Integrating Pedestrian Simulation into Early Building Design: A Deep Learning Approach for Trajectory Prediction

Patrick Berggold[1], Stavros Nousias, Ph.D.,[2] and André Borrmann, Prof. Dr.-Ing.[3]

[1]Chair of Computational Modeling and Simulation, TUM School of Engineering & Design, Technical University of Munich, Germany (corresponding author). e-mail: patrick.berggold@tum.de
[2]Chair of Computational Modeling and Simulation, TUM School of Engineering & Design, Technical University of Munich, Germany; e-mail: stavros.nousias@tum.de
[3]Chair of Computational Modeling and Simulation, TUM School of Engineering & Design, Technical University of Munich, Germany; e-mail: andre.borrmann@tum.de

ABSTRACT

The incorporation of pedestrian simulators during the early phases of a building design process remains impractical. Although they are essential tools to assess the operational and evacuation performance of a building, their integration is challenging due to time-intensive runtimes and laborious export and conversion steps when generating simulation results. Deep learning methods have demonstrated their ability to generate results instantly with sufficient, or even outstanding performance across many disciplines. In this article, we present the development of a neural network that can process both sequential and image data simultaneously to investigate its capability to reproduce simulation results via supervised learning. Unlike previous approaches, we directly predict trajectories based on floorplans derived from a parametric Building Information Modeling (BIM) model. Our findings show that the network can capture short-term relations effectively considering previous agent states and their surrounding environments, while long-term predictions remain challenging.

INTRODUCTION

Throughout the design and development phases of a building, numerous experts from the Architecture, Engineering, and Construction industry collaborate. Consequently, this leads to an enormous degree of inter-dependence between the disciplines that are involved in this process, such that insufficient or erroneous communication and collaboration in the early design stages can result in substantial temporal or economic expenses in later stages (Gervásio, Santos, Martins, & da Silva, 2014). The Building Information Modeling (BIM) methodology addresses these challenges by creating a centralized, collaborative digital platform that integrates various

aspects of the construction process throughout the building life cycle (Borrmann, König, Koch, & Beetz, 2018).

Simultaneously, in current practice, incorporating pedestrian simulations into the building design – particularly in the early stages – remains laborious due to the lack of fast and automated generation of simulation results (Clever, Abualdenien, & Borrmann, 2021), as simulation runtimes may extend to several minutes for large, complex buildings. Moreover, pedestrian simulators are often standalone applications, leading to manual and cumbersome conversion steps for preparing input data and visualizing simulation results. This involves exporting the BIM model into vendor-neutral exchange formats like IFC (Industry Foundation Classes) and subsequent conversion into the specific simulation input format. Consequently, pedestrian simulators are rarely employed to assess every building variant discussed during the initial project phases (Asriana & Aswin, 2016).

With the advances in Artificial Intelligence (AI), and particularly deep learning, several approaches have been developed to emulate pedestrian simulator results for instant prediction. Current research partially focuses on evacuation indicators (e.g. evacuation time) or macroscopic quantities of interest, such as density or flow. In contrast, the major advantage of forecasting trajectories lies in eliminating the need for any pre-/post-processing steps that convert trajectories into macroscopic quantities. Thereby, results are available at every single timestep, providing fine-grained insights into the movements of individual agents – the virtual pedestrians. While trajectory prediction of humans has also been studied in previous research, model benchmarking typically relies on the few publicly available real-world datasets and solely compares short-term predictions. This limitation is impractical for the evaluation of many floorplan variants, which requires models to be trained on large datasets, incorporating predictions over long-term simulation runtimes. In addition, there is significant interest in non-supervised approaches, such as reinforcement learning. However, these methods do not guarantee compliance with pedestrian simulations through supervision, which is crucial for ensuring safety. Thus, supervised, long-term trajectory prediction in building design context is missing in the current research landscape.

In this article, we aim to address these challenges by investigating a supervised data-driven approach that is capable of learning large quantities of synthetic trajectory data to emulate simulation results, utilizing a pedestrian simulator and a parametric BIM model. In the next section, the related works concerning evacuation simulations and trajectory prediction are discussed. Section 3 describes the methodology of our approach and introduces the neural network. Subsequently, the results and the conclusion are presented.

