A Curriculum–Based Reinforcement Learninig Approach to Pedestrian Simulation

Thomas Albericci[1], Thomas Cecconello[1], Alberto Gibertini[1] and Giuseppe Vizzari[1]

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

Abstract

Reinforcement Learning represents a way to train an agent situated in an environment what to do to maximise an accumulated numerical reward signal (received by the environment as a feedback to every chosen action). Within this paper we explore the possibility to apply this approach to pedestrian modelling: pedestrians generally do not exhibit an optimal behaviour, therefore we carefully defined a reward function (combining contributions related to proxemics, goal orientation, basic wayfinding considerations), but also a particular training *curriculum*, a set of scenarios of growing difficulty supporting the incremental acquisition of proper orientation, walking, and pedestrian interaction competences. The paper will describe the fundamental elements of the approach, its implementation within a software framework employing Unity and ML-Agents, describing the promising achieved simulation results.

Keywords

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

1. Introduction

Reinforcement Learning (RL) [1] is a machine learning approach that is being growingly investigated as way to achieve autonomous agents, where the acceptation of the term “autonomous” is closer to Russell and Norvig’s [2] than the most widely adopted ones in agent computing. Russell and Norvig state that:

> *A system is autonomous to the extent that its behavior is determined by its own experience*

RL represents a way to train an agent situated in an environment what to do to maximise an accumulated numerical reward signal (received by the environment as a feedback to every chosen action). The agent is provided with a model of perception and action, but besides these modelling elements and the reward function, the approach can autonomously explore the space of potential agent behaviours and converge to a policy (i.e. a function mapping the state and perception to an appropriate action to be carried out in that context).

A certain amount of initial knowledge (in an analogy to built-in reflexes in animals and humans, but also internalized norms, rules, and even ways to evaluate the degree of acceptability

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

t.albericci1@campus.unimib.it (T. Albericci); thomas.cecconello@unimib.it (T. Cecconello); a.gibertini@campus.unimib.it (A. Gibertini); giuseppe.vizzari@unimib.it (G. Vizzari)
0000-0002-7916-6438 (G. Vizzari)

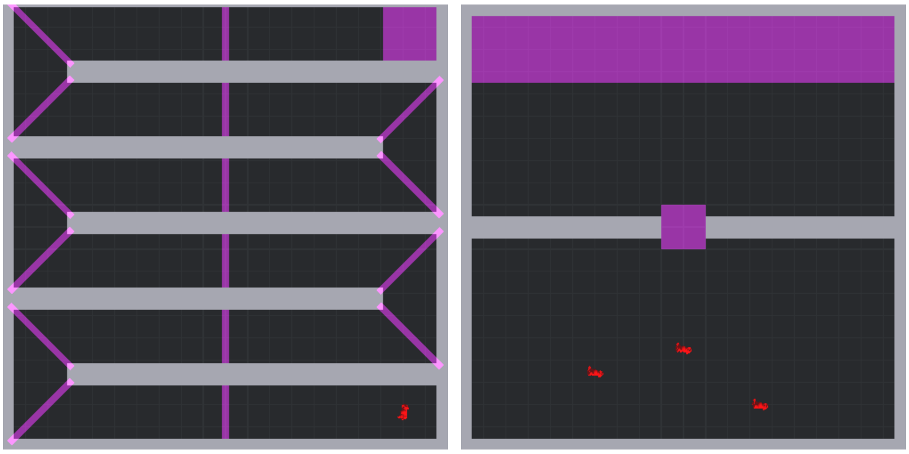
of a state of affairs) is therefore considered by the approach, but it should be sided by the ability to learn, to adjust one's behaviour to achieve a better performance. RL approaches, reinvigorated by the energy, efforts, and promises brought by the *deep learning* revolution, seems one of the most promising ways to investigate how to provide an agent this kind of autonomy. On a more pragmatic level, recent developments and results in the RL area suggest that this approach might even be a promising alternative to current agent-based approaches to the modeling of complex systems [3]: whereas currently behavioral models for agents are carefully hand crafted, often following a complicated interdisciplinary effort involving different roles and types of knowledge, as well as validation processes based on the acquisition and analysis of data describing the studied phenomenon, RL could simplify this work, focusing on the definition of an environment representation, the definition of a model for agent perception and action, and defining a reward function. The learning process could, in theory, be able to explore the potential space of the *policies* (i.e. agent behavioral specifications) and converge to the desired decision making model. While the definition of a model of the environment, as well as agent perception and action, and the definition of a reward function are tasks requiring substantial knowledge about the studied domain and phenomenon, the learning process could significantly simplify modeler's work, and at the same time it could solve issues related to model calibration. Although some relevant related work can be found in the literature (in particular [4]), results achieved so far highlight significant limitations, especially in the capability of generalization of the training phase: this is a very important aspect for this kind of application, not just because it is inconvenient to pay this computational cost for every scenario to be analyzed, but also due to the fact that results could not be actually comparable, since they would be achieved with different simulation models.

Within this line of work, and building on preliminary efforts [5], this paper describes an experimentation of this approach to pedestrian modelling. Whereas is RL agents learn how to behave to optimize their expected cumulative reward, pedestrians generally do not exhibit an optimal behaviour. Therefore we carefully defined a reward function (combining contributions related to proxemics, goal orientation, but also basic wayfinding considerations). We also employed a particular training *curriculum* [6], a set of scenarios of growing difficulty supporting the incremental acquisition of proper orientation, walking, and pedestrian interaction competences.

