DOI: 10.1111/cgf.70156

COMPUTER GRAPHICS *forum*
Volume 44 (2025), number 6, e70156

MPACT: Mesoscopic Profiling and Abstraction of Crowd Trajectories

Marilena Lemonari,[1,2] Andreas Panayiotou,[1,2] Theodoros Kyriakou,[1,2] Nuria Pelechano,[3] Yiorgos Chrysanthou,[1,2] Andreas Aristidou[1,2] and Panayiotis Charalambous[2]

[1]University of Cyprus, Nicosia, Cyprus
{m.lemonari, a.panayiotou, t.kyriakou, y.chrysanthou}@cyens.org.cy, a.aristidou@ieee.org
[2]CYENS - Centre of Excellence, Nicosia, Cyprus
totis77@gmail.com
[3]Universitat Politècnica de Catalunya, Barcelona, Spain
nuria.pelechano@office365.upc.edu

Abstract

Simulating believable crowds for applications like movies or games is challenging due to the many components that comprise a realistic outcome. Users typically need to manually tune a large number of simulation parameters until they reach the desired results. We introduce MPACT*, a framework that leverages image-based encoding to convert unlabelled crowd data into meaningful and controllable parameters for crowd generation. In essence, we train a parameter prediction network on a diverse set of synthetic data, which includes pairs of images and corresponding crowd profiles. The learned parameter space enables: (a) implicit crowd authoring and control, allowing users to define desired crowd scenarios using real-world trajectory data, and (b) crowd analysis, facilitating the identification of crowd behaviours in the input and the classification of unseen scenarios through operations within the latent space. We quantitatively and qualitatively evaluate our framework, comparing it against real-world data and selected baselines, while also conducting user studies with expert and novice users. Our experiments show that the generated crowds score high in terms of simulation believability, plausibility and crowd behaviour faithfulness.*

Keywords: animation; behavioural animation, animation; motion control, methods and applications

CCS Concepts: • Computing methodologies → Motion path planning; Intelligent agents; Real-time simulation; Neural networks

1. Introduction

Simulating movements is an important part of creating high-quality populated virtual worlds. Plausible simulations are crucial to the perceived realism of movie scenes and video game environments, as well as the success of urban planning, evacuation and architectural visualisation systems. Thus, the objective in the crowd simulation domain is to easily, intuitively and quickly create virtual crowds that move and interact realistically and naturally.

Previously, research in the field primarily emphasised the basic tasks of goal-seeking and collision avoidance, which have been progressively improved over the years [VTP21]. More recently, the focus shifted to diversifying crowds by integrating more sophisticated behaviours, closer to those observed in the real world. We note that a crowd's 'behaviour' refers to the local movements of agents in a temporal window, for example, a stationary conversation that can later transition to leaving a group. This shift led to studies on group formations, more complex static behaviours, and interactions of agents with the environmental elements, like points-of-interest (POIs) [TYK*09, KBK16]. Still, in reality, people frequently perform multiple navigation tasks concurrently instead of sequentially. Capturing this complexity requires not only modelling individual movements, but also accurately representing the intricate interplay between them within the environmental context. Data-driven and learning-based methods show promise for such an undertaking, however, they are highly constrained by the input data. Also, learned behaviour models are sometimes treated as black-box information without parameterisation, limiting the creative freedom of authors [LCL07, CPV*23].

A simulator's complexity ultimately comes down to the nature of the parameters that it incorporates. Most frameworks rely on

M. Lemonari and A. Panayiotou contributed equally to this work.

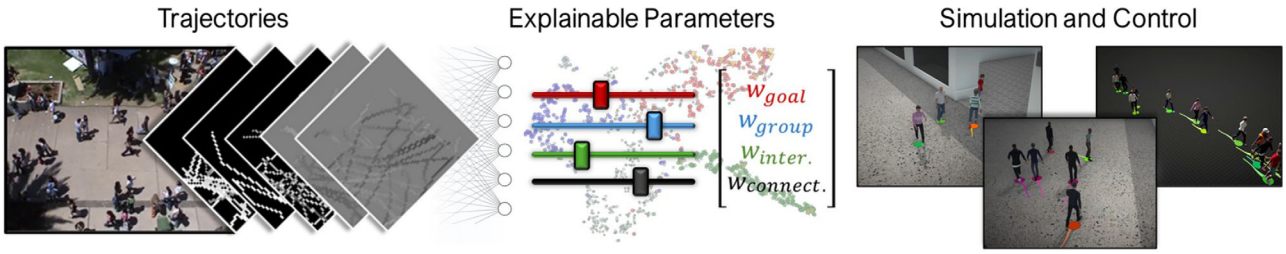

Figure 1: *The MPACT framework enables crowd generation with implicit user control. Users input a set of crowd data, with the behaviours they want to see in the generations, and author novel environments distributing the behaviours via explainable high-level behaviour profiles, optimised by the MPACT model, for example, flock-like crowd (far right).*

manual selection of the simulator parameters, which is often time-consuming, user-biased, requires expert knowledge or tedious trial-and-error experimentation. Inevitably, this paved the way for a new category of research dedicated to optimising simulator parameters. So far, optimisation methods are limited to dealing with mostly navigation/steering parameters, such as path and speed deviations, interpersonal distances and density [WJGO*14, KSHG18], and not high-level behavioural variables. Overall, achieving authentic and convincing crowds depends on the *expressiveness* of the simulator parameters and how their values are *selected*; a simulation is as good as the choice of control parameter values.

Manually defining parameter values often implies a trade-off between ease-of-use and level of achieved realism. More intuitive tools provide macroscopic control for example, flows and densities, but achieving individuality is often impossible, or comes with extensive manual tuning. Additionally, although various user interfaces (UI) have been designed to provide intuitive control to the users, their effectiveness depends on how easily users grasp the effect of changing simulation parameters on the generation. Easy-to-use UIs, that seamlessly integrate the capabilities of complex simulation frameworks, maximise usability not only for novice users but also for industry professionals.

Our goal is to deliver a comprehensive framework that represents *unlabelled* crowd data in a meaningful and controllable way, translating it into parameters for a crowd model. Specifically, we develop an inference pipeline (Figure 2) that first takes arbitrary crowd data (trajectories) and encodes it using a custom representation in the image space (Section 3.2); this representation is ideal for encoding spatial information. Our method operates *mesoscopically* - an intermediate perspective between microscopic (per-agent) and macroscopic (crowd as-a-whole), focusing on designated areas within an environment over specific time intervals. Thus, ours is a mesoscopic method for profiling and abstracting crowd trajectories (MPACT) that effectively *abstracts* behaviours found in crowd data.

Next, we deliver a model (Section 3.3), trained to navigate from the input representation space (image space) to a more compact and explainable simulator parameter space (MPACT latent space). By design, our model optimises the parameters of an underlying crowd simulator (modCCP—a modified version of CCP [PKL*22]) presented in Section 3.1. These parameter definitions are high-level and interpretable, and their combination defines a behaviour *profile* capable of describing a wide range of crowd tasks (Figure 1), making it a usable and holistic method. We explore the MPACT space and demonstrate how we can use it for profiling unknown and unseen scenarios, as well as how we operate on it, for example, interpolating between points in this space yields plausible and controllable simulations.

