CCP: Configurable Crowd Profiles

Andreas Panayiotou
CYENS - Centre of Excellence
Nicosia, Cyprus
a.panayiotou@cyens.org.cy

Theodoros Kyriakou
CYENS - Centre of Excellence
Nicosia, Cyprus
t.kyriakou@cyens.org.cy

Marilena Lemonari
University of Cyprus
Nicosia, Cyprus
lemonari.marilena@ucy.ac.cy

Yiorgos Chrysanthou
CYENS - Centre of Excellence,
University of Cyprus
Nicosia, Cyprus
y.chrysanthou@cyens.org.cy

Panayiotis Charalambous
CYENS - Centre of Excellence
Nicosia, Cyprus
p.charalambous@cyens.org.cy

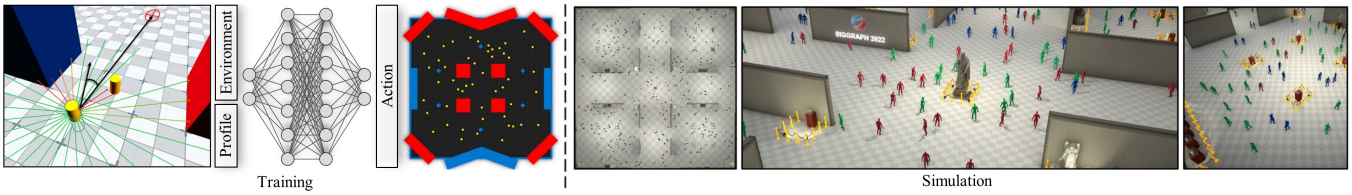

Figure 1: We train a single policy for configurable agents which we can then use to simulate crowds with diverse behaviors in complex environments.

ABSTRACT

Diversity among agents' behaviors and heterogeneity in virtual crowds in general, is an important aspect of crowd simulation as it is crucial to the perceived realism and plausibility of the resulting simulations. Heterogeneous crowds constitute the pillar in creating numerous real-life scenarios such as museum exhibitions, which require variety in agent behaviors, from basic collision avoidance to more complex interactions both among agents and with environmental features. Most of the existing systems optimize for specific behaviors such as goal seeking, and neglect to take into account other behaviors and how these interact together to form diverse *agent profiles*. In this paper, we present a RL-based framework for learning multiple agent behaviors concurrently. We optimize the agent policy by varying the importance of the selected behaviors (goal seeking, collision avoidance, interaction with environment, and grouping) while training; essentially we have a reward function that changes dynamically during training. The importance of each separate sub-behavior is added as input to the policy, resulting in the development of a single model capable of capturing as well as enabling dynamic run-time manipulation of agent profiles; thus allowing *configurable profiles*. Through a series of experiments, we verify that our system provides users with the ability to design virtual scenes; control and mix agent behaviors thus creating personality profiles, and assign different profiles to groups of agents. Moreover, we demonstrate that interestingly the proposed model generalizes to situations not seen in the training data such as a) crowds with higher density, b) behavior weights that are outside the training intervals and c) to scenes with more intricate environment layouts. Code, data and trained policies for this paper are at https://github.com/veupnea/CCP.

CCS CONCEPTS

• **Computing methodologies** → **Sequential decision making**; **Real-time simulation**; **Reinforcement learning**; **Animation**.

KEYWORDS

crowd simulation, data-driven methods, user control, crowd authoring, reinforcement learning.

ACM Reference Format:
Andreas Panayiotou, Theodoros Kyriakou, Marilena Lemonari, Yiorgos Chrysanthou, and Panayiotis Charalambous. 2022. CCP: Configurable Crowd Profiles. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings), August 7–11, 2022, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3528233.3530712

SIGGRAPH '22 Conference Proceedings, August 7–11, 2022, Vancouver, BC, Canada

ACM ISBN 978-1-4503-9337-9/22/08…$15.00
https://doi.org/10.1145/3528233.3530712

1 INTRODUCTION

Human crowds are a fundamental feature of most real-world environments. Depending on the environment, we can observe different behaviors; in a street for example, people move around (often-times) in small groups to reach their goals, whereas in a market they wander about and group around certain kiosks to buy goods. Being able to easily and efficiently simulate and author all of these context-specific behaviors is important in several domains (entertainment,

urban studies, architecture, etc.). Several crowd simulation techniques have been developed; most of them requiring artists and users to set several parameters to achieve the desired outcome. This is not an easy task, especially by non-experts.

Recently, several data-driven methods have been explored for learning behavior models in the context of crowd simulation, with the majority of them being Supervised Learning (SL) in nature. Several researchers introduced Reinforcement Learning (RL) as an attractive alternative to these techniques; the premise here is to learn via trial-and-error by optimizing for simple scalar reward signals. The data in this case is being simulated instead of given a-priori, with occasional scalar values indicating rewards or penalties for certain events such as collisions with others or reaching goals. Reward signals are both a blessing and a curse; easy to understand (e.g., +1 for reaching goal) but also notoriously difficult to balance especially when having multiple behaviors at the same time. In a typical RL setting, the rewards are defined a-priori and are fixed during training, keeping the behavior of the agent fixed both during and after training. This however introduces challenges; the relative values of the different rewards and their frequency affects the learned behavior, and often defining proper values to achieve certain behaviors is not straightforward. Consider for example the simple case of optimizing for collision avoidance and goal seeking. Setting a high penalty for collisions, might lead to hesitant agents never reaching their goals or moving in unnatural patterns, whereas a very high reward for seeking a goal might lead to agents overlapping or running over others, not caring for collisions. Researchers typically work with intuition; they try different values and keep those that yield desirable results. We note that simply selecting the model that achieves higher reward does not make much sense as each different training session of RL optimizes for different ranges of values. This is cumbersome; training takes time and thus manually selecting the best set of values is quite challenging.

Inspired by recent work on configurable game playing agents [Le Pelletier de Woillemont et al. 2021] we propose a novel RL-based framework to concurrently learn multiple diverse crowd behaviors by varying agent profiles during training. Profiles are intuitively described by a set of weights for different reward signals (i.e., goal seeking, collision avoidance, group and interactions with the environment); different mixtures of these weights essentially describe different profiles (e.g., more grouping-oriented agents are deemed more sociable). These profile values are added as control signals to the input of the policy network; therefore, learning multiple policies at the same time during training. We propose a curriculum-based approach to train crowds by modifying both the complexity of the environment and the combination of weights as training progresses. Having trained agents possessing diverse profiles concurrently, we can then create large-scale crowd simulations capable of exhibiting diverse behaviors (profiles) that are mixtures of the basic behaviors; these can easily be modified at run-time by users. An additionally important aspect of the proposed framework is its generalization properties. It suffers less from the amount of data, compared to SL approaches, and it can easily cope with higher agent density and deviating environments. Moreover, traditional crowd simulation systems are reactive in nature and weigh behaviors in the action space (i.e., weighted sum of actions returned by different behavior modules) without considering long-term consequences and interactions between the different behaviors. In the proposed approach, behaviors are defined by the dynamically changing weights of the reward signals; agents therefore learn policies that consider both long-term effects of actions and interactions between the behaviors. Ultimately, our proposed framework simulates various behaviors found in human crowds such as social interactions, collision avoidance and interactions with areas of interest (e.g., museum exhibits), creating natural-moving agents instead of optimizing individual aspects, and allows for intuitive control over their mixture.

