Kalpa Publications in Computing

Volume 22, 2025, Pages 781–791

Proceedings of The Sixth International Conference on Civil and Building Engineering Informatics

Trajectory Forecasting for Worker Safety in Construction Using Transformer and Graph Attention Networks

Mohammed Alduais[1], Xinming Li[1]*, and Qipei Mei[1]

University of Alberta, Edmonton, Alberta, Canada.
alduais@ualberta.ca, xinming.li@ualberta.ca, qipei.mei@ualberta.ca

Abstract

With the rise of residential housing demand worldwide, offsite construction emerges as a possible option to speed up construction while improving the safety of workers. However, offsite construction sites are normally a dynamic environment in which workers collaborate with various machinery and large moving objects, resulting in additional safety concerns. Accurate prediction of future trajectories is an important step in building a collision alarm system that can be utilized to mitigate such safety risks. Traditional methods, such as Kalman filters (KF) and Markov processes, rely heavily on past trajectories and handcrafted features, which fail to account for the dynamic nature of construction sites. With the rising interest in data-driven approaches, several studies have explored different methods of trajectory prediction. Long Short-Term Memory (LSTM) network is one of the major methods used for forecasting future trajectories by leveraging both past individual and contextual information. However, one of the main limitations of LSTM is error accumulation, which limits the model from providing accurate results. Inspired by the success of the transformer model in natural language processing, this paper proposes the use of transformer encoder-decoder architecture with graph attention networks (GATs) to predict worker trajectories on construction sites. The temporal interactions of the workers are captured by the transformer model, while GAT captures the spatial relationships of the workers, which allows the model to build more comprehensive view of the workers behavior. The model is able to take 8 frames, covering 3.2 seconds, and predict the next 12 frames, covering 4.8 seconds, with an average displacement error (ADE) of 1.25 m and a final displacement error (FDE) of 2.3 m. The proposed model improves performance compared to traditional methods such as LSTM.

1 Introduction

In the coming few years, the construction industry will be one of the main drivers of the global economy. Its growth is anticipated to increase by an average of 4.4% between 2020 and 2025, exceeding the growth rates of the manufacturing and services industry [12]. With the rise of new real estate construction, the modular construction market in Europe and the United States will exceed a market value of $130 billion by 2030 [3]. Modular construction is shifting operations

*Corresponding author

I. Hkust, J. Cheng and Y. Yantao (eds.), ICCBEI 2025 (Kalpa Publications in Computing, vol. 22), pp. 781–791

from an on-site to an off-site fabrication facility. The fabricated building components are then transported to the site for installation [14]. One of the main reasons for the rise of modular construction is its elevated safety standards, as emphasized by [7]. Regardless, fabrication facilities are dynamic working environments where unsafe scenarios may occur. These events result from congested working environments and the utilization of construction equipment, such as robotic machines [13]. There are several reasons for such hazardous situations: workers' lack of positional awareness when working with construction machinery, which can be due to its sudden movement and rotation, obscured sightlines, and ambient noise [13]. Monitoring and predicting the future trajectories of workers is essential for mitigating the risk of potential injuries arising from such unsafe situations. This approach yields vital information that can be employed to create a contact collision alarm system.

Several studies have proposed methods for trajectory prediction in the construction industry. For example, [21] presented a method designed to predict the trajectory of workers and mobile equipment using a Kalman filter. In this method, the trajectories were obtained using multiple video cameras. However, this method relies on past trajectories to predict future trajectories, neglecting other important information, such as the influence of other entities. The behavior of workers is influenced by their interaction with others and the task they are working on; this is due to the dynamic working environment in a construction site, where multiple entities coexist [4]. Another study proposed to model the worker's trajectory as a Markov process, and the prediction of future trajectories depends on past historical records. However, this method relies on hand-crafting the features, which is hard to implement for broad range scenarios.

