4$, the fine-tuned 3D-MotNet achieved its optimal performance. Additionally, the last row of Table 1 was the baseline calculated by directly applying the pretrained 3D-MotNet ($H = 4$) to the validation data set of data set 1. The results demonstrate that the fine-tuned model improved its 3D pose estimation performance by 29.7% compared to the pretrained counterpart. Further, Figure 7a visualized the outcomes when directly applying the pretrained 3D-MotNet ($H = 4$) to the validation data set of data set 1, while Figure 7a depicted the outcomes using the fine-tuned 3D-MotNet ($H = 4$) on the same validation data set. Notably, the fine-tuned model accurately extracted subjects’ 3D postures during construction tasks involving standing, carrying, and delivering. In contrast, the pretrained network encountered difficulties in accurately estimating several 3D poses (refer to the errors in red circles)

5.2 | Performance of the IntentionNet in motion intention inference

This section reported the performance evaluation of the IntentionNet in the IAPMP for inferring human motion. Data set 2 in Section 4.2 was implemented to train the IntentionNet. Throughout the training process, the latent vector $\boldsymbol{z}$ of the IntentionNet was initialized according to the normal distribution $\boldsymbol{z} \sim \boldsymbol{N}(0, 1)$. The Xavier initialization method (Kumar, 2017) was applied to initialize the weight matrix of the network. The optimizer applied to

TABLE 1 Validation performance of the 3D-MotNet for 3D human pose estimation on data set 1.

H	Head	Neck	L. shoulder	R. shoulder	L. elbow	R. elbow	L. wrist	R. wrist	L. hip	R. hip	L. knee	R. knee	L. ankle	R. ankle	Total (mm)
3	6.1	5.2	5.8	6.6	5.9	3.1	3.8	2.9	3.0	4.7	3.8	6.2	3.0	3.9	64.0
4	**2.8**	**2.9**	**3.6**	**5.2**	**4.1**	**2.1**	**3.1**	**1.2**	**2.3**	**1.9**	**1.9**	**4.5**	**3.1**	**3.1**	**41.8**
5	4.8	4.8	4.7	5.9	5.0	2.8	3.5	1.9	3.0	1.8	3.4	4.9	3.1	3.2	52.8
6	2.9	3.0	3.7	5.0	4.3	2.1	3.0	1.1	2.4	1.8	2.2	4.6	3.0	2.9	42
7	3.3	3.1	3.8	5.1	4.2	2.6	3.2	1.3	2.4	1.9	2.3	4.4	3.0	3.0	43.6
3DM	4.1	3.3	4.8	5.7	5.1	3.0	5.2	3.1	2.5	2.9	2.5	4.7	3.6	3.7	54.2

Note: 3DM stands for the 3D-MotNet pretrained by two public data sets (Common Objects in Context [COCO] and Human3.6 M).
Bolded terms indicate the model (3D-MotNet)'s optimal performance, which is achieved when H equals 4.

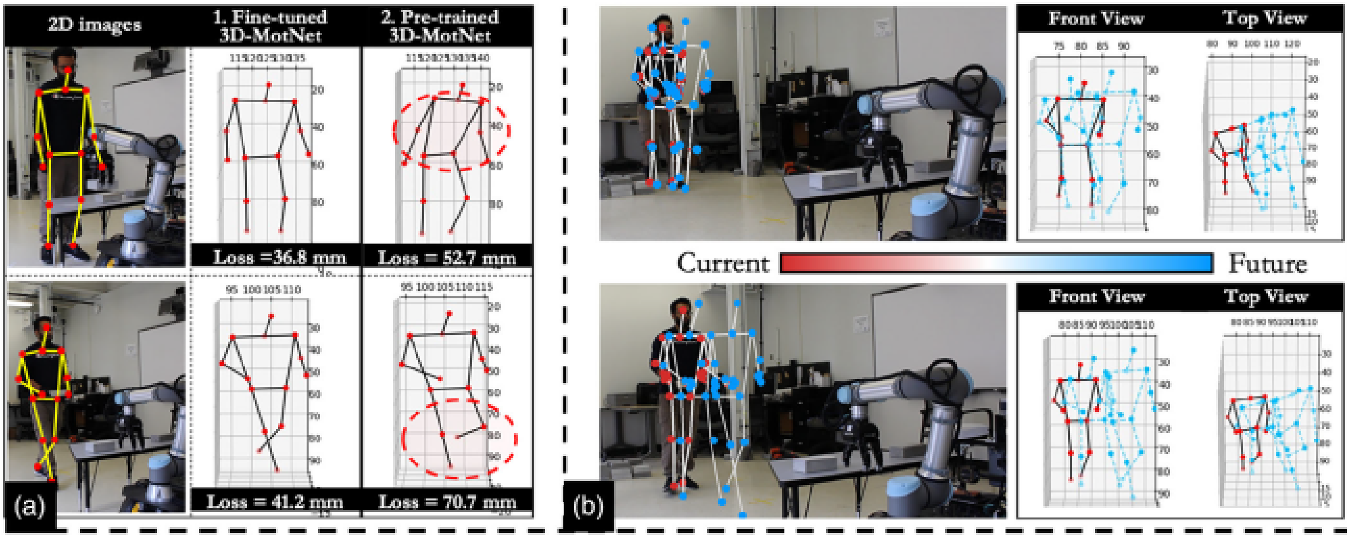


FIGURE 7 (a) Outcomes of the 3D-MotNet in 3D pose estimation; (b) outcomes of the IntentionNet in motion intention estimation.

train each part of the IntentionNet (VAE model and FC neural network; Section 3.2) was adaptive moment estimation (ADAM). The learning rate of the VAE model was set as 2.5×10^{-5} with a 10^{-4} decay. The batch size was 32. Meanwhile, the FC neural network was trained with a learning rate of 1×10^{-3} without weight decay. The corresponding batch size was 1024, and a dropout rate of 0.15 was applied. As mentioned in Section 3.2, the 3D-MotNet was applied to generate the initial pose predictions for the FC neural network of the IntentionNet. The training process of the IntentionNet utilized the fine-tuned 3D-MotNet. With these settings, IntentionNet was trained to predict the 3D postures of workers over successive T frames in the future. The optimal value of T for the IntentionNet was determined by training the network with different T values. The range of T was set as 1 to 6, with an interval of 1.

Table 2 presented the overview of the IntentionNet's performance, where each row corresponded to a value of T. The authors reported the performances (errors) of the IntentionNet in the motion goal prediction (VAE model)

TABLE 2 Performance of the IntentionNet for human motion inference on data set 2.