2 RELATED WORK

Simulating crowds and crowd behaviors has been extensively explored throughout the years, with various techniques developed to abide by a diverse range of scenarios and objectives [Pelechano et al. 2016; van Toll and Pettré 2021]. Microscopic approaches have extensively been used to achieve behavior diversity; macroscopic approaches model the crowd as a whole and therefore diversity is not a priority. Data-driven and Machine Learning approaches have gained increasing popularity in recent years due to promising results. We focus our description of related work on a) heterogeneous crowds with attention to data-driven methods, b) crowd authoring and c) RL, which are the most relevant techniques to our work.

Heterogeneity in Crowds. Generating behavioral diversity in virtual crowds has been addressed in the literature with a variety of approaches; early works use predefined weights for different behaviors to achieve such heterogeneity [Reynolds 1999; Shao and Terzopoulos 2005]. Recently, Ren et al. [2016] incorporated diverse group properties in groups of agents controllable via user constraints. More similar to our concept, several researchers [Durupinar et al. 2011; Guy et al. 2011; Kim et al. 2012] focused on behavior diversity in terms of personality traits; these expose some parameters to users to allow for control of agent behaviors. Even though there is previous research considering profiles in the context of crowd simulation, the scope of our study concerns the degree to which data can be used to represent such behavior diversity.

In data-driven methods, the crowd simulation model is implicitly defined by example data. Early data-driven approaches were graph-based [Kwon et al. 2008; Lai et al. 2005], similar to methods in the Character Animation literature [Kovar et al. 2002]. These methods however remain limited to group navigation (e.g., flocking) and fail to reflect the variations of behaviors found in human crowds. The need to simulate behavior diversity found in real-life crowds, led to more sophisticated and practical approaches. One family of such approaches creates databases of examples; agents simply try to match these during simulation time [Lee et al. 2007; Lerner et al. 2007]. Some extensions to these methods added secondary actions for increased realism [Lerner et al. 2010; Zhao et al. 2017]. Early works inspired by this reasoning, like Metoyer and Hodgins [2003] allowed the user to define specific examples of behaviors. Morphable crowds by Ju et al. [2010] focused on blending the different crowd styles represented by the input data. Charalambous and Chrysanthou [2014] used Temporal Perception Patterns to represent state and introduced PAG (Perception-Action-Graph) to improve simulation quality and performance of data-driven methods. Despite the significant improvements achieved by these methods, the results

highly depend on the variability and amount of input data, therefore accumulating errors over time and failing to consider long-term effects of agents' actions. Hence, unwanted or unnatural behaviors are observed, reducing the generalizability of these systems.

Other methods use real-world data to find optimal parameters for given systems. There is extensive research tackling simulation aspects like collision avoidance and steering, primarily estimating parameters corresponding to anticipation and time-to-collision [Paris et al. 2007; Pettré et al. 2009; van Basten et al. 2009]. These data-driven techniques are not exempt from the limitations of the underlying behavior model, despite their success in refining them. Several researchers use reference crowd data to analyze simulated crowds. Such analysis often includes comparisons of simulated results to the corresponding results derived from the data [Charalambous et al. 2014; Guy et al. 2012; He et al. 2020; Kapadia et al. 2011; Karamouzas et al. 2018; Wang et al. 2017]. Wolinski et al. [2014] find optimal sets of parameters subject to certain metrics and reference crowd data, enabling fair comparison between crowd simulators. He et al. [2020] additionally use data to guide simulations in environments that are similar to the ones in the input data.

Crowd Authoring. Controlling crowd simulations at a higher level is crucial to several scenarios since it allows users to easily and efficiently author agents' behaviors according to their wishes. This requires intuitive tools which are highly dependent on what simulation aspect the users aim to control [Lemonari et al. 2022]. For the *path-planning* aspect, past literature heavily uses sketching interfaces [Ju et al. 2010; Metoyer and Hodgins 2003]. In contrast, authoring the *animation and visualization* aspects usually entails asset and template manipulation [Maïm et al. 2009; Ulicny et al. 2004]. Editing has also been widely explored in literature, with the more user-friendly systems incorporating manipulation handles. For example, Kwon et al. [2008], propose deformation gestures as a post-processing tool for group motion editing. One of the most challenging aspects in authoring is controlling high-level behaviors such as describing agendas, desires and even personalities; these behaviors benefit the expressiveness and realism of simulations [Kraayenbrink et al. 2012]. Authoring local movements can be a consequence of controlling high-level behaviors or controlling certain local movement aspects; mainly involving coding low-level parameters such as time horizon for the avoidance strategy [Karamouzas et al. 2017].

In our work we learn a configurable crowd model that allows us to mix and learn several behaviors at the same time; to our knowledge this was never addressed in the crowd simulation literature before. Previous works weigh the output of different reactive behavior models (i.e., in the action space) [Reynolds 1999; Shao and Terzopoulos 2005], whereas in the proposed approach we use control signals as inputs to a non-linear learned policy. At simulation time we can easily modify and mix the behavior of agents in a crowd by modifying simple sliders. The effect of the change in such parameter values is immediate, logical and intuitive, allowing naïve users to define crowd profiles as they intend to.

Reinforcement Learning (RL). With the rise of Deep Learning and its wide range of applications, researchers naturally explored Deep RL for crowd simulation. In particular, RL has proven to be a useful tool for learning optimal strategies in sequential decision making problems [Sutton and Barto 2018; Szepesvári 2010]. The introduction of Deep Q-Learning (DQL) [Mnih et al. 2015] and other Deep RL approaches such as Proximal Policy Optimization (PPO) [Schulman et al. 2017] inspired further research on how these techniques could be used in the context of crowd simulation. Treuille et al. [2007] demonstrated the potential of RL in the domain of crowd simulation by producing characters navigating environments while performing collision avoidance with moving obstacles; Peng et al. [2017] expanded on this idea to physically-based characters. Several researchers developed crowd simulation policies via the development of effective RL strategies [Godoy et al. 2015; Henry et al. 2010; Lee et al. 2018; Martinez-Gil et al. 2011].