Data-driven methods have emerged as viable techniques in trajectory prediction due to rapid advances in computational power and deep learning techniques. For instance, [4] proposed a method for trajectory prediction using Long-Short-Term Memory (LSTM). This method takes advantage of both the individual trajectories, represented by the past trajectories, and the contextual information, represented by the movement of the target neighbors, information about the working group, and the final destination. However, error accumulation is one of the drawbacks of using LSTM, since the model uses the output from the last step to predict the following step, affecting its effectiveness. In 2020, [5] proposed a transformer inspired trajectory prediction model. Transformer networks showed a promising result compared to other methods. This paper expands on Giuliari et al. model by integrating graph attention networks and proposes a complete framework from detection to prediction.

2 Related Work

With the advances in the field of autonomous vehicles, trajectory prediction emerges as one of the fundamental challenges that need to be solved, especially in applications such as analyzing pedestrian behavior [4]. Bayesian models, probabilistic planning, and data driven methods are the three main approaches used in the field of trajectory prediction. Firstly, Bayesian models, [9] is considered one of the pioneers who introduced the use of the Extended Kalman filter algorithm for trajectory prediction. This algorithm alternates between a prediction step, where current internal states are computed based on previous states, and an update step, where the current states are refined using observations. In addition, [2] proposed a Discrete Choice framework, and Treuille et al. introduced continuum dynamics. One of the disadvantages that hinder the use of Bayesian models in trajectory prediction is the need for handcrafting the internal states, which can be time consuming and very difficult to implement in complex situations [19, 6]. Secondly, probabilistic planning is where the problem we have is modeled using a Markov decision process. This method aims to find the optimal solution by maximizing a reward function and ultimately

reaching the goal [10]. However, a key challenge in implementing probabilistic planning methods is accurately defining a reward function, specifically when dealing with real word trajectory prediction tasks. Thirdly, data-driven approaches show great promise for handling complicated trajectory prediction issues. For instance, [20] proposed a Convolutional Neural Network (CNN) technique for predicting pedestrian behavior in an area with high density. On the other hand, LSTM networks can detect spatiotemporal information in sequential data. Several research publications proposed using LSTM networks to handle trajectory prediction difficulties. For example, [1] added a social pooling layer to an LSTM network to allow LSTM cells to share information internally. In addition, [6] proposed combining graph attention and LSTM networks to capture the spatial and temporal interactions at each time step. Furthermore, an environment aware LSTM model was proposed by [19]. The proposed model improved the vanilla LSTM model by introducing worker to worker and environment to worker interactions to the model input. [5] introduced a new model that draws inspiration from the renowned transformer networks commonly employed in Natural Language Processing (NLP), as described in the paper "Attention Is All You Need" [15]. The suggested model represents each person independently, with no human-human or scene interaction terms. As a result, this paper aims to extend [5] research by introducing graph attention networks to capture spatial interactions.

3 Method

During construction activities, construction workers rely on their environmental awareness, such as accounting for walking paths, other construction equipment, and other construction workers, to help mitigate any risk of collision. To integrate such vital information, we propose incorporating transformer networks to capture the temporal interactions and graph attention networks to capture the spatial relationships between construction workers. This becomes particularly essential given that the original transformer model lacked the incorporation of contextual information beyond historical trajectories. Figure 1 shows the proposed methodology. In order to automate the trajectory prediction, YOLOv10 model proposed by [17] was used to for object detection and to keep track of the workers present in the scene DeepSort proposed by [18] was incorporated to the overall framework. Once worker detection and tracking are completed, further data preprocessing is implemented to prepare the data for the training phase. In the model training, the model is initially trained using ETH dataset to initialize the model parameters and using transfer learning the model is re-trained using the new construction dataset. In the following sections, each phase is presented in detail.

3.1 Worker Detection and Tracking

In order to automate the processes of detecting and tracking workers and panels, YOLOv10 model was trained using real world construction dataset. In addition to using YOLOv10, DeepSort algorithm was utilized to add another parameter to the dataset, which is tracking ID. This will help in preparing the data to match the input format.