T	Motion goal prediction error (pixel)—VAE model loss	3D human pose (all body joints) prediction error (mm)—FC network loss
1	59	161.5 (average loss per human pose: 80.75)
2	**24**	**145.3 (average loss per human pose: 48.43)**
3	89	345.9 (average loss per human pose: 86.48)
4	106	505.9 (average loss per human pose: 101.18)
5	210	699.2 (average loss per human pose: 116.53)
6	386	922.1 (average loss per human pose: 131.72)

Abbreviations: VAE, variational autoencoder; FC, fully connected.
Bolded terms indicate the model (IntentionNet)'s optimal performance, which is achieved when T equals 2.

as well as 3D human pose prediction corresponding to the motion goal (two-part FC neural network). When T was 2, the IntentionNet had the optimal performance of both. This performance was attributed to two factors: first, setting T to a value of 1 proved insufficient for IntentionNet to acquire adequate temporal information about human motion, thereby hindering its ability to accurately predict motion intention. Second, when T exceeded a value of 2, IntentionNet could have an enhanced ability to capture features for motion inference (i.e., capture more temporal features). However, increasing the number of frames (i.e., increasing T value) could potentially result in a higher cumulative error for IntentionNet as it learned features from human motion information in each subsequent frame. The accumulation of errors significantly degraded the effectiveness of the IntentionNet in predicting motion goals and corresponding 3D human poses, diminishing the overall reliability of the method in motion intention inference in real-time applications. In this study, T was 2 allowed the IntentionNet to have the ideal balance, enabling IntentionNet to effectively learn informative features while maintaining minimal errors when predicting human motion intentions. In other words, the trained IntentionNet had the optimal performance in predicting the motion intention of workers in consecutive two frames of images in the future. Given the sampling rate of data set 2, each increase of T by 1 corresponds to IntentionNet's capability to predict the intended motion of workers by an increment of 0.5 s. Figure 7b illustrates the performance of the trained IntentionNet in predicting the motion intentions (intended motion sequences) of the subject in the training session for the collaborative bricklaying tasks. The human motion intention predicted by the IntentionNet was consistent with the ground-truth of 1 s (2 × 0.5) into the future. This result also indicated that the proposed IAPMP, equipped with the trained IntentionNet, could effectively anticipate the human motion intention 1 s in advance during human–robot interactions.

5.3 | Performance of the IAPMP in generating collision-free trajectories for collaborative robots

After obtaining the trained networks (3D-MotNet and IntentionNet) with optimal parameters, the authors integrated these models with the collision-checking motion planning mechanism to generate the IAPMP. This IAPMP was applied to the testing session of the collaborative bricklaying task. Throughout the testing session, the effectiveness of the IAPMP was assessed by its ability in empowering the robot to make collision-free trajectories. Figure 8a shows the overall performance of the IAPMP across each subject participating in the testing session, measured by success rate. The success rate was quantified by the rounds of the task where the IAPMP enabled the robot to make timely collision-free adjustments. To be more specific, in a round of the task, if the IAPMP guided the robot to timely adjust its trajectory without collisions, it was labeled a success. Conversely, task rounds where the IAPMP failed to make the robot generate such adjustments were categorized as "failures." The success rate was calculated as the ratio of successful rounds to the total task rounds (20). As shown, across all participants, the IAPMP achieved an average success rate of 94.2% in allowing the robot to generate collision-free adjustments during the collaboration. In other words, in 5.8% of the 120 (= 20 rounds/subject × 6 subjects) total rounds involving participants, the IAPMP could not enable the robot to make proper adjustments. A more in-depth discussion of these failure cases will be detailed in Section 6. Moreover, during the testing session, the proposed IAPMP (using parameters H = 4 in 3D-MotNet, $T = 2$ in IntentionNet, and $\delta = 5\%$ in the collision-checking mechanism) took an average time of 242 ms to estimate the motion intentions of subjects and accordingly enable the robot to generate the collision-free adjustments. Specifically, it took about 109 ms per frame for intention estimation. Following this, based on the predicted intention, the IAPMP took approximately 133 ms to calculate the robot's optimal trajectory and allow the robot to physically adjust the trajectory. This processing speed of the IAPMP ensured timely robot adjustments.

Figure 8b showed a successful round in the testing session of the collaborative bricklaying tasks. In Figure 8b.1, a series of images sequentially showed a representative round of the task captured by the camera. From left to right, we observed subject S_4 lifting the brick, delivering it to the collaborative robot, and subsequently, the robot picking up the brick to build the construction wall. Rows from Figure 8b.2 to Figure 8b.3 showed the corresponding outcomes generated by the IAPMP in the analytical space. For each captured image, the IAPMP first extracted the 3D human pose from the scene based on the fine-tuned 3D-MotNet. Simultaneously, leveraging the collision-checking motion planning mechanism, the IAPMP estimated the volume occupied by the human subject based on the extracted 3D human pose, as shown in Figure 8b.2. Next, the IntentionNet in IAPMP predicted the motion sequences of the subject in the subsequent $T = 2$ frames (Figure 8b.2). Similarly, using the collision-checking motion planning mechanism, the IAPMP extended its estimation to the volume occupied by the human subject in the future two frames, relying on the inferred motion sequences of 3D human poses. Correspondingly, the occupied volume of the applied UR5e