However, most of the past work involving Deep RL in the crowd simulation literature focused on investigating the effect of different reward functions. These methods make simplifying assumptions or have simple *manually-defined reward functions* since they attempt to capture specific aspects of crowds such as *collision avoidance* and *goal reaching* [Lee et al. 2018; Long et al. 2017; Martinez-Gil et al. 2017; Sun et al. 2019]. Most of the time, researchers define a set of simple reward signals and hand-tune them until they get the desirable result; this is time consuming (training is typically slow) and inefficient since it requires trial-and-error of different values. Moreover, adding new behaviors in the mix requires readjustment of the different reward signals to balance between the different behaviors and therefore retraining. In the end researchers find a set of values that works well for specific scenarios and stick with them.

This leads to virtual crowds with *uniform behavior* that do not have the variability of actual crowds, and with *limited control* over behavior parameters after training. Recently, several researchers started exploring policy parameterization techniques to successfully learn more generalizable and complex tasks such as learning a broad family of motor skills from limited amounts of reference data [Lee et al. 2021] or adapting physics-based motion on characters with different body shapes [Won and Lee 2019]. Recently, Hu et al. [2021] proposed a parametric RL-based method for heterogeneous collision avoidance behaviors in crowds; agent parameters such as preferred velocities are varied during training and are input to the control policy. All of these methods add a set of control signals to the input of the policy and have a predefined reward function with constant weighting between the different learned subtasks (goal reaching, collision avoidance, walk forward, etc).

In this work, we aim to capture multiple behaviors concurrently by selecting and mixing a subset of different basic ones; we selected collision avoidance, goal seeking, grouping and interaction with objects of interest (more behaviors can easily be added if needed). We allow for the *variation of the importance (weight)* of each of these behaviors during training time; in essence we optimize a policy with a multitude of reward functions. Therefore, we learn a *single crowd model* that captures many different agent behaviors; an agent's behavior can simply and efficiently be manipulated at runtime by adjusting these weights. We got inspiration from the work by de Woillemont et al. [2021] where multiple player strategies are being trained simultaneously in a simple two player game, achieving large variety among their behaviors, while preserving the respective performance. Despite the conceptual similarities, we work in a challenging domain that includes many simulated

characters with different and diverse profiles at the same time. Managing to allow agents to balance each sub-behavior differently (i.e., having different profiles) is thus a focal point of our research.

3 THE CCP FRAMEWORK

We divide our system in two distinct phases: training, and run-time simulation (Fig. 1). During the *training* phase: a) multiple agents are concurrently exposed to varying environment conditions such as the environmental setup, the types of buildings/props, their placement, and distribution of other agents, and b) the agents' profile is varied dynamically by adjusting the relative importance of different reward signals. During *run-time simulation*, agents in a crowd are assigned goal positions and profiles which can be changed dynamically.

For training we use Proximal Policy Optimization (PPO) which is an on-policy RL technique [Schulman et al. 2017]; agents aim to optimize their behaviors to maximize the expected cumulative sum of the dictated reward signals. Instead of manually defining different rewards, we *vary the weight of different reward signals during the training phase* (inspired by [Le Pelletier de Woillemont et al. 2021]). The intuition behind this is to *train agents that are capable of doing multiple, diverse behaviors and allow for the dynamic adaptation of these behaviors*. For this to work, we concatenate these reward weights to the input of the policy network (i.e., as part of the agents' observations); these weights essentially describe the profile of an agent and affect how that agent acts in the environment. Alternatively we could train multiple policies and then combine them by e.g., a blending of the output actions; this is not always possible. Following such an approach entails exposing agents to a limited set of behaviors during training. In most cases, the behaviors that all agents have, are uniform. Consider mixing a collision avoidance behavior with a grouping behavior; agents were never exposed to a mix between the two. This is crucial; mixing policies will result in unnatural results. We model agents as circles that move on a 2D plane; contrary to most crowd models, our agents have different look and move directions to allow for more diverse actions such as sidestepping and looking towards objects of interest or group members (Fig. 2). Agents work in a decentralized manner, each having partial knowledge of the environment state. We use discrete actions and move the agents by discretely changing their velocity keeping it under the maximum speed.

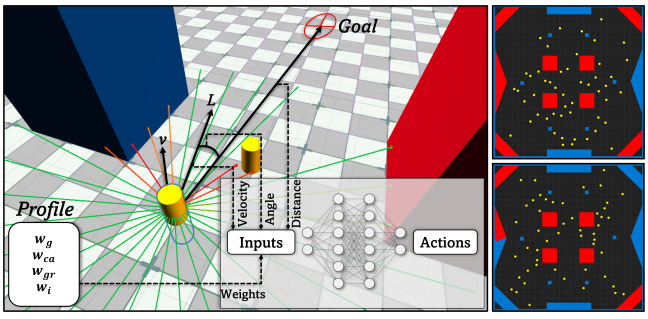

Figure 2: A character from a crowd and its observations.

Table 1: Rewards per event and corresponding weights.

Weight	Range	Behavior
w_g	[.1, 1.8]	Goal Seeking
w_{ca}	[.5, 3.5]	Collision Avoidance
w_{gr}	[−3, 5]	Grouping
w_i	[−5, 5]	Interaction

Event	Symbol	Base Reward	Weight	Dense
Reached Goal	R_g	+1.0	w_g	N
Agent Collision	R_{ca}	-.01	w_{ca}	N
Obstacle Collision	R_{co}	-.5	w_{ca}	N
Towards Goal	R_{gt}	+.00075	w_g	Y
Away from Goal	R_{ga}	-.00025	w_g	Y
In Group	R_{gr}	+.001	w_{gr}	Y
Interacting	R_i	+.001	w_i	Y
Living Penalty	R_l	-.00015	w_g	Y

3.1 Rewards

RL algorithms optimize an agent's decisions in order to maximize expected cumulative reward over time [Sutton and Barto 1998]. The reward function is very important since it defines the task the RL algorithm is optimizing for; proper selection and balancing of different reward signals is therefore crucial to achieve desirable results. Defining rewards for crowd simulation is not trivial; people in the real world balance between several different sub-objectives such as moving towards goals, avoiding collisions, moving with others or stopping to observe a street performance. Moreover, people have different profiles (aggressive, procrastinators, etc.) and moods. Using a traditional RL algorithm would require training a different model for each of these profiles; this is problematic. Instead of hard-coded weights for the different reward signals, we decided to train a model for different combinations of primitive subtasks: a) goal seeking, b) collision avoidance, c) grouping and d) interaction with points of interest (POIs). For each of these four categories we have different ranges of weights (Table 1); these were selected by experience and experimentation. A set of values $\{w_g, w_{ca}, w_{gr}, w_i\}$ for each of these weights define a *profile*; an agent having $w_g = 1$ for instance is more goal-oriented than an agent that has $w_g = .2$. During training, these values are randomized and are part of an agent's observations (Section 3.2). We define both *sparse* and *dense* reward signals with corresponding configurable weights (Table 1); the total reward R^t at any given simulation step t is:

$$R^t = w_g(R_g + R_{gt} + R_{ga} + R_l) + w_{ca}(R_{ca} + R_{co}) + w_{gr}R_{gr} + w_iR_i \quad (1)$$

