Understanding and Modeling Collision Avoidance Behavior for Realistic Crowd Simulation

Zihan Yu
Tsinghua University
Beijing, China
yuzh19@tsinghua.org.cn

Guozhen Zhang
Tsinghua University
Beijing, China
zgz18@mails.tsinghua.edu.cn

Yong Li*
Tsinghua University
Beijing, China
liyong07@tsinghua.edu.cn

Depeng Jin
Tsinghua University
Beijing, China
jindp@tsinghua.edu.cn

ABSTRACT

For walking pedestrians, when they are blocked by obstacles or other pedestrians, they adjust their speeds and directions to avoid colliding with them, which is called collision avoidance behavior. This behavior is the most complex part of pedestrians' walking processes and its modeling and simulation are the keys to realistic crowd simulation, which serves as the foundation for various applications. However, most existing methods either lack the representation power to accurately model the complex collision behavior or do not model it explicitly, which leads to a poor level of realism of the simulation. To realize realistic crowd simulation, we propose to analyze, understand, and model the collision avoidance behavior in a data-driven way. First, to automatically detect collision avoidance behavior for further analysis, we propose a domain transformation algorithm that detects it by transforming the trajectories in the spatial domain into a new domain where the behavior is much more apparent and is thus easier to detect. The new domain also provides a new perspective for understanding collision avoidance behavior. Second, since there are no mature metrics to evaluate the level of realism, we propose a new evaluation metric based on the least-effort theory, which evaluates the realism of collision avoidance behavior by its physical and mental consumption. This evaluation metric also provides the foundation of modeling. Third, for realistic crowd simulation, we design a reinforcement learning model. It trains agents with our proposed reward function that models pedestrians' intrinsic needs of "reducing effort consumption" and thus can guide agents to behave realistically when avoiding collisions. Extensive experiments show our model is 55.9% and 52.5% more realistic in collision avoidance behavior than the best baselines on two real-world datasets. We release our codes at https://github.com/tsinghua-fib-lab/TECRL.

CCS CONCEPTS

• **Computing methodologies → Agent / discrete models**.

KEYWORDS

crowd simulation, collision avoidance behavior, evaluation metrics, reinforcement learning

*Corresponding author

CIKM '23, October 21–25,2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615098

ACM Reference Format:
Zihan Yu, Guozhen Zhang, Yong Li, and Depeng Jin. 2023. Understanding and Modeling Collision Avoidance Behavior for Realistic Crowd Simulation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3615098

1 INTRODUCTION

Crowd simulation is the process of simulating how pedestrians move to avoid colliding with others or obstacles and reach their destinations, which is widely used in emergency evacuation [22], architectural design [3], traffic scheduling [15], etc. For example, for emergency evacuation, simulating how pedestrians interact with other pedestrians and obstacles helps us analyze the time consumption of an evacuation plan and further optimize its design. For the architectural design of public transport interchanges (e.g., railway stations), the simulation helps to analyze the density of pedestrians under given passenger throughput and further evaluates the capacity of different architectural designs. All of these applications depend on the realism of the crowd simulation. An unrealistic simulator can lead to poor optimization results and evaluation errors.

For personal safety and comfort, pedestrians tend to adjust their velocities when they will collide with others or obstacles. While at other times, they typically move straight toward their destinations at uniform speeds [12]. As shown in Figure 1(a), the latter is usually a simple uniform linear motion which is easy and trivial to model, while the former is usually a complex variable motion with more speed and direction adjustments and is hard to model. The comparison of their "complexity" is shown as Figure 1(b, c): when pedestrians are avoiding collisions, their behavior has larger variances in acceleration and yaw rate compared to the behavior moving toward destinations, where the acceleration and yaw rate are the adjustments of speed and movement direction per unit of time respectively and reflect the complexity of behavior. Therefore, although the behavior of moving towards destinations generally makes up the majority of pedestrians' trajectories, accurately simulating the collision avoidance behavior is the key to realistic crowd simulation [16].

Many existing crowd simulation approaches rely on experts' knowledge rather than real-world collision avoidance data to model this behavior and evaluate these models based on subjective observation rather than objective metrics[6, 7, 10, 27], which poses limitations on the level of realism in modeling the avoiding behavior. On the other hand, recent advancements in trajectory prediction models have shown promise in predicting pedestrians' near-future trajectories using neural networks and historical trajectory data[1, 5, 13, 14, 24]. However, these models are not suitable for

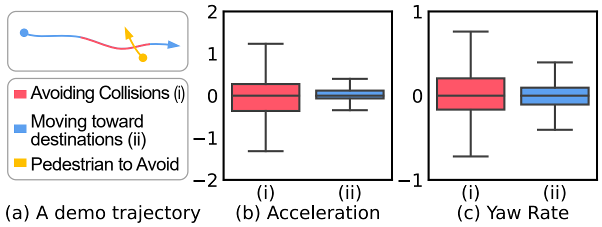


Figure 1: Demonstration of the complex motion dynamics in the collision avoidance state.

simulating longer time ranges of several minutes or hours, which are of greater importance in crowd simulation tasks, making them perform poorly in crowd simulation tasks as well[29].

To realize realistic crowd simulation, we propose to analyze, understand, and model the collision avoidance behavior in a data-driven way. This is non-trivial because of three problems: (1) There is no existing specific collision avoidance dataset that only contains pedestrians' avoiding behavior, and it's hard to autonomously annotate this behavior since pedestrians usually avoid collisions by slightly adjusting their trajectories. (2) There are no existing metrics to evaluate "the level of realism", since it heavily relies on human intuition. The metrics used in trajectory prediction such as ADE/FDE are not suitable as well since they evaluate trajectories by their similarity to the ground truth from the spatiotemporal perspective, which is, as we discussed in Secion 6, different from the realism of collision avoidance behavior. (3) The collision avoidance behavior is complex and hard to model directly: for a collision, there usually exist multiple possible routes to avoid it, and pedestrians randomly choose one of them according to their preferences, which makes it hard to be effectively modeled with universal law.

In this work, we propose to understand and model the collision avoidance behavior with a *Domain Transformation-based* (DT-based) annotation algorithm for its detection, a *Total Effort Consumption* (TEC) metric for its evaluation, and a *Total Effort Consumption-based Reinforcement Learning* (TEC-RL) model for realistic crowd simulation. Specifically, first, to automatically annotate the collision avoidance behavior for further analysis, we propose a DT-based algorithm that detects it by transforming the trajectories in the spatial domain into a collision domain where this behavior becomes much more apparent and is thus easier to annotate. Second, based on the *least-effort theory* [30], i.e. pedestrians choose routes with the least cost, we propose a TEC metric derived from human physiology and human psychology to evaluate the level of realism of a given collision avoidance process based on its physical and mental consumption. We prove that this metric can evaluate realism in a way that aligns with human intuition through an experiment. Third, to model the collision avoidance behavior, we design a TEC-based reinforcement learning model with our proposed TEC-based reward function which models the intrinsic needs of pedestrians and thus can guide agents to behave realistically during collision avoidance.