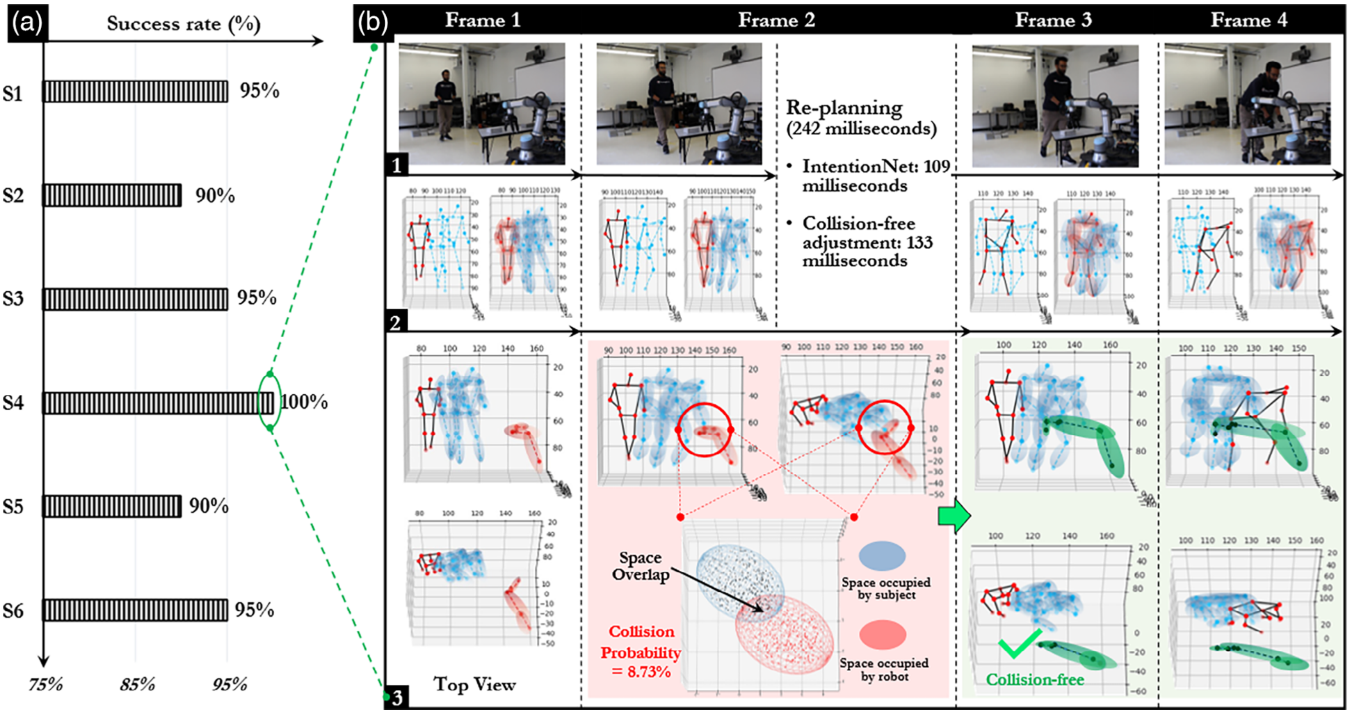


FIGURE 8 (a) The success rate of intention-aware probabilistic motion planner (IAPMP) in generating collision-free adjustments for the collaborative robot under different subjects during the testing session; (b) an example of IAPMP successfully facilitating collision-free human–robot collaboration in the testing session of the collaborative bricklaying task.

robotic arm was gauged along its trajectory (Figure 8b.3). The spatial relationship between the human subject and the robot was also visualized in Figure 8b.3. Then, the robot computed the collision probability of its preplanned path with the subject's current and future positions. The red circle of Figure 8b.3 presented that, according to the preplanned path, the robot had an 8.73% likelihood of collision with the future position of the subject. This value exceeded the threshold (5%) mentioned in Section 3.3. Hence, the IAPMP replanned the trajectory of the robot, enabling it to generate a new, collision-free trajectory (green area of Figure 8b.3) for picking up the brick and constructing the wall without interrupting the collaboration (frame 4 in Figure 8b). According to the above steps, the IAPMP demonstrated its ability to allow the robot to avoid collisions with workers during HRC.

6 | DISCUSSION

Based on the reported results, the developed IAPMP, consisting of three modules, demonstrated high potential in enabling the robot to estimate worker poses in 3D space, anticipate worker motion intentions, and generate collision-free trajectories in alignment with perceived motion intentions and corresponding collision probabilities. To further investigate the effectiveness of the IAPMP, the authors undertook several more in-depth investigations. The associated findings will be elaborated upon in the following paragraphs.

For the 3D-MotNet, the authors compared its performance with five well-known deep networks, including Martinez, Hossain, et al. (2017), Tekin et al. (2017), Mehta et al. (2018), Wang et al. (2019), and Rogez et al. (2019). All these competing networks were known for their efficient performances in extracting 3D human poses from 2D images. For each network, the authors downloaded its pretrained model and subsequently fine-tuned it using the training data set from data set 1. The validation data set of data set 1 was applied to evaluate both the accuracy and computational speed of these models in estimating the 3D postures of subjects while performing construction tasks. Table 3 reports a comparison of 3D pose estimation loss between the competing models and 3D-MotNet ($H = 4$). Notably, 3D-MotNet excelled, particularly in estimating the 3D positions of the left knee and upper arm joints. Additionally, Table 4 compares the computational speeds for 3D pose estimations between 3D-MotNet and the competing models. This table shows that 3D-MotNet ($H = 4$) offers a competitive computational speed, ranking second-best among the evaluated models. The overall performance of 3D-MotNet demonstrates its effectiveness in 3D pose estimation within the context of construction tasks.

TABLE 3 Performance comparison (loss) of 3D-MotNet and competing models on data set 1 for 3D human pose estimation.

Joints model	Head	Neck	L. shoulder	R. shoulder	L. elbow	R. elbow	L. wrist	R. wrist	L. hip	R. hip	L. knee	R. knee	L. ankle	R. ankle	Total loss (mm)
Martinez, Hossain, et al., 2017	4.1	3.8	4.2	4.9	3.8	2.7	5.1	2.9	2.4	2.4	4.3	5.8	4.1	7.3	57.8
Tekin et al., 2017	3.6	4.7	4.4	3.8	4.6	4.7	3.6	2.1	2.8	2.3	3.9	3.6	4.2	3.8	52.1
Ours: 3D-MotNet	**2.8**	**2.9**	**3.6**	**5.2**	**4.1**	**2.1**	**3.1**	**1.2**	**2.3**	**1.9**	**1.9**	**4.5**	**3.1**	**3.1**	**41.8**
Mehta et al., 2018	3.4	2.9	4.7	4.7	3.9	3.1	2.9	1.6	2.3	2.2	2.7	5.0	2.9	3.3	45.6
Wang et al., 2019	1.9	4.6	4.3	4.9	4.3	4.2	2.8	2.2	1.9	1.8	2.9	3.1	3.8	3.6	46.3
Rogez et al., 2019	3.1	1.8	4.1	3.9	3.7	3.2	3.2	3.1	2.3	1.9	2.1	3.2	3.6	3.5	42.7

Bolded terms indicate the model's optimal performance in 3D human pose estimation.

TABLE 4 Performance comparison (computational time) of 3D-MotNet and competing models on data set 1 for 3D human pose estimation.

Model	Computational speed (ms) for 3D pose estimation
Martinez, Hossain, et al., 2017	49
Tekin et al., 2017	75
Ours: 3D-MotNet	54
Mehta et al., 2018	77
Wang et al., 2019	59
Rogez et al., 2019	65

Terms in bold signify the model proposed by the authors. If the formatting leads to confusion.

The authors then conducted an ablation study to perform a detailed analysis of the IntentionNet. In the domains of AI and deep learning, an ablation study will provide insights into the individual contributions of various components within a deep network (Hu & Li, 2022; Wu et al., 2023). This is achieved by systematically removing specific components and analyzing the resultant impact on the overall performance of the network. As introduced in Section 3.2, the IntentionNet comprises three key components—the VAE model, fine-tuned 3D-MotNet, and a two-part FC neural network (Figure 3b). These components were systematically integrated to empower the IntentionNet to predict the motion intention of subjects during the collaboration. Within this framework, the VAE model was applied to predict the future goal position of the subject. The comprehensive configuration (layer, architecture, objective function) of the VAE model has been well studied (Cao et al., 2020; Dittadi et al., 2021), and its incorporation was necessary for the motion intention prediction capabilities of the IntentionNet. Then, to further check the superiority of the two-part FC neural network and the fine-tuned 3D-MotNet in enabling the IntentionNet to predict human motion intentions, the authors conducted an ablation study in two steps. First, the authors changed the setting by replacing the fine-tuned 3D-MotNet with the pretrained 3D-MotNet in the IntentionNet. The IntentionNet with the modified setting was denoted as IN_M; the original was IN_O. Figure 9a shows the comparison results between IN_M and IN_O (as per T) in predicting the motion intentions of subjects on data set 2. The results highlighted a substantial 6% enhancement in the optimal performance of the IN_O compared to IN_M. This emphasized the importance of the fine-tuning process for the pretrained 3D-MotNet in enhancing the performance of IntentionNet in motion intention inference. Second, the authors disabled the function of the upper part of the FC neural network structure (Section 3.2). Without the upper

