Curriculum-Based RL for Pedestrian Simulation: Sensitivity Analysis and Hyperparameter Exploration

Giuseppe Vizzari[1], Daniela Briola[1] and Federico Pisapia[1]

[1]*Department of Informatics, Systems and Communication (DISCo), University of Milano-Bicocca, Milan, Italy*

Abstract

Deep Reinforcement Learning (DRL) has recently shown encouraging results as a potential approach to the simulation of complex systems, in particular pedestrians and crowds. Curriculum-based approaches, in addition to reward design, represent conceptual and practical tools supporting the integration of domain knowledge and modeler's expertise into an agent training process, significantly reducing manual modeling effort while still granting the possibility to achieve plausible results in a relatively wide set of situations. Some of the workflows proposed in the literature, however, did not systematically analyze the sensitivity of the overall approach to changes in the model and in hyperparameters used to achieve proposed results. The present contribution represents a step in this direction, providing a set of experiments (i) showing the fact that curriculum based DRL models effectively grant a higher level of generalization compared to models trained even in challenging scenarios, at the cost of a relatively little overhead; (ii) showing the effect of changes both in model configuration (in particular the action model) and in hyperparameters of the learning algorithm, and suggesting lines for new research in the field to overcome current limitations.

Keywords

agent-based simulation, pedestrian simulation, reinforcement learning, curriculum learning, hyperparameter exploration

1. Introduction

Pedestrian and crowd simulation represent simultaneously an inter- and multidisciplinary research area, gathering contributions from disciplines ranging from social psychology, to applied mathematics, to engineering, as well as a consolidated context of application of commercial tools[1], used on an a daily basis by designers, planners and decision makers. Researches on this topic has developed initially as an additional area of investigation for the much more consolidated transportation research, despite the apparent difference between flows of vehicles and pedestrians, that have extremely different constraints [1], but the interest in granting the possibility to produce plausible forecasts about the actual utilization of space to designers and planners has led to the acquisition of a substantial body of knowledge about empirical evidences, modeling and applications [2], that supported an effective technology transfer. For example, the social force model [3] is officially employed within PTV Viswalk[2], a very successful commercial simulator.

Research on the field is still very active[3], aiming to improve the quality of the achieved results and to extend the range of considered phenomenologies: in particular, one research direction that has witnessed a significant growth of attention is the one exploring the possibility to employ recent results in Artificial Intelligence, and especially Deep Learning techniques, to the modeling of pedestrian and crowd behaviour. The activity of a human modeler, i.e. the user of (potentially commercial) tool implies a number of activities and decisions, to describe how the model is applied to a specific context: typically this implies importing a CAD file of an environment that is being investigated (e.g. a newly designed structure, the premises in which a crowded event takes place), and annotating it with information on how

ATT'24: Workshop Agents in Traffic and Transportation, October 19, 2024, Santiago de Compostela, Spain

giuseppe.vizzari@unimib.it (G. Vizzari); daniela.briola@unimib.it (D. Briola); f.pisapia1@campus.unimib.it (F. Pisapia)
0000-0002-7916-6438 (G. Vizzari); 0000-0003-1994-8929 (D. Briola)

[1] See an overview of platforms, libraries, fully fledged simulators – https://urban-analytics.github.io/dust/docs/ped_sim_review

[2] PTV website - Product page: https://www.ptvgroup.com/en/products/pedestrian-simulation-software-ptv-viswalk

[3] See, e.g., the web page describing the Pedestrian and Evacuation Dynamics international conference that reached its 11th edition in 2023 – https://collective-dynamics.eu/index.php/cod/ped

the pedestrians enter and circulate the area (i.e. where they go, how they use the environment). Some of these activities require information about the actual dynamics that can be observed in the environment (e.g. typical attendance) or plausible/informed assumptions. Human intervention is therefore at two levels: definition of the general model for pedestrian behaviour (e.g. the social force model) and specific application to a given context (e.g. types of pedestrians, their number, initial positions and respective goals in the environment). The pedestrian dynamics research community, however, started a systematic acquisition and sharing of empirical data from studies, observations, experiments, in an open science effort (supported by several research projects) several years ago [4]. Not surprisingly, in the past few years several researches have started investigating the possibility to create pedestrian models leveraging the available data to learn pedestrian models, first of all specific to a situation, and hopefully of more general applicability in the future. It must be noted that this kind of activity is at the same time related but different from *trajectory forecasting* [4], where the temporal window associated with the horizon of prediction is generally limited to few seconds with a focus on a specific and relatively limited area.

In line with recent research results aiming at exploiting Deep Reinforcement Learning (DRL) techniques [5] for achieving a sufficiently *general* behavioural model for a pedestrian agent positioned and moving in an environment, the present contribution employs a curriculum [6] based approach that, together with a careful reward function design phase, allowed us to exploit expertise on the simulated phenomenon and to achieve a behavioural model for a pedestrian agent showing promising results. The RL approach has been experimented for pedestrian simulation in [7]: the authors defined a perception model providing the agent with relevant information about a finite set of nearby agents, the nearest obstacle and the final goal, and they define an action model that basically regulates agent's velocity vector in terms of angle variation and acceleration/deceleration. This approach ispired a first version of the model discussed in the present paper, whose initial results [8] showed the practical possibility to achieve plausible results. This contribution presents the outcomes of a set of experiments performing a sensitivity analysis, changing some model elements, and exploring the implication of changes in the hyperparameters of the initially proposed training process. The models we want to achieve represent an alternative to already existing path planning models and pedestrian agent control mechanisms, in situations where the goal of the agent is to reach a final target, passing by intermediate targets, if necessary, showing a realistic pedestrian behaviours: an immediate exploitation of this work may be its inclusion in Unity, the platform used for this experiment, which is a largely adopted Game Engine, where the offered model for moving avatars follows the shortest path, resulting in a unrealistic movement of the avatar.