We highlight our contributions as follows:

- To the best of our knowledge, we propose to systematically analyze, understand, and model the collision avoidance behavior in a data-driven way to realize realistic crowd simulation for the first time.
- We propose a DT-based algorithm to automatedly annotate collision avoidance behavior, which provides a data basis for achieving more realistic modeling of the avoidance process. Experiments on two real-world datasets show the high accuracy of our annotation algorithm.
- We propose a TEC metric to evaluate the level of realism of the collision avoidance behavior for the first time and prove its ability to evaluate in a way that aligns with human intuition, which makes it possible to evaluate the crowd simulation model quantitatively.
- We design a TEC-RL crowd simulator with realistic collision avoidance behavior based on our proposed TEC-based reward function. Extensive experiments on two real-world datasets prove the superior performance of our model in behavioral realism compared to other state-of-the-art methods. Specifically, our model is 55.9% and 52.5% more realistic in collision avoidance behavior than the best baselines on two real-world datasets.

2 PRELIMINARIES

2.1 Crowd Simulation Concepts

In this study, we use $\boldsymbol{p}_i$ and $\boldsymbol{v}_i$ to denote the current position and velocity of pedestrian i. At each step of the simulation, we update their $\boldsymbol{v}_i$ in a manner that allows them to reach their destinations by following routes typically chosen by real humans. Similar to the majority of research conducted on microscopic crowd simulation in the past decade[26], we simplify the pedestrians in the crowd as disk-shaped particles, simplify the environment in which the crowd moves as a planar surface with polygonal obstacles, and simulate each timestep (or frame) to represent 0.1 seconds.

Additionally, in this paper, we use the term "velocity" $\boldsymbol{v}$ to represent a vector characterized by both magnitude and direction, while the term "speed" $v = \|\boldsymbol{v}\|$ is used to denote the magnitude of the velocity vector and is thus a scalar.

2.2 Collision Avoidance Concepts

Many modern crowd simulation models utilize the *Distance of Closest Approach* (DCA) and *Time to Closest Approach* (TTCA) to predict future collisions between pedestrians or between a pedestrian and an obstacle [25]. These two values are depicted in Figure 2, and can be computed by solving the following simple quadratic equations:

$$D_{ij}(t) = \max\left\{\left\|\boldsymbol{p}_i + \boldsymbol{v}_i t - \boldsymbol{p}_j - \boldsymbol{v}_j t\right\| - (R_i + R_j), 0\right\}, \quad (1)$$

$$\text{DCA}_{ij} = \min\nolimits_{0\le t\le T_{\text{clip}}} D_{ij}(t), \quad (2)$$

$$\text{TTCA}_{ij} = \text{argmin}_{0\le t\le T_{\text{clip}}} D_{ij}(t), \quad (3)$$

where $R_i + R_j$ is the sum radius of pedestrian i and j, $D_{ij}(t)$ is their distance after time t, and T_{clip} is used to prevent infinity. Note that if they will collide in the near future, the DCA is zero and TTCA is the time-to-collision.

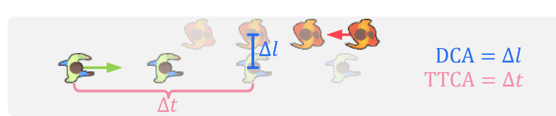


Figure 2: A demonstration of DCA and TTCA

3 MODELLING COLLISION AVOIDANCE BEHAVIOR IN A DATA-DRIVEN WAY

Collision avoidance behavior plays a crucial role in achieving realistic crowd simulation, given its inherent complexity. However, due to the lack of specialized collision avoidance datasets and established metrics for evaluating its realism, existing works have relied on expert intuition rather than real-world data to model this behavior, and evaluation of these models has largely been subjective[6, 7, 10]. To address these gaps, we propose a *Domain Transformation-based (DT-based) annotation algorithm* that automatically annotates collision avoidance behavior in real-world datasets and introduce a *Total Effort Consumption (TEC) metric* to evaluate the level of realism in a manner that aligns with human intuition.

3.1 DT-based Annotation Algorithm

3.1.1 ***DT-based Annotation Algorithm.*** Although discerning collision-avoidance processes in the spatial domain can be challenging, we have observed that the values of the *Distance of Closest Approach* (DCA) and *Time to Closest Approach* (TTAC) exhibit distinct behavior during such processes. Figure 3(a) illustrates this phenomenon. Row (1) shows two different collision avoidance processes, where the focal person (red) avoids another person (yellow) approaching from opposite and lateral directions. Row (2) depicts the corresponding changes in TTCA and DCA values, creating a trajectory in the TTCA-DCA plane, which we refer to as the *Collision Domain*. Interestingly, despite the spatial differences depicted in Row (1), they exhibit similarity in Row (2), or the collision domain. This is not a coincidence. We have found that during a collision avoidance process, the trajectory in the collision domain consistently transitions from a position near the horizontal axis to the vertical axis.

This finding becomes apparent when considering the physical meaning of TTCA and DCA. Figure 3(b) illustrates this concept. At the time t_0, a small TTCA and zero DCA indicate that a collision is imminent, prompting the focal person (red) to take evasive action. At the time t_1, a sufficiently large DCA suggests that the focal person will not get too close to the others, thereby, reducing the risk of collision. Finally, at the time t_2, a large DCA and zero TTCA indicate that these two individuals have reached their closest proximity and will start moving away from each other in the future.

Taking inspiration from this observation, we propose the definition of three regions in the collision domain: The orange region, referred to as the *collision-impending region* ($\mathcal{I}$), indicates that two individuals will get too close or potentially collide in the near future. The blue region, referred to as the *collision-free region* ($\mathcal{F}$), suggests that two individuals are moving away from each other, eliminating the risk of collision. The red region, referred to as the *collision-occurred region* ($\mathcal{C}$), represents the scenario where the two individuals have already come into close proximity or have collided with each other. These three regions can be formulated as follows:

$$\mathcal{I} = \{(\mathrm{TTCA}, \mathrm{DCA}) | \mathrm{TTCA} < \mathrm{T_{max}}, \mathrm{DCA} < \mathrm{D_{max}}\},$$
$$\mathcal{F} = \{(\mathrm{TTCA}, \mathrm{DCA}) | \mathrm{TTCA} = 0, \mathrm{DCA} \geq \mathrm{D_{max}}\},$$
$$\mathcal{C} = \{(\mathrm{TTCA}, \mathrm{DCA}) | \mathrm{TTCA} = 0, \mathrm{DCA} < \mathrm{D_{max}}\},$$

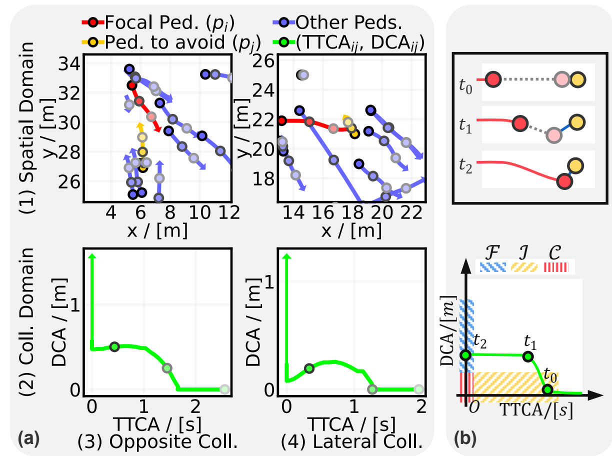
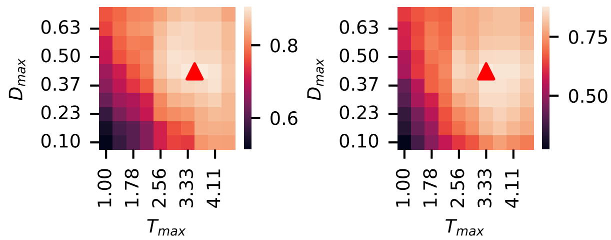


Figure 3: Comparison between trajectories in the spatial domain and collision domain. (a) Real-world trajectories, where the black disks mark every 10 simulation steps (i.e. 1 second). (b) Three regions in the collision domain.

where $T_{\max}$ represents the avoidance lead time, which is the temporal response range of pedestrians to future collisions. $D_{\max}$ represents social distancing, which is the distance people tolerate others encroaching on their side.