Our framework allows the user to guide the generation implicitly by choosing reference crowd data. MPACT then predicts the simulator's parameters, reflecting the observed behaviours in the generated simulation (Section 3.4). This removes the need for explicit behaviour definition, avoiding time-consuming experimentation and expertise in crowd dynamics. For simple scenarios (e.g., static grouping followed by goal-seeking), users could rely on modCCP or other established simulators. However, capturing intricate real-world crowd patterns with complex behaviours is challenging and inefficient through manual tuning. Our optimisation model automatically profiles trajectories while allowing fine-tuning of interpretable parameters, when needed.

It is important to note that our system is not an optimisation method for path replication; rather, we focus on achieving *behavioural similarity*. Two paths might vary spatially yet share the same behavioural intent such as moving right then grouping, or left then grouping, both reflecting a similar movement objective. Having applied our model, the user can dynamically fine-tune the predicted profiles and distribute them within custom-made environments through our intuitive and user-friendly interface. The final step involves generating a novel crowd simulation by defining the initialisation of agents and allowing them to navigate using the underlying simulator, guided by the MPACT-predicted parameters.

In a nutshell, we introduce the MPACT framework, which extracts spatio-temporal, simulator-specific parameters by learning a structured latent space. Our key **contribution** is a predictive model that maps arbitrary trajectory data to explainable parameters, guiding a crowd simulator via a custom image-based representation. In this way, users can easily and interactively leverage the predicted behaviour profiles to create and control fully-customisable crowd simulations in novel environments according to their desires. We conduct an input representation and MPACT latent space analysis, to show the abstraction and profiling capabilities of our method (Sections 4.2 and 4.3), independent of the modCCP requirements. We quantitatively and qualitatively evaluate our method in Sections 4.1 and 4.4, comparing MPACT with real data and baseline models through numerical measures and user studies.

2. Related Work

Crowd simulation can be approached from different levels, each with distinct objectives. At the *macroscopic level*, simulations focus on collective crowd movement, neglecting fine details and local navigation [JXM*10, GNL14, PvdBC*11, BSK16]. On the other hand, at the *microscopic level* the focus is on local navigation and low-level behaviour details [R*99, PAB07, KSNG17, vdBSGM11]. A *mesoscopic level* view for simulations treats agents as individual entities yet characterises their movement through aggregate relationships. Simulation methods in this category mainly focus on group dynamics and how these influence the local navigation of agents [KO12, HPNM16, RCB*17], while others incorporate social and physiological factors [DGAB16, XLL*19]. Believable crowds need to present diversity both on individual and group level, and so our mesoscopic method is suitable in this case.

2.1. Data-driven and learning-based methods

Researchers have explored *data-driven* and *learning-based* methods to enhance the plausibility of crowd simulators. Early, purely data-driven methods were mainly graph-based and were used to achieve flocking and navigation behaviours [LCF05, KLLT08, YMPT09], not particularly addressing distinct behaviours. Charalambous and Chrysanthou [CC14] extend this concept by encoding the agent states using temporal representations, constructing a perception-action graph.

Furthermore, researchers attempted to replicate the crowd dynamics observed in real crowds via letting agents learn using a set of examples [LCHL07, LCL07, LCSCO10, ZCT18, XWZ*23]. Morphable crowds by Ju et al. [JCP*10] use video-extracted trajectories with formation and trajectory models to blend input data and synthesise crowds with interpolated behaviours. Unlike their approach, we do not *manipulate* real trajectories but generate parameters that guide simulated paths to unfold input behaviours. Our framework also supports in-the-wild data without requiring behaviour annotations. Similarly, Lee et al. [LCHL07] learn an agent model from video-extracted trajectories to generate agent actions based on neighbouring movements, producing versatile crowds. While their system is highly capable, our method differs in that each generated crowd profile integrates multiple intricate behaviours. Although these methods can produce a range of behaviours, their effectiveness is highly dependent on the data quality and diversity, while their controllability is minimal. Crowd Patches [YMPT09] constructs virtual environments by linking blocks that encode periodic motion for small populations, offering control and scalability but lacking dynamic events, real-time interactivity and automated path variation—features our method handles automatically.

Recent studies have commenced exploring deep learning (DL) techniques for crowd simulation. Zhang et al. [ZYJL22] propose a framework for crowd simulation that jointly enables physics-based and DL methods to learn from each other. Additionally, reinforcement learning (RL) techniques have been explored [KKPC23]. Peng et al. [PKM*18] use imitation learning to guide an RL system so that it can follow examples from videos. However, this approach is limited by the input data and hence fails to exhibit generalisability. Hu et al. [HHB*21] introduce a multi-agent RL method that develops a parametric policy, while Talukdar et al. [TZW24] combine RL with Bayesian optimisation to navigate agents in dynamic environments. Nevertheless, these works focus mainly on predictive collision avoidance and steering, neglecting more subtle static and interacting behaviours. Panayiotou et al. [PKL*22, PAC25] propose RL-based approaches for learning diverse crowd behaviours, including the configurable crowd profiles (CCP) framework that learns multiple behaviours simultaneously, and CEDRL, a method that learns a single policy over multiple real-world datasets to capture a wide range of crowd dynamics. Charalambous et al. [CPV*23] address generalisability by using novelty detection on trajectory data to define an imitation reward for RL agent training. While it promotes diversity and novel behaviours, it lacks controllability and ignores environment structure. In contrast, our data-driven mesoscopic model optimises a defined parameter space based on input control signals.

2.2. Parameter optimisation

The majority of works on optimising crowd parameters using data design a method to analyse the data such that the analysis outcome has a compatible structure with the parameters to be optimised. Often-times, navigation and avoidance parameters are studied for example, time-to-collision, since early works focused on common simulator parameters [PPD07, POO*09, VBJK09]. The work by Wolinski et al. [WJGO*14] is capable of optimising simulation parameters to meet specific crowd navigation criteria. While their work focuses on small-to-medium scale scenarios, it can be applied to larger scale simulations by optimising a simulator's parameters to match the macroscopic features of a given fundamental diagram. Additionally, Crowd Space [KSHG18] defines a low-dimensional space employing the entropy metric to optimise the selection of different navigation algorithms. However, low densities and the diversity of input data affect its efficiency. Similarly, Berseth et al. [BKHF16] implement a framework for large-scale experiments to calibrate algorithm parameters featuring various objectives, such as minimising turbulence at bottlenecks. However, these approaches do not ensure agents behave correctly at a microscopic level, focusing mainly on steering while overlooking high-level decision-making traits. They also operate primarily in small-scale environments. In contrast, MPACT assigns dynamic, controllable agent profiles that evolve temporally and spatially, enabling life-like simulations with both moving and static interactive behaviours, as well as grouping dynamics.