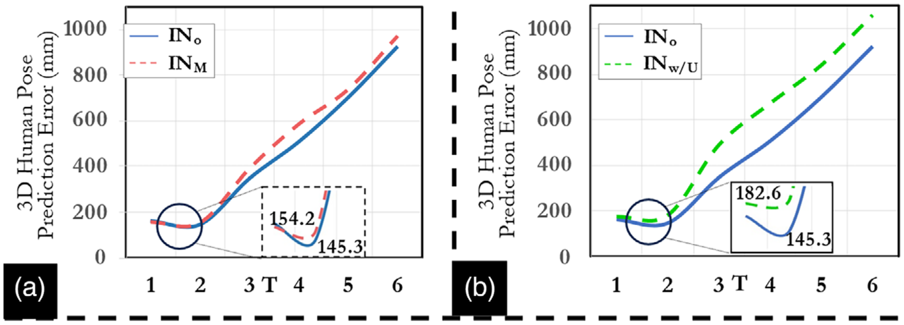


FIGURE 9 (a) Comparison between IN_M and IN_O in predicting human motion intentions on data set 2; (b) comparison between $IN_{w/U}$ and IN_O in predicting motion intentions on data set 2.

part structure, the FC network in the IntentionNet directly converted the 3D human poses from the 2D human poses, which was unable to be corrected in the 2D space using the loss function, Equation (6). The IntentionNet without the upper part structure of the FC network was denoted as $IN_{w/U}$. Figure 9b shows the performance of the $IN_{w/U}$ on the validation data set of data set 2 according to each T. As seen, compared with the optimal performance of the IN_O reported in Section 5.2, the optimal performance of the $IN_{w/U}$ degraded by around 25%. This comparison demonstrated the effectiveness of the current structure of the FC neural network in allowing the IntentionNet to predict human motion intention during HRC. All in all, according to the above ablation study for IntentionNet, the current settings presented in Section 3.2 enabled the IntentionNet to have the optimal performance in predicting the motion intention of subjects. Despite this efficiency, the authors recognized the room for improvement. Currently, the IntentionNet was designed to predict human motion intentions (represented by sequences of human movements) based on motion patterns from the training data set, data set 2. However, the IntentionNet might struggle to anticipate human movements absent from the training data set. Such an inability to predict unfamiliar movements could degrade its motion intention estimation, and impede the capacity of the IAPMP in generating suitable robot adjustments in practice, especially given the diverse motion patterns workers might exhibit when collaborating with robots. Further research should aim to refine the capability of the IntentionNet to better predict unfamiliar human movements/actions.

Next, the authors discussed two critical aspects of the proposed IAPMP in enabling the collaborative robot to generate collision-free trajectories: (1) the rationale behind the selection of the probabilistic collision-checking threshold, δ, for the IAPMP; and (2) the insights behind instances where the IAPMP failed to facilitate the robot in generating collision-free adjustments during the bricklaying task. Regarding the first aspect, the threshold δ emerged as a key factor impacting the computational efficiency of the proposed IAPMP. Figure 10a illustrates a comparison between the computational time of IAPMP allowing the robot to generate the collision-free adjustment in the collaborative task under varying δ thresholds—0.5%, 1%, 2%, 3%, 4%, 5%, 8%, 10%, 15%, and 20%. The observations suggested that a larger threshold corresponded to a reduced computational time. This is because a larger threshold imposed fewer spatial constraints on the space around the subject. With fewer space constraints, the proposed IAPMP, equipped with Algorithm 2, empowered the robot to find a suitable trajectory faster. However, it is crucial to note the inherent trade-off: While a larger threshold reduced computational time, it concurrently increased the potential collision hazard. To highlight this trade-off between computational efficiency and safety, Figure 10b reported the reliability of the robotic adjustments generated by the IAPMP across varied δ values. The measure of reliability was based on the actual success of the adjustments in avoiding collision outcomes. To be more specific, if a trajectory, as generated by the IAPMP, resulted in a collision (with the intended motion of subjects), it received a reliability score of 0; otherwise, the reliability was scored as 100%. For each δ value, the overall reliability of the adjustments generated by the IAPMP was computed as an average of 30 tests. In these tests, for each δ value, the authors used IAPMP to generate trajectories for robotic adjustments. Subsequently, the generated trajectories were assessed for collisions with the subject's intended location predicted by the IntentionNet. The safety precautions mentioned in Section 4.3 were conducted during these tests. Notably, adjustments generated at δ values exceeding 5% exhibited a sharp decline in their reliability (Figure 10b). Collating insights from Figure 10a,b, the 5% threshold was identified as the balance point, optimizing the computational efficiency of the IAPMP and safety during the bricklaying tasks. Therefore, this 5% value was chosen for δ in this study.

Regarding the second aspect, as indicated in Section 5, the IAPMP achieved a 94.2% success rate. However, there

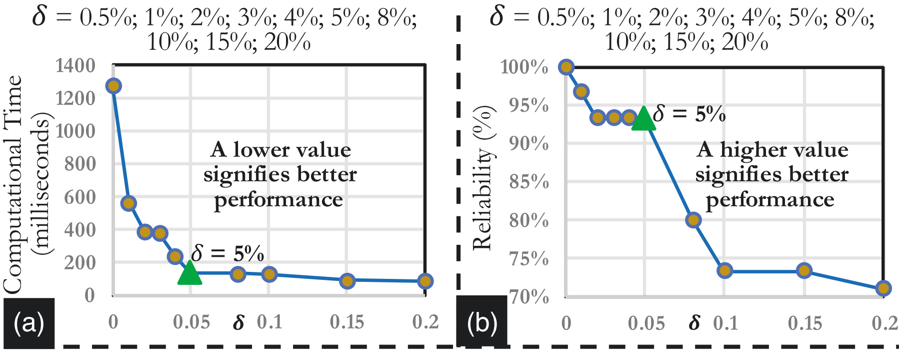


FIGURE 10 (a) Average computational time of the intention-aware probabilistic motion planner (IAPMP) in allowing robots to generate collision-free adjustment across varying δ; (b) reliability of the IAPMP in generating collision-free adjustment for the robot across varying δ.

remained 5.8% of cases (seven cases) where the IAPMP failed to produce collision-free adjustments. Feedback from the "on-site supervisors" mentioned in Section 4.3 identified three reasons for these failures. The first was errors in human motion prediction, accounting for five of the seven failure instances. Specifically, during the testing session, the trained 3D-MotNet and IntentionNet occasionally experienced inaccuracies in 3D pose estimations and subsequent motion intention inferences. Such errors resulted in the inaccurate representation of the position of the subject, leading to incorrect collision probability calculations, which in turn prevented the robot from accurately selecting the collision-free trajectory during the collaboration. The second reason was the time-lag in collision-free adjustments, which caused one of the seven observed failures. During the collaboration, while the deep networks in IAPMP enabled the robot to anticipate the subject's motion, the IAPMP sometimes lagged in generating the optimal trajectories for the robot. Such latency could be attributed to the limited computational resources of the robot, resulting in unsuccessful robotic adjustments during the testing session. The third reason, responsible for one among the seven instances, was a connectivity issue in the applied robot. In this case, the connection between the operating system and the robotic arm was disrupted. Thus, even though the IAPMP facilitated an appropriate trajectory, the robot could not execute the adjustment in the real-world setting. Though the IAPMP demonstrated promise in this study, these observed failures emphasize the importance of refining its robustness in practice. Potential directions for enhancement will be elaborated upon in Section 7.