RELATED WORK

Evacuation simulations. The simulation of pedestrian and crowd movement is essential to investigate evacuation outcomes in what-if scenarios, and to identify potential bottlenecks with minimal cost and risk compared to real-life evacuation drills (Şahin, Rokne, & Alhajj, 2019). Since the architectural layout of a building represents a principal factor in pedestrians'

wayfinding capabilities (Natapov, Parush, Laufer, & Fisher-Gewirtzman, 2022), several simulation models have been developed in recent years. The Social Force Model (Helbing & Molnar, 1995) is particularly popular; it utilizes an equation that accounts for the repulsive and attractive forces between obstacles and destinations. Moreover, another common model is the Cellular Automaton (Burstedde, Klauck, Schadschneider, & Zittartz, 2001), which discretizes space into a grid in which individuals can move from one cell to a neighboring cell. In this article, we employ a simulator that is based on the Optimal Steps Model (OSM) (Seitz & Köster, 2012). The OSM divides time into sequential frames within continuous space, in which the agent tries to maximize its utility function by reaching its assigned destination.

AI-based evacuation tools. The recent advances of AI and deep learning techniques across various domains have been extended to the domain of evacuation simulations as well. For instance, total evacuation time (TET) is predicted through a convolutional neural network based on colored floorplan images of train stations in (Clever, Abualdenien, Dubey, & Borrmann, 2022). Similarly, Abadeer, Ebeid and Gorlatch use different regression techniques to estimate TET of a university building structure plan (Abadeer, Ebeid, & Gorlatch, 2022). Both methods employ the OSM to generate the datasets for training their machine learning algorithms. The authors of (Testa, Barros, & Musse, 2019) utilize a neural network to estimate TET by evaluating evacuation time of each individual room, given its geometry and population. Furthermore, more meaningful quantities with regards to the safety – not only the speed – of an evacuation have been developed. Berggold, Nousias, Dubey and Borrmann simultaneously predict time-dependent densities and TET for office buildings using a Vision Transformer (Berggold, Nousias, Dubey, & Borrmann, 2023). A similar image-to-image approach is developed in (Nourkojouri, Dehnavi, Bahadori, & Tahsildoost, 2023), forecasting density heat maps while employing XGBoost to assess TET. Finally, Sohn et al. propose a framework to predict the aggregation of crowd densities over the entire course of a simulation (Sohn, et al., 2020).

While the aforementioned approaches provide realistic insights into crowd movement in evacuation simulations through macroscopic quantities (e.g. density) or evacuation indicators (e.g. TET), they do not discuss the prediction of microscopic quantities, essentially trajectories, which is the striking difference to our work presented in this article.

Human Trajectory Prediction (HTP). Meanwhile, trajectory prediction models exist in other fields, e.g. autonomous driving or robotics. The task of reproducing pedestrian simulator results is slightly dissimilar to forecasting real-world trajectories, mainly because agent destinations are pre-defined in simulations, whereas they must be predicted in real-world context. Furthermore, simulations typically encode social interactions as hand-crafted transition rules or potential functions, simplifying the complex nature of human behavior. Nonetheless, existing trajectory prediction models from other fields provide valuable insights from which we draw inspiration. For instance, both in (Gupta, Johnson, Fei-Fei, Savarese, & Alahi, 2018) and (Sadeghian, et al., 2019), LSTM-based GAN modules are used to predict human trajectories in crowded spaces

with different pooling mechanisms for modelling social interactions. Due to the success of Transformer-based networks, recent works have adopted this architecture. In (Giuliari, Hasan, Cristani, & Galasso, 2021), a vanilla Transformer is employed for predicting trajectories on common benchmark datasets without any social interaction terms, resulting in competitive performance compared to previous models. More advanced architectures have been developed recently, for instance in (Yuan, Weng, Ou, & Kitani, 2021), where the attentional module is altered via a masking operation to attend to inter-agent and intra-agent features differently. Furthermore, other approaches have discussed the idea of intermediate goal prediction. In (Chiara, et al., 2022), an additional module is employed to support the trajectory prediction by sampling potential intermediate goals based on the environment and observed trajectories.