Before starting the object detection phase, a pre-trained model is needed to be used during inference. Therefore, a real-world construction dataset is used to train the YOLOv10 model to detect workers. The scenes were taken from a surveillance camera inside an off-site construction facility in Edmonton, Alberta. The construction dataset contains two scenes where workers were working on different building components such as floors and panels. The construction dataset contains one class: (1) Workers. For consistency with the literature, the scene was annotated at 2.5 fps meaning each frame will be annotated every 0.4 seconds. As [8] research

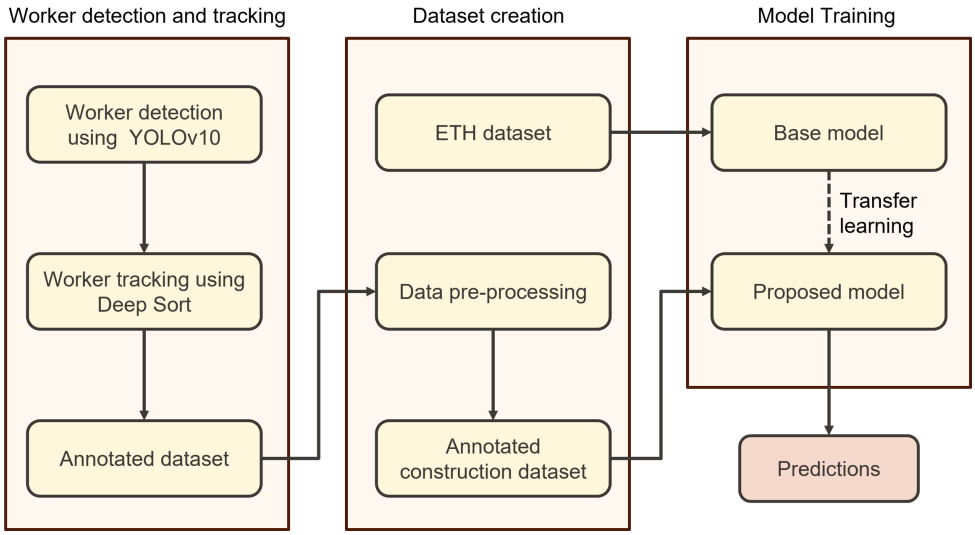

Figure 1: Overview of the proposed methodology

found that increasing the observation period does not increase the accuracy of the prediction in a construction setting. In the first scene, we annotated 1951 frames with 4 workers. In the second scene, we annotated 212 frames with 4 workers. The annotation was done using Roboflow website. With the help of augmentation functions within their website, data augmentation was implemented to enhance the model's performance. The total images used after applying data augmentation is 9388 images.

Tracking workers and moving obstacles is important to help improve the predictive performance of the proposed model. Therefore, DeepSort algorithm was utilized with a trained YOLOv10 model to track the IDs of workers. The extra parameter gained from this step is used later to aggregate the movements based on the IDs. For instance, let's assume worker number 1 appears for 2 minutes. Therefore, in the data pre-processing phase, the worker trajectory can be grouped into a single sequence and then prepare it for training.

3.2 Data Creation

In this paper, two datasets were used for training and fine-tuning the proposed model. The first dataset is described in the following section will be used to train a base model in order to initialize the model wights. Then, a new construction dataset is created to be used to fine tune the base model using transfer learning.

ETH name was derived from ETH Zurich University [11], this dataset contains two scenes in a bird's eye view from RGB cameras that are usually used for surveillance purposes: (1) Eth; and (2) Hotel. Eth scene was taken from the top of the ETH main building, and the annotation was done at 2.5 fps, which means that one frame is annotated every 0.4 seconds. This part comprises of a total of 365 different pedestrian trajectories. For Hotel, the scene was taken from the 4th floor of a hotel in Bahanhofstr in Zurich and the video was taken at a 25 fps, but the annotation was done at 2.5 fps for consistency. The original annotation format is listed below:

Figure 2: Overview of the proposed methodology

$$[\text{frame number}, \text{pedestrian\_ID}, pos_x, pos_z, pos_y, v_x, v_z, v_y]$$

However, in our case the format is changed to a new format presented below:

$$[\text{frame number}, \text{pedestrian\_ID}, pos_x, pos_y]$$

The second dataset used in this paper is a real-world construction dataset. This dataset is the output form worker detection and tracking phase after pre-processing with a total of 1900 frames. The data pre-processing comprises of taking the rectangle coordinates of the detected worker and converting it into a single point, in our case the center of the rectangle. After that the pixel coordinates are converted to real world coordinates. Resulting in [pos_x, pos_y]. For frame number can be obtained from the YOLOv10 results and pedestrian ID can be obtained from DeepSort results. Therefore, the final format will follow the same format as ETH dataset. A sample frame from the construction dataset is shown in Figure 2.