The above investigations provide insights for a better understanding of the structure, parameter settings, as well as the testing performance of the proposed IAPMP. Leveraging the current settings and structure, the IAPMP enables the robot to efficiently avoid collisions with its human collaborators in collaborative bricklaying tasks. It is worth noting that while the proposed study emphasized bricklaying tasks, the IAPMP's functionality holds promise for various established human–robot collaborative tasks in the civil and construction domains. For instance, it could seamlessly extend to tasks like timber assembly (Kramberger et al., 2022), collaborative welding (Brosque et al., 2020), and collaborative polishing (Gaz et al., 2018). In these applications, the IAPMP would enable the robot to proactively anticipate the intended motion of its human collaborators, calculate the collision probability with human collaborators, and generate collision-free and space-saving trajectories accordingly. In turn, this intelligent robot adjustment can increase the human collaborators' trust in the robot (Schaefer et al., 2016). In the long run, the proposed IAPMP can harmonize the collaboration between workers and robots, thereby contributing to the implementation of collaborative robots in the civil and construction domains.

7 | CONCLUSION

In this study, the authors introduced the IAPMP, an innovative system designed for collaborative robots operating in construction and civil environments. The IAPMP uniquely possesses dual functionalities: anticipating human motion intentions and gauging collision probability between robots and workers. Utilizing visual data from 2D cameras, IAPMP empowers robots with the ability to anticipate the workers' intended motion and associated collision risks, optimizing trajectories for safe worker–robot interaction. When compared to leading intention-aware collision avoidance methods in manufacturing, IAPMP's intention estimation capability enables it to leverage readily available field-sensing devices, such as 2D cameras, to assess workers' intended motions. Furthermore, unlike most worker–robot collision avoidance systems that rely on fixed separation distance, IAPMP adopts a probabilistic approach for collision avoidance. This enables the collaborative robots to make collision-free adjustments, ensuring continuous and smooth operation. A collaborative bricklaying task was conducted to verify the feasibility of the IAPMP. In sum, the IAPMP offers a promising way to establish technically viable and reliable HRC solutions in construction and civil engineering. Furthermore, this study produced advanced knowledge, along with a physical demonstration that elucidates both reactive and predictive aspects of workers' interaction with construction robots, which facilitates the safe implementation of collaborative construction robots.

Despite the potential of the proposed method, the current IAPMP has certain limitations that hinder its efficient deployment and operation in actual sites. In the future, the following limitations need to be addressed. First, since the case study was conducted in a well-controlled laboratory environment, the quality of the image captured by the RGB camera could be maintained at a high level. However, practical scenarios introduce image quality variations due to factors such as dust, site lighting conditions, as well as construction conditions. The fluctuations in image quality may degrade the efficiency of the proposed IAPMP in assessing 3D human pose from 2D images and predicting the motion intentions of workers. To mitigate this problem, a real-time, lightweight image enhancement algorithm (Zhang et al., 2021) could be integrated into the proposed IAPMP, allowing the robot to have stable and efficient performance in inferring the motion intentions of workers from captured images in practice. Second, expanding on the first point, to enhance the reliability of the proposed method in practice, future research can focus on integrating supplementary sensors into the current system, thereby evolving toward a multisensory-based, intention-aware worker–robot coadaptation system. The inclusion of additional sensors, such as LiDAR, can further enhance the robotic perception for surrounding environments. This enhanced system would improve the robot's operational reliability. For instance, in scenarios where vision-based systems fail to perform efficiently (dimly lit environment), distance sensors can provide the robot with crucial information about the proximity of workers, preventing potential accidents. In instances where workers might engage in unpredictable actions, especially during emergency situations that the IntentionNet in IAPMP cannot predict in a timely manner, these supplementary sensors (like distance sensors) are important. They can enable robots to quickly measure the distance to workers and execute emergency measures, such as an immediate stop, enhancing overall safety. Paired with the system, future research should consider developing a series of error and failure feedback mechanisms. Such mechanisms would enable the enhanced system to learn from and rectify errors and failures made during human–robot interactions in practice, greatly improving its reliability during continuous operations on real-world sites. Third, in the proposed IAPMP, the robot was programmed to calculate the collision probability with workers' future positions and generate collision-free adjustments in real time, which requires significant computational resources. When the robot needs to compute multiple potential adjustments to determine a feasible collision-free adjustment, this online computation may cause a delay in adjustment. In this case, the authors propose integrating a reinforcement learning (RL) step (Markov decision process in conjunction with the Q-learning algorithm; Park et al., 2019) into the IAPMP. This RL step would enable the IAPMP to generate a decision-making "policy" (Kurniawati et al., 2011) for the robot. With this policy, the robot could directly generate a suitable adjustment based on the RL results, thereby reducing the use of computational resources and time delay in generating (calculating) collision-free adjustments. These three directions can improve the error management, robustness, and computational efficiency of the proposed IAPMP, enhancing its suitability for continuous operations on actual construction sites. Fourth, the current testing of the IAPMP was limited to a single task, the collaborative bricklaying. This cannot comprehensively validate the applicability and reliability of the method. Thus, future research should evaluate the proposed IAPMP across diverse real-world collaborative tasks, such as timber assembly and collaborative welding tasks, as mentioned in Section 6. For each distinct task, the settings of the parameters in the IAPMP should be calibrated and assessed. The IAPMP should also be evaluated under a broader range of worker movements/actions not covered in this study, such as running or behaviors workers might exhibit in unforeseen scenarios, to determine if the IAPMP can promptly generate the appropriate robotic adjustments. Future assessments should include robots with a range of structures, configurations, and specifications. Testing with such a diverse array of robotic systems will offer a deeper understanding of the real-world scalability of the proposed method. Such rigorous testing would validate the effectiveness of the method and highlight potential areas for enhancement, further improving the robustness of the proposed method. Additionally, the current IAPMP focuses on the collaboration between a single robot and a single worker, which cannot be applied to the collaborations of multiple workers and robots. Therefore, future research can investigate how to expand the current methodology to accommodate the multiagent framework for human–robot interaction. Beyond the limitations and future directions related to the technical aspects, it is essential for upcoming research to delve into detailed recommendations for workers collaborating with robots. Future investigations can focus on developing essential tools (e.g., training modules) and establishing clear guidelines that safeguard the safety, well-being, and efficiency of workers in these collaborative environments. In sum, by focusing on these aspects, the practical application of the method can be significantly enhanced, ensuring a more effective implementation in real-world settings.