The paper will describe the fundamental elements of the approach, its implementation within a software framework employing Unity[1] and ML-Agents[2], describing the promising achieved simulation results: in particular, we will show that the proposed approach is able to produce plausible results in environments that were not used for sake of training, so the approach seems promising at least in terms of generality. We will finally discuss the current limits of the approach, and our current implementation, as well as ongoing future developments.

[1] https://unity.com

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


(a)

(b)

Figure 1: (a) 'Turns' environment and annotations, and (b) 'Unidirectional Door' environment and annotations.

2. The Model

2.1. Representation of the Environment

For sake of simplicity in this experimental study environments are bound to be squares of 20 × 20 metres surrounded by walls. The smaller squares (of 1 ×1 metre) that can be seen in the figures presented later on are just for sake of allowing a simpler appraisal of distances. Gray objects are walls, obstacles and anything that agents perceive as 'Wall'.

Violet rectangles are intermediate and final goals. These markers (in the vein of [7]), do not hinder the possibility of moving through them, and they are essentially a modeling tool to support agent's navigation in the environment. In fact, one of the goals of the work is to provide an alternative to Unity's path finding and (more generally) pedestrian agent control mechanisms. Later we will describe agents' perceptive model, but we anticipate that they are able to perceive these markers and to select intermediate or final movement targets; we will also see that reaching intermediate or final targets will also influence agent's reward.

Environments must therefore undergo a preparation phase before being actually used in the proposed approach; an example of an environment annotated with this rationale is shown in Figure 1(a). In this case, the targets in the middle of the horizontal corridors create an affordance for agents to move towards that direction although the actual bend, at the end of the corridor, is fairly distant, and this could confuse agents during the training phase. Moreover, oblique targets in the bends guide agents in the change of direction, also helping them to achieve a plausible trajectory [8]. Figure 1(b) shows instead an environment in which a door (an open one, of course) is present: in this case, the target is used to guide agents passing through the opening, since the final target is obstructed and not perceivable from a large portion of the Southern room.

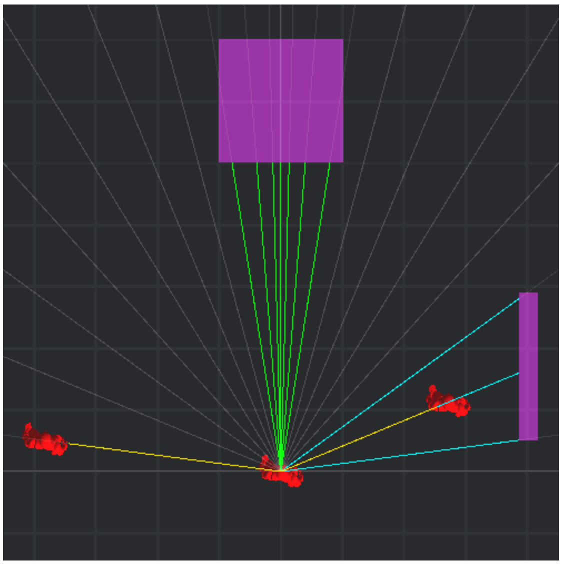


Figure 2: Rays and provided information: yellow = agent, cyan = intermediate target, green = final target, transparent = wall or none of the others. Pedestrian agents are depicted in red.

2.2. Agent Perception

Agents are provided with a limited set of *projectors* of rays, each extending up to a certain distance (10 m in these experiments) and providing specific information about what is "hit" by the ray and the associated distance from the agent.

Projectors (and therefore rays) are not uniformly distributed around the agent, but they are more densely present in from of the pedestrian, to loosely resemble real human visual perception.

The angle between the rays and the facing direction of an agent (both positive and negative) follows the rule described in Equation 1:

$$\alpha_i = Min(\alpha_{i-1} + \delta * i,\ max\_vision) \tag{1}$$

where δ has been set to 1.5, *max_vision* to 90 and α_0 to 0. As a consequence, projectors emit rays at $0°$, $\pm 1.5°$, $\pm 4.5°$, $\pm 9°$, $\pm 15°$, $\pm 22.5°$, $\pm 31.5°$, $\pm 42°$, $\pm 54°$, $\pm 67.5°$, $\pm 82.5°$ and $\pm 90°$. Figure 2 graphically depicts this distribution.

The overall number of projectors and rays would therefore be 23, but since the information associated to and conveyed by rays is different for different objects we actually have several projectors for each angle, and therefore each agent actually has 46 rays.

The overall agent's observation is summarized in Table 1. To improve the performance of neural networks typically employed in recent RL algorithms all observations have been normalized in the interval [0,1]. In particular, for normalization of walking speed we consider the maximum velocity for walking agents to be 1.7 m/s.

Information about Walls and Targets is provided to support basic wayfinding, whereas information about Agents and Walls is more detailed (including also the walking direction and

Table 1
summary of agent's observations.

Type of observation	Observation	Value
Intrinsic	Own speed	Number
Walls and Targets	Distance	Number
	Type/Tag	One Hot Encoding
Agents and Walls	Distance	Number
	Type/Tag	Boolean
	Direction	Number
	Speed	Number

speed, in case the ray 'hits" an agent) and it is provided to support more fine grained collision avoidance.

2.3. Action space

Each agent is provided with an individual desired velocity that is drawn from a normal distribution with average of 1.5 m/s and a standard deviation of 0.2 m/s. Each decision, and for these experiments we decided to grant agents three decisions per second (in line with [9], combining cognitive plausibility, quality of the achieved results, and computational costs), determines a potential change in its velocity and this is basically what agent's decision is all about for this model.