With the collision domain and the three regions defined above, we describe our *Domain Transformation-based* (DT-based) annotation algorithm as follows: In order to detect an individual's avoidance behavior towards another person, we transform their trajectories from the spatial domain into the collision domain. We annotate the process as a collision avoidance process when the transformed trajectory remains within the $\mathcal{I}$ region for a continuous duration of T_{reac} seconds, followed by a continuous stay in the $\mathcal{F}$ region for an additional duration of T_{reac} seconds. Here, $T_{\mathrm{reac}} = 300$ ms represents the human's reaction time. Moreover, the avoidance process is terminated if the other person moves out of the focal person's field of view. We also exclude avoidance processes that are shorter than 0.4 seconds and processes involving acquaintances. Acquaintances are defined as people who spend more than 70% of their time within a distance of 5 meters.

3.1.2 ***Evaluation of the DT-based Annotation Algorithm.*** To comprehensively evaluate the performance of our annotation algorithm, we consider two levels: the event level and the time-range level. At the event level, we assess whether the algorithm can detect avoidance behavior between two individuals and use the F_1 score to measure its performance in this aspect. At the time-range level, we evaluate the difference in the time series (i.e., start-stop time) between the algorithm's predictions and the ground truth. For its evaluation, we employ the affiliation metric [8], which extends the F_1 score for time series problems. To establish the ground truth, we manually marked 58 collision avoidance processes with a total duration of 266 seconds on the GC dataset and 51 processes with a total duration of 236 seconds on the ETH/UCY dataset. To determine the optimal values of $T_{\max}$ and $D_{\max}$, we conducted a grid-search optimization experiment on two datasets to maximize the sum of



(a) F_1 in GC Dataset (b) F_1 in UCY Dataset

Figure 4: The F_1 score against $D_{\max}$ and $T_{\max}$ on two datasets. The peak points are marked with red triangles.

		GC	ETH/UCY
Event Level	Precision	91.23%	91.11%
	Recall	89.66%	82.00%
	F1-Score	90.43%	86.32%
Time-Range Level	Precision (A)	96.08%	95.21%
	Recall (A)	98.66%	99.63%
	F1-Score (A)	97.35%	97.37%

Table 1: The evaluation results of our annotation algorithm, where the postfix (A) represents the affiliation metrics [8].

the F_1 scores at the two evaluation levels, as depicted in Figure 4. Based on the results, we selected $T_{\max} = 3.5$ s and $D_{\max} = 0.45$ m. These values indicate that pedestrians react to collisions occurring within 3.5 seconds and maintain a social distancing of 0.45 meters from others, which aligns with the statistical results reported by Parrillo et al. [17].

The performance of our annotation algorithm is presented in Table 1, where "True" samples are those collision-avoidance exists and "Positive" samples are that detected. The result demonstrates its high accuracy in automatically annotating avoidance processes at both the event and the time-range level.

3.2 Total Effort Consumption: A New Metric to Evaluate the Level of Realism

An appropriate evaluation metric is a foundation for effective modeling. However, there are no mature evaluating metrics for the realism of collision avoidance behavior. Most of the widely used metrics, such as the mean absolute error, evaluate trajectories by their similarities with the ground truth. However, the spatiotemporal similarity is different from the behavioral realism. For example, a trajectory that moves along the edge of an obstacle is similar to a trajectory that runs into the obstacle from the spatiotemporal perspective, but the latter is much more unrealistic because of the collisions with the obstacle. To solve this problem, we invent the *Total Effort Consumption* (TEC) metric, which evaluates the realism of behavior by its physical and mental consumption and does not rely on the ground truth, based on the *least-effort theory* [30].

When avoiding collisions, pedestrians have to adjust speed and direction, which consumes extra effort both physiologically and psychologically. Specifically, the speed adjustments require work performed, which costs physiological consumption. The direction adjustments make pedestrians deviate from destinations, which is against their will to reach the destinations and thus costs psychological consumption [2]. While on the other hand, the *least-effort theory* proposed by Zipf et al. [30] suggests that pedestrians tend to choose the route that costs the least effort, which inspires us that we can use the total effort consumption to evaluate the realism of a collision avoidance process. We suggest defining the effort as a weighted sum of three terms, including energy consumption that reflects the instantaneous motion states, process work that reflects the change of motion states, and mental effort that reflects the destination states.

3.2.1 ***Energy Consumption.*** Liu et al. [11] discovered a relationship between the power consumption of a walking human and its kinetic energy and rotational energy. Specifically, for a given route s, the energy consumption can be expressed as follows:

$$E_s = \int_0^{T_s} (me_s + e_d m\,|v(t)|^2 + e_r(\frac{1}{2}mR^2)\,|v(t)/r|^2)\ \mathrm{d}t, \tag{4}$$

Here, T_s represents the duration of the route, e_s, e_d, and e_r are coefficients associated with the pedestrian's physical characteristics (e.g., height and weight). The average values for these coefficients are $e_s = 2.23\frac{\mathrm{J}}{\mathrm{kg}\cdot\mathrm{s}}$, $e_d = 1.26\frac{1}{\mathrm{s}}$, and $e_r = 2.00\frac{1}{\mathrm{s}}$. Additionally, m and R denote the pedestrian's mass and radius, while v and r correspond to the pedestrian's current speed and turning radius, respectively.

3.2.2 ***Process Work.*** In addition to the instantaneous motion state, the modification of motion states should also be considered. For example, a trajectory with frequent or violent speed modification is usually considered less realistic than a trajectory with roughly constant speed. Therefore, we introduce the process work term to detect the abnormal modifications of motion states. According to Newtonian mechanics, for a pedestrian whose velocity is $\boldsymbol{v}(t)$, the work performed during a process s is

$$W_s = \int_0^{T_s} |P(t)|\ \mathrm{d}t = \int_0^{T_s} \left| m\frac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t}\cdot\boldsymbol{v}(t) \right|\ \mathrm{d}t. \tag{5}$$

where $\frac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t}$ is the velocity derivative, which is associated with the modification of velocity.

