buildings

MDPI

Article

Environment-Aware Worker Trajectory Prediction Using Surveillance Camera in Modular Construction Facilities

Qiuling Yang [1], Qipei Mei [2], Chao Fan [1], Meng Ma [3] and Xinming Li [1,*]

[1] Department of Mechanical Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada; qiuling@ualberta.ca (Q.Y.); cfan3@ualberta.ca (C.F.)
[2] Department of Civil & Environmental Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada; qipei.mei@ualberta.ca
[3] Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA; maxxx971@umn.edu
* Correspondence: xinming.li@ualberta.ca

Abstract: The safety of workers in modular construction remains a concern due to the dynamic hazardous work environments and unawareness of the potential proximity of equipment. To avoid potential contact collisions and to provide a safe workplace, workers' trajectory prediction is required. With recent technology advancements, the study in the area of trajectory prediction has benefited from various data-driven approaches. However, existing data-driven approaches are mostly limited to considering only the movement information of workers in the workplace, resulting in poor estimation accuracy. In this study, we propose an environment-aware worker trajectory prediction framework based on long short-term memory (LSTM) network to not only take the individual movement into account but also the surrounding information to fully exploit the context in the modular construction facilities. By incorporating worker-to-worker interactions as well as environment-to-worker interactions into our prediction model, a sequence of the worker's future positions can be predicted. Extensive numerical tests on synthetic as well as modular construction datasets show the improved prediction performance of the proposed approach in comparison to several state-of-the-art alternatives. This study offers a systematic and flexible framework to incorporate rich contextual information into the prediction model in modular construction. The observation of how to integrate construction data analytics into a single framework could be inspiring for further future research to support robust construction safety practices.

Keywords: safe workplace; trajectory prediction; interactions; long short-term memory; modular construction; data-driven approaches

Citation: Yang, Q.; Mei, Q.; Fan, C.; Ma, M.; Li, X. Environment-Aware Worker Trajectory Prediction Using Surveillance Camera in Modular Construction Facilities. *Buildings* **2023**, *13*, 1502. https://doi.org/10.3390/buildings13061502

Academic Editors: Hexu Liu, Yinghua Shen, Yuan Chen, Xianfei Yin and Bo Xiao

Received: 11 May 2023
Revised: 28 May 2023
Accepted: 8 June 2023
Published: 10 June 2023

1. Introduction

By building the infrastructure and production facilities, construction contributes to a considerable proportion of the national socio-economic development, offering tons of employment opportunities for worldwide people [1]. In recent years, modular construction has become popular because it shifts numerous building activities to factories with off-site, manufacturing-style operations. Modular construction is usually remarked as a safer workplace than traditional construction sites, where the latter reported 2.7 million nonfatal injuries and illnesses in 2020 [2]. However, indoor modular construction has its own challenges in terms of safety risks due to its compact working environment [3]. First, modular construction facilities are often very crowded and full of different safety hazards. This leads to increased risks of contact collisions and even life-threatening accidents. In addition, due to the limited working space, the frequent interaction between workers and hazards becomes a factor that cannot be ignored [4].

Among various factors that directly and indirectly contribute to fatalities and injuries in modular construction, contact collision, resulting from the proximity of construction

resources (i.e., workers, equipment, and materials), is one of the most apparent aspects [5,6]. Due to a lack of proper object coordination and solid planning, the congested modular construction gives rise to potential hazardous contact collisions and even life-threatening scenarios. Figure 1 illustrates the potential hazard in a modular construction facility. In this context, it is critical to monitor the worker's position in a timely manner and provide a warning to the involved workers before any potential stuck-by accident occurs.

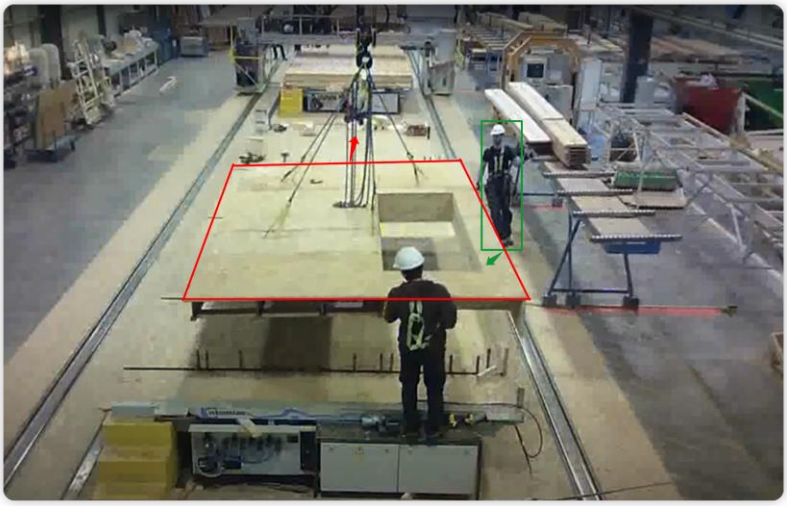


Figure 1. Potential hazard in a modular construction facility. The board in the red bounding box is guided in the direction of the red arrow, while the worker in the green bounding box is moving in the direction of the green arrow. Life-threatening contact collision may occur when the worker in the red bounding box is focusing on their allocated task and fails to recognize the proximity of the moving roof.

Recently, motivated by the remarkable performance of artificial intelligence across diverse application domains [7–14], more and more works have focused on developing data-driven neural network (NN) solutions to predict workers' position. Existing work makes an effort to obtain accurate predictions by considering the goal intent of workers, body joint angles, social relations, social rules, and norms in the fields of construction and computer vision. Although it has achieved great success, these approaches are not well suited for predicting the entity trajectory in a complicated modular construction facility as they lack adequate exploitation of environment information.

Aiming at accurately predicting the position of workers in modular factories, this paper provides a trajectory prediction method to incorporate prior knowledge of the environment into the prediction.

Specifically, an environment-aware trajectory prediction framework for modular construction workers by explicitly considering workers' movements as well as the surrounding contextual information is proposed. Every worker movement is represented by an LSTM model with a novel pooling layer that captures the spatial relationship with nearby workers (e.g., the relative distance), as well as the environment entities (e.g., the relative distance and/or direction) between the worker and surrounding objects that spread all over the facilities. In this study, we only consider the scenario where the object is stationary with no moving parts. Therefore, the spatial position of the stationary objects used in the pooling layer is regarded as prior information. To the best of our knowledge, it is the first time to involve the environment-to-worker interactions, including the distance between worker and obstacle, as well as the directional relationship between workers and obstacles in the traditional LSTM model. Our modified LSTM model is referred to as the environment-aware distance direction (EA-DD) worker trajectory prediction model. Different from the previous works [15,16], our proposed EA-DD model offers a systematic and flexible framework to incorporate general environment information into the traditional trajectory

prediction model in modular construction facilities. Leveraging the proposed EA-DD worker trajectory prediction model, an accurate forecasting of construction resources' future positions can be provided. It can aid in designing and developing a proximity warning system by releasing an alert when the workers are approaching danger zones. By reducing the number and severity of accidents in modular construction facilities, the system will contribute to significant productivity improvement.

The main contributions can be summarized as follows:

(1) A novel formulation of trajectory prediction algorithms is developed in modular construction facilities to fully exploit workplace contextual information, including not only worker-to-worker interactions but also environment-to-worker interactions;
(2) Every worker path is modeled through an LSTM network with a novel pooling that captures the interactions among workers as well as the relative distance and/or direction with the surrounding static objects;
(3) A systematic and flexible framework is offered to incorporate general environment information into the traditional trajectory prediction model in modular construction facilities.

Paper outline. Regarding the remainder of the paper, Section 2 first formulates the worker trajectory prediction problem in modular construction, followed by our proposed environment-aware solution in detail. Section 3 shows the numerical results of different trajectory prediction models using both synthetic and real modular manufacturing facility data. Discussions of the proposed model can be found in Section 4. Sections 5 and 6 provide the conclusion and possible future directions for this research.

Notation. Lower (upper)-case boldface letters denote column vectors (matrices), and normal letters represent scalars. Calligraphic letters are reserved for sets. Notation $\|\mathbf{x}\|$ is the l_2-norm of $\mathbf{x}$. Operator $\otimes$ represents Hadamard product, which is the element-wise multiplication of matrices of the same size denoted by $A \otimes B$. The sigmoid function is denoted by $\mathbb{S}$, while ϕ represents the rectified linear unit (ReLU) function.

2. Related Work

In this section, a thorough literature review on trajectory prediction is conducted, and the limitations are outlined. To obtain a reliable prediction of different subjects for various applications, considerable research efforts have been devoted [15–21]. Based on the modeling approaches, we can divide the related studies into three categories, i.e., Bayesian models, probabilistic planning, and data-driven approaches; see Table 1 for a summary. The foundation of the Bayesian model is online Bayes filters, including the Kalman filter. For example, an extended Kalman filter algorithm was proposed to update the internal state of a pedestrian in [22]. This algorithm iterated between the prediction step and update step, where the former computed the current internal states using its previous states and the latter updated the current states with observations. A dynamic Bayesian network was employed in [23] to predict whether a pedestrian would go and pass a road in front of a car. However, most Bayesian models require hand-crafted states, which hinder their applications in complicated systems. Regarding entities as intelligent agents, probabilistic planning methods [24–27] can model the problem into a Markov decision process, find the optimal solution by maximizing a reward function, and ultimately achieve the goal. The difficulty of using probabilistic planning methods to predict motion in a real scenario lies in explicitly defining an appropriate reward function that fits a specific problem, such as the path prediction task.