Agent's action space has been therefore modeled as the choice of two (conceptually) 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, a_0, causes a change in the walking speed defined by Equation 2:

$$speed_t = Max\left(speed_{min},\ Min\left(speed_{t-1} + \frac{speed_{max} * a_0}{2},\ speed_{max}\right)\right) \tag{2}$$

Where $speed_{min}$ is set to 0 and $speed_{max}$ is set to 1.7 m/s. 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, a_1, determines a change in agent's direction according to Equation 3:

$$\alpha_t = \alpha_{t-1} + a_1 * 20 \tag{3}$$

The walking direction can therefore change 20°each 0.33 s, that is plausible for normal pedestrian walking, but would be probably not reasonable for modeling running and/or sport related movements.

2.4. Reward Function

The reward function is a crucial element for a RL approach: it represents the feedback signal guiding the learning process, in a certain sense it represents a (weaker) substitute for labels in supervised learning. Moreover, here we deal with a complex form of decision making, with conflicting tendencies that are generally reconciled quickly, almost unconsciously, in a reasonable/explainable way (in retrospective) by the typical pedestrian, in a combination of individual and collective intelligence, that however leads to sub-optimal overall performance (see, for instance, the above cited [7] but also [10]).

Given the above considerations, we hand-crafted a reward function, initially in terms of components, i.e. factors generally influencing pedestrian behaviour. Later on we performed a sort of initial tuning of the related weights defining the relative importance of the different factors. A sensitivity analysis was not performed and it would be object of future works.

The overall reward function is defined in Equation 4:

$$Reward: \begin{cases} +6 & \text{Final target reached} \\ +0.5 & \text{Intermediate target reached} \\ -1 & \text{Reached a previously reached intermediate target} \\ -0.5 & \text{No target in sights} \\ -0.5 & \text{Agent in very close proximity - } < 0.6 \text{ m} \\ -0.005 & \text{Agent in close proximity } < 1 \text{ m} \\ -0.001 & \text{Agent in proximity } < 1.4 \text{ m} \\ -0.5 & \text{Wall in proximity } < 0.6 \text{ m} \\ -0.0001 & \text{Each step done} \\ -6 & \text{Reached the end of steps per episode} \end{cases} \tag{4}$$

The only ways to increase the cumulative reward are therefore the reaching of intermediate or final targets. However, reaching targets that have been previously visited brings a negative reward, since it would imply moving back from the final goal, and it makes it much less reasonable to try to “exploit” the reward to reach a formally reasonable but totally implausible policy (i.e. reach as many intermediate targets before reaching the final one before the end of the episode). Negative rewards thus are used to suggest that some actions should not be chosen unless they eventually lead to the final goal (and unless better alternatives do the same): a small negative reward granted due to the simple passage of time is usual, it pushes agents to avoid standing still and to actively look for solutions, but we also have negative rewards due to proxemics [11], and to penalize walking too close to walls (again, unless necessary). Finally, the penalization to actions leading to a position from which no target (either intermediate or final) can be seen stimulates agents to pursue the goals; one could wonder if having instead a small bonus for actually seeing a target would work analogously: all positive rewards should however be taken very carefully, since they can lead to pathological behaviours. In this case, in very complex scenarios, an agent might learn to find a target and stand still, achieving a relatively small bonus for each decision of the episode.

2.5. Adopted RL algorithm

For this research and experimentation we adopted Proximal Policy Optimization (PPO) [12], a state–of–the–art RL policy–based algorithm provided by ML-Agents. PPO is a policy gradient algorithm which works by learning the policy function π directly. These methods have a better convergence properties compared to dynamic programming methods, but need a more abundant set of training samples. Policy gradients work by learning the policy's parameters through a policy score function, $J(\Theta)$, through which is possible to apply gradient ascent to maximize the score of the policy with respect to the policy's parameters, Θ. A common way to define the policy score function is through a loss function:

$$L^{PG}(\Theta) = E_t[log\pi_{\Theta}(a_t|s_t)]A_t \tag{5}$$

which is the expected value of the log probability of taking action a_t at state s_t times the advantage function A_t, representing an estimate of the relative value of the taken action. As such, when the advantage estimate is positive, the gradient will be positive as well; through gradient ascent the probability of taking the correct action will increase, while decreasing the probabilities of the actions associated to negative advantage, in the other case.

The goal of the work was essentially to evaluate the adequacy of the approach to the problem of achieving a proper pedestrian simulation model and we did not yet analyze the performance of different RL algorithms, something that is object of future works.

3. Curriculum Learning

3.1. Rationale of the Approach

Curriculum Learning, introduced in [6], represents a strategy within machine learning initially devised with the aim of reducing the training times. The rationale is to present examples in a specific order of increasing difficulty during training, illustrating gradually more concepts and more complications to the decision. Later on, it has been employed more specifically as a *transfer learning* technique in RL and Multi–Agent RL [13]: the agent can exploit experiences acquired carrying out simpler tasks while training to solve more complex ones, in an *intra–agent* transfer learning scheme. In some situation it was also reported to support a better generalization of the overall training process [14]: achieving a good level of generalization of the acquired experience was also extremely important for our problem, since pedestrian simulation generally implies analysing the implications of different, alternative designs on the same crowding condition, without having to perform training for every specific design (which would lead to achieve incomparable results, since they would be achieved by means of different pedestrian models).