3.2.3 ***Mental Effort.*** Apart from the motion states, the destinations' states are also indispensable. Therefore, we introduce the mental effort term proposed by Bongiorno et al. [2], which describes the psychological cost of deviating from destinations. The mental effort comes from the human navigation mechanism. Specifically, when pedestrians move toward their destinations, they don't need to pay attention to destinations' locations. But when they move deviating from their destinations at an angle, they need to care about these and thus cost extra mental effort. The idea above is formulated as follows:

$$M_s = \int_0^{L_s} |\langle \boldsymbol{v}(t), \boldsymbol{d}-\boldsymbol{p}(t)\rangle|\ \mathrm{d}l = \int_0^{T_s} |\langle \boldsymbol{v}(t), \boldsymbol{d}-\boldsymbol{p}(t)\rangle||v(t)|\ \mathrm{d}t, \tag{6}$$

where $\langle\cdot,\cdot\rangle$ is the angle between two vectores, $\boldsymbol{d}$ is the destination, and $\boldsymbol{p}(t)$ is the position at time t.

3.2.4 ***TEC Metric.*** With the three terms introduced above, we define the TEC metric as a weighted sum of them:

$$\mathrm{TEC}_s = \lambda_E E_s + \lambda_W W_s + \lambda_M M_s, \tag{7}$$

where λ_E, λ_W, and λ_M are positive weighting coefficients. Since the three terms are all positive, TEC is constantly positive. Therefore, a high TEC implies some kind of unrealism in the trajectory, such as abnormally large speed, frequent velocity modification, and largely deviating from destinations. We choose the weighting coefficients λ_E, λ_W, and λ_M to make the three terms have the same weight in the evaluation. Specifically, we choose them to normalize these three terms on the real-world dataset $\mathcal{R}$:

$$\lambda_E = \frac{1}{3} \cdot \frac{1}{\mathbb{E}_{s\in\mathcal{R}} E_s}, \lambda_W = \frac{1}{3} \cdot \frac{1}{\mathbb{E}_{s\in\mathcal{R}} W_s}, \lambda_M = \frac{1}{3} \cdot \frac{1}{\mathbb{E}_{s\in\mathcal{R}} M_s}, \quad (8)$$

We observe that the values of these three coefficients vary among different datasets, but their ratios remain roughly consistent across various datasets. As presented in Table 2, the approximate ratio is $\lambda_E : \lambda_W : \lambda_M = 10^{-3} : 10^{-2} : 1$. This finding implies that pedestrians in different scenarios make similar trade-offs between energy consumption, applied work, and mental effort.

	λ_E	λ_W	λ_M
GC Dataset	4.73×10^{-4}	3.30×10^{-3}	6.17×10^{-1}
ETH/UCY Dataset	4.78×10^{-4}	5.13×10^{-3}	3.83×10^{-1}

Table 2: Values of Three Coefficients in Different Datasets.

To verify the validity of the TEC metric, i.e. whether it can measure the level of realism of a given trajectory as a human expert does, we conduct an experts' ranking experiment, and the results are presented in Section 5.2.

4 MODELING COLLISION AVOIDANCE

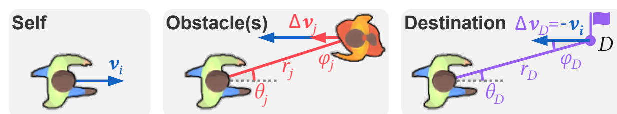
To achieve realistic crowd simulation, we take the Total Effort Consumption (TEC) metric as the optimization objective to minimize, as it serves as an indicator of trajectory realism. However, since TEC is not a function but a "function of function" (i.e., functional), this problem can be classified as a complex functional optimization problem. To tackle this challenge, we employ Reinforcement Learning (RL) techniques, which are inherently solving functional optimization problems. In our proposed approach, we define the negative TEC as the reward in RL and develop a crowd simulator called *Total Effort Consumption-based Reinforcement Learning* (TEC-RL). This framework enables us to generate trajectories with minimal TEC, leading to the generation of realistic collision avoidance behavior in crowd simulation.

4.1 Collision Detection Mechanism

Most existing reinforcement learning methods optimize the combination of two objectives, reaching destinations and avoiding collisions. However, an optimization with these two different objectives is a multi-objective optimization problem which is proven difficult to solve [19], and the hand-designed combination form is hard to describe the latent ground truth reward function. Therefore, we transform one of the objectives, avoiding collisions, into a constraint through a collision detection mechanism, and thus make our optimization a single-objective optimization problem. Specifically, we consider agents' collision entities, so they cannot pass through others or obstacles, but stop when they collide with them. As a result, agents can learn to avoid collisions from the experiences that, when they collide with obstacles, they stop, get less reward because of the abnormal sudden change of velocity, and are unable to approach destinations. Additionally, this mechanism is closer to nature: considering a baby learning to walk, it finds that when it collides with an obstacle, it stops and hurts, and is blocked to reach its destination. By accumulating this experience, the baby can learn to avoid collisions in walking, and so as our models.

4.2 TEC-based Reinforcement Learning Algorithm

In this work, we simulate pedestrians' movements by iteratively generating their velocities (i.e., norms and directions) at the next step based on their observations (i.e., positions and velocities of other pedestrians, obstacles, and destinations) at the current step, which is a Markov decision process which can be characterized by the state space $\mathcal{S}$, the action space $\mathcal{A}$, the reward function R, the transition function P and the discount factor γ.



$s = (\|\boldsymbol{v}_i\|, r_1, \theta_1, \|\Delta\boldsymbol{v}_1\|, \varphi_1, \ldots, r_N, \theta_N, \|\Delta\boldsymbol{v}_N\|, \varphi_N, r_D, \theta_D, \|\Delta\boldsymbol{v}_D\|, \varphi_D)$

Figure 5: Demonstration of the state space.

State Space As shown in Figure 5, for an agent A_i, the observation state includes three parts: the self, the obstacles, and the destination. The self's state of A_i is its current speed $\|\boldsymbol{v}_i\|$. The obstacles' states include the nearest $N = 20$ obstacles within A_i's field of view, and one of them, such as the state of the obstacle A_j, is a tuple $s_j = (r_j, \theta_j, \|\Delta\boldsymbol{v}_j\|, \varphi_j)$, where r_j is their center-to-edge distance, θ_j is the orientation of A_j with respect to A_i, $\Delta\boldsymbol{v}_j$ is their relative velocity, and φ_j is the angle between their relative velocity and relative position. For the destination's states, we similarly use a tuple $s_D = (r_D, \theta_D, \|\Delta\boldsymbol{v}_D\|, \varphi_D)$, treating the destination D as a point with zero velocity.

Action Space: Agent's action is its velocity, i.e. the speed and direction change at the next step. Note that the actual velocities may be smaller than the agents' chosen ones when collisions occur. We generate actions by sampling from a Gaussian mixture model [18], a weighted sum of $K = 3$ Gaussian distributions.