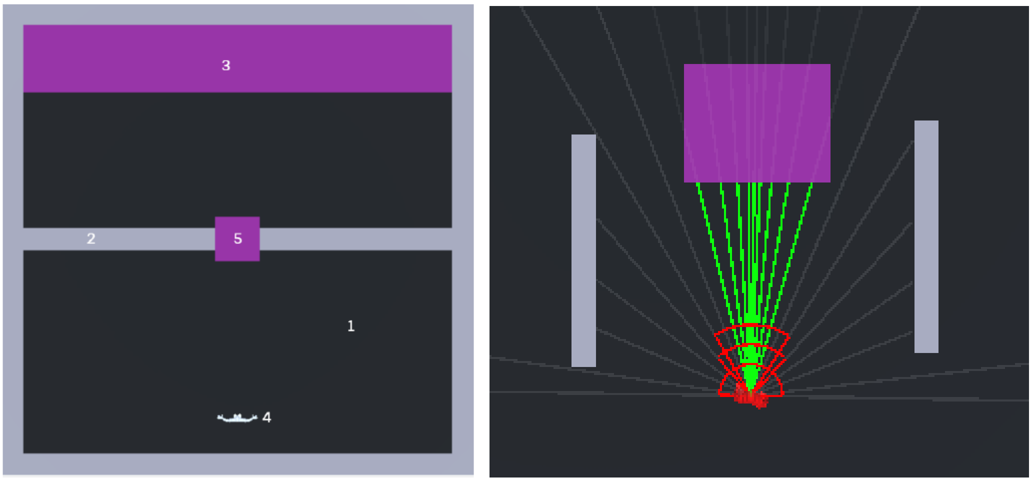
The paper is organized as follows: Section 2 presents the baseline RL model and Section 3 introduces the Curriculum Based approach. Section 4 provides an overview of the simulator supporting the experiments, while Section 5 describes the rationale of the overall analysis and presents selected results, supporting the claim that this research line is worth continuing despite the current limitations. Section 6 discusses some immediate research directions that could improve the practical applicability of the achieved results.

2. Reiforcement Learning Pedestrian Model

2.1. Representation of the Environment

For the experimental study presented in this paper, we adopted environments of 20 × 20 metres surrounded by walls, with different internal structures, although the models and simulator can work with environments of different size. Walls and obstacles are represented in gray, violet rectangles are intermediate and final targets. These violet areas are markers: they do not prevent the possibility of moving through them, but they are perceivable by agents such as gateways, mid-sections of bends, exits, and they support agent's navigation of the environment. The modeler must therefore perform an annotation of the environment before using it, as showed for example in Figure 1(a).

[4] https://ped.fz-juelich.de/da/doku.php



(a) Environment elements

(b) Rays of perception

Figure 1: (a) Environment elements: walkable area (1), walls (2), final target (3), agent (4), intermediate target (5); (b) rays of agent perception.

2.2. Agent Perception

Agents perceive the environments by means of a set of projectors generating *rays* extending up 14 m (in these experiments) that provide indications on what is hit and its distance from the agent. Projectors are distributed around the agent according to this rule: $\alpha_i = Min(\alpha_{i-1} + \delta * i,\ max\_vision)$ where δ has been set to 1.5, *max_vision* to 90 and α_0 to 0. This grants a more granular perception in the direction of movement and more sparse perception of the sides of the agent (see Fig. 1(b)). There are thus 23 angles, and for each of them two rays are projected to collect information supporting both navigation among different rooms and within rooms, and agents avoidance. The agent is also provided with cones in which it can detect the presence of walls and other agents, for supporting close range avoidance behaviours.

The overall agent's observation is summarized in Table 1: in addition to the above mentioned information, it includes basic agent's state (current velocity). To improve the performance of neural networks typically employed DRL algorithms, all numerical observations have been normalized in the interval [0,1].

Type	Observation	Value
Self Information	Own velocity	Number
Walls and targets	Distance	Number
	Type/tag	One Hot Encoding
Pedestrian	Distance	Number
	Direction	Angle
	Speed	Scalar

Table 1
Summary of agent's observations.

2.3. Action Model

The regulation of the velocity vector related to agent's movement (magnitude and direction of walking speed) is the only action managed by the action model. In line with the literature [9], agents take three decisions per second. Each agent is provided with an individual desired velocity sp_{des} that is drawn from a normal distribution with average of 1.5 m/s and a standard deviation of 0.2 m/s. Agent's action space has been therefore defined as the choice of two continuous values in the [-1,1] interval that are used to determine a change in velocity vector, respectively for magnitude and direction. The first element causes a change in the walking speed defined by Equation 1:

$$sp_t = Max\left(sp_{min},\ Min\left(sp_{t-1} + \frac{sp_{des} * a_0}{2},\ sp_{des}\right)\right) \tag{1}$$

where $speed_{min}$ is set to 0. According to this equation, the agent is able to reach a complete stop or the maximum velocity is two actions (i.e. about 0.66 s).

The second element of the decision determines a change in agent's direction; in particular, $\alpha_t = \alpha_{t-1} + a_1 * 25$. The walking direction can therefore change 25° each 0.33s; while this angle is plausible for normal pedestrian walking, this value is arbitrary and one of the experiments that will be described in Section 5 is about an evaluation of different values for the maximum turning angle.

2.4. Reward Function

Any RL approach heavily relies on its reward function, which is the only feedback signal guiding the learning process. The form of decision making we are dealing with is complex, comprising conflicting tendencies (e.g. imitation but proxemic tendency to preserve personal space) that are generally reconciled quickly, almost unconsciously, in a combination of individual and collective intelligence, that generally leads to sub-optimal overall performance [10].

Considering this, we hand-crafted a reward function considering factors recognized to be generally influencing pedestrian behavior, and performing a tuning of the related weights defining the relative importance of the different factors. The overall reward function is defined in Equation 2:

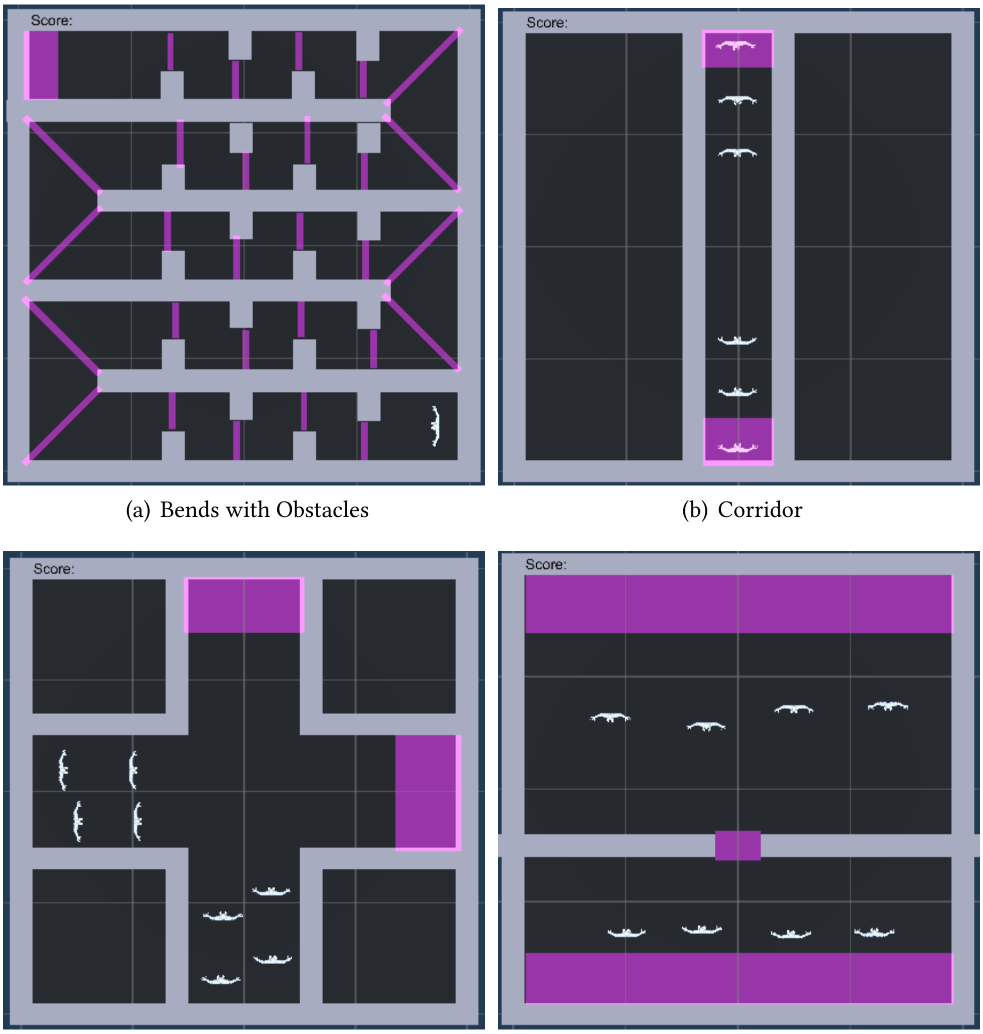
$$Reward = \begin{cases} \text{Final target reached} & +6 \\ \text{Intermediate target reached for the first time} & +0.5 \\ \text{Intermediate target reached again} & -1 \\ \text{No target in sight} & -0.5 \\ \text{Wall in proximity} < 0.6 \text{ m} & -0.5 \\ \text{Pedestrian in proximity} < 0.6 \text{ m} & -0.5 \\ \text{Pedestrian in proximity} < 1 \text{ m} & -0.005 \\ \text{Pedestrian in proximity} < 1.4 \text{ m} & -0.001 \\ \text{Elapsed timestep} & -0.0001 \end{cases} \tag{2}$$

The cumulative reward can only increase reaching the final target or a valid intermediate target (one that leads towards the final target, but reached only once). Other actions (associated to an implausible choice or simply to the fact that another decision turn has passed without reaching the final target) will instead yield a negative reward. Reaching the end of an episode of training (we will provide more information about them later on) without completing the scenario will lead to a substantial negative reward.

2.5. Adopted RL algorithm

We exploited the Proximal Policy Optimization (PPO) [11], a state–of–the–art RL policy–based algorithm, as implemented by ML-Agents [5]. PPO is a policy gradient algorithm that directly learns the policy function π, responsible for selecting actions in a given situation, without the need for a value function (which estimates the expected return of an action in a given state). Compared to dynamic

[5] https://github.com/Unity-Technologies/ml-agents



(a) Bends with Obstacles

(b) Corridor

(c) Intersection

(d) Bidirectional Door

Figure 2: A selection of training environments.

programming methods, which rely on value functions, policy gradient methods generally exhibit better convergence properties but require a larger set of training samples. Policy gradients work by learning the policy's parameters through a policy score function, denoted as $J(\Theta)$, where Θ represents the policy's parameters. This score function is optimized using gradient ascent, aiming to maximize the policy's performance. A common way to define the policy score function is through a loss function:

$$L^{PG}(\Theta) = E_t[log\pi_\Theta(a_t|s_t) \cdot A_t] \tag{3}$$

where $log\pi_\Theta(a_t|s_t)$ represents the log probability of taking action a_t given state s_t, and A_t is the advantage function, estimating the relative value of the taken action. When the advantage estimate is positive, the gradient is positive as well, leading to an increase in the probability of selecting the correct action. Conversely, the probabilities of actions associated with negative contributions are decreased. Through this mechanism, the policy gradually improves by iteratively updating its parameters. An exploration of the effect of actions in different situations is therefore necessary, but the approach is fundamentally different from supervised learning, since no annotated dataset is necessary.

Table 2
Training Environments Curriculum.

Behaviour	Environment	Retrain
Steer and walk towards a target	StartEz	×
	Start	✓
Steer to face target	Observe	✓
Reach the target in narrow corridors	Easy Corridor	×
Walk through bends avoiding walking too close to walls	Bends	×
	Bends with Obstacles	✓
Walk towards target but preserve social distance from agents moving in compatible or conflicting directions	Corridor	✓
	Unidirectional door	✓
	Intersection	✓
	T Junction	✓
	Bidirectional Door	✓