A naive application of a curriculum approach, however, initially led to issues somewhat resembling the so-called *vanishing gradient* problem [15]: technically here we do not have a recurrent neural network (or an extremely deep one like those employed for classification of images trained on huge annotated datasets) but, as we will show later on, the training is relatively long and the "oldest experiences" would be overridden by the more recent ones. The finally adopted approach, therefore, proceeds training agents in a set of scenarios of growing complexity, one at a time, but it also provides a final retraining in a selected number of earlier scenarios before the end of the overall training, to refresh previously acquired competences.

3.2. Details of the Curriculum

Starting from the above considerations, we defined a specific curriculum for RL-pedestrian agents based on this sequence of tasks of increasing complexity that are sub–goals of the overall training:

- Steer and walk towards a target;
- Steer to face target;
- Reach the target in narrow corridors;
- Walk through bends avoiding walking too close to walls;
- Avoid collisions with agents walking in the same direction;
- Avoid collisions with agents walking in conflicting directions;
- Combine all behaviours.

We defined this sequence thanks to expertise in the context of pedestrian simulation, as well as to a preliminary experimental phase (for instance the second step – steering to face a target – was introduced quite late, when we realized that, as a consequence of training in more geometrically complex scenarios, agents had sometimes difficulties in finding their targets when the environment was not essentially "guiding them"). It would be interesting to evaluate to which extent this sequence is robust, if it can be improved or if it is close to the optimum, but such an analysis was not performed at this stage of the research (we were interested in evaluating the adequacy of the approach and the possibility to achieve promising results on the domain of pedestrian simulation), and it is object of future works.

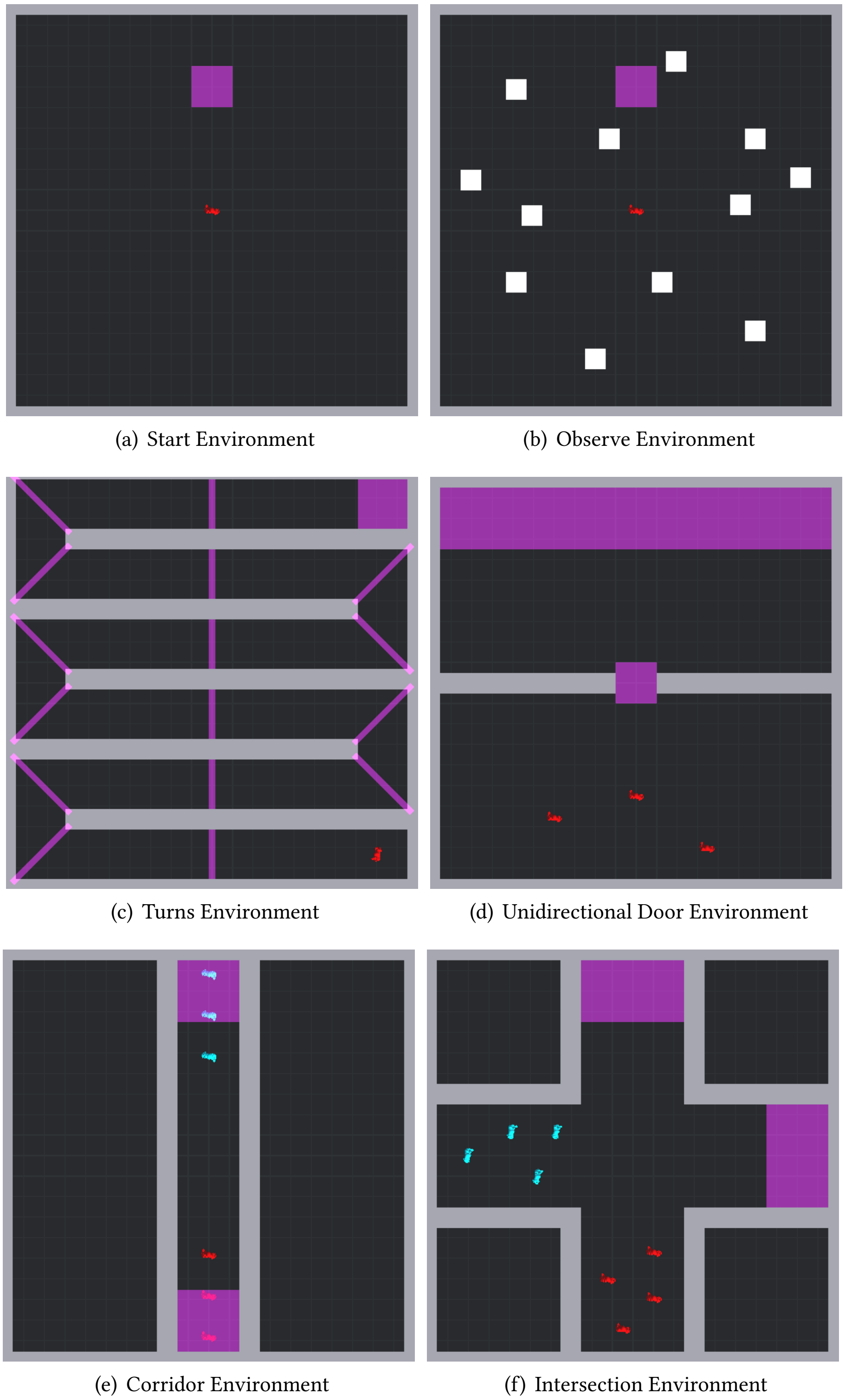
For sake of automation of the curriculum execution, we consider a step of the curriculum to be successfully completed whenever (i) a sufficiently high number of agents has been trained in the scenario and (ii) the average cumulative reward for trained agents, excluding the top and bottom 10% (for avoiding being excessively influenced by a small number outliers), exceeds a given threshold, specifically configured for every step of the curriculum.