Reward Function: The reward function evaluates the action under given observations. Inspired by the TEC metric we proposed in Section 3.2, we use the effort consumption of an action as its cost function (i.e., the negative of the reward function). Nevertheless, we find that since the cost function, TEC, is always positive, agents prefer keeping still rather than moving to destinations to minimize their effort consumption. Therefore, we change the angle difference term $|\langle \boldsymbol{v}, \boldsymbol{d} - \boldsymbol{p} \rangle|$ in Equation (6) to its negative cosine $-\pi \cos \langle \boldsymbol{v}, \boldsymbol{d} - \boldsymbol{p} \rangle$, which enables agents to get positive rewards by getting close to their destinations, rather than getting penalties when they deviate from destinations. Although we can also solve this by giving extra rewards for agents reaching destinations, it is a task with sparse rewards and thus harder to train than this negative cosine scheme.

Transition Function: Our simulator operates deterministically, and the transition function is thus a deterministic function rather than a random distribution. Given the agents' current positions

Statistics	GC	ETH/UCY
Avg. duration of CAB (s)	2.76	2.40
Avg. speed of CAB (m/s)	1.25	1.42
Avg. percent of CAB in whole trajectory	19.8%	22.4%
Pedestrian density ($1/m^2$)	0.094	0.058

Table 3: The basic statistics of the datasets, where CAB means collision avoidance behavior.

and the velocities they choose for the next step, we calculate their subsequent positions using the equation $\boldsymbol{p}^{(t+1)} = \boldsymbol{p}^{(t)} + \boldsymbol{v}^{(t+1)}\Delta t$. If any collisions occur, we adjust $\boldsymbol{p}^{(t+1)}$ to resolve them.

Discount Factor: In RL, the agents are trained to optimize the cumulative reward $\sum_t \gamma^t R_t$, where R_t is the reward of a single step t, and γ is the discount factor. Since we use the effort consumption of a single step as the negative reward, by choosing a discount factor near 1, the cumulative reward approximates the negative total effort consumption of the trajectory, and the optimization object thus approximates the least total effort consumption.

To find the optimal policy, we use an actor-critic algorithm: we first extract features from the states of self, destination, and obstacles. Then, for each obstacle, we generate a weight from its feature and sum the obstacles' features up with these weights, which enables our model to deal with obstacles in arbitrary numbers. Finally, we concatenate the features together as the actor and critic's input. The critic generates a predicted accumulated reward. The actor generates a Gaussian mixture model and samples it to get the action at the next step. The policy is shared by all agents. We train the actor and critic with Proximal Policy Optimization [21]. See our code and Appendix Section A for more details.

5 EVALUATION AND EXPERIMENTS

We perform two experiments. The first experiment aims to investigate whether our TEC metric can assess the realism level of a given trajectory in a manner consistent with human intuition. The second experiment focuses on evaluating the performance of our TEC-RL model, specifically, examining its ability to generate trajectories with realistic collision avoidance behavior.

5.1 Experiment Setup

5.1.1 ***Datasets***. To evaluate our model, we use two real-world datasets, including the GC and ETH/UCY. Their basic statistics about the collision avoidance process are shown in Table 3, and the details are shown as follows:

GC Dataset[1]: GC is a large pedestrians' trajectories dataset containing 12684 trajectories from a public transport interchange. For the GC dataset, we select five minutes with rich collision avoidance, use one for evaluation, and split the remaining into 3:1 for the training and validation of baselines that use supervised learning.

ETH/UCY Dataset[2]: ETH/UCY is a dataset containing 528 trajectories from an outdoor scene. We select all 216 seconds, use 54 seconds for evaluation, and split the remaining into 2:1 for the training and validation of baselines that use supervised learning.

[1] https://www.dropbox.com/s/7y90xsxq0l0yv8d/cvpr2015_pedestrianWalkingPathDataset.rar
[2] https://paperswithcode.com/dataset/ucy,https://paperswithcode.com/dataset/eth

5.1.2 ***Metrics***. We utilize four metrics in two types to evaluate the performance of our model.

Total Effort Consumption (TEC) measures behavioral realism and has been discussed in Section 3.2.

#Collision calculates the *mean* of the total number of frames in which a pedestrian collides with other objects while in motion.

Average Displacement Error (ADE) calculates the mean Euclidean distance difference between each predicted position and each ground-truth position. This metric evaluates trajectories in the spatiotemporal perspective and is widely used in trajectory prediction research[13, 14, 29].

Final Displacement Error (FDE) calculates the Euclidean distance difference between the final predicted position and the final ground-truth position, which is a spatiotemporal metric as well and is often used in conjunction with the ADE metric.

5.1.3 ***Baseline Methods***. We compare our model with six state-of-the-art models, including four *crowd simulation* models and two *trajectory prediction* models, where the former models collision avoidance behavior explicitly while the latter does not. we discuss their difference detailedly in Section 6.

Social Force Model (SFM)[6] models the interaction between pedestrians as repulsive forces. It is the most widely-used crowd simulation model and is extensively adopted by various commercial crowd simulation software, including MassMotion[3], VISSIM[4], and AnyLogic[5]. In this work, we calibrate the parameters used by SFM to minimize the occurrences of collision and blockages, where a group of people obstruct each other and remain stationary.

Machine-Learning-Aided Physical Model (MLAPM)[29] is an improvement on SFM obtained by conducting symbolic regression on the trained neural network.

Crowd Simulation Reinforcement Learning (CSRL)[10] is an RL model with a hand-engineered reward function. It gives rewards for agents approaching destinations and gives penalties for agents colliding with others or moving/turning too fast.

Heterogeneous Crowds using Parametric RL (HOP-RL)[7] is similar to CSRL, but adds extra penalties for impending collisions.

Social Spatio-Temporal Graph Convolutional Neural Network (Social-STGCNN)[6][14] is a supervised learning model which predicts trajectories with a GCN. We predict 1 step based on the historical 8 steps and rollout to generate the whole trajectory.

Predicted Endpoint Conditioned Network (PECNet)[7][13] is a supervised learning model considering destinations. It first predicts destinations and then future trajectories. For a fair comparison, we skip the first step and use the real destination directly.

5.1.4 ***Experiment Settings and Reproducibility***. In our experiments, we employ metrics for collision-avoiding trajectories instead of entire trajectories. Specifically, we initially identify all collisions in real-world datasets using our domain transformation-based annotation algorithm. Subsequently, we simulate the process by which a pedestrian avoids colliding with another pedestrian, starting from their initial states and continuing until the former

[3] https://www.arup.com/services/digital/massmotion
[4] https://www.myptv.com/en/mobility-software/ptv-vissim
[5] https://www.anylogic.com/
[6] https://github.com/abduallahmohamed/Social-STGCNN
[7] https://github.com/HarshayuGirase/Human-Path-Prediction

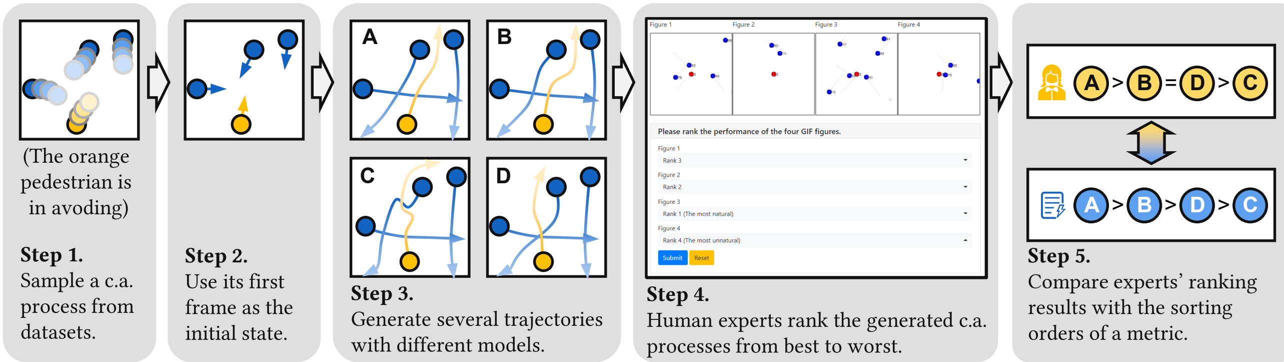


Figure 6: Human experts' ranking experiment, where c.a. means collision avoidance.