3.3 Model Training

We propose incorporating Graph Attention Networks (GAT) [16] into the transformer encoder-decoder architecture to effectively decode spatial information of workers, as depicted in Figure 3. GAT operates on graph-based data using self-attention to quantify the influence of each neighboring node on a target node. GAT comprises two main components: (1) nodes, representing individual data points (workers), and (2) edges, denoting relationships between nodes.

The transformer architecture consists of three components: positional encoding, encoder, and decoder. Positional encoding compensates for the transformer's inability to inherently process temporal data, employing sine and cosine functions to encode positional information:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right) \tag{1}$$

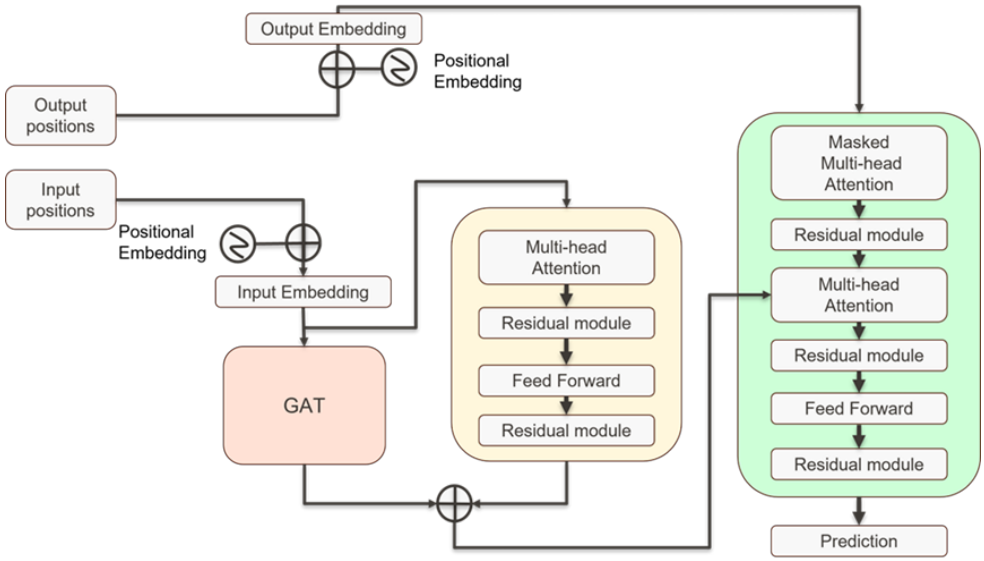

Figure 3: Overview of proposed model architecture

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{2}$$

where pos is the token's position, i is the dimension index, and d represents the model dimension.

The encoder's primary function is encoding source inputs for subsequent decoding. It comprises N identical layers (typically $N = 6$), each containing multi-head attention and fully connected feed-forward networks (FFN), interconnected by residual connections and layer normalization. The attention mechanism employed is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{3}$$

where Q, K, and V denote the query, keys, and values matrices, respectively, and d_k represents the key dimension.

The decoder leverages encoder and GAT outputs to generate target sequences, employing two attention modules: one using encoded encoder outputs and another applying masked output embeddings.

The model was implemented using PyTorch on a desktop with an RTX 3060 GPU, AMD Ryzen 9 5900 CPU, and 32GB RAM. Training utilized the Adam optimizer with a learning rate of 0.001, a batch size of 64, and 600 epochs.