Sparse reward signals include a large positive reward R_g when agents reach their goal, and penalties when agents collide with other agents or obstacles (R_{ca}, R_{co}); colliding with obstacles has a larger penalty. Dense rewards are given at every step; we positively reward agents when a) they move towards their goal (R_{gt}), b) are in groups (R_{gr}) or c) interact with POIs (R_i), whereas we penalize them when they do not move towards goals (R_{ga}). We additionally have a living penalty R_l to motivate agents to move based on their desire to reach a goal. An agent is moving towards its goal if the distance to the goal decreased from the previous step and if the look direction is towards the goal (we set $|\theta| \leq 45^o$) . An agent is part of a group if a) the number of nearest neighbors N is less than a predefined maximum number N_T and b) the angle to the center

Table 2: Default values for hyperparameters

Parameter	Value	Description
r	$.5m$	Agent Radius
R_s	$7m$	Maximum search distance
T	$.04s$	Simulation step
Learning Rate	3e-4	For Gradient Descent Updates
γ	.99	Discount factor
H	15000	Maximum steps per episode
Epochs	3	Training Epochs
Batch Size	1024	Batch Size
Buffer Size	10240	Buffer Size
β	5e-3	Entropy Regularization Strength
ϵ	.2	Divergence Threshold

of mass of the neighbors is small. N_T is a parameter that models discomfort of agents being in dense crowds; we set $N_T \in [3, 5]$ in our experiments; this parameter could potentially be added in the profile parameters. Similarly, an agent is interacting with a POI if it is looking towards the center of that object and has $N \leq N_T$ agents around it to avoid overcrowding. We note that we set $w_{ca} > 0$ and $w_g > 0$ so that all agents have some sense of goal seeking and collision avoidance. Large values of w_{ca} and w_g indicate how aggressively agents avoid collisions and pursue goals. We note that for easier relative comparison between behaviors and for the demos we use *normalized weights* $w \in [0, 1]$; e.g., a value 0 for w_i, corresponds to -5. During training, Equation 1 uses the weight ranges shown in Table 1.

3.2 Observations

Agents make decentralized decisions by collecting partial observations of the environment state (Fig. 2); these contain relevant and sufficient information for successful learning. The local coordinate system of an agent is aligned with its look direction. Observations include a) goal oriented features, b) distinct sets of rays that measure distances to other agents and POIs (obstacles, interaction areas), and c) profile parameters. More precisely, we collect: a) *Local velocity* $\mathbf{v}$ relative to the look direction, b) *Relative goal position* described by the tuple (ρ, θ) ($\rho \in [0, 1]$ is the distance to the goal normalised by a maximum distance and $\theta \in [0, 2\pi]$ is the angle between the agent's look direction and the vector towards the goal position), c) *profile parameters* are the *normalized* weights $\{w_g, w_{ca}, w_{gr}, w_i\}$ for the different reward signals and d) three sets of *distance sensors* for agents, obstacles and POIs (we cast $3 * 30$ rays uniformly around the agent up to a maximum distance of $7m$).

3.3 Actions

Agents can take one out of seven available actions: Stand Still (SS), Move Forwards and Backwards (MF, MB), Rotate Left and Right (RL, RR) and Move Left and Right (ML, MR). We set a maximum forward moving (MF) speed of $1.3m/s$; for all other moving directions (MB, ML and MR) this is set to $.13m/s$ since it is more natural for people to move in a forward direction.

3.4 Training Strategy

We use a simple Fully Connected Neural Network to model the policy with two hidden layers each having 128 nodes. It takes as input the agents' observations (Section 3.2) and outputs the action to be taken by the agent. We use the PPO algorithm to train the network with the parameters shown in Table 2.

We use a curriculum-based approach to train agents to behave as part of diverse crowds; instead of starting immediately to train agents to handle every possible situation, we gradually increase the difficulty of the environment and the behaviors the agents have to face. First, we set up a training environment similar to the one in Fig. 2 (right) with obstacles (blue boxes), POIs (red boxes) and source/destination areas (green boxes). During training, we randomly initialize agents (yellow circles) in the specified areas and set random destination points inside another such area. Each agent is trained individually; when a single agent reaches its goal, collides with an obstacle (not other agents) or a maximum number of steps is reached its episode finishes, a new agent is initialized in the same environment, and the other agents simply continue their training. To gradually introduce more difficult tasks to the agents, we a) start in a simple environment with a small number of agents, POIs and obstacles and gradually introduce more complex environments with more agents, POIs and obstacles, b) randomize weights of only one of the behaviors before we start combining them, c) randomize weights near their minimum and maximum values before we cover the entire spectrum, and d) randomize POIs and obstacles. To avoid rapidly changing policies during training, the weights are kept constant for several training episodes instead of randomizing them every step or episode.

4 EXPERIMENTS AND EVALUATION

We trained our models on a single thread of a PC having an i5-9600K CPU, an NVidia RTX 2070 GPU and 16GBs of RAM using PPO with the parameters shown in Table 2; we use the PPO implementation found in the ml-agents framework by Unity [Juliani et al. 2018]. Model training took four days. For most experiments we use a simplified cylindrical representation of agents that have separate look and move directions; movement is on a 2D plane. For visualisations with humanoid characters we use Motion Matching [Clavet 2016; Unity 2021] which allows for skeletal animation from unstructured motion capture data. For visual interpretation of results, we color code characters based on profile. To quantitatively compare the simulations in the following paragraphs, we measure the following statistics for all agents: a) speed, b) density, c) distance to the closest neighbor (DCN), and d) distance to the closest point of interest (DPOI). Please refer to the supplemented video for animated results.

4.1 Environment setups

We use four simulation environments for this study; hall, museum, crossing and circular (Fig. 3). The first 3 are associated with the experiments for behavior sensitivity (Section 4.2) and generalization (Section 4.5), while the fourth was developed to demonstrate the trained model on a large scene with multiple diverse behaviors (Section 4.6). The hall and crossing environments consist of agents walking in 2 and 4 opposing directions, respectively. In the circular setup, the agents start from the perimeter of a circle and move towards opposing points. To test interaction and avoidance behaviors, some obstacles and other POIs were added. The museum scene consists of 400 agents moving in a museum-like environment

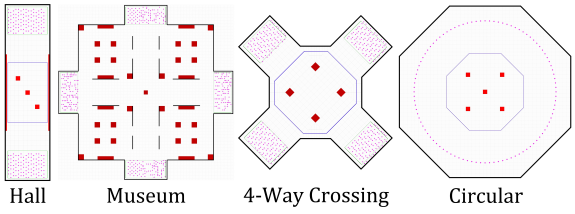

Figure 3: Simulation Environments.

that has several artefacts; we demonstrate results with different distributions of behaviors amongst the agents. We note that in Fig. 3 the red boxes indicate POIs and the blue regions indicate where the profile values are set – we consider statistics only in these regions.