		GC Dataset			ETH/UCY Dataset			Synthetic Scenario		
		precision	recall	F_1-score	precision	recall	F_1-score	precision	recall	F_1-score
Classical Metric	#Collision	0.858	0.225	0.356	0.891	0.342	0.494	0.684	0.066	0.120
	MAE	0.867	0.757	0.808	0.845	0.733	0.785	-	-	-
Our Metric	TEC	**0.959**	**0.910**	**0.934**	**0.969**	**0.940**	**0.954**	**0.952**	**0.942**	**0.947**
	TEC without EC	0.949	0.893	0.920	0.959	0.924	0.941	0.823	0.796	0.809
	TEC without PW	0.944	0.884	0.913	0.932	0.878	0.904	0.931	0.919	0.925
	TEC without ME	0.920	0.844	0.880	0.948	0.904	0.926	0.902	0.886	0.894

Table 4: The consistent degree between the ranking results of the experts and the different metrics. Note that since there are no ground truth trajectories in synthetic scenarios, the result corresponding to MAE is omitted.

successfully avoids the latter or reaches a time limit of 100 steps. We then calculate metrics based on the trajectory followed by the pedestrian who is avoiding the collision. Further implementation details can be found in Appendix Section A.

5.2 Experts' Ranking Experiment

To assess whether our proposed Total Effort Consumption (TEC) metric can evaluate the level of realism of a given trajectory in a way that is consistent with human intuition, we conduct an experts' ranking experiment as Figure 6 illustrates. We selected 600 collision avoidance processes from two real-world datasets, as well as one synthetic scenario featuring a 5 m × 5 m square with 10 randomly generated obstacles. Then, we generated multiple avoiding processes using the starting frames of these selected processes as initial states, visualized them as GIF animations (as shown in Step 4's screenshot), and enlisted experts who are familiar with real-world human walking behavior to rank them based on their level of realism. Lastly, we conducted a comparison between the ranking results obtained from the experts and the ranking provided by our TEC metric to get their consistency. This comparison was carried out using the precision/recall metric: We break the ranking results of K animations into $\binom{K}{2}$ pairs, considering those "human expert indicates the former as more realistic than the latter" as true samples and those "TEC metric indicates the former as more realistic than the latter" as positive samples.

Except for our TEC metric, we also evaluate the validity of the #Collision, Mean Absolute Error (MAE), and three ablation versions of our TEC metric in the same way. The result is shown in Table 4. Our TEC metric performs better than other metrics, especially the widely used MAE metric. The #Collision metric has a low recall rate because it cannot compare the level of realism between two trajectories that neither has a collision. The three ablation versions that ablate the Energy Consumption (EC), Process Work (PW), and Mental Effort (ME) terms respectively perform worse than the TEC metric, which proves that all three terms are effective.

5.3 Overall Performance Comparison

To examine the performance of our model, we compare our TEC-RL with six state-of-the-art baselines in different types on two datasets. Table 5 shows the comparing results for four metrics, including two behavioral realism metrics (TEC, #Collision) and two spatiotemporal similarity metrics (ADE/FDE). We summarize our key observations and insights as follows:

The Superior Performance of TEC-RL: The TEC-RL model outperforms all state-of-the-art baselines across all metrics. In terms of behavioral realism, TEC-RL achieves a significant relative performance improvement of 55.9% and 51.1% on the GC dataset, as well as 52.5% and 73.1% on the ETH/UCY dataset. Additionally, in terms of spatiotemporal similarity, TEC-RL achieves a relative performance gain of 16.84% and 26.75% on the GC dataset, as well as 4.10% and 6.19% on the ETH/UCY dataset. These findings provide compelling evidence for the efficacy of our proposed model.

Analyses on the Performance of Trajectory Prediction Baselines: All trajectory prediction baselines perform worse than the crowd simulation baselines. This distinction comes from the fundamental differences in objectives and temporal scales between the crowd simulation task and the trajectory prediction task, which we discussed in Section 6. Zhang et al. give a deeper discussion about the limited performance of trajectory prediction models, such as Social-LSTM[1], SGAN[5], STGCNN[14], etc., when applied to

		GC Dataset				ETH/UCY Dataset			
		TEC	#Collision	ADE	FDE	TEC	#Collision	ADE	FDE
CS Models	SFM [6]	2.921	0.317	1.860	3.034	4.379	0.420	2.426	3.035
	MLAPM [29]	2.896	0.321	1.823	2.987	4.337	0.295	2.245	3.047
	CSRL [10]	3.520	0.647	1.996	3.297	3.756	0.485	2.370	3.436
	HOP-RL [7]	4.101	0.647	3.208	5.026	3.698	0.182	2.801	4.022
TP Models	STGCNN [14]	63.30	2.502	7.420	15.30	297.1	3.930	7.846	16.30
	PECNet [13]	10.45	4.383	4.869	10.94	6.191	1.903	5.709	12.39
Ours	TEC-RL	**1.277** (+55.9%)	**0.155** (+51.1%)	**1.516** (+16.84%)	**2.188** (+26.75%)	**1.755** (+52.5%)	**0.049** (+73.1%)	**2.153** (+4.10%)	**2.847** (+6.19%)

Table 5: The performance evaluation results on two datasets. (CS - Crowd Simulation, TP - Trajectory Prediction)

	GC Dataset		ETH/UCY Dataset	
	TEC	#Coll.	TEC	#Coll.
w/o Collision Detection	2.702	7.119	3.210	4.378
w/o Energy Consumption	3.853	**0.144**	3.883	0.059
w/o Process Work	2.768	3.132	2.486	0.923
w/o Mental Effort	∞	-	∞	-
TEC-RL	**1.277**	0.155	**1.755**	**0.049**

Table 6: Ablation Study

the crowd simulation tasks, and provide evidence to substantiate that their error tends to accumulate during the rollout process[29].

Analyses on the Performance of Spatiotemporal Similarity: It's notable that although we design our model focusing on behavioral realism, it performs well in terms of spatiotemporal similarity (ADE/FDE) compared to other baselines. This success can be attributed to our data-driven research approach. Specifically, the reward function used to train TEC-RL is derived from data analysis and calibrated on real-world datasets. In contrast, other crowd simulation baselines rely on knowledge-driven approaches, which limits their ability to leverage real-world data to improve spatiotemporal similarity. For trajectory prediction baselines, although they are data-driven as well, they are not well-suited for long-term crowd simulation tasks and thus perform poorly as well.