Initially, the model was trained on the ETH dataset, then transfer learning was applied, and the model was retrained using a real construction dataset. The dataset was split as follows: 70% for training, 20% for validation, and 10% for testing the model performance. The model will take eight frames as input and predict twelve future frames. We conducted tests using actual construction operations data to assess the trajectory prediction accuracy of the trained models. Our evaluation metrics include both average displacement error (ADE) and final displacement error (FDE) [11]. ADE is calculated as the mean square error across all predicted positions

of the trajectory and the corresponding ground-truth trajectory, represented by the following formula:

$$\text{ADE} = \frac{\sum_{i=1}^{N} \sum_{t=T_{obs}}^{T_{pred}} \|(\hat{x}_i^t, \hat{y}_i^t) - (x_i^t, y_i^t)\|}{N \times (T_{pred} - T_{obs} - 1)} \tag{4}$$

In the given formula, N represents the number of workers, and $(\hat{x}_i^t, \hat{y}_i^t)$ and (x_i^t, y_i^t) denote the predicted coordinates of worker i at time instant t and the corresponding ground-truth coordinates of worker i at time instant t, respectively. The notation $\|\cdot\|$ represents the Euclidean distance. FDE is determined as the distance between the predicted final destination and the corresponding ground-truth destination, represented by the following formula:

$$\text{FDE} = \frac{\sum_{i=1}^{N} \|(\hat{x}_i^{T_{pred}}, \hat{y}_i^{T_{pred}}) - (x_i^{T_{pred}}, y_i^{T_{pred}})\|}{N} \tag{5}$$

Where N represents the number of workers, and $(\hat{x}_i^{T_{pred}}, \hat{y}_i^{T_{pred}})$ and $(x_i^{T_{pred}}, y_i^{T_{pred}})$ denote the predicted coordinates of worker i at time instant T_{pred} and the corresponding ground-truth coordinates of worker i at time instant T_{pred}, respectively. The notation $\| \cdot \|$ indicates the Euclidean distance. The units of ADE and FDE are in meters, given that both the ETH dataset and the construction datasets utilize real-world coordinates.

4 Results and Discussion

In this section, the results obtained from the proposed model are discussed. First, the results from the training YOLOv10 are presented. Second, the results obtained from the proposed model are compared with the literature to identify the improvements.

4.1 YOLOv10 Results

The training setup was as follows: a total of 9,288 images were used for training, validation, and testing, with a split of 70%, 20%, and 10% for training, validation, and testing, respectively. The training was conducted on a PC equipped with an NVIDIA RTX 3060 GPU with 12 GB of memory, an AMD Ryzen 9 5900 12-Core Processor at 3.00 GHz, and 32 GB of RAM. The model was trained for 400 epochs with a batch size of 64 and an image size of 640.

The model achieved the following results: a precision of 99.39%, a recall of 99.4%, a mean Average Precision (mAP) of 93.7%, and a mean Average Precision at 50 (mAP50) of 99.4%. The results indicated effectiveness of the model, as it achieved high values of precision, recall, mAP, and mAP50. Figure 4 shows three training losses for three parameters: (1) Box loss, (2) Class loss, and (3) DFL loss. Where Box loss (bounding box regression loss), is the difference between the predicted bounding boxes and the actual. For Class loss, is the loss associated with classification of objects within the predicted bounding box. These two help the model in accurately learn how to predict an approximate location of the bounding box and the class associated with it. Lastly, DFL (Distribution Focal Loss), is a loss function that is used to enhance the bounding box regression localization accuracy. Also, Figure 5 shows the variation of mAP and mAP50 across the training epochs.

The YOLOv10 model results demonstrated its ability to accurately detect both classes with high accuracy. However, performance challenges arise when applying the model to a different working environment. This can be attributed to variations in lighting, different equipment, and

changes in camera positioning. These factors result in an under-performing model that cannot accurately detect the presence of workers.