2.3. Authoring and control

Having an accessible and intuitive authoring tool is key for a successful framework. Commonly, this involves user control of the simulation process, or fine-tuning of the generated results via some type of UI for example, Microsoft Office interface for controlling agent responsibilities [All10]; the survey by Lemonari et al. [LBC*22] provides a more comprehensive review on this matter. Depending on the nature of the simulation parameters, the authoring process may affect different simulation components, such as high-level behaviours [DPA*09, RPP21], path-planning [MM17] and local movements [DMCN*17]. In previous works, crowd behaviour was controlled using deformation gestures for formations and flows [JPCC14], weight manipulation for grouping [RCB*17],

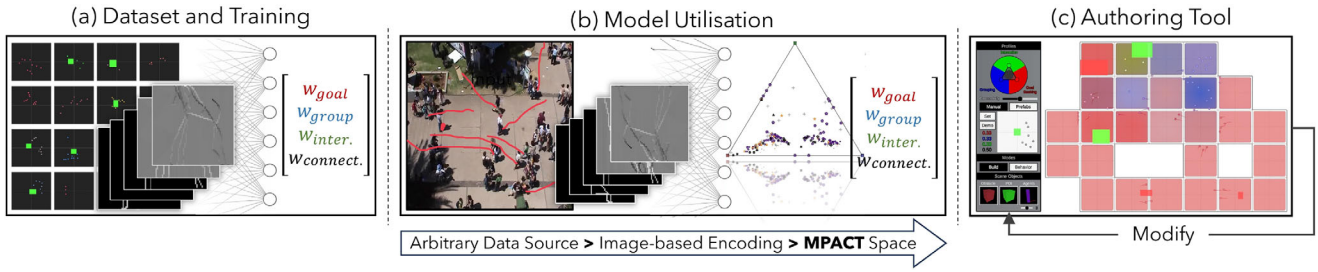

Figure 2: *MPACT framework. (a) Create the image-based encodings, based on the extracted trajectories from synthetic crowd data, and train our model. (b) Crowd data from arbitrary sources are encoded and passed through MPACT creating a behaviour profile space. (c) Sample the MPACT space enabling the distribution of profiles in the environment, simulating crowds in custom layouts with user-controlled behaviours.*

a custom sketching language [MBA22] and user-friendly tools such as brushes, storyboards and sliders [UCT04, KFS*16, PKL*22]. Interaction Fields by Colas et al. [CvTH*20] provides an intuitive authoring interface where users can draw curves, defining local navigation patterns, centred around a source point, creating an "interaction field" describing the crowd behaviour of the agents. However, users need to familiarise themselves with the system through trial-and-error, setting up the simulation from scratch. In contrast, our interface presents suggested behaviours to users that were driven by real behaviours observed in videos. Consequently, users need only to refine the results according to their desires, using intuitive functionalities like environment augmentation buttons and slider manipulation for dominant behaviour weights.

2.4. Crowd analysis

Beyond simulating and controlling virtual crowds, understanding crowd dynamics and their correlation with generated simulations is also important. Researchers have proposed various analysis methods, from trajectory-level comparisons to high-level behavioural insights. Guy et al. [GVDBL*12] introduce an entropy-based statistical metric to compare trajectory sets, aligning simulations with real data to measure residual error. Charalambous et al. [CKGC14] propose a Pareto depth analysis approach for outlier detection to identify anomalous trajectories and localised behaviours under multiple conflicting criteria. For large-scale simulations, Wang et al. [WOO16] apply path clustering to extract latent Path Patterns, while later He et al. [HXZW20] incorporate shape, speed and timing features. Amirian et al. [AZC*21] compile the majority of real world crowd datasets and analyse their statistical properties. However, focusing on high-level behaviours, the majority of works aim to cluster "similar" crowd patterns. While this approach identifies resemblance, it often fails to determine the specific nature or context of a given scenario. In contrast, MPACT can be applied to infer and describe high-level behavioural patterns in a given dataset, including their spatial distribution and how they evolve over time.

Framing our work within the literature, we describe our approach as mesoscopic since we focus on the state of an area rather than individual agents. In contract to the original work (CCP), in our framework the users only need to provide input trajectories, presenting the desired crowd scenario and then the specific behaviour weights are predicted directly. Our emphasis lies in the aspiration to facilitate novel simulations in customised environments, incorporating new crowd movements inspired by real-world dynamics rather than solely replicating behaviours captured in videos.

3. The MPACT Pipeline

This section provides a breakdown of the MPACT pipeline: the training setup (Figure 3), and inference applications (Figure 2). More specifically, we discuss (a) how we generate our synthetic training data using modCCP—our underlying simulator, (b) how we represent the crowd data in the image space and obtain image-profile pairs, (c) the learning framework and (d) how the model is utilised to serve the MPACT framework, including the proposed authoring tool.

3.1. Synthetic crowd data generation

Customising the simulator. As mentioned in Section 1, we modify the RL-based crowd model CCP [PKL*22], and use this 'modCCP' as our underlying crowd simulator. Prior to describing the core technical aspects of the modified simulator, it is important to mention that CCP inherently functions as a per-agent policy/network. At every decision step, it takes as input: (a) a profile (a set of behaviour weights), (b) a goal position in a 2D space and (c) the current state of the agent. Using these inputs, it determines an appropriate action to direct the agent's movement. CCP consists of goal-seeking w_g, grouping w_{gr}, POIs interaction w_i and collision avoidance w_{ca} weights, to capture the effect of having multiple behaviours in a single simulation, easily, with intuitive and adjustable profiles that is, $\{w_g, w_{gr}, w_i, w_{ca}\}$. A profile does not imply a weighted distribution of its behaviours amongst the affected agents, but rather, all agents exhibiting the same mixture of behaviours.

Even though the original CCP implementation provides most of the features we require, we have made enhancements to better align it with our needs. First, the absence of a continuous action space and the necessity for managing collision avoidance manually—through its dedicated weight—affect the movement quality. Thus, we incorporate continuous actions in the RL policy, and also integrate RVO [VdBLM08] to handle collisions automatically, eliminating the need for a collision avoidance weight; more details in *Actions* section below. At this stage we have $\{w_g, w_{gr}, w_i\}$-type profiles, and

Lemonari and Panayiotou et al. / MPACT

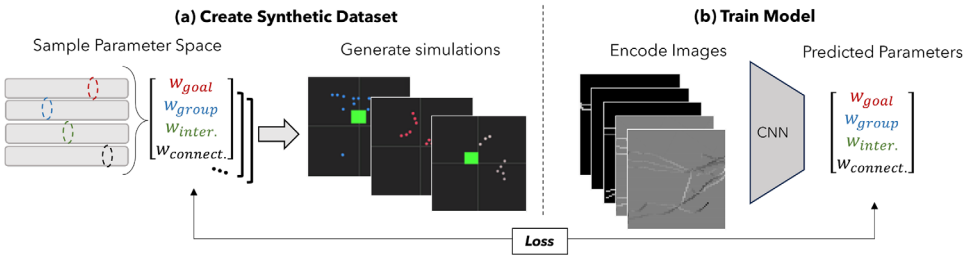

Figure 3: *Training Setup. We create the training dataset by (a) sampling the modCCP parameter space and obtaining the corresponding paths. Then, (b) we employ our image-based encoding to abstract the crowd data and use our model to predict the ground truth crowd parameters.*

to further improve behaviour blending, we adjust the reward function and impose a constraint that ensures the total of all core behaviour weights is always equal to 1 at any moment. Finally, we introduce a new weight called 'connectivity' w_c, that controls agents' proximity, offering a higher spectrum of group dynamics; this final stage yields $\{w_g, w_{gr}, w_i, w_c\}$-type profiles. Note that the concept of connectivity is similar to [RCB*17], however, our agents adjust their future velocity to keep a desired distance from their neighbours, as defined by w_c. The following part elaborates on the primary changes that have been made.