3. Curriculum Based Learning Process

3.1. Curriculum Learning for Reinforcement Learning

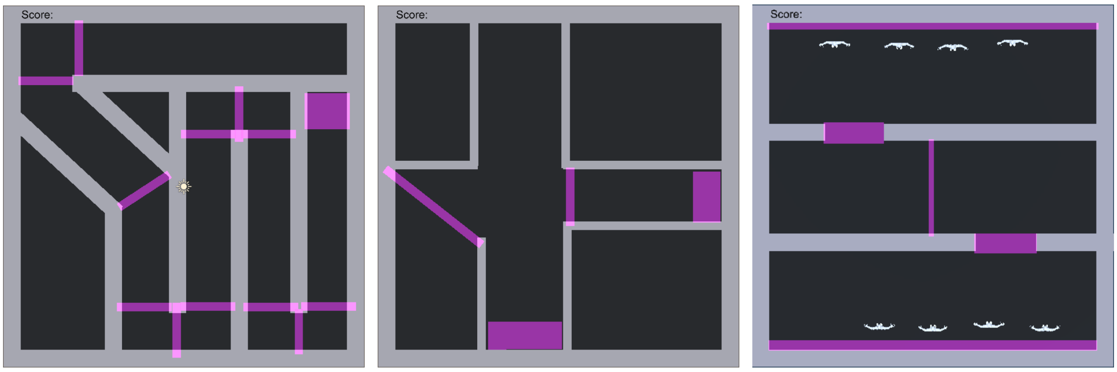
Curriculum Learning [6] was defined with the aim of reducing the training times for supervised ML approaches by adopting a cognitively plausible approach: proposed examples increase in difficulty during the training, illustrating gradually more complicated situations to the learning algorithm. Within the RL context, it has been employed as a *transfer learning* technique [12]: the idea is that the agent exploits experiences acquired on simpler scenarios when facing more complex ones within the training process, in an *intra–agent* transfer learning scheme. In addition to improving convergence, it has been reported that in some situations it supported better generalization properties in the learned policies [13]. We adopted this approach especially considering this final aspect: we wanted to train a single model directly applicable to new environments, without having to perform training for every specific one. The finally adopted approach first trains agents in a set of scenarios of growing complexity, one at a time, but then it also provides a final simultaneous retraining of the agent in a selected number of scenarios before the end of the overall training, to refresh previously acquired competences.

For sake of clarity in the remainder of the paper, we define more clearly some key concepts:

- *scenario (or step) of the curriculum*: a specific environment and specific conditions for considering this part of the training process completed;
- *episode*: each scenario generally needs to be experienced several times, each of them called an *episode*, to accumulate experience; each episode ends after a maximum time or by achieving the goal of the scenario;
- each episode therefore leads to the achievement of a *cumulative reward*, that is the summation of instant rewards achieved as a consequence of each decision and action step;
- for a given scenario, the above mentioned *completion condition* is modeled as a mathematical test over the episodes's *cumulative rewards*: a typical completion condition could be, for example, "the average cumulative rewards of the last 10 episodes is higher than threshold th_i".

3.2. The Proposed Curriculum

The proposed curriculum is described in Table 2: it includes very simple environments in which the agent learns how to steer to look for the final target and walks towards it with just perimetral walls, then it has to face situations in which the environment is narrow (a basic corridor) and in which bends are present. Then social interaction is introduced, first of all with agents with compatible directions, then with conflicting ones, in geometries presenting bends and even bottlenecks. A selection of training



(a) Omega bends environment (b) Blind bend environment (c) Double door counterflow

Figure 3: A selection of test environments considered for this work.

environments is shown in Figure 2, while the complete list of environments is described in Table 2. Most scenarios are characterised by a certain degree of stochasticity: for instance, the initial position (and facing) of the agent is randomly determined (within a given area), and the environment is flipped during the training process, with the goal of avoiding overfitting. After all the environments have been proposed and the training has brought the agent to achieve the necessary level of average accumulated reward in a defined number of consecutive episodes, a retraining phase starts. In this phase, a selection of scenarios starts simultaneously: this phase ensures that the experience brought by the first scenarios is not forgotten, and that the final policy can successfully face any of the training scenarios.

To evaluate the level of generalization achieved through this learning process, we also defined a set of *test scenarios*, that are not part of the curriculum, and used afterwards to evaluate the final policy. Figure 3 shows a selection of these environments: bends with different angles, in various combinations, with pedestrian flows (with different final targets) crossing and even counterflows were considered.

The choice and order of scenarios is based on knowledge about the simulated phenomena and on preliminary tests to evaluate the practical convergence of the process, but it is arbitrary: some of the experiments that were carried out (Section 5) were related to both the evaluation of the higher level of generalization of the curriculum based approach and changes in the curriculum structure.

4. The Simulation and Training System

The system developed to experiment the proposed approach is based on Unity[6]: in particular, the scenarios, agents, perceptive capabilities, as well as the necessary monitoring components for acquiring data about the pedestrian dynamics and data structures representing concepts related to the approach (e.g. curriculum), are implemented as Unity Prefabs[7] and C# scripts. Unity does not directly include components supporting DRL techniques, but the ML-Agents toolkit[8] provides both an extension to the Unity environment as well as a set of Python components enabling training and using DRL based agents. In particular, ML-agents provides a Python (and PyTorch) based trainer able to receive inputs associated to environmental signals (available actions, observations, and rewards), to manage DRL learning processes. To connect Unity and the trainer, ML-Agent needs to wrap Unity, defining a communicator component realizing an inter-process communication with the trainer through a Python API. The overall architecture is depicted in Figure 4.

During training, scenarios are run in Unity, while in parallel the ML-Agents trainer process must be also running, to receive and process signals from the environment, to perform DRL training. Then, the

[6] https://unity.com
[7] https://docs.unity3d.com/Manual/Prefabs.html
[8] https://github.com/Unity-Technologies/ml-agents

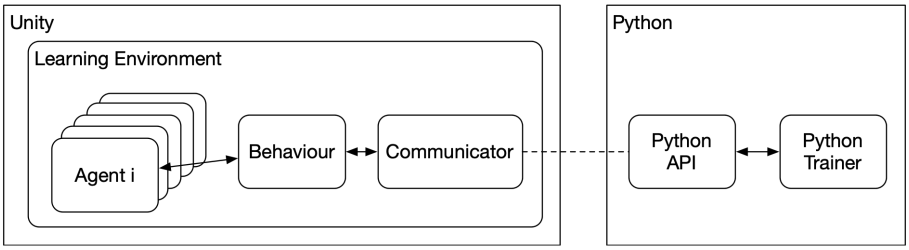


Figure 4: Unity - MLAgents components and interaction.

achieved policies can be saved and used directly within Unity without the need to have the ML-Agent trainer running (or even installed locally in the machine running the specific Unity instance), thus realizing the aim of this work, that is, generating realistic pedestrian models to be used, for example, in simulations where autonomous avatars are needed. We are in the process of organizing an open repository in which we will share the developed code as well as the training and test scenarios.

5. Achieved Results

First results of this approach confirmed the possibility complete the training process achieving plausible results also in situations not included in the training curriculum [8]. However, several choices were arbitrary, and a sensitivity analysis for some modeling elements (in particular the maximum turning angle) were not carried out, as well as an exploration of hyperparameters of the training process. In this work we describe a set of experiments exploring these aspects, acquiring additional insights on the most promising ways to continue this line of research and current (or definitive) limits.