We also included specific test scenarios, that is, environments that are not included in the training curriculum but that are used to evaluate the ability of agents to exhibit plausible behaviours in scenarios that were not experienced in the training phase, rather that just showing that they memorized the environments they had seen.

3.3. Training Environments

Table 2 reports the different environments that were defined for each of the sub–goals of the overall training. It also shows whose environment are included in the final retraining phase, that must be carried out before using the trained agents for simulation in new environments.

For sake of space, we cannot describe every environment and scenario included in the curriculum, but a selection of these training environments is shown in Figure 3. Several of these scenarios replicate experiments that were carried out with real pedestrians to study specific behaviours (see, e.g., [16] or [17]), although we currently did not investigate high–density



(a) Start Environment

(b) Observe Environment

(c) Turns Environment

(d) Unidirectional Door Environment

(e) Corridor Environment

(f) Intersection Environment

Figure 3: A selection of training Environments.

Table 2
Training Environments Curriculum.

Behaviour	**Environment**	**Retraining**
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	Turns	×
	Turns with Obstacles	✓
Avoid collisions with agents walking in the same direction	Unidirectional Door	✓
Avoid collisions with agents walking in conflicting directions	Corridor	✓
	Intersection	✓
	T Junction	✓
Combine all behaviours	Crowded Bidirectional Door	✓

situations, that moreover seem difficult to simulate with a tool such as Unity (which includes 3D models for pedestrians and components for the management of physics that should be overridden for managing significant levels of density - e.g. higher than 1 pedestrian per square metre).

We also do not have the space for commenting the training in all of these scenarios, however we can highlight some stylized facts we did observe:

- within the Corridor Environment agents learn to walk in lanes that, due to the low density, are quite stable;
- the Turns and Turns with Obstacles Environments produce plausible results in terms of trajectories, but this is mostly due to the placement of intermediate target helping agents in having smooth and plausible paths (as suggested in subsection 2.1);
- all the environments in which agents had to face narrow passages were crucial in leading them to accept the trade off between choosing some actions leading to an immediate negative reward (i.e. passing close to a wall) and achieving a longer term positive reward (i.e. reaching the final target);
- all the environments in which agents had to interact with others were analogously crucial but for helping them understand how to properly balance the need of slowing down and sometimes even waiting (when steering is simply not possible or not sufficient) to avoid collisions, but still reach the final target.

```
behaviors:
Pedestrian:
  trainer_type: ppo
  hyperparameters:
    batch_size: 512
    buffer_size: 5120
    learning_rate: 0.003
    beta: 0.01
    learning_rate_schedule: constant
  network_settings:
    hidden_units: 256
    num_layers: 2
  reward_signals:
    extrinsic:
      gamma: 0.99
      strength: 1.0
  max_steps: 100000000000000000
  time_horizon: 64
```

Listing 1: Training configuration file.

3.4. Training Configuration

Listing 1 reports the defined training configuration file[3]. The employed ML-Agents version we adopted is 0.25.1 for Python and 1.0.7 for Unity.

Once again, we were interested here in evaluating the adequacy of the approach, so we did not perform a systematic analysis of the effect of changing the different hyperparameters and this task will be object of future works. We just comment here some of the adopted choices:

- the neural network employed within the PPO algorithm is a fully connected network with 2 hidden layers of 256 nodes each; a bigger network leads to much longer training times but it does not improve the quality of the achieved results, whereas a smaller network does not converge to a reasonable policies;
- we employed a basic PPO without curiosity mechanisms [18], therefore we have essentially just extrinsic reward signals;
- we adopted a very high number for max_steps to let the curriculum guide the actual training, rather than predefined parameters. We also let time_horizon to the default value.

3.5. Reward Trend During Training

The preliminary tests we conducted before reaching this configuration for the curriculum, that were based on a single scenario or however that were based on curricula significantly

[3] Detailed descriptions of different fields are reported in https://github.com/Unity-Technologies/ml-agents/blob/release_16_docs/docs/Training-Configuration-File.md