Actions: we handle agent movement by calculating a future velocity $\mathbf{v_t}$ at each decision step $t = 0.2\,s$, based on the actions generated by the policy network, which includes a moving distance $d_t \in [-.7, 2]\,m$ and a rotation angle $r_t \in [-45, 45]^\circ$. Then, v_t is set as the preferred future velocity for the RVO simulator, which adjusts it accordingly to avoid collisions between agents and obstacles.

Observations: Besides the existing laser sensor used in CCP, we additionally include visual observations using a grid sensor. The laser sensor casts rays to detect objects and measure distances, while the grid sensor provides a structured observation by encoding occupancy of objects in a grid around the agent; both sensors are provided by Unity's ML-agents framework [JBT*20]. This modification allows agents to get a better 'sense' of the environment's state, and plan their future actions in a more sophisticated way.

Reward design: As collision avoidance is now handled by a dedicated simulator (RVO), we eliminate its weight from the framework, thus, for each simulation step t the total reward R^t is calculated as $R^t = gR_g + w_{gr}R_{gr} + w_iR_i + w_cR_c + R_s + R_l$, where,

- R_g: moving towards goal position.
- R_{gr}: standing near other agents, looking towards the centre of the group, # of neighbours lower than a threshold equal to 3 m.
- R_i: standing near and looking towards a POI, # of neighbours lower than a threshold equal to 4m.
- R_c: maintain low speed variance and close proximity to the agent cluster's center-of-mass.
- R_s, R_l: smooth navigation, and living penalty.

For details on individual reward terms and conditions, see Section G of the supplementary material.

Training modCCP: We randomly spawn 25–40 agents in the square environment (25×25 m) in clusters of 1–5, we randomise rectangular obstacles and POIs with size in range [1.5 *m*, 8 *m*], and apply the curriculum-based strategy as described in the original work [PKL*22].

Applying the simulator. We proceed by choosing behaviour weights, all within the range [0,1], obeying the additional constraint that $w_g + w_{gr} + w_i = 1$. We choose to use these four parameter weights, as described above, since we believe they are capable of spanning a wide range of intermediate and sophisticated behaviours. Note that the connectivity weight w_c is not included in this constraint, by design. The three included weights have an inherent dependency on each other, for example, interacting with an object requires stopping, and so, in that case, there is no strong goal-seeking behaviour. Hence, to make this dependency more clear to the user when presenting the profile weights, we introduce this 'sum up to 1' calculation. It is not needed for w_c since the behaviour itself does not have conceptual dependencies with the rest, for example, two people can 'goal-seek' closely together, or separately. For a given behaviour profile, we *randomly* generate the environment setup (i.e., obstacles and POIs), initialise agents, and allow them to move for up to 20 s, sufficient to exhibit their assigned behaviour, while documenting their trajectories. Simulations were conducted in a 14×14 m area, appropriately sized for agents to execute their behaviours. Within these areas, we spawn 4–10 agents, ensuring they can exhibit their behaviours while maintaining a scale comparable to real-world data. Trained on this range, our model is best suited for relatively low-density crowds. Testing model predicted profiles against ground truths for synthesised behaviours with out-of-training number of agents (20,30,50,80), confirms an expected decrease in model accuracy, but, even so, an average prediction accuracy of four behaviour samples with 20 agents reaches 70.5% (bounded Euclidean distance loss projection); this is merely an indicative result as a comprehensive study for this would involve more than four samples and multiple iterations for each one. Nevertheless, MPACT could be re-trained on a wider range of possible agent numbers, making it more suitable for denser crowds, if needed. We also note that, during data collection, we use a smaller area than in the 'modCCP' training phase to avoid dynamic profile changes. At this stage we collect samples containing (agent trajectories, behaviour profile) pairs. Finally, we argue that this multi-phased randomisation enables our synthetic samples to capture a wide range of varia-

tions, enhancing model generalisability and reflecting the diversity found in real-world crowd data.

3.2. Data representation

Next, we create the dataset to be used during training. Having the (agent trajectories, behaviour profile) pairs from modCCP, we aim to obtain ground truth pairings that connect a crowd data representation (representation space) with a set of simulator parameters (parameter space). So, for the *parameter space*, we choose the four behaviour weights defined in Section 3.1.

For the *representation space*, we argue that images are a suitable option to spatially encode crowd data in a structured way so that we are able to train a model to extract modCCP parameters. Likewise, since we aim to predict localised behaviour parameters, we find it necessary to incorporate spatial information in the inputs. Our image-based encoding consists of five channels, and each one of these encodes specific properties of the simulation:

1. the *horizontal velocity*,
2. the *vertical velocity*,
3. the most efficient path to the goal point (*optimal path-OP*),
4. the *interaction clusters (IC)* between agent groups and around POIs,
5. and lastly the values of *interpersonal distances (ID)* between groups throughout the simulation.

Note that we refrain from using the trajectory image alone since it does not provide a complete understanding of the current state, and is missing details such as the direction of movement; an agent moving left-to-right and right-to-left will have the same visual effect. Thus, we encode **velocity** instead, which helps the model understand the navigation component. The remaining channels are encoded as follows. The **optimal path** is constructed by drawing a straight line connecting agent's spawn and goal positions. The **interaction cluster** is generated by connecting the positions of agents, in a radius of 3.6 m (social distance as defined by Hall [Hal63]), that are stationary in similar time intervals; we also draw the POIs in this channel (if any in the scene). Finally, the **interpersonal distance** visualises the centre-of-mass trajectory of each cluster; agents are assigned to clusters based on distance ($\leq 2.5m$) and the overall trajectory similarity.

We build the *synthetic training dataset* by mapping the generated trajectories to the image space via generating and stacking the five channels, thus producing the respective (5,64,64) images. Figure 4 shows an example of these image encodings from a reference trajectory image (for more on the randomness in the encodings, refer to supplementary material Section A). Since the images are intended to be used as inputs to the resulting model, we encode additional information via the *pixel intensity*. First, for the velocity channels, pixel intensity corresponds to positive or negative normalised speeds; we set the intensity midpoint at 0.5, and we increase (positive) or decrease (negative) pixel value accordingly. Second, for the OP channel, lighter pixels represent lower path deviation and overall path distance. Third, for the IC channel, lighter pixel values represent longer grouping duration between agents. Finally, for the ID channel, lighter pixels imply closer proximity between agents in the same group. We emphasise that the generated images exclude the spawning regions, concentrating solely on the agents' movements by capturing just the 10×10 m central portion of the environment. At this stage we generate samples containing (image encodings, behaviour profile) pairs. The resulting synthetic dataset consists of 150K samples, where the 75%, and 25% is used for training and validation, respectively.

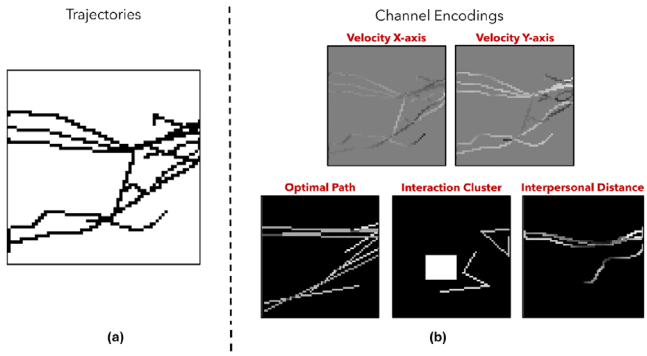

Figure 4: *Image-based encoding. We demonstrate an encoding's individual channels (b) from an example set of trajectories (a). The first two channels encode agent velocity while the others encode, respectively, the shortest path to the goal, the clustering observed both between agents and around POIs, and the interpersonal distances kept by agents in relatively close proximity.*

Table 1: *Input channels ablation study.*

	Losses ↓		Accuracies ↑			
Channels	Train	Val.	w_{goal}	w_{group}	$w_{inter.}$	$w_{conn.}$
All (5)	1.019	1.613	0.786	0.669	0.733	**0.763**
w/o OP	1.031	1.620	0.756	0.642	0.733	0.762
w/o IC	1.150	1.708	0.731	0.559	0.568	0.749
w/o ID	**1.018**	**1.602**	**0.797**	**0.682**	**0.747**	0.755
only OP	1.160	1.703	0.735	0.543	0.553	0.612
only IC	1.035	1.617	0.767	0.661	0.726	0.737

Bold text represents the best value for each column.