Figure 5 summarizes the carried out experiments. First of all, we focused on the curriculum, backing up the claim that the curriculum based approach grants a higher level of generalization compared to agents trained in a single scenario; we also evaluated the impact of the retraining phase. A second block of training processes and experiments evaluated alternative choices in the maximum turning angle for each decision step. Then, we evaluated the impact of performing simpler or more complex training processes by tuning the hyperparameters of the ML-Agent trainer: within this block of experiments we also considered a discrete action model (supporting only discrete changes of the velocity vector), but also an increased presence of stochastic elements in the training scenarios. For sake of space, we will only visually present some of the achieved results and we will only comment the results of some experiments.

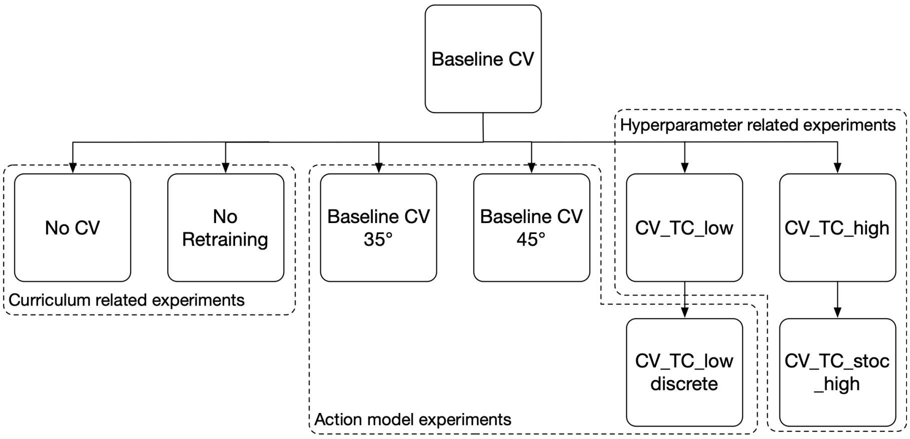


Figure 5: Performed experiments.

5.1. Curriculum Related Experiments

The first experiment related to the presence and structure of the curriculum in the training process compared an agent trained with the baseline curriculum (*Baseline CV* in Figure 5) with an agent trained on just the Bidirectional Door scenario (*No CV* in Figure 5). This experiment might seem unnecessary, since single scenarios of the proposed curriculum are not sufficiently large and varied to represent all the variety and difficulties proposed by the union of the scenarios. However, it would be very hard to create a single environment in which all of the situations faced by trained agents in the whole curriculum would be faced. Moreover, this experiments provides a quantitative idea of the effects of curriculum on training process duration as well its capability to support a good level of generalization. Figure 6 (a) and (b) describe the trend of the cumulative reward acquired by trained agents in the baseline curriculum (*Baseline CV*) and in a training carried out on just the Bidirectional Door scenario (*No CV*): the duration of the training process for just the Bidirectional Door scenario is comparable to the duration of the whole curriculum based process (more or less 3500 seconds), excluding the retraining and consolidation phase. A more complicated single training scenario, more generally covering additional competences compared to Bidirectional Door that does not propose bends, would require an even longer training process. These results corroborate the idea that a gradual acquisition of experiences simplifies and makes more rapid the convergence of the overall training process.

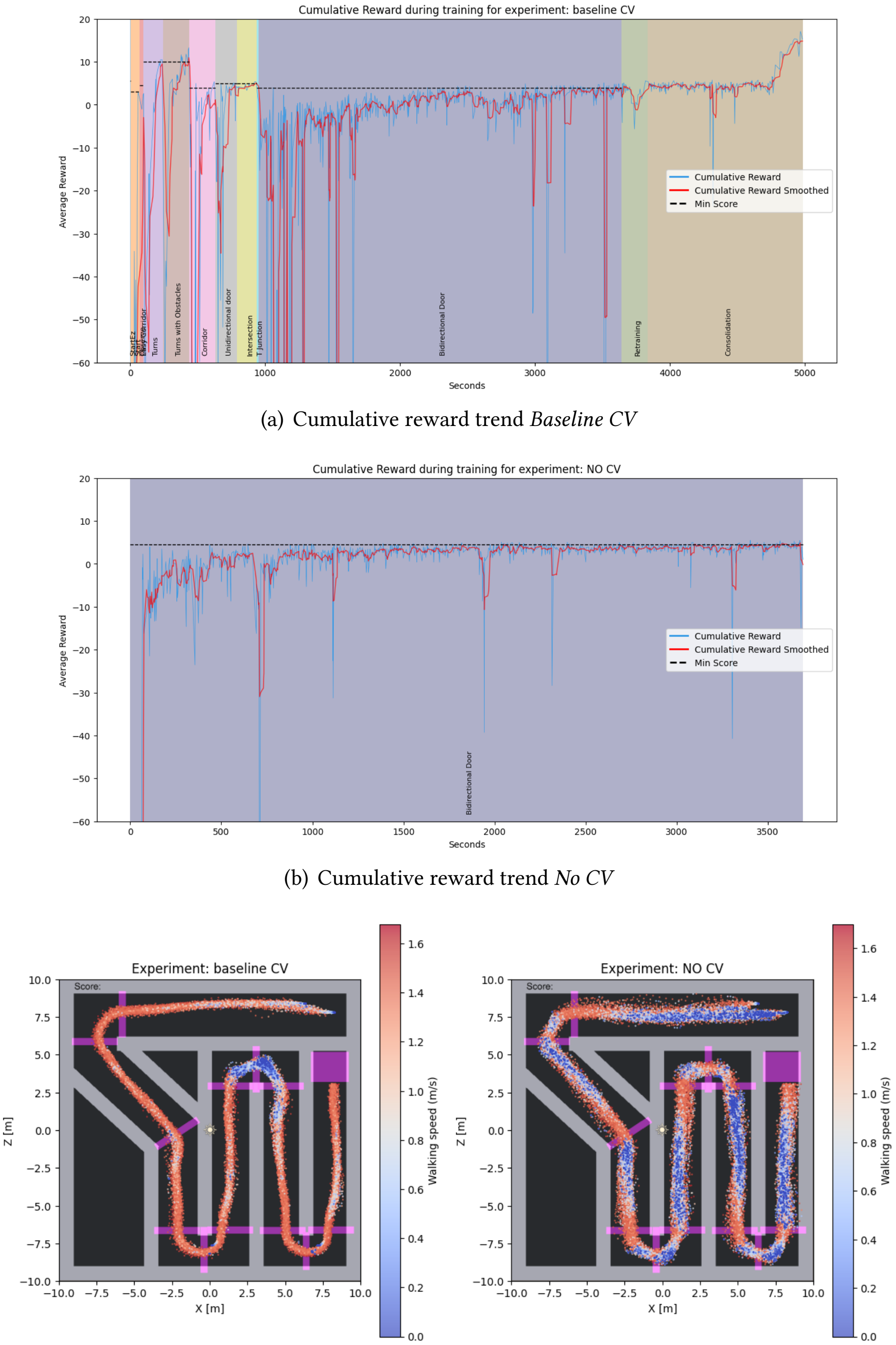
Comparing the two models in one of the test environments we can see how the model achieved with the *No CV* procedure performs poorly. Figure 6(c) shows the quality of both models when facing the Omega bends environment, simultaneously showing trajectories and walking speeds of 50 runs in which two agents move from the upper right corner to the lower right one: the quality is much higher for the *Baseline CV* model, the agents trained solely on the Bidirectional door scenario have learned how to manage social interaction, at least in that kind of environment, but they have a hard time navigating through bends without slowing down frequently and significantly.