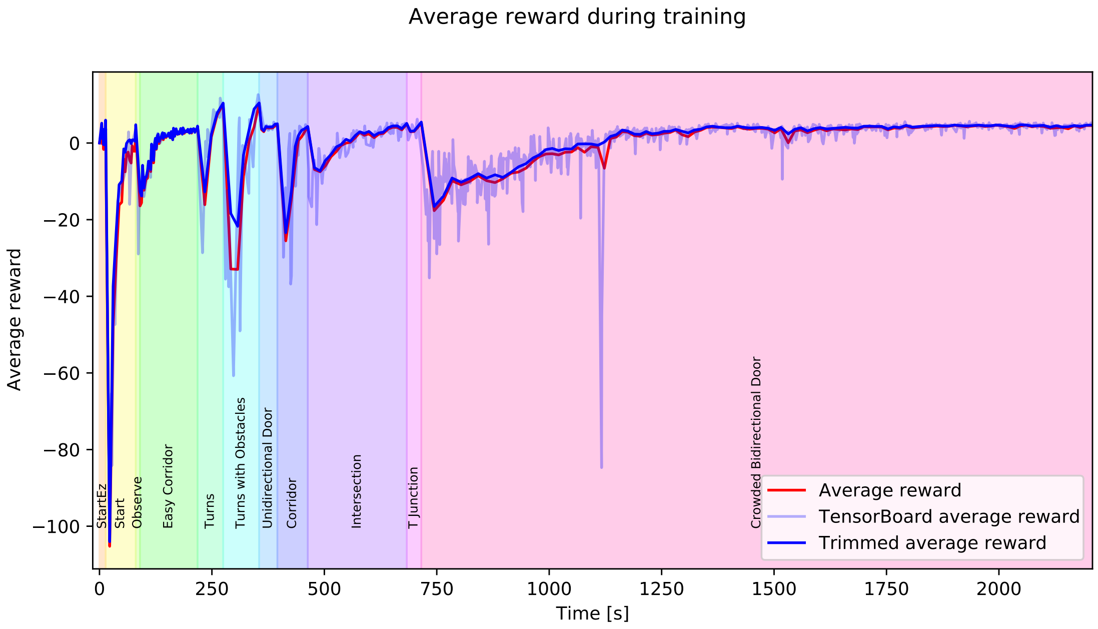


Figure 4: Average reward throughout training.

more compact than the one described above, were unsuccessful (at least they did not produce good results within the same time frame associated to training with this configuration for the curriculum).

The overall training time with the defined curriculum varies according to different factors, but on a Windows based PC employing an Intel Core i7-6820HL @ 2.70GHz, employing only the CPU[4] would require around 37 minutes to reach the final retraining phase (which is significantly shorter). Technically, agents have been trained in 9 equal environments at the same time, with a Unity velocity set to 100 (i.e. one second of simulation execution corresponds to 100 simulated seconds). The available hardware would not allow further compression of simulated time, but future developments in the ML-Agents framework could bring significant improvements (especially if they would fully exploit GPUs), but we could need to change the training phase workflow.

Figure 4 shows the trend of the cumulative reward. The Tensorboard average reward is the raw measure provided by Tensorboard, while the *Average reward* is computed averaging out the cumulative reward achieved by agents in 36 episodes within an environment. The *Trimmed average reward* actually removes respectively the 10% top and 10% bottom performing episodes.

The different colors highlight the duration of the different scenarios of the curriculum: as expected the reward drops (sometimes dramatically) when agents change the environment, but through time the training converges. It also clearly shows that environments in which agents have more significant interactions are tougher for the training algorithm. We were

[4] The adopted version of ML-Agents suggests doing so, since it would not properly exploit a GPU.

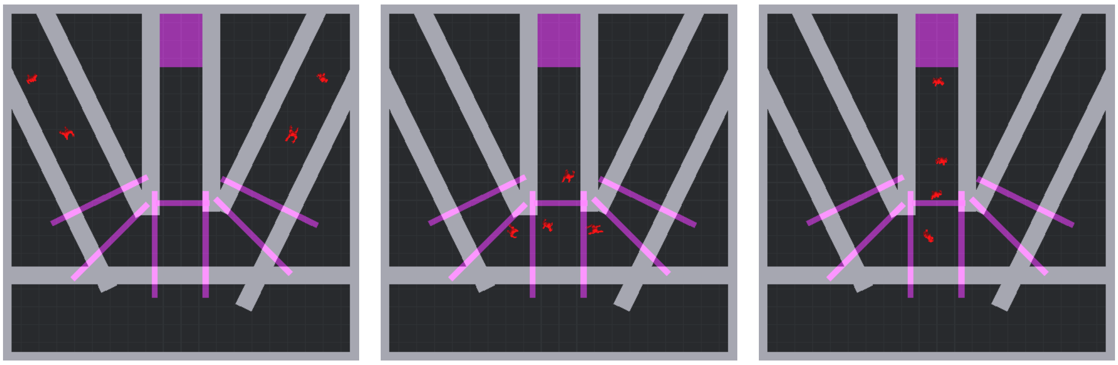


Figure 5: Anchor environment execution.

actually almost surprised by the fact that a vanilla PPO was able to successfully converge in such situations, that are much closer to situations that call for specific Multi-Agent RL algorithms. In these situations, the basic approaches often fail due the instability in the reward trend that depends on more factors outside the scope of control of the trained agent; specific reward functions that balance individual and aggregated level evaluation of the situation and new algorithms are typically employed. We also conducted an analogous experimentation considering groups of pedestrians, a situation that makes pedestrian to pedestrian interaction both more complex and much more frequent (essentially uniformly present in each step of the training) than the type described in the present work, and PPO was not able to converge. The description of this additional experimentation is out of the scope of the present work.

4. Analysis of Achieved Results

4.1. Qualitative Analysis of Generalization in Test Scenarios

After the training phase we tested the learned pedestrian model in some specific environments that were not "shown" to agents during the training, to understand if the approach was able to grant pedestrian a general capability to produce realistic behaviours even in newly encountered situations.

In particular, Figure 5 shows the *"Anchor"* environment, in which agents enter from the NE and NW corners, make a sharp bend and move North (a movement patter with a junction between two flows that is not that different from the T Junction environment): agents do not have particular problems, although they might have an hesitation close to the point in which the flows merge, due to the interaction and coordination process that must take place between pedestrians coming from the two entrances (something that is also plausible and that can be qualitatively observed in real world experiments).