Table 1. Categories of trajectory prediction approaches.

Category	Method	Input	Reference
Bayesian models	Kalman filter	Coordinate, velocity	[22]
	Dynamic Bayesian network	Coordinate, head orientation, distance to road	[23,28]
Probabilistic planning	Markov model	Coordinate and moving direction	[24,29]
	Markov decision process	Environment-aware coordinate	[25,26,30]
Data-driven methods	Convolutional neural network	Coordinate	[31]
	Inverse reinforcement learning	Coordinate, goal	[32–34]
	Recurrent neural network	Video	[35]
	Social-LSTM	Coordinate, neighbor coordinate	[15]
	Encoder-decoder LSTM	Coordinate, neighbor coordinate, group, goal	[17]
	Social generative adversarial network	Coordinate, neighbor coordinate	[36,37]

In recent years, data-driven methods have also shown great potential in several challenging trajectory tracking and prediction tasks [20,38]. Storing the two-dimensional coordinates of a pedestrian into three-dimensional sparse data and then encoding the data through convolution and pooling layers of a convolutional neural network (CNN), the pedestrian walking behavior in crowded scenes could be predicted in [31]. To overcome the difficulty of reward designing, an inverse reinforcement learning method was introduced to vision-based trajectory prediction [30,32–34], where estimated rewards are employed to reproduce the optimal behavior sequences. On the other hand, the long short-term memory (LSTM) network is particularly well suited for modeling temporal dependency among sequential features. Several studies thus used the LSTM network to handle path prediction problems. For instance, an LSTM was introduced for predicting self-driving vehicle trajectories in [39] to offer safe and efficient driving on public roads. Considering surrounding neighbors' influence, a social-LSTM was proposed in [15] with a novel pooling layer to capture the interactions among different pedestrians and therefore avoid collisions. This work was further extended in [36], entailing generative adversarial networks to learn the social interactions that could influence the path of pedestrians. In the construction domain, worker trajectory was predicted through an encoder-decoder LSTM architecture considering neighbor and the goal information in [17]. A hybrid learning model integrating convolution neural networks and LSTM was proposed in [40] to detect construction workers' unsafe behaviors.

3. Method

In real-world working scenarios, workers in modular construction can mitigate potential risks by incorporating observations of the surrounding environment as well as prediction of behaviors of other co-workers and adapting their own actions accordingly. Due to limited attention and obstructed sight, workers are prone to errors, and a single chance of failure to notice could lead to disastrous consequences. Therefore, we try to offload this task to an automated environment-aware system that can predict the trajectories of workers in the modular construction, and alerts and warnings could be issued correspondingly.

3.1. Problem Formulation

Modular construction facilities, although arranging operational tasks in a controlled environment, still include dynamic and complex environments that require lots of lifting and moving all over the place. This increases the risk of contact collisions and even life-threatening accidents. The trajectory prediction problem can be mathematically formulated as follows: considering a modular construction video clip of N workers $\left(x_i^t, y_i^t\right)$ is used to represent the position of worker $i \in \{1, 2, \ldots N\}$ at time t. Given the historical positions of workers $\left\{\left(x_i^t, y_i^t\right)\right\}_{i=1:N}^{t=1:T_{obs}}$ from time 1 to T_{obs}, the goal is to predict the trajectories of workers over time $T_{obs} + 1$ to T_{pred}.

3.2. Environment-Aware Trajectory Prediction

To predict workers' paths in future several time instants, every worker's coordinate data are fed into an LSTM network. As a particular implementation of a recurrent neural network (RNN), the LSTM network is a prudent choice for time-series data. LSTM network has demonstrated remarkable performance in many sequential problems, such as machine language translation, acoustic modeling, and activity recognition [41]. When dealing with time-series problems, traditional RNNs (see Figure 2) suffer from difficulty in handling long-term dependencies. As a variation of RNN, the LSTM network can avoid this long-term dependency problem by regulating the flow of information with internal mechanisms; see Figure 3 for an illustration of the LSTM structure. As shown in Figure 3, the cell state in the LSTM is for keeping the information unchanged; the input gate contributes new information to be stored in the memory, the forget gate decides the forget degree of the internal state, and the output gate computes output from previous states. These gates can learn what information in the input sequence is useful to keep or throw away. Upon these learnable gates designs, the LSTM network can manage the internal memory state when dealing with long time series and pass the relevant information through the long-term dependency to make predictions [42].

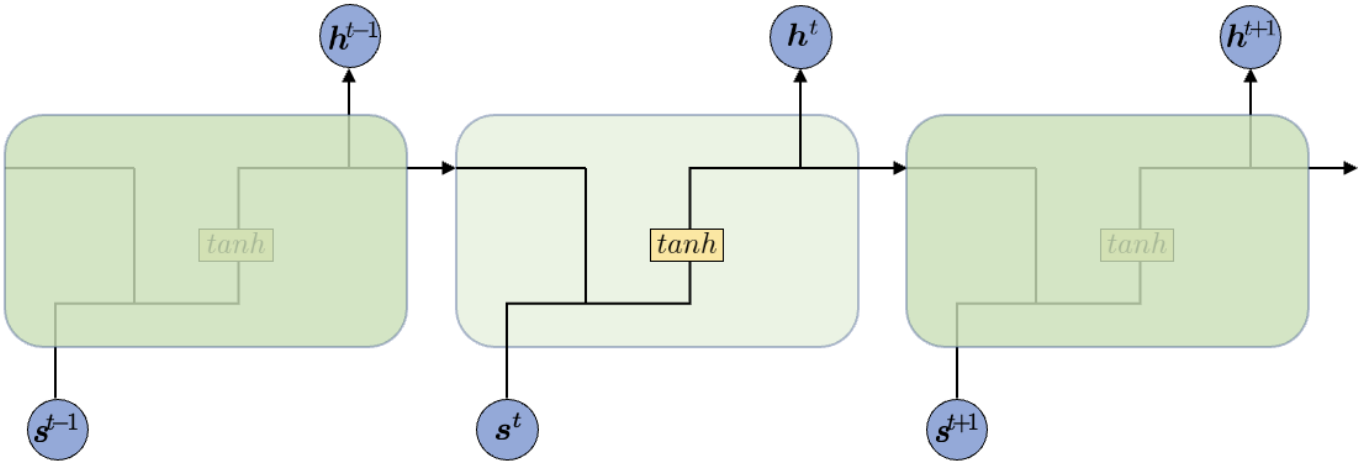


Figure 2. Internal structure of an RNN.

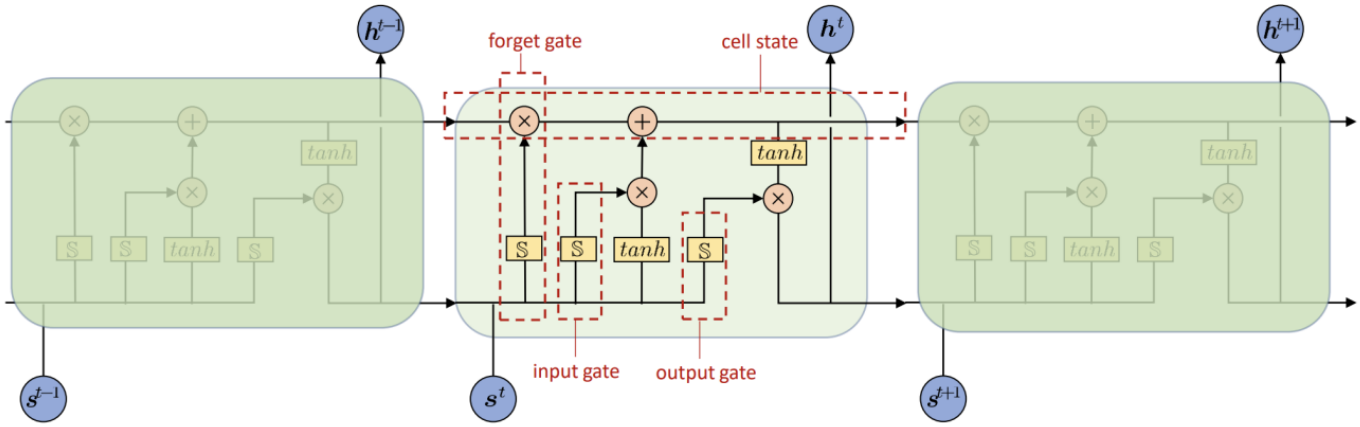


Figure 3. Internal structure of an LSTM.