An **ablation study** on the input channels has been conducted to assess the contribution of the encodings, documenting model performances in Table 1, the aim of this study is to provide clarity and justification regarding the choice of the encodings as our data representation. Note that we did not ablate the velocity channels as we consider them essential for the trajectory representations. We carry out the experiments with a smaller dataset of 25K samples, while keeping the hyperparameters constant throughout the experiments for fair comparisons that is, 150 epochs, batch size of 256, and learning rate of $5e^{-4}$; more training details are discussed later in Section 3.3.

Based on the findings of Table 1, we can conclude that the least significant is the ID channel since removing it yields the lowest losses and highest accuracies, except, of course the connectivity value accuracy. Still, it is seen that its differences with the 'full inputs' model are almost negligible, and so, for the sake of w_c, we choose to proceed by utilising all five encodings as input

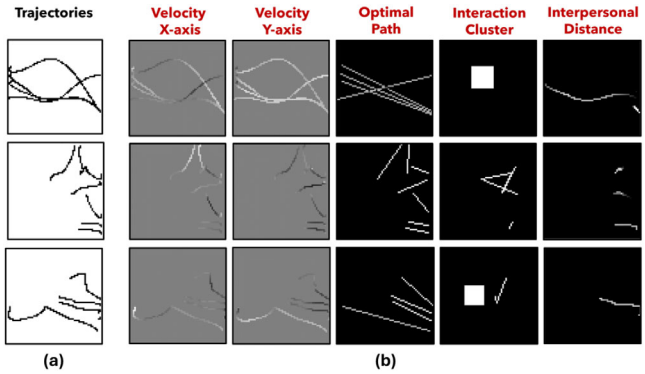

Figure 5: *Behaviour-dominant Encodings. We present per-channel encodings (b), along with the reference trajectory images (a), for examples of goal-dominant (first row), group-dominant (second row) and interaction-dominant behaviours (third row).*

channels to our model. From the three ablated channels, Table 1 reveals IC as the most important one. which is reasonable considering it provides information both for grouping and interaction with POIs. For further visualisation of the encodings, Figure 5 illustrates examples of trajectories and encodings for per-weight dominant behaviours that is, goal-dominant, group-dominant, and interaction-dominant profiles. These examples confirm our intuition that the OP channel is useful in assessing how prominent the goal weight is, whereas the IC channel gives valuable information regarding the impact of the group and interaction weights.

3.3. Training the MPACT model

Having built the synthetic dataset, we proceed to train the MPACT model to take as input the image encodings (introduced in Section 3.2) and output the underlying behaviour profile that matches the crowd task observed in the image. Our objective is to find a mapping between the image space and the crowd parameter space so that it is possible to observe behaviours in-the-wild and parameterise them in a way that is: *understandable* (weights of intuitive behaviours), *adjustable* (easy user intervention), and can be used as building blocks for novel, custom simulations (via the UI).

During training, we further augment our data by randomly applying transformation operations to the images that is, rotation and flipping. Specifically, we do not allow for full random rotations as this can lead to information loss for example, a rotation of 30° in an image with values at its corner pixels, will result in loss of information. So, we define four ranges [80°, 100°], [170°, 190°], [260°, 280°], [350°, 370°] from which a random angle can be selected. As mentioned before, the input of our model is a five-channel 64×64 image and the final output corresponds to the four behaviour weight values. The model has three convolutional layers (kernel size = 3, dropout = 0.2), followed by four fully-connected layers (dropout = 0.5). The output of the last linear layer is a six-value vector. A snapshot of our model's input/output setup is shown in Figure 6.

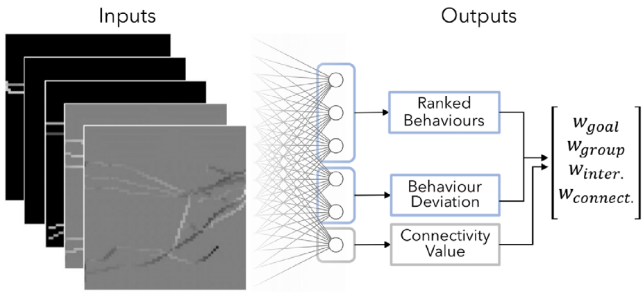

Figure 6: *MPACT input/output. Image encodings are given as input, while the network makes six predictions. The first three rank the goal, group, and interaction, according to dominance, and the next two find their difference. Weight deviations and rankings compute w_g, w_{gr}, w_i, while w_c is directly taken from the last prediction.*

By design, the model does not directly regress the four behaviour weights. The first three values of the output are used for ranking the dominant behaviours, trained with a cross entropy loss. Essentially, they represent the confidence of each core behaviour being the most dominant one. The next two values (d_1, d_2) are designed to represent the differences of the most dominant, and second and third most dominant behaviours respectively that is, $d_1 = w_{1st} - w_{2nd}$, and $d_2 = w_{1st} - w_{3rd}$. From the outputted differences, the predicted weights w_g, w_{gr}, w_i can be calculated according to Equation (1), and are trained with L1 loss. We approach the problem using multi-task learning (MTL), where one task is to predict the dominant behaviour, and the other to estimate the numerical relationship between them. Similarly, some works apply MTL for different tasks, such as combination of classification and regression [CRH*24, KJBS24], or ranking and regression [ZY10]. This model design, performs better and needs less training data. We presume that the higher performance is due to the limited prediction space; by first having the ranking, the model is given further context and so limits the kind of combinations to predict.

$$\begin{aligned} w_{1st} &= (1 + d_1 + d_2)/3.0, \\ w_{2nd} &= w_{1st} - d_1, \\ w_{3rd} &= w_{1st} - d_2. \end{aligned} \tag{1}$$

Lastly, the final output is set to correspond to the connectivity value w_c, and also trained with L1 loss; we isolate connectivity since it can coexist with all other behaviours and serves more as a distance modifier rather than a core behaviour. Hence, our final loss function is given by:

$$\begin{aligned} \mathcal{L}_{\text{total}} = .7 \cdot \mathcal{L}_{R1} + .3 \cdot \mathcal{L}_{R2} \\ + \mathcal{L}_{\ell_1}(w_{1st}) + \mathcal{L}_{\ell_1}(w_{2nd}) + \mathcal{L}_{\ell_1}(w_{3rd}) \\ + \mathcal{L}_{\ell_1}(w_c), \end{aligned} \tag{2}$$

where $\mathcal{L}_{R1}$, $\mathcal{L}_{R2}$ are the cross entropy losses of the most-dominant and second-most dominant behaviours, respectively. We prioritise the regression weights (L1 losses) by giving them higher importance as they correspond to our ultimate outputs, while we balance the losses of first (0.7) and second (0.3) most dominant behaviour,

Table 2: *Individual behaviours ablation study.*

Datasets	Zara	Students	Church	Average
Model	Diff. distance covered (m) ↓			
Full Model	2.908	**4.972**	3.258	**3.713**
w/o Goal	10.173	6.555	10.728	9.152
w/o Group	4.267	7.773	8.096	6.712
w/o Inter.	4.175	6.043	**2.711**	4.310
Fixed Conn.	**2.888**	5.761	5.196	4.615

Bold text represents the best value for each column.

because the dominant is the one having the largest impact on the resulting behaviour blend. We note that the specific balancing values have been chosen via trial-and-error.