4.2 Weight Sensitivity on Crowd Behavior

The first set of experiments concerns the effect of each of the weights on the agents' behavior; goal seeking, collision avoidance, grouping, and environment interactions (Fig. 4). To assess the impact of each behavior on the outcome of the simulation, we gradually increase each behavior's weight in the 3 different environments – the weights apply to all the agents in the environment. We ran 4 sets of experiments, starting with the bidirectional hall setup with 6 agents, and proceeding with additional three sets of experiments containing 75 agents in each of the simulation environments;hall, cross, circular. For all 4 sets of experiments, we set the respective weight to 0, .33, .67 and 1, *while fixing the remaining weights to .5 throughout the experiments.* An overview of the effect of these weights on the speed, DCN and DPOI for the Hall scenario with 75 agents can be seen in Fig. 5; please refer to the video for animated results of all the simulations for all scenarios.

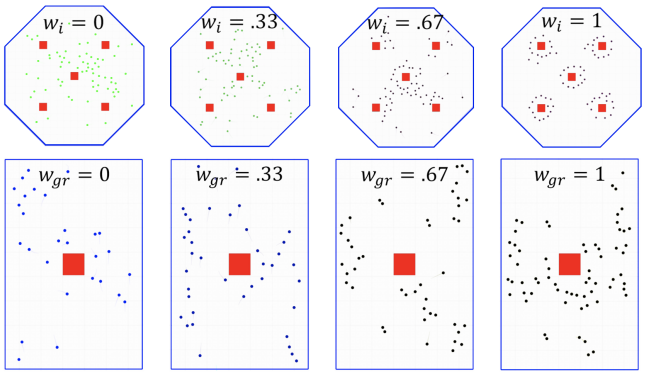

Figure 4: Effect of weight changes for the Interaction (Top) and Grouping (Bottom) Behavior.

Goal Seeking is a basic, important behavior for agents in crowds. In all scenarios, we find that as w_g increases, so does the average speed, DCN and DNOI; this indicates that agents prefer to move rather than group for long or stop at POIs. The deviations from the mean are explained by the mixture of other behaviors since we set .5 for all of them (Fig. 5). More specifically, when agents come in contact with other agents and POIs, they slow down and exhibit

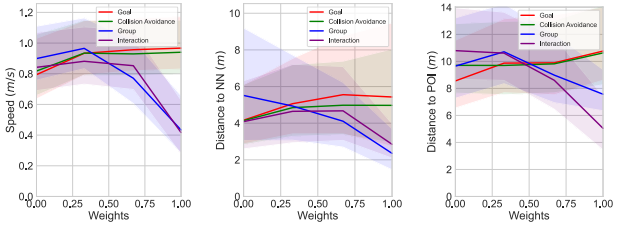

Figure 5: Weight Sensitivity in the Hall environment. We examine statistics for Speed, Distance to NN and DPOI. Shaded areas indicate half a standard deviation from the mean.

mixtures of collision avoidance, grouping and interaction behaviors. The agents finally reaching their goal depends on their incentive to do so, which is depicted by the value of w_g, as anticipated, as well as the environment conditions (number of interaction areas and nearby agents). *Collision Avoidance* is a fundamental crowd simulation behavior. Our experiments verify that as w_c increases, agents move faster and keep larger distances to the other agents and objects. However, since we also set $w_g = w_{gr} = .5$ we occasionally have grouping and interactions with the environment. The *Grouping* experimental results demonstrate that as w_{gr} increases, agents get significantly slower as they merge with others to form groups. Moreover, as expected, the closest distances to neighbors decrease since agents tend to stay together for prolonged periods of time. Interestingly, the DPOI also decreases slightly; since we have $w_i = .5$, some agents approach and stay next to others for both reasons. Finally, *Interaction with POIs* is an interesting behavior, enabling the development of simulations like crowds in museums and exhibitions. Similar to the grouping behavior, as w_i increases, agents move slower, keep short distances to each other (with little variation) and of course keep short distances to the POIs.

To summarize, we demonstrate in several environments that even when trained with multiple behaviors at the same time, we can control the behavior intuitively by manipulating profile parameters (weights). We also recorded a demonstration of dynamic *real-time control and modification* of the profiles in the crossing environment; the reader is referred to the video for the results.

4.3 Comparison with Power Law and Real-World Data

We compare CCP against simulation data from Power Law (PL) [Karamouzas et al. 2014] and real-world data of pedestrians (RW) [Lerner et al. 2007]. The RW data consist of 25 minutes of 1317 tracked trajectories from a) a medium density bidirectional flow commercial street (Zara) and b) a dense university scene. We simulate a total of 2800 trajectories in a bidirectional scenario (Hall) using different number of concurrently simulated agents (50-350) using both PL and CPP; both simulators had the same a) agent radius ($r = .5m$), b) update periods (we tested both .2s and .04s) and c) neighborhood radius ($R_s = 7m$). Since PL is a collision avoidance method, we set $w_{ca} = 1, w_g = 1, w_{gr} = 0, w_i = 0$ for CCP to be as close as possible. In total, we have 32 minutes of data for PL and 34 for CCP. We build the fundamental diagram (Fig. 6) for all three datasets; the local density d_a^t of an agent a at time t is estimated

using the equation in Helbing et al. [2007]:

$$d_a^t = \sum_{i \in N(a^t)} \frac{1}{\pi R_s^2} exp(-||\mathbf{p}_i^t - \mathbf{p}_a^t||^2 / R_s^2). \quad (2)$$

$\mathbf{p}_a^t$ and $N(a^t)$ are the position and all neighbors of agent a at time t. In all datasets, speed decreases as density increases; PL and CCP exhibit on average the same overall behavior. Additionally, the real-world data are richer in nature as compared to both approaches, indicating directions for future work and research. We have indicative simulations in the provided video.

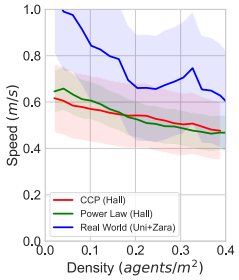

Figure 6: Fundamental Diagram for CCP, Power Law and Real-World Data.

4.4 Comparison with Baseline RL Model

On the grounds of fair evaluation, we train a baseline RL model only including goal seeking and collision avoidance. We set the normalized weights of our CCP model accordingly, to correspond to the baseline simulation ($w_g = 1, w_{ca} = .5$); side-by-side visualisations are shown in the video. Visually, we get similar results, on one side indicating the need for further exploration of how the behaviors are mixed and how to optimize the effect of the prominent behavior (e.g., finding optimal routes to reach the goal) and on the other hand showing that even with the presence of multiple behaviors a simple manipulation of weights yields comparable results.