METHOD

Overview. Our approach focuses on developing a deep learning framework that can accurately and almost instantly predict a coordinate sequence of maximum length for all agents present in the simulation, based on a fixed number of previously observed agent states. For creating a comprehensive training dataset, we construct a parametric BIM model that resembles office building floors. To achieve a sufficient degree of geometric variability in the dataset, we incorporate several building geometry variations, ranging over the building length and width, hallway width, number of rooms, as well as the presence or absence of hallway obstacles and bottlenecks (essentially doors) in front of the destination areas.

Length	Width	Hallway width	Number of rooms	Destination bottleneck	Obstacle presence	Agents per origin
25 m	20 m	2 m	6	{present, absent}	{present, absent}	{10,20,30}
35 m	20 m	3 m	6	{present, absent}	{present, absent}	{10,20,30}
40 m	25 m	4 m	7	{present, absent}	{present, absent}	{10,20,30}

Table 1. Overview of the input parameters to our dataset

Table 1 provides an overview of the dataset's geometric variations. After generating the BIM model in Autodesk Revit, we must first export its floorplan into an IFC file, and subsequently convert it both into RGB image and simulator input format. To further increase the diversity in our dataset, we introduce variations with respect to the simulation parameters as well. Every room may or may not serve as origin area for the agents (denoted in red color), while either end or both ends of the hallway may serve as destination area (denoted in green color) to simulate evacuation scenarios. Specifically, in case of an emergency, individuals must leave their offices at once to reach the exits or staircases. For each floorplan, we include every combination of

origin and destination areas in our dataset. Finally, to cover different agent volumes, we use 10, 20 and 30 agents per origin area for each variant. In total, our dataset encompasses 9,108 floorplan variations and simulation runs, incorporating the associated trajectory data. Each simulation is essentially represented by a four-dimensional table with the timestamp, agent id, x- and y-coordinates as columns, where the agent states are sampled every 0.5 seconds.

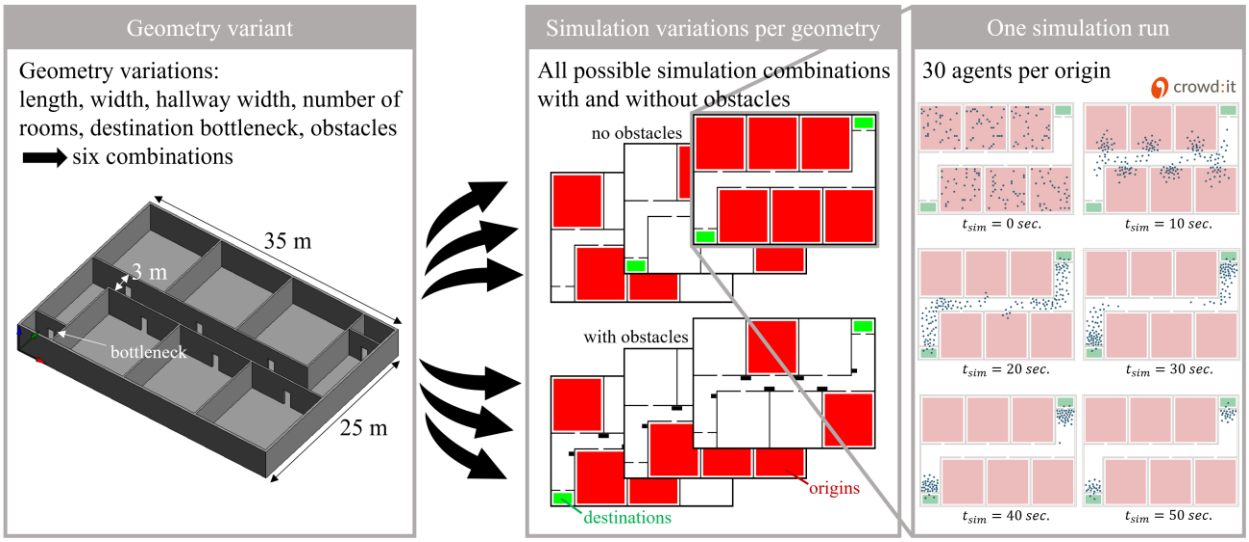

Figure 1. The parametric BIM model represents basic office building floors. From the BIM model, colorized floorplans and the associated simulation files are exported and converted to generate the dataset.