After the model is trained, we use the validation set and a tolerance threshold of 0.1 to compute the individual accuracies. We choose this tolerance since we observed that smaller differences in weight values do not have noticeable impact on the resulting behaviours. Our model is able to predict the most-dominant and 2nd-most-dominant behaviours with accuracies 87% and 74%, respectively. Furthermore, regarding the individual behaviours, the model predicts the goal-seeking, grouping, interaction and connectivity weights with accuracies 79%, 68%, 75% and 77%, respectively.

An additional **ablation study** has been conducted to emphasise the need for the four aforementioned behaviours (Table 2). We fix hyperparameter values and train smaller models, documenting the difference between the distance covered *DC* by the agents following our models and the true *DC* by ground truth agents. We compute distance covered as $DC = \sum_{i=1}^{N-1} \|\mathbf{p}_{i-1} - \mathbf{p}_i\|$, comparing each real agent to its corresponding simulated agent. We use three snippets of real-world data from a busy street, a university campus, and a church yard (Zara, Students, and Church [LCL07, CC14]). Quantitatively examining behaviour diversity is challenging, thus, we need a simple yet expressive metric to use in this study. Our choice to use the 'distance covered' metric stems from the need of a representative metric that shows *behaviour* faithfulness since we study the usefulness of individual behaviours. For instance, if we used density, then grouping of three versus ten people would be considered wrong even though the underlying behaviour is shared. Table 2 shows the need of our chosen behaviours as, on average, the full model performs best. The per-dataset results are also sensible since the Zara dataset mostly consists of people walking to their goal, thus removing the goal behaviour hurts the performance the most. In a similar manner, the Church dataset does not contain any interaction behaviours, hence removing the interaction weight is actually beneficial for the performance. Having in mind the real world behaviour diversity, we base our choice of using all four behaviours on the average metric of the three data clips.

3.4. Utilising the trained model

At this stage, we have a trained MPACT model ready to be applied to unseen data from real, video-extracted trajectories. Each scenario is discretised both in time and space. For this, we use an appropriate

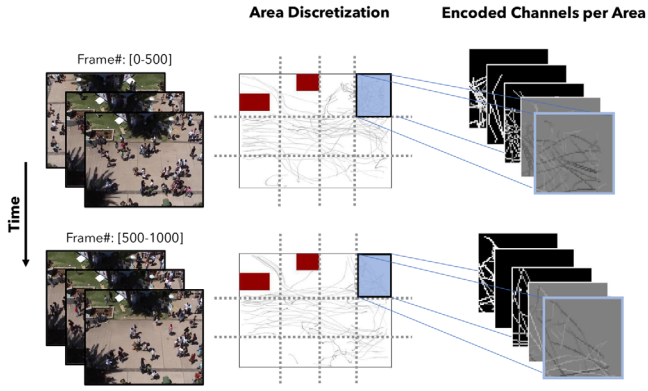

Figure 7: *Spatial and temporal discretisation. Spatial (horizontal) and temporal (vertical) discretisation of the areas in reference videos/trajectories.*

time window for each dataset (i.e., 125, 250 or 500 video frames), and according to the video aspect ratio (4:3 in our case) we divide the space into smaller areas that will each have their own behaviour profile, as hinted in Figure 7. In this stage we aim to predict the profile of each area, for each time window, given only the extracted trajectories. Here, we remind that the synthetic training samples encode simulations conducted in a 10×10 m environment, so we discretise to bring the test data closer to the training data. With our test videos, we found it best to split it in those 12 sections. Of course, this is not universal across all possible data, motivating further research as to how this discretisation should happen. We note that for each dataset, we construct the environment map by positioning obstacles and POIs as specified. Given the space and time discretisation, the agent trajectories are extracted specifically for every behaviour area and for each time-frame, to create an input image set (for an experiment with shifting window predictions, refer to supplementary material Section B). We note that, given the real world dimensions (in metres) of the environment, we can detect how each behaviour area extends and collect its trajectories. Then, the image sets for each area are fed into the trained model resulting in a series of predicted profiles per area.

As a last component in our pipeline, we allow user intervention via an authoring interface which facilitates (a) synthesising new crowds that behave according to the input behaviours, and (b) manual tweaking of parameter weights. Given a crowd dataset, our model extracts spatio-temporal behaviour profiles that can either be directly used to reconstruct the high-level crowd task observed in the input data, or strategically adjusted and distributed within the environment to generate novel crowd scenarios with similar context. Figure 8 shows an overview of our authoring tool, hinting at its functionalities: (a) the user creates a custom environment by creating the layout of the area, setting the positions and scales of obstacles/POIs, and spawn/goal positions of agents, (b) then a clustering algorithm presents to the user the suggested behaviours, as those were predicted by our model and finally (c) the user assigns them to specific environment areas for specific time-frames. A demonstration of each UI feature is provided in the accompanying supplementary video material.

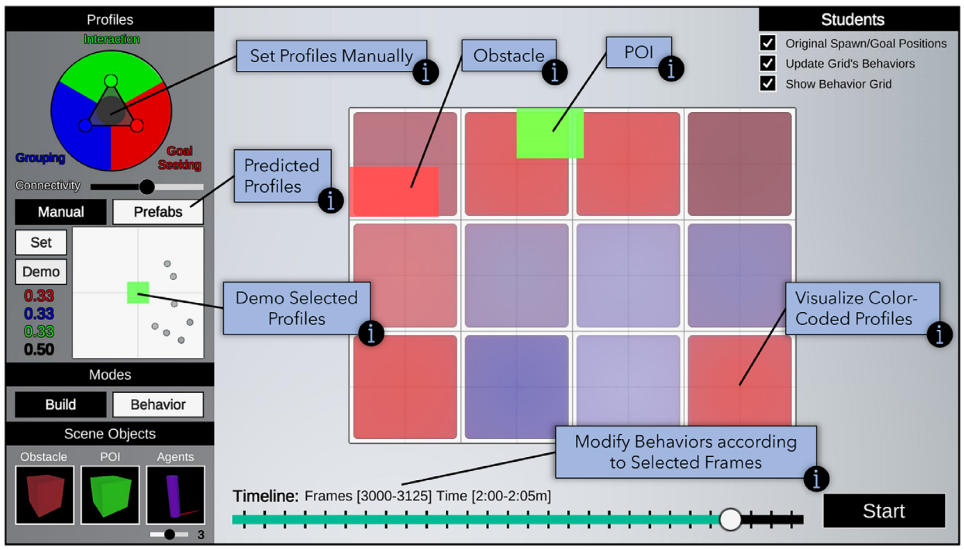

Figure 8: *Authoring user interface.*

Additionally, we use the DBSCAN clustering algorithm [EKS*96], with parameters *epsilon* = *0.125, minSamples* = *1*, to group together similar predicted behaviour profiles and show the user the cluster centres as the behaviour-dominant profiles (prefabs) to further ease authoring. Besides this, users can modify the replicated behaviours according to their preferences, as we allow them to select a particular area—choosing the preferred time-frame and grid location—and adjust the predicted parameter values using easy-to-use sliders. We alleviate the need to manually specify the parameter values from scratch, which would mean that the user should have been able to find the link between the observed behaviours and the respective parameters via trial-and-error.