Letting $\mathbf{x}_i^t = (x_i^t, y_i^t)$ be the location of the i-th worker at time instant t, the input $\mathbf{s}_i^t$ can be obtained by $\mathbf{s}_i^t = \phi(\mathbf{x}_i^t, \mathbf{W}_{xy})$, where $\mathbf{W}_{xy}$ is the input weight matrix. At time t, each feature is fed into its corresponding LSTM cell, and the cell state is updated through the following equations:

$$\mathbf{i}^t = \mathbb{S}(\mathbf{W}_{ii}\mathbf{s}_i^t + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h}^{t-1} + \mathbf{b}_{hi}) \tag{1}$$

$$\mathbf{f}^t = \mathbb{S}(\mathbf{W}_{if}\mathbf{s}_i^t + \mathbf{b}_{if} + \mathbf{W}_{hf}\mathbf{h}^{t-1} + \mathbf{b}_{hf}) \tag{2}$$

$$\mathbf{g}^t = \tanh(\mathbf{W}_{ig}\mathbf{s}_i^t + \mathbf{b}_{ig} + \mathbf{W}_{hg}\mathbf{h}^{t-1} + \mathbf{b}_{hg}) \tag{3}$$

$$\mathbf{o}^t = \mathbb{S}(\mathbf{W}_{io}\mathbf{s}_i^t + \mathbf{b}_{io} + \mathbf{W}_{ho}\mathbf{h}^{t-1} + \mathbf{b}_{ho}) \tag{4}$$

$$\mathbf{c}^t = \mathbf{f}^t \otimes \mathbf{c}^{t-1} + \mathbf{i}^t \otimes \mathbf{g}^t \tag{5}$$

$$\mathbf{h}^t = \mathbf{o}^t \otimes \tanh(\mathbf{c}^t) \tag{6}$$

where $\mathbf{W}$, $\mathbf{b}$ are the corresponding weight and bias for each gate, and $\mathbf{i}^t$, $\mathbf{f}^t$, and $\mathbf{o}^t$ represent input gate, forget gate, and output gate, respectively; $\mathbf{c}^t$ is the cell state, and $\mathbf{g}^t$ collects all information to the cell state. For simplicity, let θ collect all the weights and bias; that is,

$$\begin{aligned}\theta := [\ & \mathbf{W}_{ii}, \mathbf{b}_{ii}, \mathbf{W}_{hi}, \mathbf{b}_{hi}, \mathbf{W}_{if}, \mathbf{b}_{if}, \mathbf{W}_{hf}, \mathbf{b}_{hf}, \mathbf{W}_{ig}, \mathbf{b}_{ig}, \\ & \mathbf{W}_{hg}, \mathbf{b}_{hg}, \mathbf{W}_{io}, \mathbf{b}_{io}, \mathbf{W}_{ho}, \mathbf{b}_{ho}]\end{aligned} \tag{7}$$

After proper training, the LSTM gates could learn what information to forget and what to keep and pass on to the next level, making the retention of the message in a long sequence possible.

The prediction of worker i's position at time $t+1$ is modeled as a bivariate Gaussian distribution $N\left(\mu_i^{t+1}, \sigma_i^{t+1}, \rho_i^{t+1}\right)$ parameterized by the mean μ_i^{t+1}, standard deviation σ_i^{t+1}, and correlation coefficient ρ_i^{t+1}. Instead of predicting the positions directly, the parameters are estimated as the output of the LSTM network by applying a linear transformation $\mathbf{W}_o$ to the output hidden state $\mathbf{h}_i^t$:

$$[\mu_i^{t+1}, \sigma_i^{t+1}, \rho_i^{t+1}] = \mathbf{W}_o\mathbf{h}_i^t. \tag{8}$$

Consequently, the loss function can be chosen as the negative log-likelihood [15]:

$$L_i(\mathbf{W}_{xy}, \theta, \mathbf{W}_o) = -\sum_{t=T_{obs}+1}^{T_{pred}} \log\left(\mathbb{P}(x_i^t, y_i^t \mid \mu_i^t, \sigma_i^t, \rho_i^t)\right). \tag{9}$$

As a powerful tool for modeling time-dependent data, this LSTM network provides a suitable prediction for the trajectories of workers. However, the environment of modular construction is dynamic and complicated, and multiple aspects can influence the path of workers. This LSTM algorithm fails to capture the rich interactions among workers and their surroundings [20]. To address this issue, we propose environment-aware trajectory prediction algorithms to incorporate context information in the model.

Worker-to-worker interactions: Many tasks in construction cannot be accomplished by a single worker and often require the collaboration of two or more. Therefore, worker-to-worker interactions are the primary factors to be considered while making trajectory predictions. To take such interactions into consideration, a pooling layer connecting neighboring LSTM cells is added to each worker's LSTM network, explicitly incorporating other workers' states into the estimation of the position of the worker of interest. A worker of interest may have a varying number of neighbors over time. Hence, it is critical to handle different numbers of neighbors consistently in the model. To this end, a local grid of predefined size is constructed around the worker of interest. Each grid cell is responsible for aggregating the state information of workers inside this cell. As a result, only a fixed number of states needs to be computed in order to obtain predictions of the successive positions. This idea is manifested in two popular variants of grid-based LSTM networks, namely, social-LSTM (S-LSTM) and LSTM with occupancy map pooling (O-LSTM).

Given the position history of all the neighbors of the worker i as $\left\{(x_j^t, y_j^t)\right\}_{j\in\mathcal{N}_i}$, where $\mathcal{N}_i$ represents the collection of neighbors around the worker i, the occupancy map pooling can be calculated by

$$\mathbf{O}_i^t(m,n) = \sum_{j\in\mathcal{N}_i} \mathbf{1}_{mn}[x_j^t - x_i^t, y_j^t - y_i^t] \tag{10}$$

where $\mathbf{1}_{mn}[a,b]$ is an indicator function taking the value of $\mathbf{1}$ if (a,b) is in the (m,n) cell of the gird and 0 otherwise. The occupancy map $\mathbf{O}_i^t$ is then embedded into a vector.

$$\mathbf{e}_{oi}^t = \phi(\mathbf{O}_i^t, \mathbf{W}_{eo})$$

which is concatenated with the input $\mathbf{s}^t$ to form the aggregated input $\widetilde{\mathbf{s}}_i^t = [\mathbf{s}_i^t\ \mathbf{e}_{oi}^t]$. The future positions of the worker i can be estimated by feeding input $\widetilde{\mathbf{s}}_i^t$ into the LSTM model (1)–(6). Leveraging the position information of neighbors, the O-LSTM model can make predictions and avoid immediate collisions.

The O-LSTM model only considers the relative positions of neighbors and completely ignores their state information, which can be critical to the accurate prediction of future actions. The S-LSTM was designed with this idea in mind and found a way to combine the states into its own estimation, resulting in better prediction accuracy. The way S-LSTM incorporates states of neighbors is similar to that of O-LSTM with the corresponding hidden state.

$$\mathbf{H}_i^t(m,n,:) = \sum_{j\in\mathcal{N}_i} \mathbf{1}_{mn}\left[x_j^t - x_i^t, y_j^t - y_i^t\right]\mathbf{h}_j^{t-1}. \tag{11}$$

Similarly, these social pooling results $\mathbf{H}_i^t$ are then fed into a ReLU network to obtain the embedding vector $\mathbf{e}_{hi}^t = \phi(\mathbf{H}_i^t, \mathbf{W}_{eh})$, which again is used to obtain the aggregated input vector $\widetilde{\mathbf{s}}_i^t$. The predictions of the worker i's trajectory can be obtained by updating the LSTM model using Equations (1)–(6).

The grid-based models consider neighbors using a grid constructed around the primary worker, thus transforming the varying number of neighboring workers into a fixed number of cells.

Environment-to-worker interactions: Worker-to-worker interaction is a suitable predictor of future action because workers' behaviors are collision avoidance oriented. However, there is another factor that is not captured in worker-to-worker interactions, that is, the interaction between workers and static objects located along the path. In the presence of such obstacles, workers will naturally take a detour path to avoid a collision. Thus, the environment-to-worker interactions are equally important and should be factored in when making predictions. In this context, the present study focuses on environment-to-worker interactions by investigating collision avoidance with the presence of static obstacles, while our proposed framework can also account for other types of static objects, including walls, fixed equipment, and so on.

As obstacles are static and spread all over the facilities, the spatial coordinates of these obstacles can be regarded as known prior information. A straightforward way to consider obstacle information in the current problem formulation is to extend the grid cell information in Equations (10) or (11). Typically, modular construction facilities are usually congested with various obstacles. Modeling environment-to-worker interactions of all obstacles (even at far distances) can lead to the model learning spurious correlations. Thus, we propose to consider only the closest obstacles. Specifically, considering top-K static obstacles around the worker i, (x_{o_k}, y_{o_k}) is denoted as the position of the obstacle, $k \in K$. The Euclidean distance between worker i and obstacle k is defined as

$$\mathbf{d}_i^t(k) = \sqrt{(x_i^t - x_{o_k})^2 + (y_i^t - y_{o_k})^2}. \tag{12}$$

The embedding vector of the distance model $\mathbf{e}_{di}^t$ can be obtained by $\mathbf{e}_{di}^t = \phi(\mathbf{d}_i^t, \mathbf{W}_{ed})$.