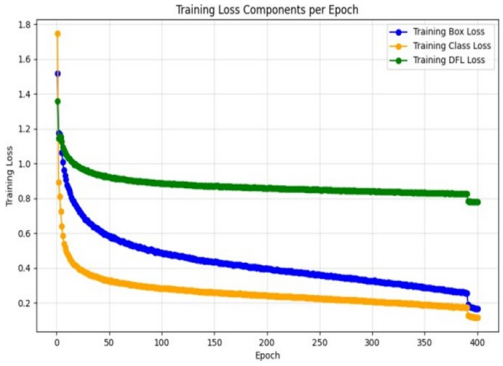

Figure 4: Training loss curves for YOLOv10

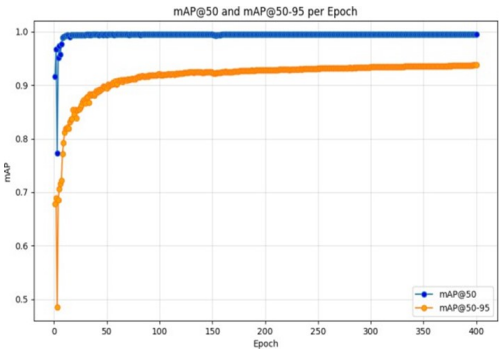

Figure 5: mAP50 and mAP50-95 curves for YOLOv10

4.2 Proposed Model Results

To measure the performance of the proposed model, four models are used as a baseline. The training for all base models was conducted in accordance with a set of parameters to reduce any discrepancies. The observation and the prediction windows were kept the same. As this parameter changes, the overall performance will dramatically change. As discussed in previous sections the observation window will be 8 frames and the model predicts the next 12 frames. Similarly, the batch size and the epochs used are 4 and 600, respectively. The results are summarized in Table 1, which shows that the proposed model outperformed all the baseline models with an improvement percentage range from approximately 27.92% to 31.51% for MAD and an improvement percentage range from approximately 26.57% to 29.84% for FAD.

Model Name	MAD	FAD
LSTM	1.753	3.332
SGAN	1.767	3.343
EA-Distance	1.825	3.349
EA-Direction	1.734	3.201
Proposed Model	1.250	2.350

Table 1: Comparison of MAD and FAD across baseline and proposed models

Figure 6 presents a comparison between a single prediction made by the proposed model and the corresponding ground-truth values. The figure illustrates the actual trajectories of three workers alongside their predicted trajectories. The results indicate that the model successfully captures the overall movement pattern of the workers, even when irregular movements occur, as seen with worker 3. However, in some cases, the predicted trajectories' lengths do not align with the ground truth. A possible explanation for this discrepancy lies in the structure of the dataset itself. Specifically, the workers' movements are significantly different from those of pedestrians. Workers typically have prolonged stops while completing tasks before moving to

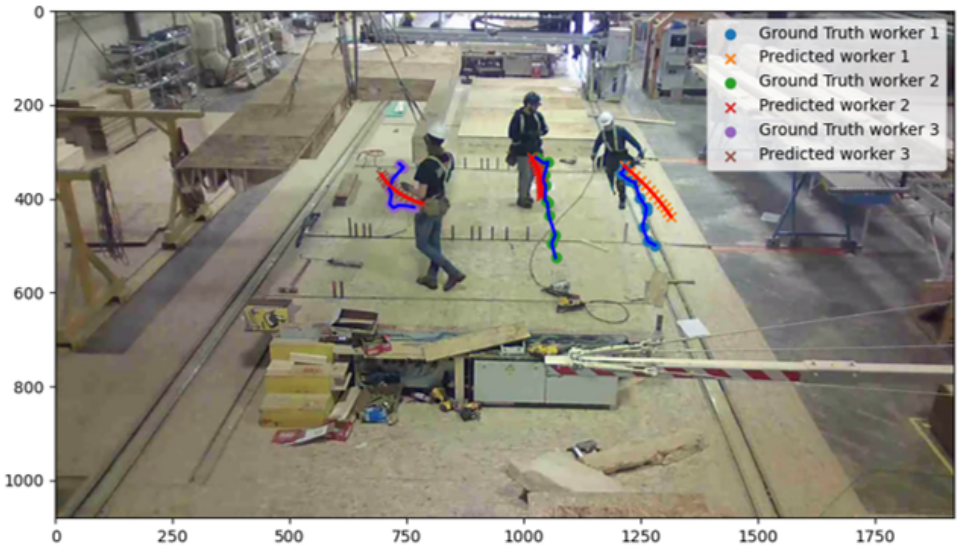

Figure 6: Predicted vs ground truth trajectories of the proposed model

the next location. As a result, their trajectories should reflect a balance between long and short movements.