We have also evaluated the impact of the retraining and consolidation process (not reported due to space limitation): while they surely have a noticeable cost in terms of training time, results showed that they have an important effect in allowing the *Baseline CV* agents in remembering how to face situations encountered in the early steps of the curriculum.

5.2. Modeling Alternatives and Hyperparameters

An arbitrary modeling choice was about the maximum angle for pedestrian turning per step, set to 25°: within this experiment we trained agents that could turn up to 35° (*Baseline CV 35°*) and 45° *(Baseline CV 45°*) per decision step. Differences in the duration of the training process were minimal, but the observable behaviour of the achieved policies was instead very different, as shown in Figure 7. Figure 7(a) shows the different paths followed by two agents (first and second group) simultaneously present in the Omega bends environment (again, 50 episodes are shown): *Baseline CV 35 °* agents have a hard time moving in the second U turn, as well as in some of the corridors, slowing down unnecessarily and often moving very close to the walls; *Baseline CV 45 °* agents move more regularly, but their trajectories have a quite high variability, and they sometimes slow down without apparent reasons. Figure 7(b) shows the frequency of adopted rotation angles: *Baseline CV* agents often have small regulation of the angle, and they also have all right/all left turning decisions; *Baseline CV 35°* almost only make all right/all left turning decisions; finally, *Baseline CV 45°* agents instead never take all right turning decisions, and they rarely take all left turning decisions. The overall density of the achieved cumulative reward per episode is shown in 7(c), confirming that the 25°maximum rotation yields the best results.

Instabilities in the results of some of the models described so far pushed us to consider evaluating the effect of allowing a longer, more through training process for the neural network employed by PPO algorithm. We considered performing training with a smaller network (just one hidden layer, made up of 128 units) and smaller batch size and buffer size; we also tried training the same network of the baseline model (two hidden layers, with 256 hidden units) with a larger batch size and buffer size. The buffer size is the number of experiences to collect before updating the policy model, and the batch size



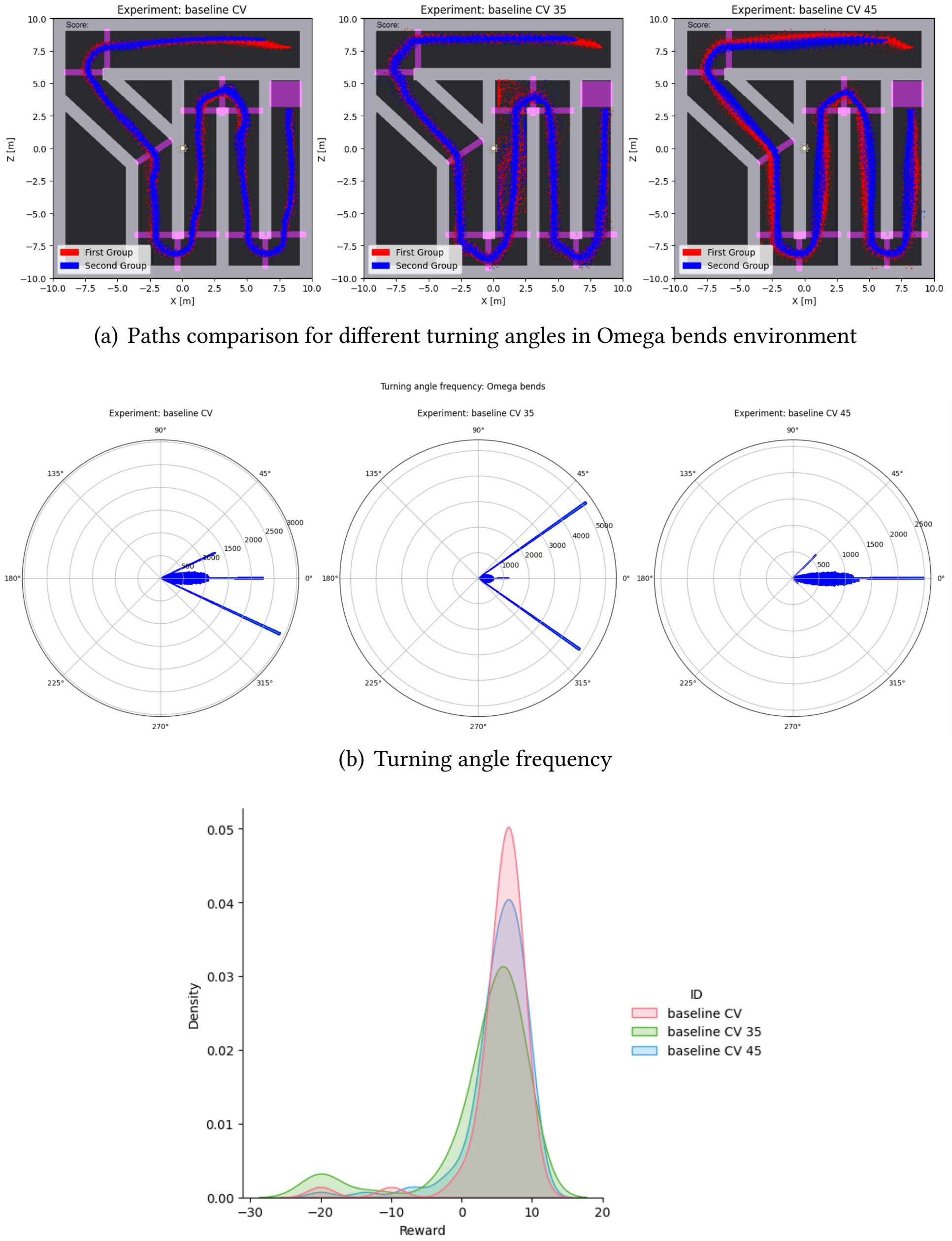
(a) Cumulative reward trend *Baseline CV*