In addition to distance, the directional relationship between obstacles and the workers also plays an important role in predicting future movements. For instance, a specific obstacle has much less influence on a worker's path when the worker deviates from it. Therefore, we further leverage a directional vector $\mathbf{q}_i^t(k)$ to capture this directional information of obstacle k relative to the worker i as

$$\mathbf{q}_i^t(k) = \left(\frac{x_i^t - x_{o_k}}{\|x_i^t - x_{o_k}\|}, \frac{y_i^t - y_{o_k}}{\|y_i^t - y_{o_k}\|} \right). \tag{13}$$

The direction embedding vector is $\mathbf{e}_{qi}^t = \phi(\mathbf{q}_i^t, \mathbf{W}_{eq})$. Our proposed EA-DD model captures this obstacle distance and direction information relative to the worker. Concatenating worker-to-worker interactions vector $\mathbf{e}_{hi}^t$ as well as environment-to-worker interactions vector $\mathbf{e}_{di}^t$ and $\mathbf{e}_{qi}^t$, the new input $\widetilde{\mathbf{s}}_i^t$ of the EA-DD model can be addressed as

$$\widetilde{\mathbf{s}}_i^t = [\mathbf{s}_i^t \ \mathbf{e}_{hi}^t \ \mathbf{e}_{di}^t \ \mathbf{e}_{qi}^t]. \tag{14}$$

This new input $\widetilde{\mathbf{s}}_i^t$ is fed into the LSTM model (1) to obtain predictions of worker i's trajectory.

It should be noted that the loss function of worker i in (4) introduces new parameters $\mathbf{W}_{eh}$, $\mathbf{W}_{ed}$, and $\mathbf{W}_{eq}$ during back-propagation as

$$L_i(\mathbf{W}_{xy}, \mathbf{W}_{eh}, \mathbf{W}_{ed}, \mathbf{W}_{eq}, \theta, \mathbf{W}_o) = -\sum_{t=T_{obs}+1}^{T_{pred}} \log\left(\mathbb{P}(x_i^t, y_i^t \mid \mu_i^t, \sigma_i^t, \rho_i^t)\right). \tag{15}$$

The proposed EA-DD worker trajectory model is summarized in Figure 4.

The overall implementation process consists of two stages, namely offline training and online inference. Specifically, in the offline training phase, the model is fed with the observations $\left\{\mathbf{x}^t, \mathbf{y}^t\right\}^{t=1:T_{obs}}$ of all the workers at time instants $t = 1, \ldots, T_{obs}$ and trained to output the forecasting. In the online inference process, the observations of all workers $\left\{\mathbf{x}^t, \mathbf{y}^t\right\}^{t=1:T_{obs}}$ are fed into the trained model, and near-future forecasting

$\{\hat{\mathbf{x}}^t, \hat{\mathbf{y}}^t\}^{t=T_{obs}+1:T_{pred}}$ can be obtained by considering worker-to-worker interactions as well as environment-to-worker interactions.

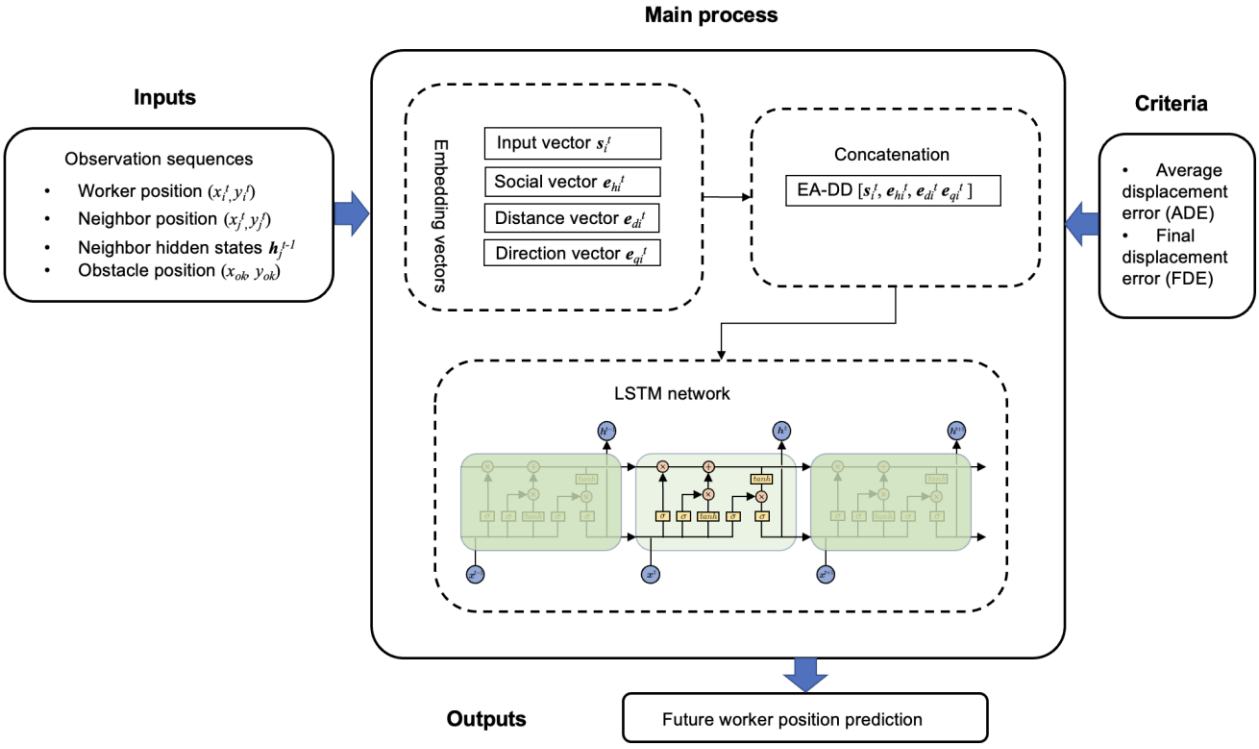


Figure 4. Overview of our EA-DD worker trajectory prediction model. The model takes as input the primary worker's position, neighbor's position, neighbor's hidden states, and obstacles' position. By embedding individual movement-related vectors as well as the environment information vectors, the EA-DD model generates its output in the future position of the primary worker. The prediction performance is evaluated by ADE and FDE.

4. Results

In this section, the implementation details and two evaluation metrics are first described. Then, extensive experiments on both a synthetic dataset and a real modular construction facility dataset are implemented to demonstrate the effectiveness of our proposed EA-DD worker trajectory prediction method.

4.1. Implementation Details

The proposed EA-DD trajectory prediction model was trained by Pytorch [43] on an NVIDIA RTX A5000 GPU. To reduce over-fitting, two techniques were adopted, which are data augmentation and early stopping. Data augmentation was used to ensure the model's fit to predict future positions, while early stopping can terminate the training process when the total loss on the validation set cannot go down after certain epochs and save the model, which can provide the smallest loss. Both the synthetic dataset and real modular manufacturing facility dataset are split into a training set (70%) for model training, a validation set (20%) for early stopping and hyper-parameter selection, and a testing set (10%) for performance assessment of the prediction model. The network was trained with the Adam optimizer [44], with a learning rate of 1×10^{-3} and a batch size of 8. In the experiments, different combinations of the above hyperparameters, as well as the number of hidden units, were used, and the optimal ones that resulted in the highest accuracy in the validation set were selected.

We consider 3.6 s observations and make 4.8 s predictions, which align with relevant studies on trajectory prediction in both construction and other domains [15,16]. With the frame rate setting as 0.4, we use the past 9 frames to predict 12 future frames. The dimension

of the embedding vector for the worker velocity is 64, while the dimension of the interaction vector is 256. We use a 16 × 16 grid to capture the neighbors. We choose K = 2, meaning the top 2 obstacles are considered around workers. The spatial coordinates of those obstacles are known as prior information. To assess the performance of the proposed EA-DD model, we have simulated three trajectory prediction policies as baselines, namely the LSTM model, the O-LSTM model [15], and the S-LSTM model [15]. Notice that the solution based only on LSTM does not consider any kind of interaction while predicting the worker trajectory. On the other hand, O-LSTM and S-LSTM are able to consider worker-to-worker interactions in the forecasting. In order to not only investigate how workers influence each other, we then extend the S-LSTM model with our environment-to-worker pooling in our previous work [45] to become the EA-Distance model, which considers the distance of obstacles relative to workers, and the EA-Direction model, which considers the direction information. These two models are also tested here for a comprehensive comparison.

4.2. Evaluation Metrics

To evaluate the performance of the proposed trajectory prediction framework performance, we use the two most common metrics, which are average displacement error (ADE) and final displacement error (FDE) [15,46–48]. ADE is the mean square error overall predicted positions of the path and the ground-truth path as follows:

$$ADE := \frac{\sum_{i=1}^{N} \sum_{t=T_{obs}+1}^{T_{pred}} \|(\hat{x}_i^t, \hat{y}_i^t) - (x_i^t, y_i^t)\|}{N \times (T_{pred} - T_{obs} - 1)} \tag{16}$$

where N is the number of workers, $(\hat{x}_i^t, \hat{y}_i^t)$ and (x_i^t, y_i^t) are the predicted coordinate of worker i at time instant t and the ground-truth coordinate of worker i at time instant t, respectively. ADE is the MSE over all positions of predicted trajectories and the actual ones. It is the measure to show how close the prediction and actual trajectories are. Therefore, ADE is important to ensure the overall accuracy of the prediction.