Figure 6 shows the *"Omega"* environment, a maze–like structure in which 90° and u-turns to the right and to the left are present without choices among different passages. We emphasize that the training environments do not include all of these configurations for bends. Trained agents exhibit a reasonable behaviour, slowing down before the bends to avoid collisions with

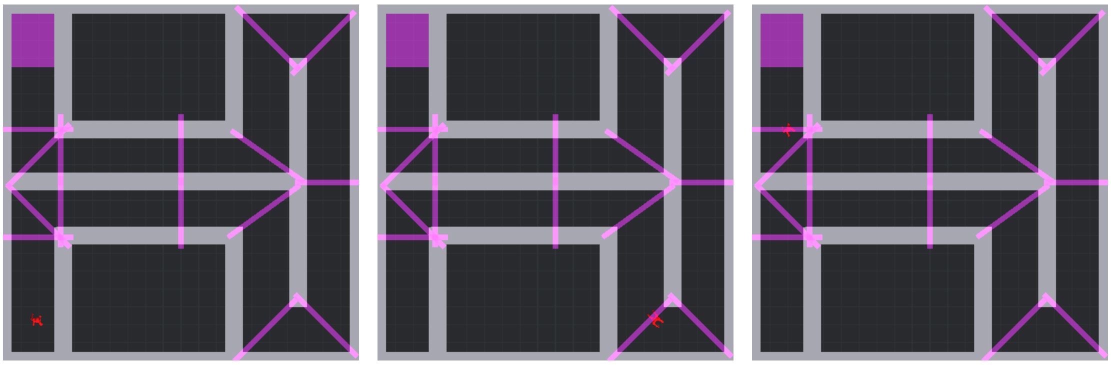


Figure 6: Omega environment execution.

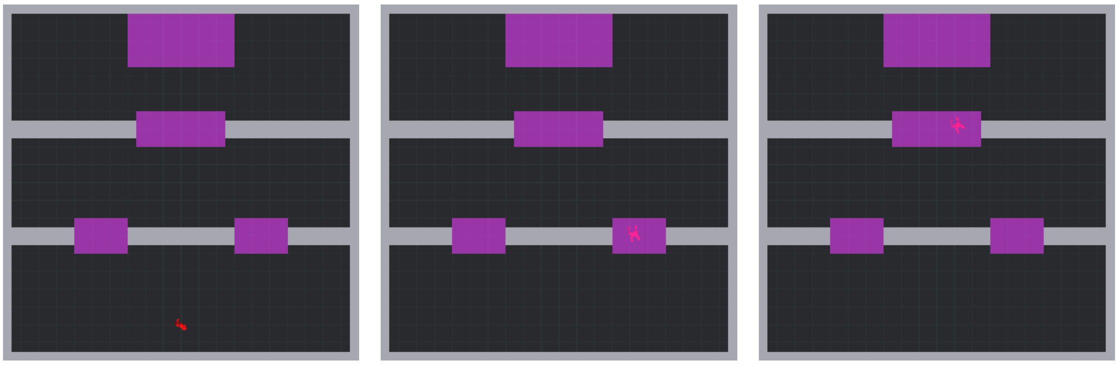


Figure 7: “Door choice” environment execution.

walls.

Figure 7 shows the *“Door choice”* environment, a relatively simple situation that however includes the choice of a passage from the Southern to the central region, in addition to a single passage to the Northern region that includes the final target. Within the training environments agents never face a situation in which they have to choose among two or more intermediate targets, and we wanted to find out if this kind of situation would instead be necessarily included in a proper curriculum for training pedestrian dynamics.

Trained agents actually do not have a problem in performing a plausible movement pattern in this scenario: they do not always choose the closest passage, but (i) real world experiments show that real pedestrians are not necessarily optimizing the expected travel time (although this generally happens when additional factors to distance, such as congestion, influence their decisions), and (ii) additional modifications to the model and to the training curriculum would be necessary to improve wayfinding behaviour to be competitive with hand–written and calibrated models.

Figure 8 finally shows the *“Bidirectional Door”* environment, a variant of the “Crowded Bidirectional Door” employed in the training. The lower number of pedestrians from the region beyond the passage, and their random initial position, paradoxically can represent a problem

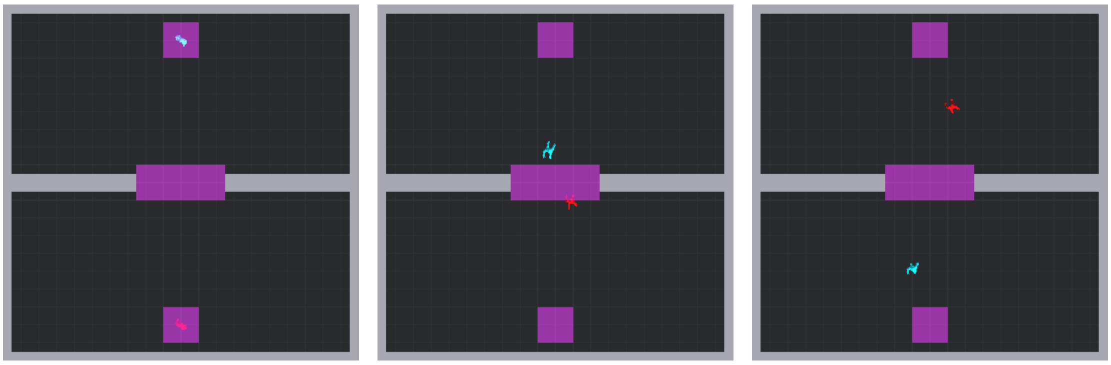


Figure 8: “Bidirectional Door” environment execution.

for the agents, since they cannot perceive the potential conflict until the very last moments. This scenario was therefore aimed at finding out if the trained agents were able to move at free–flow speed and then slow down when they perceive a conflicting pedestrian, avoiding it and, at the same time, don not completely disrupt the trajectory.