(b) Cumulative reward trend *No CV*

(c) Paths in Omega bends environment

Figure 6: *Baseline CV* vs *NO CV*.

is the number of experiences in each iteration of the gradient descent. Results of the training with the smaller network (*CV_TC_low*) showed significantly lower training times, associated to good results in test scenarios in which social interaction was absent or simple (compatible goals, few conflicts), but bad results in scenarios in which social interactions are frequent and require slowing down and/or performing sharp collision avoidance maneuvers. Considering the simpler structure of the network, we also tried adopting a simpler action model, with discrete instead of continuous, would improve the situation (*CV_TC_low_discrete*). This prediction turned out to be true, yielding improvements that led to



(a) Paths comparison for different turning angles in Omega bends environment

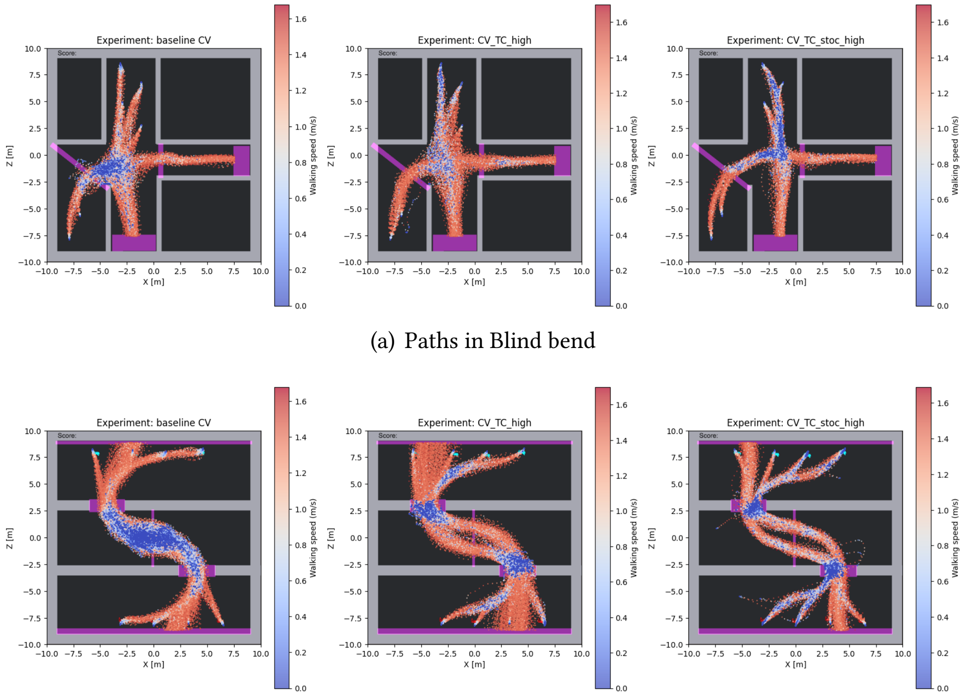
(b) Turning angle frequency

(c) Overall reward comparison

Figure 7: Models with different maximum turning angles.

comparable results with the baseline model with lower training costs: the *CV_TC_low_discrete* training process represents a potentially useful approach when social interactions are not frequent (i.e. very low density scenarios).

The training process employing larger buffer and batch sizes (*CV_TC_high*) was expected to grant improvements in case of intense social interaction, due to the possibility to better explore the complex patterns arising from the interactions of multiple agents. Results, however, showed longer training times (see Figure **??**), improvements in environments with frequent and difficult social interactions, but also instabilities also in relatively simple test environments. We interpreted this as a sign of overfitting. We decided, as a final attempt in this set of experiments, to further increase the intensity of the stochastic elements in the individual initialization of the training scenarios, increasing the frequency and extent of random changes in the initial position of agents, flips and changes in the spatial structure of the



(a) Paths in Blind bend

(b) Paths in Double door counterflow

Figure 8: Paths for test environments for odels with larger buffer and batch size, and with higher degree of stochasticity.

environments. This interestingly led to a decreased training time (from about 8000s to less than 6500s): the increased variety of the situations simplified the convergence to a policy able to stably achieve higher rewards. Regarding the quality of the dynamics generated by the model, 8(a) and 8(b) show patterns of interactions respectively in the Blind bend (4 agents moving from the upper side and two agents from the lower left corner) and Double door (8 agents, equally divided in the upper and lower rooms) counterflow test scenarios, again 50 episodes for each graph. Agents of the *CV_TC_high_stoc* model have more stable trajectories, with fewer drops in walking speed, suggesting that increasing the variety of situations encountered by the agents in the training process is even more important with the growth of the network capacity and with the thoroughness of the training process for the neural network employed by the PPO algorithm.

5.3. Limitations

Even though all the performed experiments employed components, mechanisms, and functionalities offered by Unity, in principle nothing prevents employing open source alternatives such as Godot [9]. In any case, due to difficulties in managing complicated patterns of movement for the 3D models of the agents, we do not believe that this kind of model could scale to levels of density consistently higher than 2 pedestrians per square meter.

The performed experiments do not consider situations in which agents need to reach specific intermediate points of the environment in movement plans. Agents trained through this process act here and now: they essentially depend on the environment and annotations to reach the final target of their movement. We have worked on extensions of the model supporting exploration the environment to reach specific intermediate targets before the final one [14], but these preliminary results do not consider the social interaction part that is instead central in this work. An integration of these aspects would be necessary for a realistic application of this kind of approach in real world pedestrian simulation

[9] thttps://godotengine.org/

systems, and it is object of current and future works. Validation of the model represents a task that will be more seriously tackled afterwards, the preliminary results are encouraging but still partial.

We did not (yet) take a Multi-Agent [15] perspective to Reinforcement Learning, and in a sense this work represents an exploration of the limits that the single agent approach can reach in situations that are actually characterized by the simultaneous presence of autonomous agents influencing each other's actions and performances.