The left side of Figure 1 visualizes our parametric BIM model, which is shaped based on its input parameters. On the right side, the simulation process is displayed, initializing all agents at once into their associated origin areas, from which each agent moves towards its destination. As mentioned above, these simulation settings are supposed to replicate an evacuation scenario. When the destination is reached, the agent terminates, leaving the simulation scope. We utilize the pedestrian simulator *crowd:it* (accu:rate, 2024), which is configured in our experimental setup to provide a Gaussian agent velocity distribution between 0.5 and 1.6 m/s, with standard deviation of 0.26 m/s and mean of 1.34 m/s that represents the approximate normal walking speed of pedestrians (Weidmann, 1992). The agent torso size is uniformly distributed in a radius between 0.42 and 0.46 meters.

Neural network architecture. Our approach integrates both global and local scene information for each agent involved in the simulation. These inputs are provided as N semantic maps with one channel each, where N represents the number of agents. Each global agent map comprehensively represents the entire scene, encompassing non-walkable or repulsive areas, such as walls and obstacles, while also incorporating the agent's specific destination designated as attractive area. The local maps are centralized perspectives for each agent, represented as a square of fixed size with each agent at the center. These squares contain an occupancy grid

around the agents, in which the agent itself, as well as its neighbors, are represented as a Gaussian distribution whose mean is placed at the agent location. Its standard deviation is the agent's torso size, which is a simulation parameter. In the maps, repulsive areas are assigned a value of +1, contrasting with destination areas, which are marked with -1. The rest of the walkable regions within the scene are denoted by zeros.

As displayed in Figure 2, the maps are passed through a Resnet-18 backbone (He, Zhang, Ren, & Sun, 2015) to extract and encode semantic image information, for instance to consider nearby agents or obstacles. While the hidden states are combined with the sequence information for predicting the next steps, a reconstruction head is attached to the backbone to enable pre-training the weights via image reconstruction. Notably, the image encoder's weights remain frozen after pre-training to reduce computational load and convergence effectiveness when performing sequence prediction. To encode the previous coordinates of N agents, we utilize a Transformer encoder. The output of the batch-wise self-attention is concatenated with the hidden states of the semantic map input, and passed through a multilayer perceptron (MLP) to predict each agent's next step represented by its subsequent x- and y-coordinates, which also results in a new set of centralized local occupancy maps. For the next timestep prediction, the output coordinates are appended to the observed states. Meanwhile, previous local maps are replaced with current ones for the next predictions to update interaction information continuously together with the agent state. This autoregressive training method is quite common in trajectory prediction tasks, enabling the model to iteratively refine predictions based on the evolving context.

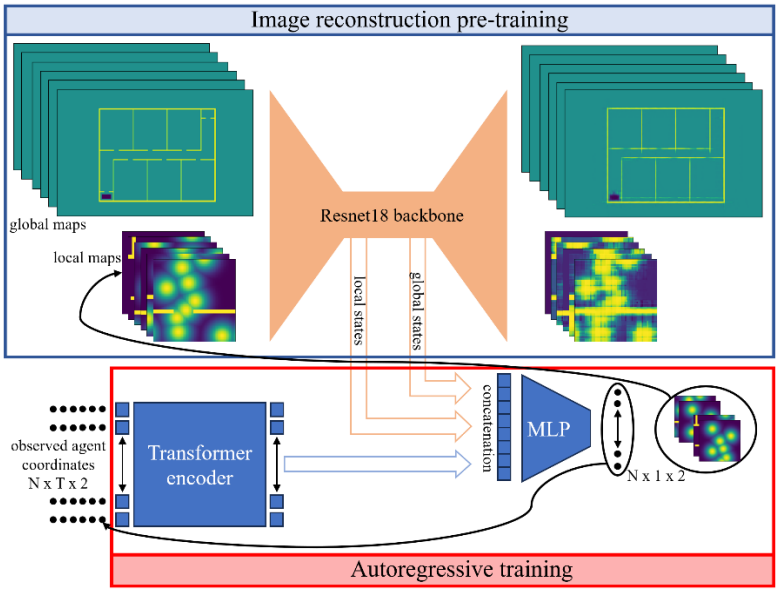

Figure 2. Our neural network architecture, including a pre-trained image encoder, and a Transformer encoder for capturing observed sequences. The lower part is trained autoregressively, with previous predictions used as inputs for subsequent predictions.