FDE is the distance between the predicted final destination and the ground truth one as

$$FDE := \frac{\sum_{i=1}^{N} \|(\hat{x}_i^{T_{pred}}, \hat{y}_i^{T_{pred}}) - (x_i^{T_{pred}}, y_i^{T_{pred}})\|}{N}. \tag{17}$$

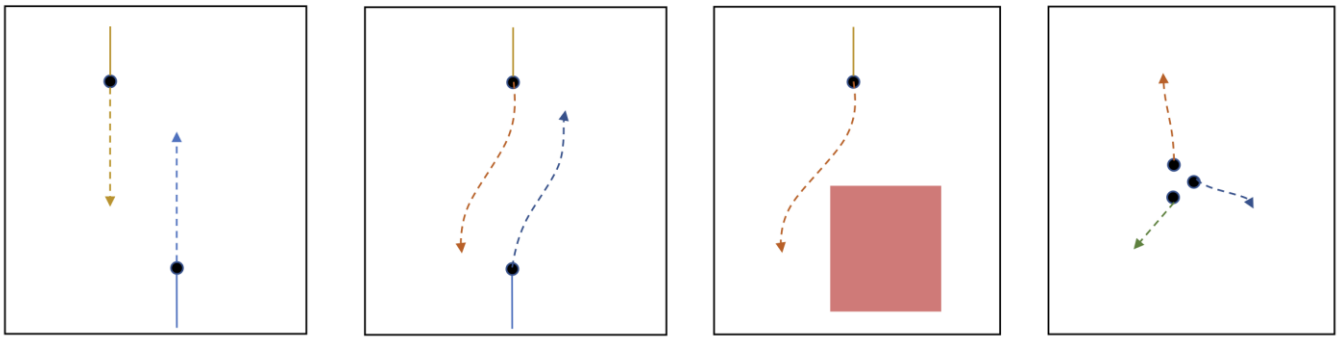
FDE is the MSE over the final actual position and the final predicted position. It is the measure to show the accuracy in predicting a worker's final location and is important to detect potential collisions.

ADE and FDE essentially measure the accuracy of the overall predicted path and the accuracy of the predicted final location, which are critical to avoid potential collisions in modular construction.

4.3. Synthetic Experiments

We first verify our proposed model on a synthetic dataset. The dataset is generated according to a collision avoidance approach called optimal reciprocal collision avoidance (ORCA) [49], which offers sufficient conditions for multiple independent agents making collision-free movements in a common workspace without communication with each other. In this synthetic dataset which is available on github https://github.com/Rachelmy/EA-DD (accessed on 10 August 2022)), a total of 1000 scenes were generated for five workers as well as two obstacles. The trajectories reflect various worker-to-worker interactions as well as environment-to-worker interactions, including (i) crossing each other, (ii) worker collision avoidance, (iii) obstacle collision avoidance, and (iv) dispersing; see Figure 5 for example scenarios. The vertices of two obstacles are (−1, −2), (1, −2), (1, 2), (−1, 2), and (3, −3.5), (3.5, −3.5), (3.5, −3), (3, −3), respectively. Regarding the center position as the position of an obstacle, we obtain the spatial coordinates of two obstacles, i.e.,

$(x_{o1}, y_{o1}) = (0, 0)$ and $(x_{02}, y_{o2}) = (3.25, -3.25)$; see Figure 6 for the illustration of the synthetically generated samples and obstacles.



(**a**) (**b**) (**c**) (**d**)

Figure 5. Example scenarios. Dotted lines indicate the path of worker in the future time slots, and arrows are the directions. (**a**) crossing each other; (**b**) worker collision avoidance; (**c**) obstacle collision avoidance; (**d**) dispersing.

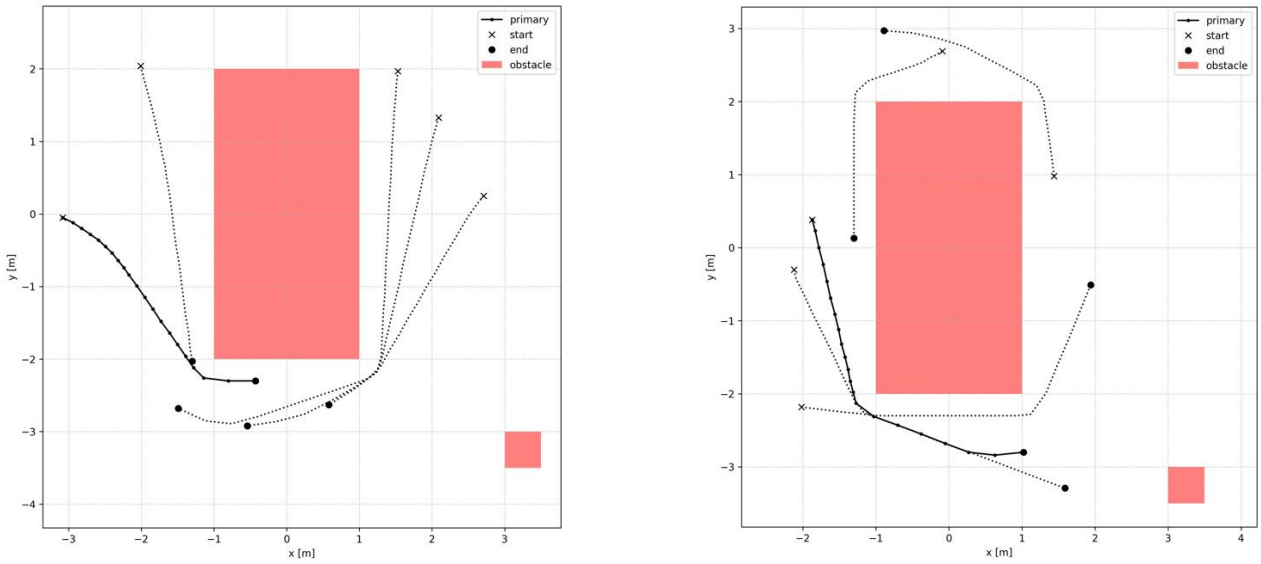


Figure 6. Illustration of the synthetically generated samples and obstacles.

Table 2 summarizes the ADE and FDE results for the different models on the test dataset. Evidently, by incorporating worker-to-worker interactions and environment-to-worker interactions into the prediction model, three environment-aware models can attain smaller errors compared with the one that does not consider worker-to-worker interactions and/or environment-to-worker interactions. Furthermore, the EA-DD model fully exploiting the environment information on the job site can achieve the most competitive performance relative to the alternatives. Noticing FDEs are higher than ADEs, the reason is that forecasting a worker's position at a far time instant is more challenging than the close one given the same observations. Figure 7 depicts the qualitative results of the proposed schemes in comparison to alternatives. Curves show that the prediction trajectories of our proposed scheme are close to the ground truth. It should be noted that the running time of EA-DD is longer than other methods due to the complex nature of the method. Further studies will be conducted to improve the execution efficiency of EA-DD.

Table 2. Prediction performance of different models on a synthetic dataset.

Model Name	ADE (m)	FDE (m)	Time (s)
LSTM	0.36	0.89	0.16
O-LSTM [15]	0.31	0.74	0.39
S-LSTM [15]	0.28	0.67	0.82
EA-Distance [45]	0.30	0.69	0.93
EA-Direction [45]	0.24	0.55	0.95
EA-DD	0.22	0.50	1.03