6. Conclusions and Future Developments

This paper presented a curriculum based DRL approach to pedestrian modeling and simulation. The model and training approach were presented, as well as a set of experiments (i) showing the fact that curriculum based DRL models effectively grant a higher level of generalization compared to models trained even in challenging scenarios, at the cost of a relatively little overhead; (ii) showing the effect of changes both in model configuration (in particular the action model) and in hyperparameters of the learning algorithm, and suggesting lines for new research in the field to overcome current limitations.

Future works are aimed at extending the model to embed the capability to explore and plan paths in the environment granting the possibility to reach specific intermediate targets in a more complicated environment, as well as a more thorough validation of the achieved results that might suggest extensions to the current curriculum by adding further scenarios that should support the acquisition of additional movement competences to the agents. The proposed morel, as of this moment, only takes a relatively shallow approach to the evaluation of mutual distances among pedestrians: the recent COVID19 outbreak has shown that contextual conditions can call for more granular and individual consideration of interpersonal distances, potentially considering affective states [16].

Besides model improvement, and in addition to supporting designers' and decision makers' activities implying the need to simulate pedestrians, these models could be also used in Virtual Reality systems for guiding avatars [17] that should exhibit plausible behaviors.

Acknowledgements

This work was partly developed within the Spoke 8 — MaaS and and Innovative services of the National Center for Sustainable Mobility (MOST) set up by the "Piano nazionale di ripresa e resilienza (PNRR)—M4C2, investimento 1.4, "Potenziamento strutture di ricerca e creazione di "campioni nazionali di R&S" su alcune Key Enabling Technologies" funded by the European Union. Project code CN00000023, CUP: D93C22000410001. This work was also partially supported by the MUR under the grant "Dipartimenti di Eccellenza 2023-2027" of the Department of Informatics, Systems and Communication of the University of Milano-Bicocca, Italy.

References

[1] M. Batty, Agent-based pedestrian modeling, Environment and Planning B: Planning and Design 28 (2001) 321–326. URL: https://doi.org/10.1068/b2803ed. doi:10.1068/b2803ed. arXiv:https://doi.org/10.1068/b2803ed.

[2] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, A. Seyfried, Evacuation Dynamics: Empirical Results, Modeling and Applications, in: Encyclopedia of Complexity and Systems Science, Springer New York, 2009, pp. 3142–3176. URL: https://link.springer.com/referenceworkentry/10.1007/978-0-387-30440-3_187. doi:10.1007/978-0-387-30440-3_187.

[3] D. Helbing, P. Molnár, Social force model for pedestrian dynamics, Phys. Rev. E 51 (1995) 4282–4286. doi:10.1103/PhysRevE.51.4282.

[4] P. Kothari, S. Kreiss, A. Alahi, Human trajectory forecasting in crowds: A deep learning perspective, IEEE Transactions on Intelligent Transportation Systems (2021) 1–15. doi:10.1109/TITS.2021.3069362.

[5] R. S. Sutton, A. G. Barto, Reinforcement Learning, an Introduction (Second Edition), MIT Press, 2018. ISSN: 01406736.

[6] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 41–48. URL: https://doi.org/10.1145/1553374.1553380. doi:10.1145/1553374.1553380.

[7] F. Martinez-Gil, M. Lozano, F. Fernández, Emergent behaviors and scalability for multi-agent reinforcement learning-based pedestrian models, Simulation Modelling Practice and Theory 74 (2017) 117–133. URL: https://www.sciencedirect.com/science/article/pii/S1569190X17300503. doi:https://doi.org/10.1016/j.simpat.2017.03.003.

[8] G. Vizzari, T. Cecconello, Pedestrian simulation with reinforcement learning: A curriculum-based approach, Future Internet 15 (2023). URL: https://www.mdpi.com/1999-5903/15/1/12. doi:10.3390/fi15010012.

[9] S. Paris, S. Donikian, Activity-Driven Populace: A Cognitive Approach to Crowd Simulation, IEEE Computer Graphics and Applications 29 (2009) 34–43.

[10] M. Haghani, M. Sarvi, Imitative (herd) behaviour in direction decision-making hinders efficiency of crowd evacuation processes, Safety Science 114 (2019) 49–60. URL: https://www.sciencedirect.com/science/article/pii/S0925753518309275. doi:https://doi.org/10.1016/j.ssci.2018.12.026.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, CoRR abs/1707.06347 (2017). URL: http://arxiv.org/abs/1707.06347. arXiv:1707.06347.

[12] F. L. D. Silva, A. H. R. Costa, A survey on transfer learning for multiagent reinforcement learning systems, Journal of Artificial Intelligence Research 64 (2019) 645–703. doi:10.1613/jair.1.11396.

[13] B. Baker, I. Kanitscheider, T. M. Markov, Y. Wu, G. Powell, B. McGrew, I. Mordatch, Emergent tool use from multi-agent autocurricula, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkxpxJBKwS.

[14] G. Vizzari, D. Briola, T. Cecconello, Curriculum-based reinforcement learning for pedestrian simulation: Towards an explainable training process?, in: R. Falcone, C. Castelfranchi, A. Sapienza, F. Cantucci (Eds.), Proceedings of the 24th Workshop "From Objects to Agents", Roma, Italy, November 6-8, 2023, volume 3579 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 32–48. URL: https://ceur-ws.org/Vol-3579/paper3.pdf.

[15] K. Zhang, Z. Yang, T. Başar, Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms, Springer International Publishing, Cham, 2021, pp. 321–384. URL: https://doi.org/10.1007/978-3-030-60990-0_12. doi:10.1007/978-3-030-60990-0_12.

[16] S. Bandini, D. Briola, A. Dennunzio, F. Gasparini, M. Giltri, G. Vizzari, Distance-based affective states in cellular automata pedestrian simulation, Nat. Comput. 23 (2024) 71–83. URL: https://doi.org/10.1007/s11047-023-09957-y. doi:10.1007/S11047-023-09957-Y.

[17] D. Briola, F. Tinti, G. Vizzari, Creating virtual reality scenarios for pedestrian experiments focusing on social interactions, in: M. Alderighi, M. Baldoni, C. Baroglio, R. Micalizio, S. Tedeschi (Eds.), Proceedings of the 25th Workshop "From Objects to Agents", Bard (Aosta), Italy, July 8-10, 2024, volume 3735 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 161–176. URL: https://ceur-ws.org/Vol-3735/paper_13.pdf.