The transformer encoder encompasses three self-attention layers with hidden dimension 512. The MLP includes three linear layers with two intermediate dimensions of size 256, each combined

with the corresponding layer norm and leaky ReLU activation with factor 0.2. Our image encoder is built through four Resnet-blocks, plus an image reconstruction head consisting of three transposed convolutional layers that increase resolution and decrease dimensionality from the hidden space. In total, our network has approximately 18M parameters, with 12.5M parameters for the image encoder and 5.5M parameters for the Transformer and MLP layers.

RESULTS

We run several experiments with the network architecture displayed in Figure 2, with each run fixed to 50 epochs. Specifically, we split each simulation into distinct sequences, where a sequence consists of O observed steps and T future steps for all agents involved in the simulation. Therefore, one dataset sample encompasses O observed steps for all agents, plus the semantic maps as input to the network, while T future steps of N agents represent the targets. In our experiments, we vary O and T to investigate how well the network can reproduce simulator output for short and long sequences, and thereby its capability to model complex agent-agent and agent-environment relations across different floorplans. To the best of our knowledge, no comparable supervised trajectory prediction experiments are available on such large, synthetic datasets.

A set of augmentations are applied to both trajectories and semantic maps. Initially, a padding operation fixes the global input maps to a uniform size. Subsequently, translations and flipping along both axes, as well as transposing and random rotations around 90 degrees are applied. The network is trained with an initial learning rate of 3e-4 and a ReduceOnPlateau scheduler with factor of 0.5 and 7 epochs patience. We use the Mean-Squared-Error (MSE) as loss function, and present the results in terms of the Average Distance Error (ADE) and Final Distance Error (FDE) metrics in Figure 3, which are well established in the Human Trajectory Prediction community and more intuitive than MSE. Notably, we present the results solely from the test set that comprises 15% of the original dataset (where the remaining samples are used for training and validation). ADE evaluates the average distance between predicted coordinate $\hat{y}_{n,t}$ and actual coordinate $y_{n,t}$ at each time step *t* for each agent *n*, while FDE measures the distance between the predicted position and the ground truth position of a trajectory at final timestep T:

$$ADE = \frac{\sum_{n \in N} \sum_{t \in T} \|\hat{y}_{n,t} - y_{n,t}\|}{N \cdot T} \quad , \quad FDE = \frac{\sum_{n \in N} \|\hat{y}_{n,T} - y_{n,T}\|}{N}$$

We run all combinations on 2, 4, 8, 12 and 16 observed and predicted steps, as displayed in Figure 3, with the colormap aligned to the ADE. Firstly, we observe a realistic pattern in the predictions, namely the network's ease to predict short-term sequences for two and four predicted timesteps, with ADE and FDE values around 0.3 meters and 0.5 meters, respectively. Intuitively, we also see increasing errors with fewer observed steps, as the self-attention mechanism of the Transformer encoder relies on the input sequence to establish relationships between states across observed agent trajectories. As the number of observed steps decreases, the

model's ability to capture essential context diminishes, making it progressively harder to predict accurate trajectories for longer sequences. This becomes problematic particularly closely after agent initialization, when only few previous states are observed.

Finally, although it is difficult to compare our results to state-of-the-art HTP baselines due to the inherent differences in the datasets (large, synthetic ones vs. small, real ones), we do observe that the ADE is confidently below one meter for the specific O=8, T=12 combination that is the most common benchmark sequence split. For even longer predictions, for instance T=16 or more, the network struggles to make accurate predictions due to error propagations, with the best performing FDE of approximately 3 meters (provided 16 observed steps). It is evident that long-term predictions remain challenging, as previous literature reviews pointed out.