4.5 Stress Testing the model

During the training of the model, the agents were exposed to the environment, shown in Fig. 2, with no more than 100 agents present in the simulation at the same time and using only the weight intervals in Table 1. We stress test the trained model; could it generalize to situations never faced before, like increased number of agents? Could it extrapolate behavior outside the trained weights intervals?

4.5.1 Density Sensitivity. The first aspect we study is the effect of increasing the crowd size on each of the trained behaviors. We performed 4 rounds of experiments, one for every behavior having a set of weights reflecting its effect profoundly. For both environments (i.e., hall–Fig. 7 and crossing) we find that our system is generalizable to larger crowds (up to 250 agents in our tests). As seen in Fig. 8, which regards the crossing setup, changes in density affect the goal and collision avoidance behaviors more intensely. For instance, the speed of the agents decreases with the increase in density while for interaction and grouping seem to remain constant, on average; this

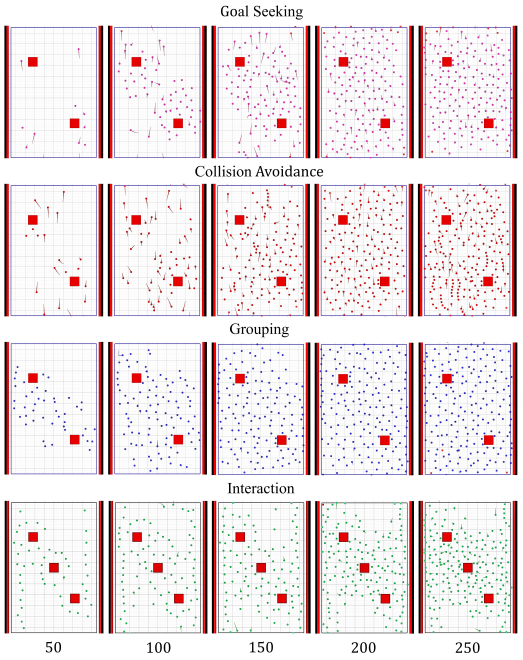

Figure 7: Density Sensitivity on the Hall Scenario.

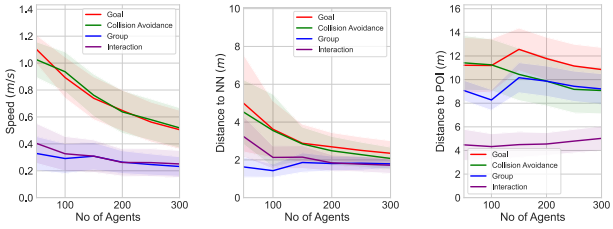

Figure 8: Density Sensitivity in the Crossing environment. We examine statistics for Speed, Distance to NN and Distance to POI. Shaded areas are half a standard dev. from the mean.

is sensible since adding more agents in the same space will cause them to slow down no matter their profile. However, this effect on grouping and interaction is more subtle since speed does not help agents achieve these behaviors better. We observe a similar trend in Distance to NN, again representative of realistic phenomena as agents would become more dense as the crowd size increases. Regarding the DPOI, for interaction it remains roughly the same since the goal is to be close to such objects, whereas for avoidance there is a trend for small decrease. For the goal and grouping behaviors, there are slight fluctuations but distance remains fairly constant, which makes sense as these behaviors are unaffected by their surrounding environmental features.

4.5.2 Weights outside training intervals. The second aspect of the generalisability experiments considers normalized weights outside the $[0, 1]$ interval. In Fig. 10 we demonstrate the effect of these changes on the Crossing Scenario for the DPOI; trends are similar

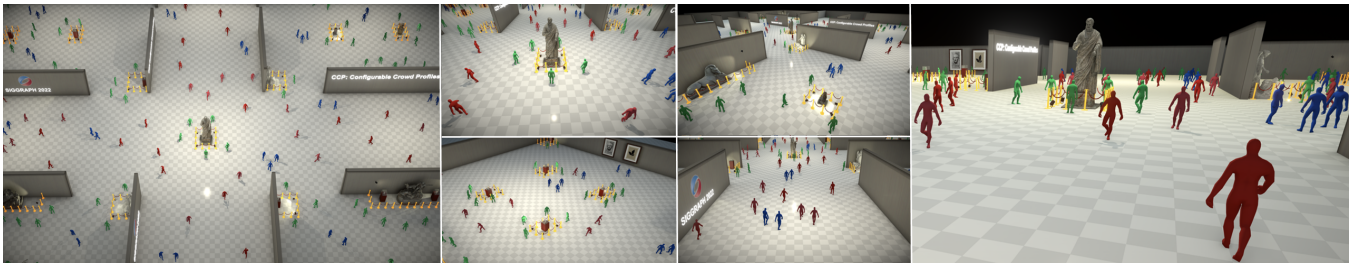

Figure 9: Museum Exhibition. This scene consists of 400 agents with different behaviors.

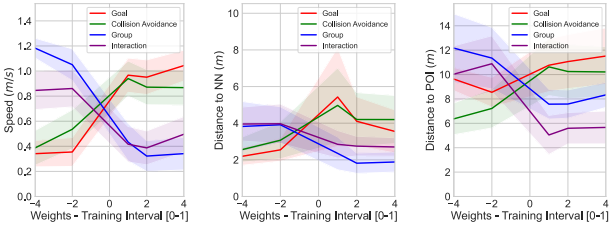

Figure 10: Statistics for ranges of profiles not seen in training. Shaded areas are half a standard dev. from the mean.

for the other statistics and scenarios also. For collision avoidance, a decrease in the respective weight corresponds to smaller distance to nearest object, indicating the inability of agents to confidently avoid obstacles. For grouping and interaction, such decrease results in larger distances whereas for larger behavior weights the distance remains fairly the same. This reflects the fact that outside the training interval the agents are not as close to the desired objects (weights smaller than 0) or remain close enough (weight larger than 1). The goal behavior's distances remain the constant on average, since such distances do not have an effect on the agents' goal seeking abilities.

4.6 Museum Exhibition

Following the analysis of the individual aspects of our system and the extent to which they reflect the intended outcome, we demonstrate its practicality when designing and authoring realistic scenes; we showcase one case study, a museum exhibition Fig. 9. The system's capabilities are reflected in a) designing a large, complex environment with abstract layout, multiple obstacles and interaction areas (walls, exhibits), b) creating custom profiles for museum visitors by mixing the trained behaviors, and c) controlling the effects of each constructed profile to the crowd (i.e., 70% of agents having dominant interaction behavior).

5 DISCUSSIONS AND FUTURE WORK

We present CCP, a learning-based framework that allows for the simulation of diverse crowd behaviors using a single policy network. Our system is useful for creating heterogeneous crowds with agents exhibiting a mixture of simpler (e.g., collision avoidance, goal seeking) and more sophisticated behaviors (e.g., grouping, interaction with environmental objects) according to users' desires. We implement a RL-based training method, able of successfully learning multiple behaviors concurrently. We demonstrate the system's effectiveness in a complex, realistic scenario of a museum exhibition.