4. Results

Our model has been trained on a personal computer, using a 12-core CPU, 64GBs of RAM and an NVIDIA RTX 2070 GPU (8GB). Training took approximately 9.5 hours for 500 epochs, using batch size and learning rate equal to 256 and $5e^{-4}$, respectively. For optimisation, we use the AdamW algorithm and set the weight decay equal to $1e^{-4}$. Prior to presenting our main analysis, results, and evaluation, we perform a self-consistency test to confirm that MPACT can indeed recognise the dominant behaviours (for this analysis, refer to supplementary material Section C).

Initially, we quantitatively asses our method by measuring its performance against both real world data, selected baseline works, and recent data-driven simulators (GREIL, CEDRL [CPV*23, PAC25]). For the baseline **quantitative** comparisons, we choose the existing frameworks of RVO, CCP 'baseline', and Random Walk [VdBLM08, PKL*22, GAK62]. For Random Walk, each agent navigates by setting a random moving direction (-45°, 0° or 45°), with the same frequency as 'modCCP' selects a new action. We use identical RVO parameters as the ones used in the underlying parameter model (Section 3.1), and the CCP 'baseline' refers to selecting CCP weights that best fit the video based on empirical observation and qualitative assessment; we list these values in Section 4.1. Next, we assess our input representation (five-channel image encodings) and its capability to represent crowd data in a standardised manner (see Section 4.2). Then, in Section 4.3, we explore the capabilities of the MPACT latent space and the extend to which it provides a mapping to an explainable and usable parameter space. Finally, for the **qualitative** evaluation, we perform two separate user studies (Section 4.4) that inherently compare MPACT with real data, RVO and manually-defined CCP weights; snippets of the animated results can be found in the supplementary video.

For our experiments, we utilise the three datasets: Students, Zara, and Church (as introduced in Section 3.3). For each video, we use the real agent trajectories to construct the corresponding image-based encodings, each representing a certain area and frame range.

4.1. Quantitative evaluation

4.1.1. *Testing on real world data*

We apply our model again on the three real world datasets to obtain the crowd parameters (behaviour profiles) of each area at each time window. Then, we use these parameters and run simulations keeping the original agents' spawn/goal positions, documenting the generated trajectories. In order to quantitatively assess our framework, we use four metrics: density, speed [Wei93, Fru71], distance to nearest neighbour (DNN) [HJAA07] and movement direction diversity (MDD). We compute the local density d_a^t of an agent a at time t as:

$$d_a^t = \sum_{i \in N(a^t)} \frac{1}{\pi R_s^2} exp(-||\mathbf{p}_i^t - \mathbf{p}_a^t||^2 / R_s^2). \quad (3)$$

For DNN we use a KDTree to efficiently find the nearest neighbour (within a radius of 3.6 m) of each agent at every timestep t and collect the corresponding distance. Finally, for the MDD, we construct a movement vector $\hat{\mathbf{v}}_\mathbf{i}$ every 4 s, and then calculate MDD (list of

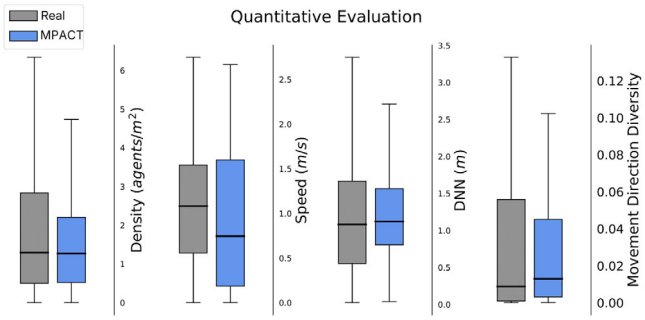

Figure 9: *Statistical metrics. We compute density, speed, DNN and MDD, using the ground truth data (real) and the MPACT-generated simulations (MPACT).*

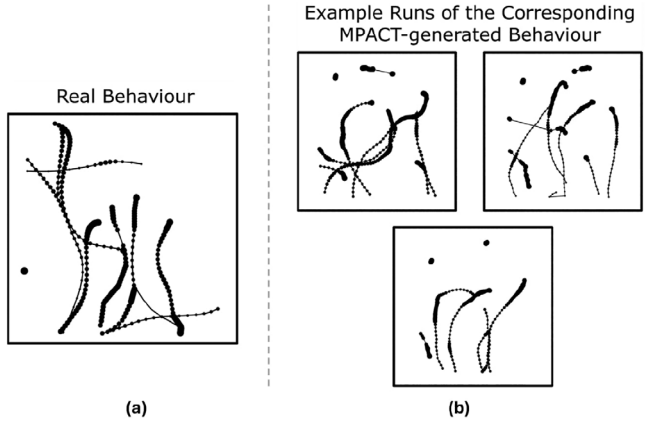

Figure 10: *Behaviour consistency. We show real trajectories for a single behaviour area (10 s clip, students dataset) (a), and different runs of generated trajectories from the MPACT-predicted profile* {.11, .5, .39, .3} *(b); thicker line implies lower agent speeds.*

values) for an agent i using:

$$MDD_i = \sum_{j=1}^{n-1} \left(1 - \left(\left(\hat{\mathbf{v}}_j^i \cdot \hat{\mathbf{v}}_j^{i+1}\right) + 1\right)/2\right). \quad (4)$$

Essentially, these metrics are used to quantify the behaviour similarity between real and simulated scenarios; for each metric, we aggregate the values for all agents, across all timesteps. Mainly, the agents in the videos walk, turn, and stand still. Combinations of these movements reveal their behaviour, for example, if agents were walking and then stopped for a long period of time, then they are likely to be grouping or inspecting a nearby POI. Conversely, if two agents move constantly while having low DNN, they possibly have goal-dominant behaviour with a high connectivity value. Hence, if two populations (real and MPACT-generated) have similar distributions of these selected metrics, we believe it is an indicative measure of overall behaviour similarity. We visualise said metric distributions corresponding to the real and simulated trajectories in Figure 9. The results show that the overall trend of the real distributions is compatible with the simulated ones, especially in median and interquartile range (IQR) [Moo09]. That said, the similarity of the box-plot whiskers is partly subjective; they provide a simplified global view of the data so we cannot yet claim with absolute certainty faithfulness to real behaviours.

Additionally, in Figure 10 we provide visual correspondence of real and simulated trajectories with the sampled underlying profile; MPACT was applied on real-world data to predict the crowd profile which was then used to generate three simulations with randomised spawn and goal positions. We show that all runs are contextually similar to the corresponding real scenario, in terms of crowd behaviour. In all cases, some agents remain stationary while others move around, occasionally pausing to meet and interact with others.