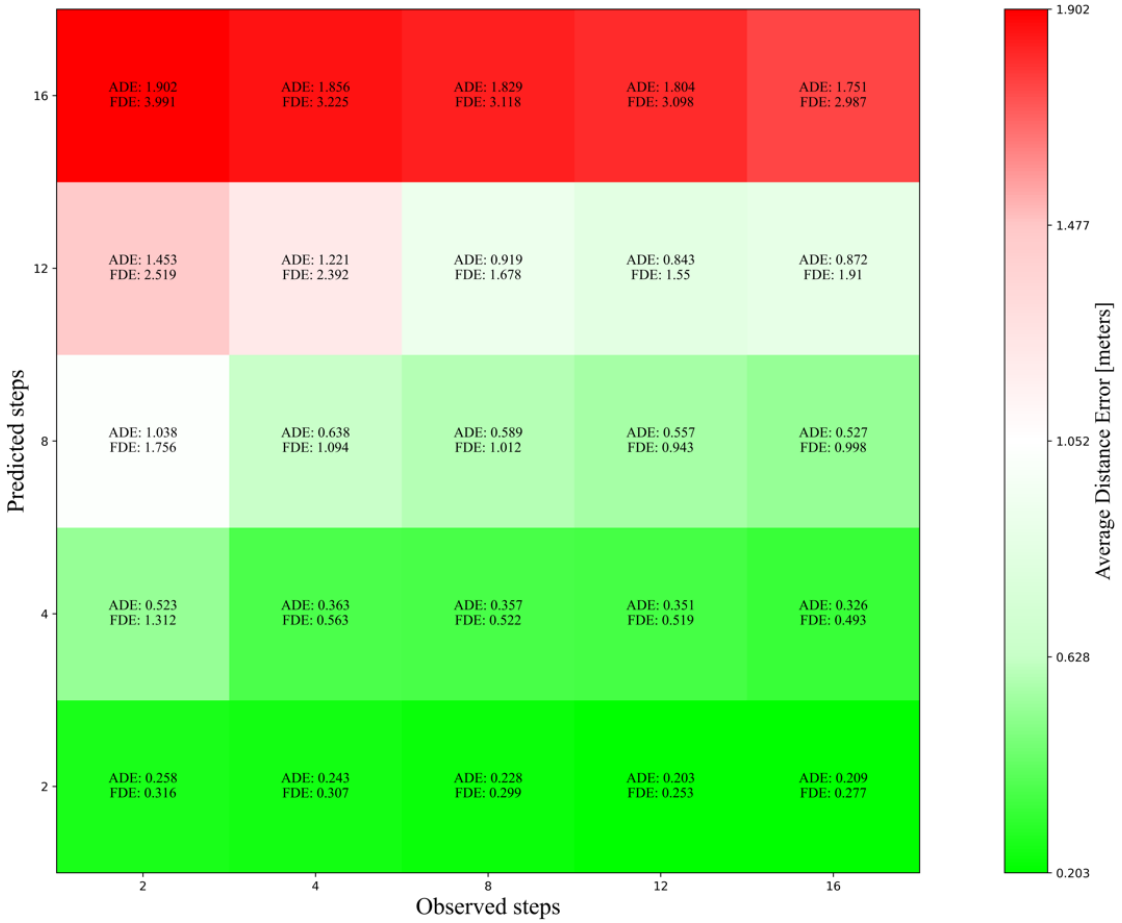

Figure 3. Trajectory prediction results in terms of ADE and FDE for different combinations of observed and predicted steps. The colormap aligns with the ADE.

CONCLUSION AND OUTLOOK

In this article, we investigate a novel approach in predicting pedestrian simulator output via supervised deep learning. Specifically, unlike previous approaches, our network architecture is capable of capturing sequential relations to predict future agent trajectories based on previous

ones, as well as considering complex agent-agent and agent-environment relationships through an image encoder. The dataset to train our neural network is generated through a parametric BIM model, which demonstrates that the network can be integrated into the BIM-driven building design process. The results indicate that our approach is feasible and can accurately predict short-term trajectories up to 12 sequential timesteps. Nonetheless, long-term predictions remain challenging. Furthermore, in order to completely emulate pedestrian simulator output, the network must be able to reconstruct entire trajectories from only the agents' initialization coordinates, which still remains challenging in concurrent research.