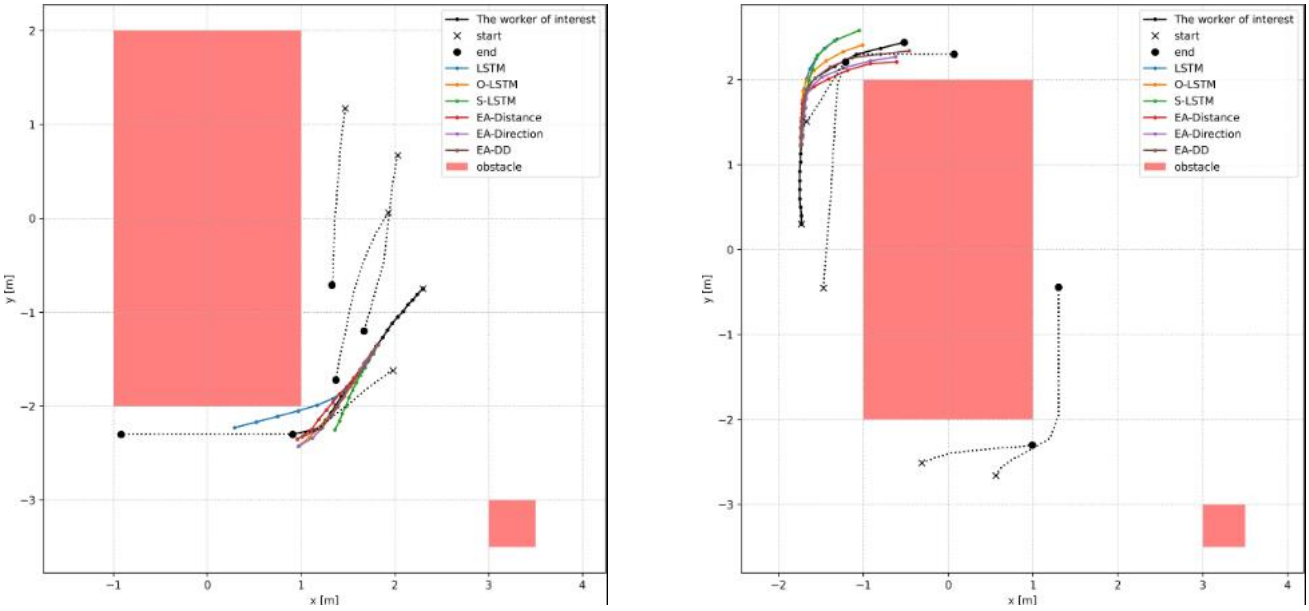


Figure 7. Two example scenes of the worker trajectory predicted by LSTM (blue line), Occupancy-LSTM (orange line), Social-LSTM (green line), EA-Distance (red line), and EA-Direction (purple line), EA-DD (brown line) models on a synthetic dataset. The black line represents the ground truth, which is the worker's actual trajectory, while the red zones indicate static obstacles on the job site. Workers navigate their trajectories according to different prediction models to avoid potential collisions with each other as well as obstacles.

4.4. Modular Construction Experiments

The second experiment validates the performance of different models on a real modular manufacturing facility dataset from a leading off-site construction facility located in Edmonton, Canada. We use a video of the construction facility as the data source, and it provides rich environmental information. Leveraging the temporal continuity property of the construction video, different frames were randomly selected in each video clip, generating training examples. Figure 8 gives two sample scenes of the video. The pixel coordinates of workers and obstacles were manually annotated over the whole frame, and a complete inspection ensured the validity of the annotations. As a result, we obtain an annotated real dataset consisting of 13 moving workers and 2 static obstacles in a total of 1826 scenes. The models are trained by synthetic data with the same static obstacle positions and then tested on this real construction dataset.

When constructing a real dataset, the challenge is obtaining the actual point locations of workers and static elements of the workplaces from the construction video. The video is obtained by a perspective view, where we can only obtain the entity positions in an image in terms of pixels. To transfer the image positions to real-world coordinates, we need to find the homography matrix by four manually selected points on the ground with estimated

measurements. Upon this homography matrix, the positions of workers and static obstacles are extracted. Specifically, extracting the object's actual point locations entails the following three steps.

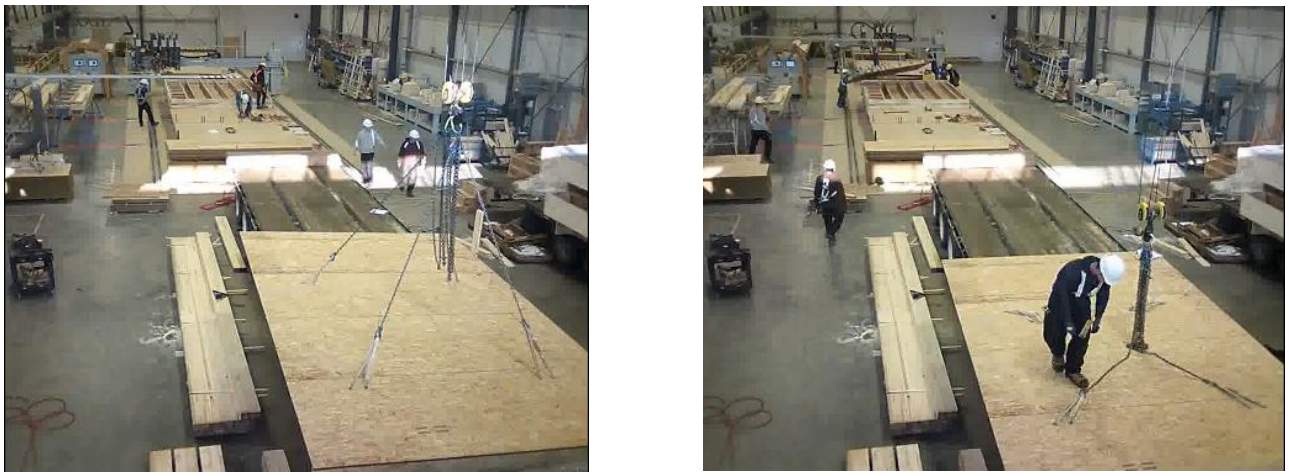


Figure 8. Two example scenes in real construction video.

(Step 1) Estimate the homography matrix.

Before getting an estimate of the homography matrix, we need first to undistort the image using estimated camera parameters, which are obtained from camera calibration. Next, we obtain the image (pixel) coordinates by detecting a pattern, such as a chessboard. In this construction data, we use the four corners of the trailer as the pattern. The real measurements of the trailer's corner are estimated based on prior information. Then, we obtain the corresponding real-world coordinates of the pattern by selecting a proper coordination system. Finally, the estimate of the homography matrix can be conducted by using four coordinate pairs of the trailer's corner from the image and the real world.

(Step 2) Obtain pixel coordinates in the video.

The object's pixel coordination can be obtained in two ways: (i) manually annotate objects frame by frame, and (ii) use object detection and object tracking methods to automatically obtain the pixel coordinates of the target objects. The obtained object coordination should be grouped to form trajectories, either manually or automatically, via object tracking.

(Step 3) Obtain real-world coordinates using the homography matrix.

Finally, the actual locations of workers and static elements can be obtained by applying the homography matrix (obtained from Step 1) to the corresponding pixel coordinates (obtained from Step 2).

Table 3 provides the performance evaluation of different models. Again, the result indicates that the incorporation of environment information in trajectory prediction leads to lower ADE and FDE values. Figure 9 illustrates four example scenes from a bird-eye view, where the worker's trajectory is influenced by other workers and/or static obstacles, such as trailers. Figure 9a shows the trajectory prediction performance of the interested worker without being influenced by neighbors and static elements. The same worker's trajectory influenced by a neighbor is depicted in Figure 9b. The LSTM, which does not consider worker-to-worker interactions, fails to predict the correct trajectory. Similarly, in Figure 9c, the prediction of another worker's trajectory validates that the LSTM algorithm cannot avoid collisions. Figure 9d demonstrates the trajectory prediction performance of a worker proceeding into an obstacle. The O-LSTM, which does not model environment-to-worker interactions, cannot stop the collision with the obstacle. In summary, our environment-aware prediction model fully exploits the existing interactions in the scene, leading to better performance in collision avoidance.

Table 3. Prediction performance of different models on a real construction dataset.

Model Name	ADE (m)	FDE (m)
LSTM	1.98	3.52
O-LSTM [15]	1.62	2.91
S-LSTM [15]	1.60	2.86
EA-Distance [45]	1.56	2.03
EA-Direction [45]	1.50	1.91
EA-DD	1.48	1.85