ACKNOWLEDGMENTS

The work presented in this paper was supported financially by two National Science Foundation Awards (No. 2222654 and No. 2221167). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Adamczyk, J., & Malawski, F. (2021). Comparison of manual and automated feature engineering for daily activity classification in mental disorder diagnosis. *Computing and Informatics*, *40*(4), 850–879.

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., & Khatib, O. (2018). Progress and prospects of the human–robot collaboration. *Auton Robots*, *42*(5), 957–975.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*(12), 8675–8690.

Altobelli, F., Taylor, H. F., & Bernold, L. E. (1993). Prototype robotic masonry system. *Journal of Aerospace Engineering*, *6*(1), 19–33.

Balan, L., & Bone, G. (2006). Real-time 3D collision avoidance method for safe human and robot coexistence. In *2006 IEEE/RSJ international conference on intelligent robots and systems* (pp. 276–282). IEEE. https://doi.org/10.1109/IROS.2006.282068

Bauer, A., Wollherr, D., & Buss, M. (2008). Human-robot collaboration: A survey. *International Journal of Humanoid Robotics*, *05*(01), 47–66.

Bock, T., & Linner, T. (2016). Single-task construction robots by category. In Tomas Bock, Tomas Linner (Eds.), *Construction robots: elementary technologies and single-task construction robots* (pp. 14–290). Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9781139872041.002

Brosque, C., Galbally, E., Khatib, O., & Fischer, M. (2020). Human-robot collaboration in construction: Opportunities and challenges. In *2020 International congress on human-computer interaction, optimization and robotic applications (HORA)* (pp. 1–8). IEEE. https://doi.org/10.1109/HORA49412.2020.9152888

Cao, Z., Gao, H., Mangalam, K., Cai, Q.-Z., Vo, M., & Malik, J. (2020). Long-term human motion prediction with scene context. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. M. (eds), *Computer Vision ECCV 2020. ECCV 2020. Lecture Notes in Computer Science*, vol 12346. Springer, Cham. https://doi.org/10.1007/978-3-030-58452-8_23

Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2021). OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(1), 172–186.

Cao, Z., Simon, T., Wei, S.-E., & Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. In Rama Chellappa, Zhengyou Zhang, Anthony Hoogs (eds.), *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1302–1310). IEEE.

Casas, S., Luo, W., & Urtasun, R. (2021). IntentNet: Learning to predict intention from raw sensor data.

Chan, F. K. (2008). *Spacecraft collision probability*. American Institute of Aeronautics and Astronautics, Inc.

Dittadi, A., Dziadzio, S., Cosker, D., Lundell, B., Cashman, T., & Shotton, J. (2021). Full-body motion from a single head-mounted device: Generating SMPL poses from partial observations. In Tamara Berg, James Clark, Yasuyuki Matsushita & Camill J. Taylor (eds.), *2021 IEEE/CVF international conference on computer vision (ICCV)* (pp. 11667–11677). IEEE.

Du Toit, N. E., & Burdick, J. W. (2011). Probabilistic collision checking with chance constraints. *IEEE Transactions on Robotics*, *27*(4), 809–815.

Fang, Q., Li, H., Luo, X., Li, C., & An, W. (2020). A sematic and prior-knowledge-aided monocular localization method for construction-related entities. *Computer-Aided Civil and Infrastructure Engineering*, *35*(9), 979–996.

Fragkiadaki, K., Levine, S., Felsen, P., & Malik, J. (2015). Recurrent network models for human dynamics. In Ruzena Bajcsy, Greg Hager & Yi Ma (eds.), *2015 IEEE international conference on computer vision (ICCV)* (pp. 4346–4354). IEEE.

Gaz, C., Magrini, E., & De Luca, A. (2018). A model-based residual approach for human-robot collaboration during manual polishing operations. *Mechatronics*, *55*, 234–247.

Goody, E. N. (1995). *Social intelligence and interaction: Expressions and implications of the social bias in human intelligence*. Cambridge University Press.

Haddadin, S., & Croft, E. (2016). Physical human–robot interaction. In Siciliano, B., & Khatib, O. (eds), *Springer Handbook of Robotics. Springer Handbooks*. Springer, Cham. https://doi.org/10.1007/978-3-319-32552-1_69

Haddadin, S., Haddadin, S., Khoury, A., Rokahr, T., Parusel, S., Burgkart, R., Bicchi, A., & Albu-Schäffer, A. (2012). On making robots understand safety: Embedding injury knowledge into control. *International Journal of Robotics Research*, *31*(13), 1578–1602.

Haghighat, A., & Sharma, A. (2023). A computer vision-based deep learning model to detect wrong-way driving using pan–tilt–zoom traffic cameras. *Computer-Aided Civil and Infrastructure Engineering*, *38*(1), 119–132.

Holmström, E., & Ahlborg, B. (2005). Morning warming-up exercise—Effects on musculoskeletal fitness in construction workers. *Applied Ergonomics*, *36*(4), 513–519.

Hu, D., & Li, S. (2022). Recognizing object surface materials to adapt robotic disinfection in infrastructure facilities. *Computer-Aided Civil and Infrastructure Engineering*, *37*(12), 1521–1546.

Hu, N., Bestick, A., Englebienne, G., Bajscy, R., & Kröse, B. (2016). Human intent forecasting using intrinsic kinematic constraints. In *IEEE international conference on intelligent robots and systems, 2016-Novem* (pp. 787–793). IEEE. https://doi.org/10.1109/IROS.2016.7759141

Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). 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.

Jang, K., An, Y., Kim, B., & Cho, S. (2021). Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot. *Computer-Aided Civil and Infrastructure Engineering*, *36*(1), 14–29.

Khajwal, A. B., Cheng, C., & Noshadravan, A. (2023). Post-disaster damage classification based on deep multi-view image fusion. *Computer-Aided Civil and Infrastructure Engineering*, *38*(4), 528–544.

Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational dropout and the local reparameterization trick. Proceedings of the 28th International Conference on Neural Information Processing Systems – Volume 2, NIPS'15, 2575–2583. Cambridge, MA, USA: MIT Press.

Kirschner, R. J., Jantalia, J., Mansfeld, N., Abdolshah, S., & Haddadin, S. (2021). CSM: Contact sensitivity maps for benchmarking robot collision handling systems. In *2021 IEEE international conference on robotics and automation (ICRA)* (pp. 3590–3596). IEEE. https://doi.org/10.1109/ICRA48506.2021.9561528

Kramberger, A., Kunic, A., Iturrate, I., Sloth, C., Naboni, R., & Schlette, C. (2022). Robotic assembly of timber structures in a human-robot collaboration setup. Front Robot AI, *8*, 768038. https://doi.org/10.3389/frobt.2021.768038

Kratzer, P., Midlagajni, N. B., Toussaint, M., & Mainprice, J. (2020). Anticipating human intention for full-body motion prediction in object grasping and placing tasks. In Silvia Rossi & Adriana Tapus (eds.), *2020 29th IEEE international conference on robot and human interactive communication (RO-MAN)* (pp. 1157–1163). IEEE.