In future research, addressing the challenges with respect to long-term sequence predictions requires an enhanced network architecture, more efficient input data formats or advanced training techniques, particularly in the decoding process, given the inherent error propagation that may lead to significant divergence in long sequences. Promising options are network architectures such as Generative Adversarial Networks, Variational Autoencoders or cross-attentional modules in the Transformer decoder for tackling this issue. Furthermore, the recent success in Natural Language Processing suggests that adopting pre-training methods like Next Sequence Prediction or Masked Language Modeling may prove beneficial for trajectory prediction as well.

ACKNOWLEDGEMENTS

We express gratitude for the support from the TUM Georg Nemetschek Institute in funding this work that is part of the FORWARD project.

REFERENCES

Abadeer, M., Ebeid, F., & Gorlatch, S. (2022). Integration of Machine Learning Methods into Agent-based Simulations for Predicting Evacuation Time in Disaster Scenarios. 2022 3rd Asia Symposium on Signal Processing.

accu:rate. (2024). crowd:it – the software for state-of-the-art planners. Last accessed 2024/01/15, from https://www.accu-rate.de/en/

Asriana, N., & Aswin, I. (2016). Making Sense of Agent-Based Simulation: Developing design strategy for pedestrian-centric urban space. Proceedings of eCaade 2016, S. 342-352.

Berggold, P., Nousias, S., Dubey, R. K., & Borrmann, A. (2023). Towards predicting Pedestrian Evacuation Time and Density from Floorplans using a Vision Transformer. 30th Int. Conference on Intelligent Computing in Engineering.

Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). Building Information Modeling: Why? What? How?: Technology Foundations and Industry Practice. Springer.

Burstedde, C., Klauck, K., Schadschneider, A., & Zittartz, J. (2001). Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Physica A: Statistical Mechanics and its Applications.

Chiara, L. F., Coscia, P., Das, S., Calderara, S., Cucchiara, R., & Ballan, L. (2022). Goal-driven Self-Attentive Recurrent Networks for Trajectory Prediction. Conference on Computer Vision and Pattern Recognition.

Clever, J., Abualdenien, J., & Borrmann, A. (2021). Deep learning approach for predicting pedestrian dynamics for transportation hubs in early design phases. EG-ICE Workshop on Intelligent Computing in Engineering.

Clever, J., Abualdenien, J., Dubey, R. K., & Borrmann, A. (2022). Predicting occupant evacuation times to improve building design. ECPPM 2022-eWork and eBusiness in Architecture, Engineering and Construction.

Gervásio, H., Santos, P., Martins, R., & da Silva, L. S. (2014). A macro-component approach for the assessment of building sustainability in early stages of design. Building and Environment, 73, 256-270.

Giuliari, F., Hasan, I., Cristani, M., & Galasso, F. (2021). Transformer Networks for Trajectory Forecasting. International Conference on Pattern Recognition.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., & Alahi, A. (2018). Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. CoRR.

Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics. Phys. Review E.

Natapov, A., Parush, A., Laufer, L., & Fisher-Gewirtzman, D. (2022). Architectural features and indoor evacuation wayfinding: The starting point matters. Safety Science.

Nourkojouri, H., Dehnavi, A. N., Bahadori, S., & Tahsildoost, M. (2023). Early design stage evaluation of architectural factors in fire emergency evacuation of the buildings using Pix2Pix and explainable XGBoost model. Journal of Building Performance Simulation.

Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., & Savarese, S. (2019). SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints. IEEE Conference on Computer Vision and Pattern Recognition.

Şahin, C., Rokne, J., & Alhajj, R. (2019). Human behavior modeling for simulating evacuation of buildings during emergencies. Physica A: Statistical Mechanics and its Applications.

Seitz, M. J., & Köster, G. (2012). Natural discretization of pedestrian movement in continuous space. Physical Review E.

Sohn, S. S., Zhou, H., Moon, S., Yoon, S., Pavlovic, V., & Kapadia, M. (2020). Laying the foundations of deep long-term crowd flow prediction. ECCV 2020.

Testa, E., Barros, R., & Musse, S. (2019). Crowdest: a method for estimating (and not simulating) crowd evacuation parameters in generic environments. The Visual Computer.

Weidmann, U. (1992). Transporttechnik der Fussgänger. Schriftenreihe des Instituts für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau, 2 ed.

Yuan, Y., Weng, X., Ou, Y., & Kitani, K. (2021). AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting. International Conference on Computer Vision.