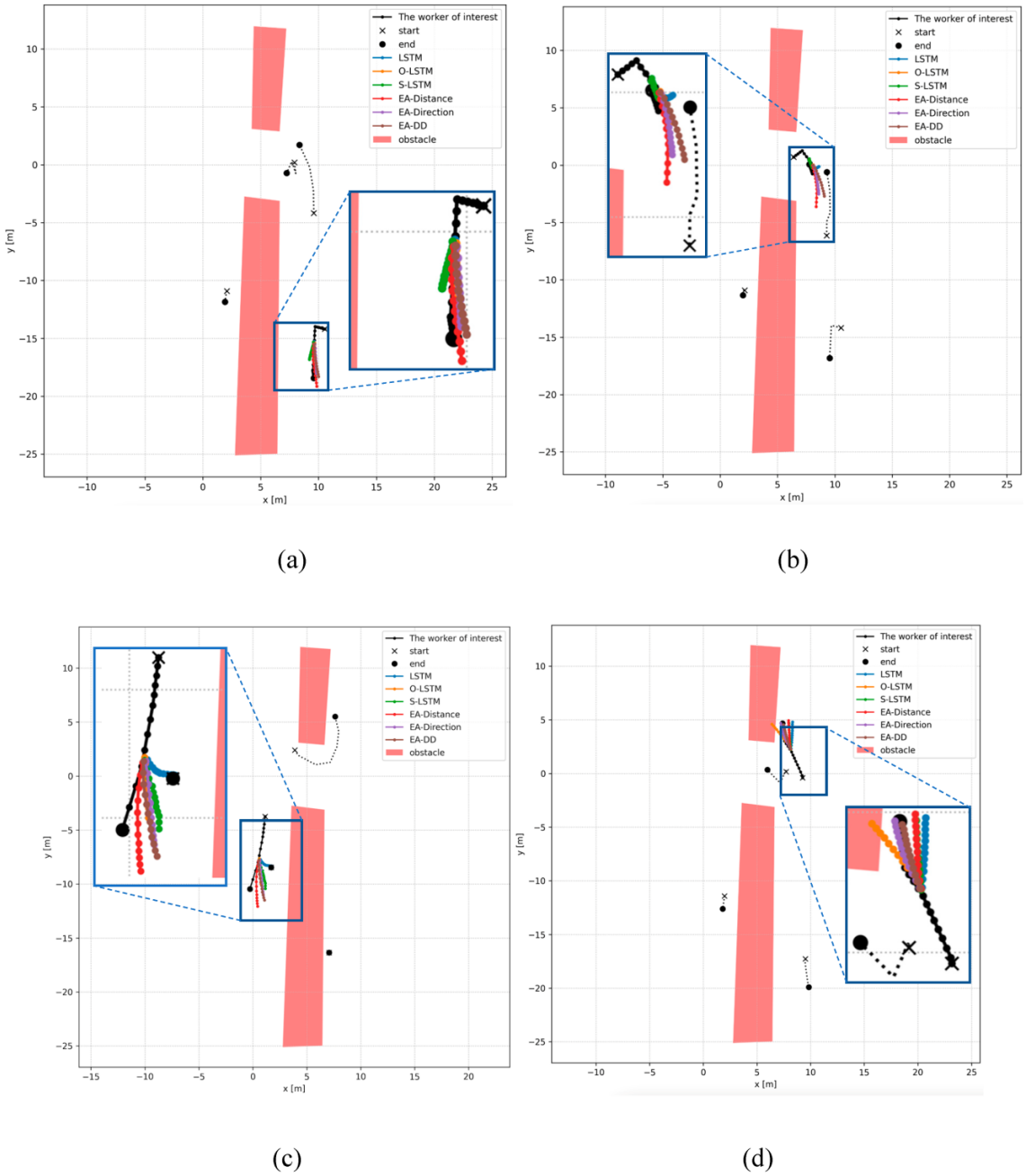


Figure 9. Prediction performance of different models in a bird's-eye view on construction dataset. (**a**) trajectory prediction performance of the interested worker without being influenced by neighbors and static elements, (**b**) worker's trajectory influenced by a neighbor, (**c**) the prediction of another worker's trajectory, (**d**) trajectory prediction performance of a worker proceeding into an obstacle.

5. Discussion

To develop a long-term trajectory prediction model for modular construction facilities, two important hyperparameters, including prediction length and observation length, need to be carefully tuned. To check the system performance on different combinations of the prediction and observation length, we conducted an experiment on a synthetic dataset. Specifically, to achieve the prediction model capable of predicting 4.8 s, the observation length is changed from 2.4 s to 6 s. Therefore, a total of seven different tuning scenarios were established and applied in training. Table 4 summarizes the ADE and FDE of our EA-DD model with different hyperparameters of observation length and prediction length. Results show that a longer observation length does not necessarily provide higher accuracy in the performance. This might be because the earlier observations have more noise and less relevancy than the later ones, which can lead to a negative impact on the performance.

Table 4. Prediction performance.

Number	Observation (s)	Prediction (s)	ADE (m)	FDE (m)
1	2.8	4.8	0.35	0.93
2	3.2	4.8	0.26	0.72
3	3.6	4.8	0.22	0.50
4	4.4	4.8	0.23	0.52
5	4.8	4.8	0.30	0.68

It is also interesting to compare the performance of the EA-DD model and the S-LSTM (and O-LSTM) model. The reason that our proposed EA-DD performs superior to the S-LSTM is mainly due to the complex nature of workplace movements as well as the noisy measurements. The design of social pooling in S-LSTM (and O-LSTM) is not capable of learning the important notion of preventing collisions since the model is trained to just minimize ADE/FDE without avoiding any possible collisions. On the contrary, by focusing explicitly on the collision of any static objects, our proposed EA-DD model obtain enough domain knowledge of the environment while designing the interaction encoder. In practice, there is a trade-off between complexity and compute efficiency. The design of environment-to-worker interactions hampers compute efficiency as we incorporate rich contextual information in our model. However, safer forecasting can be achieved as the objective of the model, including collision avoidance. Improving computational efficiency without losing any contextual information will be an interesting and meaningful direction in our future research.

In the absence of proper site coordination, the congested workspace can lead to potentially hazardous and life-threatening situations. Therefore, to improve risk management, in addition to a commitment to safety, supervision, and PPE, new measures that can collect data in a timely manner and provide alerts to workers before a potential hazard occurs will become an asset in the construction site. To this end, the proposed EA-DD worker trajectory prediction model provides accurate forecasting of construction resources' future positions. It can aid in designing and developing a proximity warning system by releasing an alert when the workers are approaching danger zones.

By incorporating contextual information and worker movement information into the LSTM network, this work contributes to the body of knowledge by creating a novel environment-aware deep learning method for worker trajectory prediction in modular construction facilities. The proposed framework not only considers the spatial interactions between the worker and the neighbors but also innovatively incorporates the relative distance and direction between the worker and static elements. The results of both the synthetic data experiments as well as the real modular construction data experiments in this paper show that integrating the above contextual information outperforms the pure position-based prediction model. Based on accurate position forecasting, an early warning

when two entities are expected to get too close can be provided. Workers can then actively plan a safe path to avoid collisions while ensuring smooth operations.

Although the proposed method provides an accurate and proactive prediction for the worker's future position, there remain a few limitations of the findings that deserve more effort in future work. First, a large dataset is required for training LSTM models and forecasting the future positions of various entities. However, the datasets used in the experiment are relatively small due to the lack of publicly available datasets in modular construction. To expand the capability of the dataset, more statistical tests need to be conducted, which, therefore, further justify the performance of the model. More modular construction videos carrying distinguishable movements of workers and interactions with surrounding entities need to be collected and annotated as well. Second, in theory, there should not be a limitation on the number of obstacles, workers, and the size of the environment. The number of obstacles is chosen to align with our real dataset, where we have two obstacles in the modular manufacturing facility. Studies on other numbers of obstacles can be performed in the future when a more extensive dataset for modular construction facilities is available. Along with collecting a large dataset, exploring a more crowded environment to train LSTM models and forecast the future positions of various entities is deferred to our future research. Third, taking advantage of the public dataset in other domains, such as crowds [48,50], transfer learning can be adopted to relieve the heavy burden of manually annotating images in the modular construction video. Fourth, if the task requires the entity to move back and forth, the workers may move in the opposite direction, where our proposed method has limited ability to make an accurate prediction. Adding more information, such as destination, specific task, and group, may help improve the performance in this situation. Another limitation of this study is that we only consider the scenario where the object is static with no moving parts. The position of the obstacle is simplified as prior knowledge to examine its influence on trajectory prediction. Developing a more general approach to incorporating the moving entities, including equipment and activities, in the prediction model will be an important direction in future research. Finally, in addition to designing and developing a trajectory prediction framework that can provide reliable forecasting results, a key question that also needs to be addressed in the construction safety area is how to alert a worker of potential hazardous encounters in real-time to avoid an accident effectively. To this end, a thoroughly proactive hazard detection system will need further attention in future work.

6. Conclusions

Modular construction facilities are complex and information-intensive environments that contain various static and dynamic entities, including workers, equipment, buildings, and materials. This congested workplace increases the risk of contact collisions and even life-threatening accidents. To prevent struck-by accidents and mitigate potential injuries, accurate forecasting of construction resources' future positions is critical in modern modular construction facilities. In this work, we have designed an environment-aware worker trajectory prediction framework using a computer vision-based method. The proposed prediction model exploits rich job-site contextual information as well as individual movement information in the environment. Specifically, every worker path is modeled through an LSTM network. We design a novel pooling method that captures the worker-to-worker interactions as well as the environment-to-worker interactions. Numerical tests showcase that by fully exploiting the environmental information, our proposed prediction model can achieve competitive performance relative to several existing ones. This research provides a systematic and flexible framework to incorporate rich contextual information into a trajectory prediction model to improve the current practice of worker path forecasting in modular construction.