Our novelty lies in the development of *learned configurable agents* which allow for different profiles, thus enabling users to assign behaviors and have intuitive control over them. This introduces a high degree of heterogeneity, even in a single simulation, which is controllable by users via mixing the trained behaviors; this is arbitrary and reflects different profiles; agents appear friendly, aggressive, curious etc. The generalizable qualities of our approach, both in terms of crowd size and environmental layout, further solidify our system as an efficient and practical authoring tool. Our framework also manages to extend the effect of agents' profiles beyond the training interval, intensifying or damping their effects accordingly.

Despite the fact that our method responds well to the training scenarios, the lack of a wide variety of such scenarios during training strain its impact. Moreover, the amount of distinct behaviors we used for training is limited; even though, in principle, incorporating additional behaviors would be straightforward and effective.

There are a lot of promising directions for future work stemming from this study. One interesting area we would like to investigate is how our simulated results can correlate and integrate with real-world data as well as the extent to which these correlations can be examined and deciphered. Additionally, we aim to further enrich the training process by more carefully exploring the reward function, action space, and observations. This entails a number of aspects; non-linear or DL-based reward functions instead of the linear combination of the individual rewards, different or more degrees of freedom of the individuals, and the addition of more complex features in the observations. Alternatively, in the context of character animation, it would be interesting to explore the concept of our configurable agents for style control. These tactics will either help in testing our system from different standpoints, or enhance its abilities making it more concrete, realistic and impactful.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 739578 and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy. This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement No 860768 (CLIPE project).

REFERENCES

Panayiotis Charalambous and Yiorgos Chrysanthou. 2014. The PAG Crowd: A Graph Based Approach for Efficient Data-Driven Crowd Simulation. *Computer Graphics Forum* 33, 8 (2014), 95–108. https://doi.org/10.1111/cgf.12403

Panayiotis Charalambous, Ioannis Karamouzas, Stephen J. Guy, and Yiorgos Chrysanthou. 2014. A Data-Driven Framework for Visual Crowd Analysis. *Computer Graphics Forum* 33, 7 (2014), 41–50. https://doi.org/10.1111/cgf.12472

Simon Clavet. 2016. Motion Matching and The Road to Next-Gen Animation (GDC 2016). Retrieved January 25, 2022 from https://archive.org/details/GDC2016Clavet/page/n9/mode/2up.

Funda Durupinar, Nuria Pelechano, Jan Allbeck, Uğur Güdükbay, and Norman I. Badler. 2011. How the Ocean Personality Model Affects the Perception of Crowds. *IEEE Computer Graphics and Applications* 31, 3 (2011), 22–31. https://doi.org/10.1109/MCG.2009.105

Julio E. Godoy, Ioannis Karamouzas, Stephen J. Guy, and Maria Gini. 2015. Adaptive learning for multi-agent navigation. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1577–1585.

Stephen J. Guy, Sujeong Kim, Ming C. Lin, and Dinesh Manocha. 2011. Simulating Heterogeneous Crowd Behaviors Using Personality Trait Theory. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vancouver, British Columbia, Canada) *(SCA '11)*. ACM, New York, NY, USA, 43–52. https://doi.org/10.1145/2019406.2019413

Stephen J. Guy, Jur van den Berg, Wenxi Liu, Rynson Lau, Ming C. Lin, and Dinesh Manocha. 2012. A Statistical Similarity Measure for Aggregate Crowd Dynamics. *ACM Trans. Graph.* 31, 6, Article 190 (Nov. 2012), 11 pages. https://doi.org/10.1145/2366145.2366209

Feixiang He, Yuanhang Xiang, Xi Zhao, and He Wang. 2020. Informative Scene Decomposition for Crowd Analysis, Comparison and Simulation Guidance. *ACM Trans. Graph.* 39, 4, Article 50 (July 2020), 15 pages. https://doi.org/10.1145/3386569.3392407

Dirk Helbing, Anders Johansson, and Habib Zein Al-Abideen. 2007. Dynamics of crowd disasters: An empirical study. *Physical review E* 75, 4 (2007), 046109.

Peter Henry, Christian Vollmer, Brian Ferris, and Dieter Fox. 2010. Learning to navigate through crowded environments. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Anchorage, AK, USA, 981–986.

Kaidong Hu, Michael Brandon 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* PP (2021), 1–1. https://doi.org/10.1109/TVCG.2021.3139031

Eunjung Ju, Myung Geol Choi, Minji Park, Jehee Lee, Kang Hoon Lee, and Shigeo Takahashi. 2010. Morphable Crowds. *ACM Trans. Graph.* 29, 6, Article 140 (Dec. 2010), 10 pages. https://doi.org/10.1145/1882261.1866162

Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. 2018. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627* abs/1809.02627 (2018).

Mubbasir Kapadia, Matt Wang, Shawn Singh, Glenn Reinman, and Petros Faloutsos. 2011. Scenario space: characterizing coverage, quality, and failure of steering algorithms. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vancouver, British Columbia, Canada) *(SCA '11)*. ACM, New York, NY, USA, 53–62. https://doi.org/10.1145/2019406.2019414

Ioannis Karamouzas, Brian Skinner, and Stephen J. Guy. 2014. Universal power law governing pedestrian interactions. *Physical review letters* 113, 23 (2014), 238701.

Ioannis Karamouzas, Nick Sohre, Ran Hu, and Stephen J. Guy. 2018. Crowd space: a predictive crowd analysis technique. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–14.

Ioannis Karamouzas, Nick Sohre, Rahul Narain, and Stephen J. Guy. 2017. Implicit Crowds: Optimization Integrator for Robust Crowd Simulation. *ACM Trans. Graph.* 36, 4, Article 136 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073705

Sujeong Kim, Stephen J. Guy, Dinesh Manocha, and Ming C. Lin. 2012. Interactive Simulation of Dynamic Crowd Behaviors Using General Adaptation Syndrome Theory. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Costa Mesa, California) *(I3D '12)*. Association for Computing Machinery, New York, NY, USA, 55–62. https://doi.org/10.1145/2159616.2159626

L. Kovar, M. Gleicher, and F. Pighin. 2002. Motion graphs. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 473–482.

Nick Kraayenbrink, Jassin Kessing, Tim Tutenel, Gerwin Haan, Fernando Marson, Soraia Musse, and Rafael Bidarra. 2012. Semantic Crowds: Reusable Population for Virtual Worlds. *Procedia Computer Science* 15 (Dec. 2012), 122–139. https://doi.org/10.1016/j.procs.2012.10.064

T. Kwon, K. H. Lee, J. Lee, and S. Takahashi. 2008. Group motion editing. In *ACM Transactions on Graphics (TOG)*. ACM, New York, NY, United States, 80.