The Generalizability of the Hyperparameter Calculation Method: Hyperparameters $\lambda_E, \lambda_W, \lambda_M$ calculated by Equation 8 exhibit different values across two datasets (see Table 2). Nonetheless, they all yield remarkable performances. This substantiates the generalizability of the hyperparameter computation method.

5.4 Ablation Study

We further examined how different parts in our model, including the three terms in our reward function and the collision detection mechanism, contribute to the performance. The result is shown in Table 6, where the *w/o Collision Detection* is the model trained in a simulator without collision detection mechanism, *w/o Energy Consumption*, *w/o process Work* and *w/o Mental Effort* are the models trained with a reward function that set λ_E, λ_W and λ_M to 0 respectively. We have four observations from the ablation results:

No Redundant Components: Removing any component causes a performance decrease compared with our TEC-RL model, which shows that all of the components are effective.

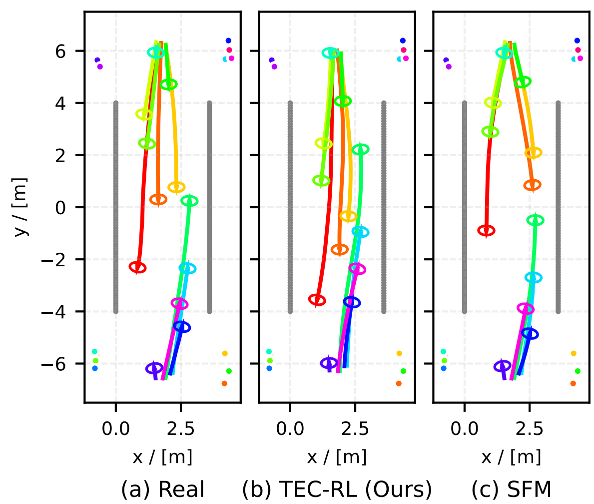


Figure 7: A case study in a bidirectional corridor scenario

The Critical Role of the Mental Effort Term: We find that when the mental effort term is removed, agents lose the rewards they get by getting close to their destinations and thus learn to stay still to minimize their effort consumption. Since they cannot move at all, we mark the TEC as ∞.

The Components Guiding Agents to Avoid Collisions: Removing the collision detection mechanism or the process work term in the reward function causes #Collision increase, which suggests that these two components work together to guide agents to avoid collisions. Specifically, the collision detection mechanism converts the collision into a sharp velocity change, and the process work term further converts it to extra effort in the reward function.

The Role of the Energy Consumption Term: When the energy consumption term is removed, the trained agents move at abnormally fast speeds but are still able to avoid collisions, which is because the term is related to only the instantaneous motion states but not their change. Therefore, the energy consumption term serves a role to make the pedestrian's speed more realistic.

5.5 Case Study

To further validate the realism of our model's collision avoidance behavior, we conducted a case study as depicted in Figure 7. The

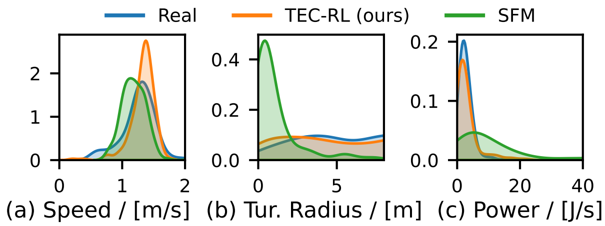


Figure 8: Comparison of the Kernel Density Estimation (KDE) of speed, turning radius, and instantaneous power.

scenario is based on the BO experiment of the HERMES Project conducted by Keip et al. [9]. In this scenario, pedestrians (represented by ellipses of different colors) are given instructions to navigate through a bidirectional corridor and leave from their assigned sides (either left or right, indicated by corresponding colored dots).

Figure 7(a) shows trajectories of real-world humans recorded by Keip et al., while Figures 7(b) and 7(c) display the simulated trajectories generated by our model and the Social Force Model (SFM), respectively. In our model, agents are equipped with the capability to proactively avoid collisions with pedestrians approaching from the opposite direction, which is similar to real-world trajectories. In contrast, SFM produces trajectories that exhibit unrealistic behavior: The orange and green agents on the right-hand side fail to avoid each other in advance, leading to a situation where they have to make sudden turns to prevent a collision.

To provide a comprehensive comparison, we analyzed the speed, turning radius, and instantaneous power in collision avoidance processes. Their distributions are depicted in Figure 8. The simulation results obtained from our TEC-RL model demonstrate distributions that closely resemble real-world data, while the SFM exhibits lower speeds, smaller turning radii, and higher instantaneous power compared to real-world trajectories, indicating its lack of realism.

6 RELATED WORK

Crowd simulation simulates how pedestrians move from their initial states to their designated destinations following patterns typically observed in human behavior[29]. The modeling of collision avoidance behavior is the most crucial part due to its complexity. Most of the crowd simulation models focus on simulating this behavior[26]. For example, force-based models such as Social Force[6] model the tendency of pedestrians to maintain a distance from each other as repulsive forces, which is widely used and processes many variations[23]. Velocity-based models such as Velocity Obstacle[4] directly calculate the velocity without the risk of collision. Agent-based models that adopted RL such as CSRL[10] and HOP-RL[7] guide agents to avoid collisions by penalizing them when a collision occurs. However, due to the lack of specific collision avoidance datasets and appropriate metrics for evaluating the level of realism, the modeling of this behavior heavily relies on expert observation and intuition rather than real-world data. The evaluation of these models is predominantly subjective, relying on qualitative observations rather than quantitative objective metrics. This restricts the performance of these models.

Trajectory prediction is another task raised these years that seems similar to crowd simulation, which utilizes neural networks to predict the future trajectories of pedestrians based on their historical trajectory data[1, 5, 13, 14]. However, trajectory prediction primarily serves the purpose of supporting autonomous driving systems rather than urban planning that crowd simulation concerns, which makes it inherently different from crowd simulation. Specifically, trajectory prediction focuses on short-term predictions within a few seconds and aims to achieve spatiotemporal similarity between the predicted trajectories and the ground-truth trajectories. In contrast, crowd simulation is concerned with a broader time range, and its primary objective is to generate realistic trajectories that align with human intuition[20, 27]. The difference between spatiotemporal similarity and realism can be seen from such an extreme example: Let's assume a real pedestrian is faced with an obstacle ahead and chooses to bypass it from the left. We have two models, A and B, simulating this process. Model A, symmetrically to the ground truth, chooses to bypass the obstacle from the right; while model B first approaches the obstacle, then makes a left turn of 90° and navigates along its edge. It is evident that the result produced by model A is more realistic than that of model B. However, a metric of spatiotemporal similarity such as ADE would suggest model B is better since its trajectory is closer to the ground truth, which contradicts human intuition. With the differences discussed above, trajectory prediction models perform poorly when being adopted to crowd simulation tasks[29].

7 CONCLUSION

In this work, we propose to understand and model collision avoidance behavior in a data-driven way. Specifically, we propose a *Domain Transformation-based* annotation algorithm to provide the data basis, a *Total Effort Consumption* metric for its evaluation, and a *Total Effort Consumption-based Reinforcement Learning* model for crowd simulation with realistic collision avoidance behavior. In our future work, we shall consider the heterogeneity in pedestrian behavior, akin to Yuan et al. [28].