Author Contributions: Methodology, Q.Y., Q.M., M.M. and X.L.; validation, Q.Y. and C.F.; formal analysis, Q.Y.; investigation, Q.Y. and M.M.; resources, Q.M. and X.L.; data curation, Q.Y. and C.F.; writing—original draft preparation, Q.Y.; writing—review and editing, X.L. and Q.M.; visualization, Q.Y.; supervision, X.L.; project administration, Q.M. and X.L.; funding acquisition, Q.M. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Sciences and Engineering Research Council of Canada Alliance—Alberta Innovates Advance Program ALLRP 561120-20.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: The authors would like to acknowledge the support from their industry partner in Canada.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; Savarese, S. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 961–971.
2. Alazzaz, F.; Whyte, A. Uptake of Off-Site Construction: Benefit and Future Application. *Int. J. Civ. Archit. Struct. Constr. Eng.* **2014**, *8*, 5.
3. Altche, F.; de La Fortelle, A. An LSTM network for highway trajectory prediction. In Proceedings of the 20th IEEE International Conference on Intelligent Transportation Systems, Yokohama, Japan, 16–19 October 2017; pp. 353–359. [CrossRef]
4. Ballan, L.; Castaldo, F.; Alahi, A.; Palmieri, F.; Savarese, S. Knowledge Transfer for Scene-Specific Motion Prediction. In *Computer Vision*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 697–713. [CrossRef]
5. Bartoli, F.; Lisanti, G.; Ballan, L.; Del Bimbo, A. Context-Aware Trajectory Prediction. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1941–1946. [CrossRef]
6. Bokhari, S.Z.; Kitani, K.M. Long-Term Activity Forecasting Using First-Person Vision. In *Computer Vision*; Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2017; pp. 346–360. [CrossRef]
7. Bureau of Labor Statistics. *Employer-Reported Workplace Injuries and Illnesses—2020*; Bureau of Labor Statistics: Washington, DC, USA, 2020; p. 9.
8. Cai, J.; Zhang, Y.; Yang, L.; Cai, H.; Li, S. A context-augmented deep learning approach for worker trajectory prediction on unstructured and dynamic construction sites. *Adv. Eng. Inform.* **2020**, *46*, 101173. [CrossRef]
9. Ding, L.; Fang, W.; Luo, H.; Love, P.E.; Zhong, B.; Ouyang, X. A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. *Autom. Constr.* **2018**, *86*, 118–124. [CrossRef]
10. Dong, C.; Li, H.; Luo, X.; Ding, L.; Siebert, J.; Luo, H. Proactive struck-by risk detection with movement patterns and randomness. *Autom. Constr.* **2018**, *91*, 246–255. [CrossRef]
11. Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; Alahi, A. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2255–2264.
12. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
13. Jeong, G.; Kim, H.; Lee, H.-S.; Park, M.; Hyun, H. Analysis of safety risk factors of modular construction to identify accident trends. *J. Asian Arch. Build. Eng.* **2021**, *21*, 1040–1052. [CrossRef]
14. Karasev, V.; Ayvaci, A.; Heisele, B.; Soatto, S. Intent-Aware Long-Term Prediction of Pedestrian Motion. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2543–2549. [CrossRef]
15. Kim, D.; Liu, M.; Lee, S.; Kamat, V.R. Trajectory Prediction of Mobile Construction Resources Toward Pro-Active Struck-by Hazard Detection. In *International Symposium on Automation and Robotics in Construction*; International Association for Automation and Robotics in Construction (IAARC): Banff, AB, Canada, 2019. [CrossRef]
16. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv* **2017**, arXiv:1412.6980. [CrossRef]
17. Kitani, K.M.; Ziebart, B.D.; Bagnell, J.A.; Hebert, M. Activity Forecasting. In *Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 201–214. [CrossRef]
18. Kooij, J.F.P.; Schneider, N.; Flohr, F.; Gavrila, D.M. Context-Based Pedestrian Path Prediction. In *Computer Vision*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2014; pp. 618–633. [CrossRef]
19. Kothari, P.; Kreiss, S.; Alahi, A. Human Trajectory Forecasting in Crowds: A Deep Learning Perspective. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 7386–7400. [CrossRef]

20. Lasota, P.A.; Fong, T.; Shah, J.A. A Survey of Methods for Safe Human-Robot Interaction. *Found. Trends Robot.* **2014**, *5*, 261–349. [CrossRef]
21. Lee, N.; Choi, W.; Vernaza, P.; Choy, C.B.; Torr, P.H.; Chandraker, M. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 336–345.
22. Lerner, A.; Chrysanthou, Y.; Lischinski, D. Crowds by Example. *Comput. Graph. Forum* **2007**, *26*, 655–664. [CrossRef]
23. Luo, H.; Xiong, C.; Fang, W.; Love, P.E.; Zhang, B.; Ouyang, X. Convolutional neural networks: Computer vision-based workforce activity assessment in construction. *Autom. Constr.* **2018**, *94*, 282–289. [CrossRef]
24. Ma, M.; Nikolakopoulos, A.N.; Giannakis, G.B. Hybrid ADMM: A unifying and fast approach to decentralized optimization. *EURASIP J. Adv. Signal Process.* **2018**, *2018*, 73. [CrossRef]
25. Ma, W.C.; Huang, D.A.; Lee, N.; Kitani, K.M. Forecasting Interactive Dynamics of Pedestrians with Fictitious Play. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 774–782.
26. Mei, Q.; Gül, M. A cost effective solution for pavement crack inspection using cameras and deep neural networks. *Constr. Build. Mater.* **2020**, *256*, 119397. [CrossRef]
27. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
28. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: San Jose, CA, USA, 2019; Volume 32.
29. Pellegrini, S.; Ess, A.; Schindler, K.; Van Gool, L. You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 261–268. [CrossRef]
30. Rashid, K.M.; Behzadan, A.H. Risk Behavior-Based Trajectory Prediction for Construction Site Safety Monitoring. *J. Constr. Eng. Manag.* **2018**, *144*, 04017106. [CrossRef]
31. Rashid, K.M.; Louis, J. Activity identification in modular construction using audio signals and machine learning. *Autom. Constr.* **2020**, *119*, 103361. [CrossRef]
32. Rhinehart, N.; Kitani, K.M. First-Person Activity Forecasting with Online Inverse Reinforcement Learning. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3696–3705.
33. Rudenko, A.; Palmieri, L.; Herman, M.; Kitani, K.M.; Gavrila, D.M.; O Arras, K. Human motion trajectory prediction: A survey. *Int. J. Robot. Res.* **2020**, *39*, 895–935. [CrossRef]
34. Sadeghi, A.; Wang, G.; Giannakis, G.B. Deep Reinforcement Learning for Adaptive Caching in Hierarchical Content Delivery Networks. *IEEE Trans. Cogn. Commun. Netw.* **2019**, *5*, 1024–1033. [CrossRef]
35. Saleh, K.; Hossny, M.; Nahavandi, S. Intent Prediction of Pedestrians via Motion Trajectories Using Stacked Recurrent Neural Networks. *IEEE Trans. Intell. Veh.* **2018**, *3*, 414–424. [CrossRef]
36. Schneider, N.; Gavrila, D.M. Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study. In *Pattern Recognition*; Weickert, J., Hein, M., Schiele, B., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; pp. 174–183. [CrossRef]
37. Seo, J.; Han, S.; Lee, S.; Kim, H. Computer vision techniques for construction safety and health monitoring. *Adv. Eng. Informatics* **2015**, *29*, 239–251. [CrossRef]
38. Song, J.; Haas, C.T.; Caldas, C.H. Tracking the Location of Materials on Construction Job Sites. *J. Constr. Eng. Manag.* **2006**, *132*, 911–918. [CrossRef]
39. Teizer, J.; Venugopal, M.; Walia, A. Ultrawideband for Automated Real-Time Three-Dimensional Location Sensing for Workforce, Equipment, and Material Positioning and Tracking. *Transp. Res. Rec. J. Transp. Res. Board* **2008**, *2081*, 56–64. [CrossRef]
40. Berg, J.V.D.; Lin, M.; Manocha, D. Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1928–1935. [CrossRef]
41. Vemula, A.; Muelling, K.; Oh, J. Social Attention: Modeling Attention in Human Crowds. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, QLD, Australia, 21–25 May 2018; pp. 4601–4607. [CrossRef]
42. Wells, J. *The Construction Industry in Developing Countries: Alternative Strategies for Development*; Croom Helm: Beckenham, Kent, UK, 1986.
43. Xiao, B.; Xiao, H.; Wang, J.; Chen, Y. Vision-based method for tracking workers by integrating deep learning instance segmentation in off-site construction. *Autom. Constr.* **2022**, *136*, 104148. [CrossRef]
44. Xu, Y.; Piao, Z.; Gao, S. Encoding Crowd Interaction with Deep Neural Network for Pedestrian Trajectory Prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5275–5284.
45. Yang, Q.; Mei, Q.; Fan, C.; Ma, M.; Li, X. Environment-Aware Worker Trajectory Prediction Using Surveillance Camera on Modular Construction Sites. In Proceedings of the Modular and Offsite Construction Summit, Edmonton, AB, Canada, 27–29 July 2022; pp. 1–8.
46. Yang, Q.; Wang, G.; Sadeghi, A.; Giannakis, G.B.; Sun, J. Two-Timescale Voltage Control in Distribution Grids Using Deep Reinforcement Learning. *IEEE Trans. Smart Grid* **2019**, *11*, 2313–2323. [CrossRef]

47. Yi, S.; Li, H.; Wang, X. Pedestrian Behavior Understanding and Prediction with Deep Neural Networks. In *Computer Vision*; Springer International Publishing: Cham, Switzerland, 2016; pp. 263–279. [CrossRef]
48. Yu, Y.; Guo, H.; Ding, Q.; Li, H.; Skitmore, M. An experimental study of real-time identification of construction workers' unsafe behaviors. *Autom. Constr.* **2017**, *82*, 193–206. [CrossRef]
49. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
50. Ziebart, B.D.; Ratliff, N.; Gallagher, G.; Mertz, C.; Peterson, K.; Bagnell, J.A.; Hebert, M.; Dey, A.K.; Srinivasa, S. Planning-Based Prediction for Pedestrians. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 3931–3936. [CrossRef]