When agents had initial positions granting them immediate mutual perception they would start moving cautiously, and they cross the door keeping their right, then move to the final target. Otherwise, agents start moving at full velocity until they perceive each other, slowing down, and again change position to avoid each other when passing through the door, generally keeping their right. Sometimes agents do not follow the most direct path to the final target after passing through the door, but the overall behaviour is acceptable.

We did not test if the side preference is random and due to the randomness in the training process, or if there is some systematic bias (maybe due to the spatial structure of the training environments) that leads to an uneven distribution of this preference.

5. Conclusions and Future Developments

The paper has presented a research effort aimed at experimenting the adequacy of applying RL techniques to pedestrian simulation, especially considering the need to achieve general models applicable to a wide range of situations without the need of performing a training for each analyzed scenario. The achieved results are promising and encouraging. There are, of course, several limits of the current state of the research, representing lines for future research:

- we did not show a quantitative analysis of the achieved results, also for sake of space: this analysis, representing a first step in the direction of model validation, is object of current and future works;
- we intend to release as an open source project the developed software and the environments used for the curriculum and for the tests; at the moment of submission of the camera ready the repository is not ready, especially due to the lack of proper documentation, but anyone interested in accessing it can contact the authors (and plausibly future works on this line of research will include a reference to the accessible software repository);

- analysis of the effects of changes in RL algorithm, hyperparameters, configuration of the curriculum: we reached the presented solution performing some comparisons with alternative settings, but a systematic analysis of each of these aspect would require a focused specific work;
- additional quantitative experiments to improve the evaluation of the achieved results on the side of pedestrian simulation, towards a validation of the model or the acquisition of new objectives for model improvement;
- overcoming some current limits: modeling groups within the simulated pedestrian population is not possible, and preliminary work in this direction suggests that a change in the adopted RL algorithm would be necessary, due to the more systematic presence of agent to agent interaction; dealing with high density situations; going deeper in the capability of the model to perform wayfinding, possibly achieving the capability to adapt to the perceived level of congestion [19].

References

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

[2] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (4th ed.), Pearson, 2020.

[3] S. Bandini, S. Manzoni, G. Vizzari, Agent based modeling and simulation: An informatics perspective, Journal of Artificial Societies and Social Simulation 12 (2009) 4.

[4] 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.

[5] N. Habbash, F. Bottoni, G. Vizzari, Reinforcement learning for autonomous agents exploring environments: an experimental framework and preliminary results, in: R. Calegari, G. Ciatto, E. Denti, A. Omicini, G. Sartor (Eds.), Proceedings of the Workshop on 21st Workshop "From Objects to Agents", Bologna, Italy, September 14-16, 2020, volume 2706 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 84–100. URL: http://ceur-ws.org/Vol-2706/paper5.pdf.

[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] L. Crociani, G. Vizzari, S. Bandini, Modeling environmental operative elements in agent-based pedestrian simulation, Collective Dynamics 5 (2020) 508–511. URL: https://collective-dynamics.eu/index.php/cod/article/view/A85. doi:10.17815/CD.2020.85.

[8] L. Crociani, K. Shimura, G. Vizzari, S. Bandini, Simulating pedestrian dynamics in corners and bends: A floor field approach, in: G. Mauri, S. El Yacoubi, A. Dennunzio, K. Nishinari, L. Manzoni (Eds.), Cellular Automata, Springer International Publishing, Cham, 2018, pp. 460–469.

[9] S. Paris, S. Donikian, Activity-driven populace: A cognitive approach to crowd simulation, IEEE Computer Graphics and Applications 29 (2009) 34–43. doi:10.1109/MCG.2009.58.

[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] E. T. Hall, The hidden dimension, Doubleday New York Ed., 1966.

[12] 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.

[13] 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.

[14] 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.

[15] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 06 (1998) 107–116. URL: https://doi.org/10.1142/S0218488598000094. doi:10.1142/S0218488598000094. arXiv:https://doi.org/10.1142/S0218488598000094.

[16] J. Zhang, A. Seyfried, Comparison of intersecting pedestrian flows based on experiments, Physica A: Statistical Mechanics and its Applications 405 (2014) 316–325. URL: https://www.sciencedirect.com/science/article/pii/S0378437114001988. doi:https://doi.org/10.1016/j.physa.2014.03.004.

[17] J. Zhang, W. Klingsch, A. Schadschneider, A. Seyfried, Transitions in pedestrian fundamental diagrams of straight corridors and t-junctions, Journal of Statistical Mechanics: Theory and Experiment 2011 (2011) P06004. URL: https://doi.org/10.1088/1742-5468/2011/06/p06004. doi:10.1088/1742-5468/2011/06/p06004.

[18] D. Pathak, P. Agrawal, A. A. Efros, T. Darrell, Curiosity-driven exploration by self-supervised prediction, in: D. Precup, Y. W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 2778–2787. URL: http://proceedings.mlr.press/v70/pathak17a.html.

[19] G. Vizzari, L. Crociani, S. Bandini, An agent-based model for plausible wayfinding in pedestrian simulation, Engineering Applications of Artificial Intelligence 87 (2020) 103241. URL: https://www.sciencedirect.com/science/article/pii/S0952197619302246. doi:https://doi.org/10.1016/j.engappai.2019.103241.