Kulić, D., & Croft, E. A. (2005). Safe planning for human-robot interaction. *Journal of Robotic Systems*, *22*(7), 383–396.

Kumar, S. K. (2017). On weight initialization in deep neural networks. ArXiv, arXiv:1704.08863.

Kurniawati, H., Du, Y., Hsu, D., & Lee, W. S. (2011). Motion planning under uncertainty for robotic tasks with long time horizons. *International Journal of Robotics Research*, *30*(3), 308–323.

Lasota, P. A., & Shah, J. A. (2015). Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *57*(1), 21–33.

Lee, S., & Adams, T. M. (2004). Spatial model for path planning of multiple mobile construction robots. *Computer-Aided Civil and Infrastructure Engineering*, *19*(4), 231–245.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2014). Microsoft COCO: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T. (eds), *Computer Vision - ECCV 2014. ECCV 2014. Lecture Notes in Computer Science*, vol *8693*. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, H., & Wang, L. (2021). Collision-free human-robot collaboration based on context awareness. *Robotics and Computer-Integrated Manufacturing*, *67*, 101997.

Liu, M., Hong, D., Han, S., & Lee, S. (2016). Silhouette-based on-site human action recognition in single-view video. In José L. Perdomo-Rivera, Antonio Gonzáles-Quevedo, Carla López del Puerto, Francisco Maldonado-Fortunet, & Omar I. Molina-Bas (eds.), *Construction research congress 2016* (pp. 951–959). American Society of Civil Engineers.

Liu, Y., Habibnezhad, M., Jebellli, H., & Monga, V. (2021). Worker-in-the-loop cyber-physical system for safe human-robot collaboration in construction. In Raymond Issa, J.D., P.E. (ed.), *International conference on computing in civil engineering*. pp. 1075–1083 American Society of Civil Engineers.

Liu, Y., & Jebelli, H. (2022). Intention estimation in physical human-robot interaction in construction: Empowering robots to gauge workers' posture. In Farrokh Jazizadeh, Tripp Shealy & Michael J. Garvin (eds.), *Construction research congress 2022* (pp. 621–630). American Society of Civil Engineers.

Liu, Y., Ojha, A., Shayesteh, S., Jebelli, H., & Lee, S. (2022). Human-centric robotic manipulation in construction: Generative adversarial networks (GAN) based physiological computing mechanism to enable robots to perceive workers' cognitive load. *Canadian Journal of Civil Engineering*, *50*(3), 224–238.

Lu, L., & Dai, F. (2023). Automated visual surveying of vehicle heights to help measure the risk of overheight collisions using deep learning and view geometry. *Computer-Aided Civil and Infrastructure Engineering*, *38*(2), 194–210.

Luo, R., Hayne, R., & Berenson, D. (2018). Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces. *Autonomous Robots*, *42*(3), 631–648.

Luo, R. C., & Mai, L. (2019). Human intention inference and on-line human hand motion prediction for human-robot collaboration. In Allison Okamura (ed.), *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5958–5964). IEEE.

Maeda, G., Ewerton, M., Neumann, G., Lioutikov, R., & Peters, J. (2017). Phase estimation for fast action recognition and trajectory generation in human–robot collaboration. *International Journal of Robotics Research*, *36*(13–14), 1579–1594.

Mao, W., Liu, M., & Salzmann, M. (2020). History repeats itself: Human motion prediction via motion attention. In: Vedaldi, A., Bischof, H., Brox, T. & Frahm, J. M. (eds), *Computer Vision ECCV 2020. ECCV 2020. Lecture Notes in Computer Science*, vol 12359. Springer, Cham. https://doi.org/10.1007/978-3-030-58568-6_28

Martinez, J., Black, M. J., & Romero, J. (2017). On human motion prediction using recurrent neural networks.

Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017). A simple yet effective baseline for 3d human pose estimation. In Katsushi Ikeuchi, Gerard Mediono & Marcello Pelillo (eds.), *2017 IEEE international conference on computer vision (ICCV)* (pp. 2659–2668). IEEE.

Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., & Theobalt, C. (2018). Single-shot multi-person 3D pose estimation from monocular RGB. In Andrea Fusiello (ed.), *2018 international conference on 3D vision (3DV)* (pp. 120–130). IEEE.

Mihelj, M., Bajd, T., Ude, A., Lenarčič, J., Stanovnik, A., Munih, M., Rejc, J., & Šlajpah, S. (2019). Homogenous transformation matrices. In Matjaž Mihelj,Tadej Bajd, Aleš Ude, Jadran Lenarčič, Aleš Stanovnik, Marko Munih, Jure Rejc& Sebastjan Šlajpah (eds.), *Robotics* (pp. 11–25). Springer International Publishing.

Mule 135 Owner's Manual. (2018). *Mule 135 owner's manual*. https://www.construction-robotics.com/wp-content/uploads/2019/02/MULE-135-MANUAL.pdf

Pan, X., & Yang, T. Y. (2023). 3D vision-based out-of-plane displacement quantification for steel plate structures using structure-from-motion, deep learning, and point-cloud processing. *Computer-Aided Civil and Infrastructure Engineering*, *38*(5), 547–561.

Park, J. S., & Manocha, D. (2020). Efficient probabilistic collision detection for non-Gaussian noise distributions. *IEEE Robotics and Automation Letters*, *5*(2), 1024–1031.

Park, J. S., Park, C., & Manocha, D. (2019). I-Planner: Intention-aware motion planning using learning-based human motion prediction. *International Journal of Robotics Research*, *38*(1), 23–39.

Patera, R. P. (2001). General method for calculating satellite collision probability. *Journal of Guidance, Control, and Dynamics*, *24*(4), 716–722. https://doi.org/10.2514/2.4771

Petters, S., & Belden, R. (2014). SAM, the robotic bricklayer. *SMART/Dynamics of Masonry*, *1*, 10–14.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Rihani, R. A., & Bernold, L. E. (1994). Computer integration for robotic masonry. *Computer-Aided Civil and Infrastructure Engineering*, *9*(1), 61–67.

Robla-Gomez, S., Becerra, V. M., Llata, J. R., Gonzalez-Sarabia, E., Torre-Ferrero, C., & Perez-Oria, J. (2017). Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access*, *5*, 26754–26773.

Rogez, G., Weinzaepfel, P., & Schmid, C. (2019). LCR-Net++: Multi-person 2D and 3D pose detection in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*, 1146–1161.