Yu-Chi Lai, Stephen Chenney, and ShaoHua Fan. 2005. Group motion graphs. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, Los Angeles, California, 281–290.

Pierre Le Pelletier de Woillemont, Rémi Labory, and Vincent Corruble. 2021. Configurable Agent with Reward as Input: A Play-Style Continuum Generation. In *2021 IEEE Conference on Games (CoG)*. IEEE, Copenhagen, Denmark, 1–8. https://doi.org/10.1109/CoG52621.2021.9619127

Jaedong Lee, Jungdam Won, and Jehee Lee. 2018. Crowd simulation by deep reinforcement learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*. ACM, Limassol, Cyprus, 1–7.

Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. 2007. A Data-driven Approach to Crowd Simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '07)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 109–118. http://dl.acm.org/citation.cfm?id=1272690.1272706

Seyoung Lee, Sunmin Lee, Yongwoo Lee, and Jehee Lee. 2021. Learning a Family of Motor Skills from a Single Motion Clip. *ACM Trans. Graph.* 40, 4, Article 93 (July 2021), 13 pages. https://doi.org/10.1145/3450626.3459774

Marilena Lemonari, Rafael Blanco, Panayiotis Charalambous, Nuria Pelechano, Marios Avraamides, Julien Pettré, and Yiorgos Chrysanthou. 2022. Authoring Virtual Crowds: A Survey. *Computer Graphics Forum* (2022). https://doi.org/10.1111/cgf.14506

Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. 2007. Crowds by Example. *Computer Graphics Forum* 26, 3 (2007), 655–664. https://doi.org/10.1111/j.1467-8659.2007.01089.x

Alon Lerner, Yiorgos Chrysanthou, Ariel Shamir, and Daniel Cohen-Or. 2010. Context-Dependent Crowd Evaluation. *Computer Graphics Forum* 29, 7 (2010), 2197–2206. https://doi.org/10.1111/j.1467-8659.2010.01808.x

Pinxin Long, Tingxiang Fan, Xinyi Liao, Wenxi Liu, Hao Zhang, and Jia Pan. 2017. Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning.

Jonathan Maïm, Barbara Yersin, and Daniel Thalmann. 2009. Unique Character Instances for Crowds. *IEEE Computer Graphics and Applications* 29, 6 (2009), 82–90. https://doi.org/10.1109/MCG.2009.129

Francisco Martinez-Gil, Miguel Lozano, and Fernando Fernández. 2011. Multi-Agent Reinforcement Learning for Simulating Pedestrian Navigation. In *International Workshop on Adaptive and Learning Agents*. Springer, Berlin, Heidelberg, Berlin, Heidelberg, 53.

Francisco Martinez-Gil, Miguel Lozano, and Fernando Fernández. 2017. Emergent behaviors and scalability for multi-agent reinforcement learning-based pedestrian models. *Simulation Modelling Practice and Theory* 74 (2017), 117–133. https://doi.org/10.1016/j.simpat.2017.03.003

Ronald A. Metoyer and Jessica K. Hodgins. 2003. Reactive Pedestrian Path Following from Examples. In *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*. IEEE Computer Society, Washington, DC, USA, 149.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

Sebastien Paris, Julien Pettre, and Stephane Donikian. 2007. Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach. *Computer Graphics Forum* 26, 3 (2007), 665–674.

Nuria Pelechano, Jan M. Allbeck, Mubbasir Kapadia, and Norman I. Badler. 2016. *Simulating heterogeneous crowds with interactive behaviors*. CRC Press, USA.

Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. 2017. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 41.

Julien Pettré, Jan Ondrej, Anne-Hélène Olivier, Armel Crétual, and Stéphane Donikian. 2009. Experiment-based Modeling, Simulation and Validation of Interactions between Virtual Walkers. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, Louisiana, New Orleans, 189–198.

Z. Ren, Panayiotis Charalambous, Julien Bruneau, Qunsheng Peng, and Julien Pettré. 2016. Group Modeling: A Unified Velocity-Based Approach. *Computer Graphics Forum* 36, 8 (2016), 45–56.

Craig W. Reynolds. 1999. Steering behaviors for autonomous characters. , 763–782 pages.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR* abs/1707.06347 (2017).

Wei Shao and Demetri Terzopoulos. 2005. Autonomous Pedestrians. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Los Angeles, California) *(SCA '05)*. Association for Computing Machinery, New York, NY, USA, 19–28. https://doi.org/10.1145/1073368.1073371

Libo Sun, Jinfeng Zhai, and Wenhu Qin. 2019. Crowd Navigation in an Unknown and Dynamic Environment Based on Deep Reinforcement Learning. *IEEE Access* 7 (2019), 109544–109554. https://doi.org/10.1109/ACCESS.2019.2933492

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement learning: An introduction*. Vol. 1. MIT press Cambridge.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Csaba Szepesvári. 2010. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning* 4, 1 (2010), 1–103.

Adrien Treuille, Yongjoon Lee, and Zoran Popović. 2007. Near-optimal Character Animation with Continuous Control. *ACM Trans. Graph.* 26, 3 (July 2007). https://doi.org/10.1145/1276377.1276386

Branislav Ulicny, Pablo de Heras Ciechomski, and Daniel Thalmann. 2004. Crowdbrush: Interactive Authoring of Real-Time Crowd Scenes. In *Symposium on Computer Animation* (Grenoble, France) *(SCA '04)*. Eurographics Association, Goslar, DEU, 243–252. https://doi.org/10.1145/1028523.1028555

Unity. 2021. Kinematica. https://docs.unity3d.com/Packages/com.unity.kinematica@0.8/manual/index.html

B. van Basten, S. Jansen, and I. Karamouzas. 2009. Exploiting motion capture to enhance avoidance behaviour in games. *Motion in Games* 5884 (2009), 29–40.

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

He Wang, Jan Ondrej, and Carol O'Sullivan. 2017. Trending Paths: A New Semantic-Level Metric for Comparing Simulated and Real Crowd Data. *IEEE Transactions on Visualization and Computer Graphics* 23, 5 (May 2017), 1454–1464. https://doi.org/10.1109/TVCG.2016.2642963

D. Wolinski, S. J. Guy, A.-H. Olivier, M. Lin, D. Manocha, and J. Pettré. 2014. Parameter estimation and comparative evaluation of crowd simulations. *Computer Graphics Forum* 33, 2 (2014), 303–312. https://doi.org/10.1111/cgf.12328

Jungdam Won and Jehee Lee. 2019. Learning Body Shape Variation in Physics-Based Characters. *ACM Trans. Graph.* 38, 6, Article 207 (Nov. 2019), 12 pages. https://doi.org/10.1145/3355089.3356499

M. Zhao, W. Cai, and S. J. Turner. 2017. CLUST: Simulating Realistic Crowd Behaviour by Mining Pattern from Crowd Videos. *Computer Graphics Forum* 37 (2017), 184–201.