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under 2022YFF0606904, the National Natural Science Foundation of China under U21B2036, U20B2060, and Beijing National Research Center for Information Science and Technology (BNRist).

A APPENDIX FOR REPRODUCIBILITY

We train the model in a $20m \times 20m$ region with $N = 3$ agents and $N_O = 3$ obstacles. In an epoch, we randomly initialize agents' and obstacles' positions and simulate no longer than 200 steps until any agent reaches its destination. After collecting 6000 steps, we update the model 64 times with the learning rate 10^{-5}. We multiply the entropy regularity coefficient l_2 by 0.1 for every 64×10000 update and finish the training after $64{\times}50000$ updates. The feature extractor, weights generator, actor, and critic we mentioned in Section 4.2 are all multilayer perceptrons with ReLU activation functions. We implement all the models and baselines with PyTorch on a server with a Ryzen 2990WX CPU and an NVIDIA GeForce RTX 2080 GPU. The training can be finished in 5000 episodes, taking about half an hour. The trained simulator can simulate scenarios with hundreds of pedestrians in real time. See our codes https://github.com/tsinghua-fib-lab/TECRL for more details.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 961–971.

[2] Christian Bongiorno, Yulun Zhou, Marta Kryven, David Theurel, Alessandro Rizzo, Paolo Santi, Joshua Tenenbaum, and Carlo Ratti. 2021. Vector-Based Pedestrian Navigation in Cities. *Nature Computational Science* 1, 10 (Oct. 2021), 678–685.

[3] Tian Feng, Lap-Fai Yu, Sai-Kit Yeung, KangKang Yin, and Kun Zhou. 2016. Crowd-Driven Mid-Scale Layout Design. *ACM Trans. Graph.* 35, 4 (jul 2016), 14 pages. https://doi.org/10.1145/2897824.2925894

[4] Paolo Fiorini and Zvi Shiller. 1998. Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research* 17, 7 (July 1998), 760–772. https://doi.org/10.1177/027836499801700706

[5] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2255–2264.

[6] Dirk Helbing and Péter Molnár. 1995. Social Force Model for Pedestrian Dynamics. *Physical Review E* 51, 5 (May 1995), 4282–4286.

[7] Kaidong Hu, Michael Haworth, Glen Berseth, Vladimir Pavlovic, Petros Faloutsos, and Mubbasir Kapadia. 2021. Heterogeneous Crowd Simulation Using Parametric Reinforcement Learning. *IEEE Transactions on Visualization and Computer Graphics* (2021).

[8] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. 2022. Local Evaluation of Time Series Anomaly Detection Algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 635–645.

[9] C Keip and K Ries. 2009. *„Dokumentation von Versuchen zur Personenstromdynamik ".* Technical Report. Technical Report, Project Hermes.

[10] Jaedong Lee, Jungdam Won, and Jehee Lee. 2018. Crowd Simulation by Deep Reinforcement Learning. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*. Association for Computing Machinery, 1–7.

[11] Chi Liu, Rui Ye, Liping Lian, Weiguo Song, Jun Zhang, and Siuming Lo. 2018. A Least-Effort Principle Based Model for Heterogeneous Pedestrian Flow Considering Overtaking Behavior. *Physics Letters A* 382, 20 (May 2018), 1324–1334.

[12] Celine Loscos, David Marchal, and Alexandre Meyer. 2003. Intuitive Crowd Behaviour in Dense Urban Environments Using Local Laws. In *Proceedings of the Theory and Practice of Computer Graphics 2003 (TPCG '03)*. IEEE Computer Society, USA, 122.

[13] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. 2020. It Is Not the Journey But the Destination: Endpoint Conditioned Trajectory Prediction. In *Computer Vision – ECCV 2020*. Vol. 12347. Springer International Publishing, 759–776.

[14] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. 2020. Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 14412–14420.

[15] Takuma Otsuka, Hitoshi Shimizu, Tomoharu Iwata, Futoshi Naya, Hiroshi Sawada, and Naonori Ueda. 2019. Bayesian Optimization for Crowd Traffic Control Using Multi-Agent Simulation. IEEE Press, 1981–1988. https://doi.org/10.1109/ITSC.2019.8917496

[16] Jin Hyoung Park, Francisco Arturo Rojas, and Hyun Seung Yang. 2013. A Collision Avoidance Behavior Model for Crowd Simulation Based on Psychological Findings: A Collision Avoidance Behavior Model for Crowd Simulation. *Computer Animation and Virtual Worlds* 24, 3-4 (May 2013), 173–183.

[17] Vincent N Parrillo and Christopher Donoghue. 2013. The national social distance study: Ten years later. In *Sociological forum*, Vol. 28. Wiley Online Library, 597–614.

[18] Douglas A Reynolds et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics* 741, 659-663 (2009).

[19] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113.

[20] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. 2020. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research* 39, 8 (2020), 895–935.

[21] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. (2017).

[22] Ameya Shendarkar, Karthik Vasudevan, Seungho Lee, and Young-Jun Son. 2008. Crowd Simulation for Emergency Response Using BDI Agents Based on Immersive Virtual Reality. *Simulation Modelling Practice and Theory* 16, 9 (2008).

[23] Hongzhi Shi, Jingtao Ding, Yufan Cao, Li Liu, Yong Li, et al. 2022. Learning Symbolic Models for Graph-structured Physical Mechanism. In *The Eleventh International Conference on Learning Representations*.

[24] Hongzhi Shi, Quanming Yao, and Yong Li. 2023. Learning to Simulate Crowd Trajectories with Graph Networks. In *Proceedings of the ACM Web Conference 2023*. 4200–4209.

[25] Wouter van Toll, Fabien Grzeskowiak, Axel López-Gandía, Javad Amirian, Florian Berton, Julien Bruneau, Beatriz Cabrero Daniel, Alberto Jovane, and Julien Pettré. 2020. Generalized Microscropic Crowd Simulation using Costs in Velocity Space. *Symposium on Interactive 3D Graphics and Games* (2020).

[26] W. van Toll and J. Pettré. 2021. Algorithms for Microscopic Crowd Simulation: Advancements in the 2010s. *Computer Graphics Forum* 40, 2 (2021), 731–754. https://doi.org/10.1111/cgf.142664

[27] Shanwen Yang, Tianrui Li, Xun Gong, Bo Peng, and Jie Hu. 2020. A review on crowd simulation and modeling. *Graph. Model.* 111 (2020), 101081.

[28] Yuan Yuan, Huandong Wang, Jingtao Ding, Depeng Jin, and Yong Li. 2023. Learning to Simulate Daily Activities via Modeling Dynamic Human Needs. In *Proceedings of the ACM Web Conference 2023*. 906–916.

[29] Guozhen Zhang, Zihan Yu, Depeng Jin, and Yong Li. 2022. Physics-Infused Machine Learning for Crowd Simulation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2439–2449.

[30] G. K. Zipf. 1950. Human Behaviour and the Principle of Least Effort. *The Economic Journal* 60, 240 (Dec. 1950), 808.