Schaefer, K. E. (2016). Measuring trust in human robot interactions: development of the "Trust Perception Scale-HRI". In: Mittu, R., Sofge, D., Wagner, A., Lawless, W. (eds), *Robust Intelligence and Trust in Autonomous Systems*. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7668-0_10

Schneider, S. (1995). Ergonomics: OSHA's draft standard for prevention of work-related musculoskeletal disorders. *Applied Occupational and Environmental Hygiene*, *10*(8), 665–674.

Shim, S., Lee, S., Cho, G., Kim, J., & Kang, S. (2023). Remote robotic system for 3D measurement of concrete damage in tunnel with ground vehicle and manipulator. *Computer-Aided Civil and Infrastructure Engineering*, *38*(15), 2180–2201.

Tadic, V., Toth, A., Vizvari, Z., Klincsik, M., Sari, Z., Sarcevic, P., Sarosi, J., & Biro, I. (2022). Perspectives of RealSense and ZED depth sensors for robotic vision applications. *Machines*, *10*(3), 183.

Tekin, B., Marquez-Neila, P., Salzmann, M., & Fua, P. (2017). Learning to fuse 2D and 3D image cues for monocular body pose estimation. In Katsushi Ikeuchi, Gerard Medioni & Marcello Pelillo (eds.), *2017 IEEE international conference on computer vision (ICCV)* (pp. 3961–3970). IEEE.

Thomaz, A., Hoffman, G., & Cakmak, M. (2016). Computational human-robot interaction. *Foundations and Trends in Robotics*, *4*(2–3), 104–223. https://doi.org/10.1561/2300000049

Vasic, M., & Billard, A. (2013). Safety issues in human-robot interactions. In Rudger Dillmann & Roland Siegwart (eds.), *2013 IEEE international conference on robotics and automation* (pp. 197–204). IEEE.

Vicentini, F., Giussani, M., & Tosatti, L. M. (2014). Trajectory-dependent safe distances in human-robot interaction. In Antoni Grau, Richard Zurawski & Herminio Martinez (eds.), *Proceedings of the 2014 IEEE emerging technology and factory automation (ETFA)* (pp. 1–4). IEEE.

Villani, V., Pini, F., Leali, F., & Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, *55*(June 2017), 248–266.

Vit, A., & Shani, G. (2018). Comparing RGB-D sensors for close range outdoor agricultural phenotyping. *Sensors*, *18*(12), 4413.

Wang, K., Lin, L., Jiang, C., Qian, C., & Wei, P. (2019). 3D human pose machines with self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*, 1069–1082.

Wang, Y., Sheng, Y., Wang, J., & Zhang, W. (2018). Optimal collision-free robot trajectory generation based on time series prediction of human motion. *IEEE Robotics and Automation Letters*, *3*(1), 226–233.

Wang, Y., Ye, X., Yang, Y., & Zhang, W. (2017). Collision-free trajectory planning in human-robot interaction through hand movement prediction from vision. In Michael Mistry & Ales Leonardis (eds.), *2017 IEEE-RAS 17th international conference on humanoid robotics (humanoids)* (pp. 305–310). IEEE.

Wang, Z., Mülling, K., Deisenroth, M. P., Ben Amor, H., Vogt, D., Schölkopf, B., & Peters, J. (2013). Probabilistic movement modeling for intention inference in human-robot interaction. *International Journal of Robotics Research*, *32*(7), 841–858.

Wei, X., Wan, X., Huang, S., & Sun, W. (2017). The application of motion capture and 3D skeleton modeling in virtual fighting. In: Chang, J., Zhang, J., Magnenat Thalmann, N., Hu, S. M., Tong, R. & Wang, W. (eds), *Next Generation Computer Animation Techniques. AniNex 2017. Lecture Notes in Computer Science*, vol 10582. Springer, Cham. https://doi.org/10.1007/978-3-319-69487-0_8

Wu, C., Li, X., Jiang, R., Guo, Y., Wang, J., & Yang, Z. (2023). Graph-based deep learning model for knowledge base completion in constraint management of construction projects. *Computer-Aided Civil and Infrastructure Engineering*, *38*(6), 702–719.

Xu, J., Chen, X., Lan, X., & Zheng, N. (2021). Probabilistic human motion prediction via a Bayesian neural network. In *2021 IEEE international conference on robotics and automation (ICRA)* (pp. 3190–3196). IEEE. https://doi.org/10.1109/ICRA48506.2021.9561665

Xu, X., Holgate, T., Coban, P., & García de Soto, B. (2022). Implementation of a robotic system for overhead drilling operations: A case study of the Jaibot in the UAE. *International Journal of Automation & Digital Transformation*, *1*(1), 19–35.

Xu, Z., Zhang, F., Wu, Y., Yang, Y., & Wu, Y. (2023). Building height calculation for an urban area based on street view images and deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *38*(7), 892–906.

Yan, X., Li, H., Wang, C., Seo, J., Zhang, H., & Wang, H. (2017). Development of ergonomic posture recognition technique based on 2D ordinary camera for construction hazard prevention through view-invariant features in 2D skeleton motion. *Advanced Engineering Informatics*, *34*, 152–163.

Yoo, J., & Langari, R. (2019). A predictive perception model and control strategy for collision-free autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, *20*(11), 4078–4091.

You, S., Kim, J.-H., Lee, S., Kamat, V., & Robert, L. P. (2018). Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments. *Automation in Construction*, *96*, 161–170.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, *31*(7), 1235–1270.

Zhang, Z., Jiang, Y., Jiang, J., Wang, X., Luo, P., & Gu, J. (2021). STAR: A structure-aware lightweight transformer for real-time image enhancement. In Camillo J. Taylor, Tamara Berg, James

Clark & Yasuyuki Matsushita (eds.), *2021 IEEE/CVF international conference on computer vision (ICCV)* (pp. 4086–4095). IEEE.

Zhao, J., Li, Y., Xu, H., & Liu, H. (2019). Probabilistic prediction of pedestrian crossing intention using roadside LiDAR data. *IEEE Access*, *7*, 93781–93790.

Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., Kehtarnavaz, N., & Shah, M. (2023). Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, *56*(1), 1–37.

Zheng, P., Wieber, P.-B., Baber, J., & Aycard, O. (2022). Human arm motion prediction for collision avoidance in a shared workspace. *Sensors*, *22*(18), 6951.

Unhelkar, V. V., Lasota, P. A., Tyroller, Q., Buhai, R.-D., Marceau, L., Deml, B., & Shah, J. A. (2018). Human-aware robotic assistant for collaborative assembly: integrating human motion prediction with planning in time. *IEEE Robot Autom Lett*, *3*(3), 2394–2401.

Zhang, X., Wong, Y., Kankanhalli, M. S., & Geng, W. (2019). "Unsupervised domain adaptation for 3D human pose estimation." Proceedings of the 27th ACM international conference on multimedia, 926–934. New York, NY, USA: ACM. https://doi.org/10.1145/3343031.3351052

How to cite this article: Liu, Y., & Jebelli, H. (2023). Intention-aware robot motion planning for safe worker–robot collaboration. *Computer-Aided Civil and Infrastructure Engineering*, 1–28. https://doi.org/10.1111/mice.13129