5 Conclusion

In conclusion, this study has presented an innovative approach to predicting worker trajectories on dynamic construction sites by integrating transformer encoder-decoder architecture with graph attention networks (GATs). The proposed model effectively captures both the temporal and spatial interactions of workers, enabling more accurate trajectory predictions. By leveraging both transformers for temporal analysis and GATs for spatial context, our model outperforms traditional methods, such as LSTM, particularly in mitigating error accumulation and addressing the complex, interactive environments characteristic of construction sites. Testing with real-world data validated the model's superior performance, demonstrating lower average displacement error (ADE) and final displacement error (FDE) than baseline methods.

This trajectory forecasting framework holds substantial potential for enhancing worker safety by forming the basis for collision warning systems in construction environments. Future work could explore adapting the model to various construction environments and expanding it to incorporate additional dynamic elements, such as machinery or varying site conditions, to further enhance robustness.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[2] Gianluca Antonini, Michel Bierlaire, and Matthieu Weber. Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological*, 40(8):667–687, 2006.

[3] Nick Bertram, Stefan Fuchs, Jan Mischke, Richard Palter, Gernot Strube, and Jonathan Woetzel. Modular construction: From projects to products. Technical report, McKinsey & Company, 2019. Retrieved from https://www.mckinsey.com/~/media/mckinsey/business%20functions/operations/our%20insights/modular%20construction%20from%20projects%20to%20products%20new/modular-construction-from-projects-to-products-full-report-new.pdf.

[4] Jinhui Cai, Yifan Zhang, Liang Yang, Hao Cai, and Shuang Li. A context-augmented deep learning approach for worker trajectory prediction on unstructured and dynamic construction sites. *Advanced Engineering Informatics*, 46:101173, 2020.

[5] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10335–10342, 2021.

[6] Yuxiao Huang, Hao Bi, Zhen Li, Tian Mao, and Zhaoxiang Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6271–6280, 2019.

[7] Ruoyu Jin, Jianzhong Hong, and Jian Zuo. Environmental performance of off-site constructed facilities: A critical review. *Energy and Buildings*, 207:109567, 2020.

[8] Dong Kim, Mingxiang Liu, SangHyun Lee, and Vineet R. Kamat. Trajectory prediction of mobile construction resources toward pro-active struck-by hazard detection. In Mohamed Al-Hussein, editor, *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 982–988, 2019.

[9] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[10] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. arXiv preprint arXiv:1604.01431, 2017.

[11] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, pages 261–268, 2009.

[12] Graham Robinson, Jim Leonard, Tom Whittington, Oxford Economics, Marsh, Guy Carpenter, and Jane Turner. Future of construction. Technical report, Oxford Economics, 2021. Retrieved from https://www.oxfordeconomics.com/wp-content/uploads/2023/08/Future-of-Construction-Full-Report.pdf.

[13] Jochen Teizer and Tao Cheng. Proximity hazard indicator for workers-on-foot near miss interactions with construction equipment and geo-referenced hazard areas. *Automation in Construction*, 60:58–73, 2015.

[14] Huu-Tai Thai, Tuan Ngo, and Brian Uy. A review on modular construction for high-rise buildings. *Structures*, 28:1265–1290, 2020.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[16] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[17] Alex Wang, Hang Chen, Lu Liu, Kai Chen, Zhihang Lin, Jingkuan Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.

[18] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017.

[19] Qijun Yang, Qipei Mei, Chuang Fan, Ming Ma, and Xiangyu Li. Environment-aware worker trajectory prediction using surveillance camera in modular construction facilities. *Buildings*, 13(6):1502,

2023.

[20] Shuai Yi, Hongsheng Li, and Xiaogang Wang. Pedestrian behavior understanding and prediction with deep neural networks. In *Computer Vision – ECCV 2016*, volume 9905, pages 263–279. Springer International Publishing, 2016.

[21] Zhenhua Zhu, M.-W. Park, Christian Koch, Morteza Soltani, Amin Hammad, and Kaveh Davari. Predicting movements of onsite workers and mobile equipment for enhancing construction site safety. *Automation in Construction*, 68:95–101, 2016.