4.1.2. *Comparisons*

We carry out further experiments to evaluate the quality of our framework compared to existing 'baselines'. Specifically, we use the following models, (a) RVO, (b) random walk and (c) CCP baseline that is, having a single static manually-defined profile across all data. For the latter, we estimate the universal profile of {.45, .35, .2, .25} by analysing the occurrence of each core behaviour in the three datasets. We use (b) to serve as the control baseline, (a) to serve as the well-established and widely-used method and (c) as the direct utilisation of the underlying simulator, without any parameter optimisation. All three choices are intentional to provide several perspectives on the logical alternatives. A more careful choice of profiles for each dataset, each frame range, and each area could be made, which we explore further during our 'experts' study in Section 4.4.2. Since judging distribution similarity visually for example, box-plots, is somewhat subjective, we also calculate the Jensen–Shannon divergence (JSD) [Lin91] between the distributions of each metric type as calculated from the real trajectories and the per-model simulated trajectories. JSD is a variation of the KL-divergence and is bounded in range [0,1] with values closer to 0 suggesting similar probability distributions. Table 3 presents the JSD score between the real and simulated per-metric distributions, when using various clips from all available datasets; we use 95 s in total. Note that each per-metric distribution contains all the individual values for all agents, across all timesteps. MPACT has the lowest JSD score for three out of four metrics suggesting that it could serve as the new baseline or at least be comparable with the CCP baseline and RVO.

Table 3: *Models' comparison based on real data similarity.*

Metrics	Density	Velocity	DNN	Direction Diversity
Model	JSD ↓			
MPACT	**0.0167**	**0.173**	**0.0355**	0.0137
CCP baseline	0.0189	0.315	0.0421	**0.0113**
RVO	0.0171	0.483	0.0476	0.0753
Random walk	0.0651	0.461	0.0978	0.246

Bold text represents the best value for each column.

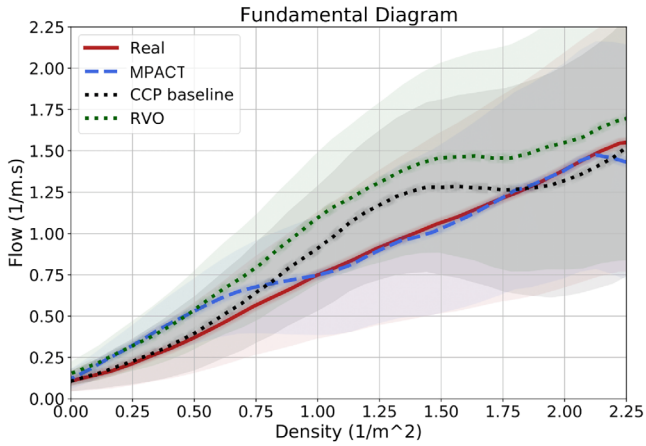

Figure 11: *Fundamental diagram. We present real world, MPACT, CCP baseline and RVO data, consisting of the combination of Students and Zara datasets.*

Investigating further, we build the fundamental diagram (FD) [FMN18, Fru71] comparing the trend of real world and simulated data. FDs are widely used in the crowd simulation domain either in the velocity/density or flow/density form. They are a robust way to visualise the interplay of key macroscopical crowd characteristics. For the FD in Figure 11, we combine the Students and Zara datasets resulting to a total of 562 agent trajectories and 9.5 min of simulation. The local density of each agent a at each timestep t is again calculated by Equation (3), and all values are aggregated. The FD reveals a higher similarity between the real world and MPACT simulations, compared to RVO and CCP baseline. The latter methods seem to face higher flows in medium densities.

State-of-the-art crowd simulators. While MPACT is not a dedicated crowd simulator, we also assess how it performs against two recent data-driven models that offer behaviour diversity: GREIL [CPV*23] and CEDRL [PAC25]. We construct the FDs using two real-world datasets and present the results in Figure 12. The first, Zara, was used during the training of both GREIL and CEDRL. Simulations with all three models—using real spawn times, positions, and goals—show that MPACT outperforms GREIL, in this scenario, and performs comparably to CEDRL, despite being trained solely on synthetic data. To further evaluate generalization, we use the ETH Hotel dataset [PESVG09], which was not included in CEDRL's nor MPACT's training. In this setting, the MPACT-generated simulation aligns more closely with real-world statistics, whereas CEDRL exhibits higher flow at higher densities; GREIL is excluded from this setup, as it requires a dataset-specific trained policy that is not available. Additionally, we plot the paths of the generated simulations in Figure 12, where lower colour intensity indicates higher agent speed. In line with the FD results, CEDRL agents tend to move faster and exhibit more direct, goal-oriented behaviour, while MPACT agents seems to better capture the intermediate behaviours presented in the data.

While the FD reflects overall crowd dynamics, it may overlook finer aspects such as smooth navigation or fine-grained interactions. Still, it suggests that our approach performs competitively with state-of-the-art methods, while also offering authoring and controllability features that are limited or absent in the compared models.

4.2. Representation analysis

Next, we investigate the extend to which our image-based representation (formulated in Section 3.2) abstracts crowd data into a structured representation space, where data with similar underlying behaviours fall close to each other.

To measure how 'close' the representations are, we use a contrastive learning framework (simple contrastive learning of representations—SimCLR [CKNH20]) to learn a structured latent space based on (image-based encodings, behaviour profiles) pairs. Visualising the learned latent space (Figure 13-top), we expect the compressed representations of similar behaviours to form clusters for example, goal-seeking data generated from RVO are close to goal-dominant data from the original CCP simulator, whereas group or interaction data differ. We present the latent vectors of three types of data. Inverted triangles denote encodings of dominant behaviours from different generators (CCP, RVO, modCCP) and instances (2 CCP runs for the same profile). We also illustrate encodings representing the same 'complex' scenarios (Zara, Church, Students), taken from real data, experts, and modCCP, and lastly some of the training data with dominant behaviours (3-coloured dots). The space is structured, as representations of dominant behaviours fall within common clusters. Samples with intriguing behaviours are more widespread but still structured (Student points form smaller clusters) and distributed according to the crowd task they resemble most; we do not expect them to gather around the dominant latent clusters, otherwise the scenarios would have been straightforward to begin with, superseding the need for an alternative representation. We also do not expect distinct clustering for example, all Student samples together, because real data naturally face diversity, and the behaviours (and thus the encoding) change with time and location.

Since the trained network only looks at the image-based representation that is, does not have information about the behaviour driving the data, a structured latent space implies that our proposed image-based representation contains suitable features, encoding the underlying crowd task, even if the data itself (trajectories) vary. As additional insights, in Figure 13-bottom, we provide the feature maps (as extracted from convolution layers in three different depths) of goal-dominant scenarios, grouping, and interaction with POIs. We observe that the trained model tends to focus on similar features when the representation encodes a comparable scenario (e.g., goal-seeking in [a] and [b]), whereas different behaviours exhibit distinct features. Hence, our representation is meaningful, effective and generalisable, that is, independent of the underlying data generator. Compared to alternative crowd data representations for example, feature-based [KSHG18], our encodings are model-compatible (MPACT model), expanding its general utilisation opportunities; this representation can then be used to author new crowds via implicit high-level control ques.

4.3. MPACT latent space analysis

We additionally assess the degree to which the MPACT model maps crowd data (encodings) into a generalised, interpretable and

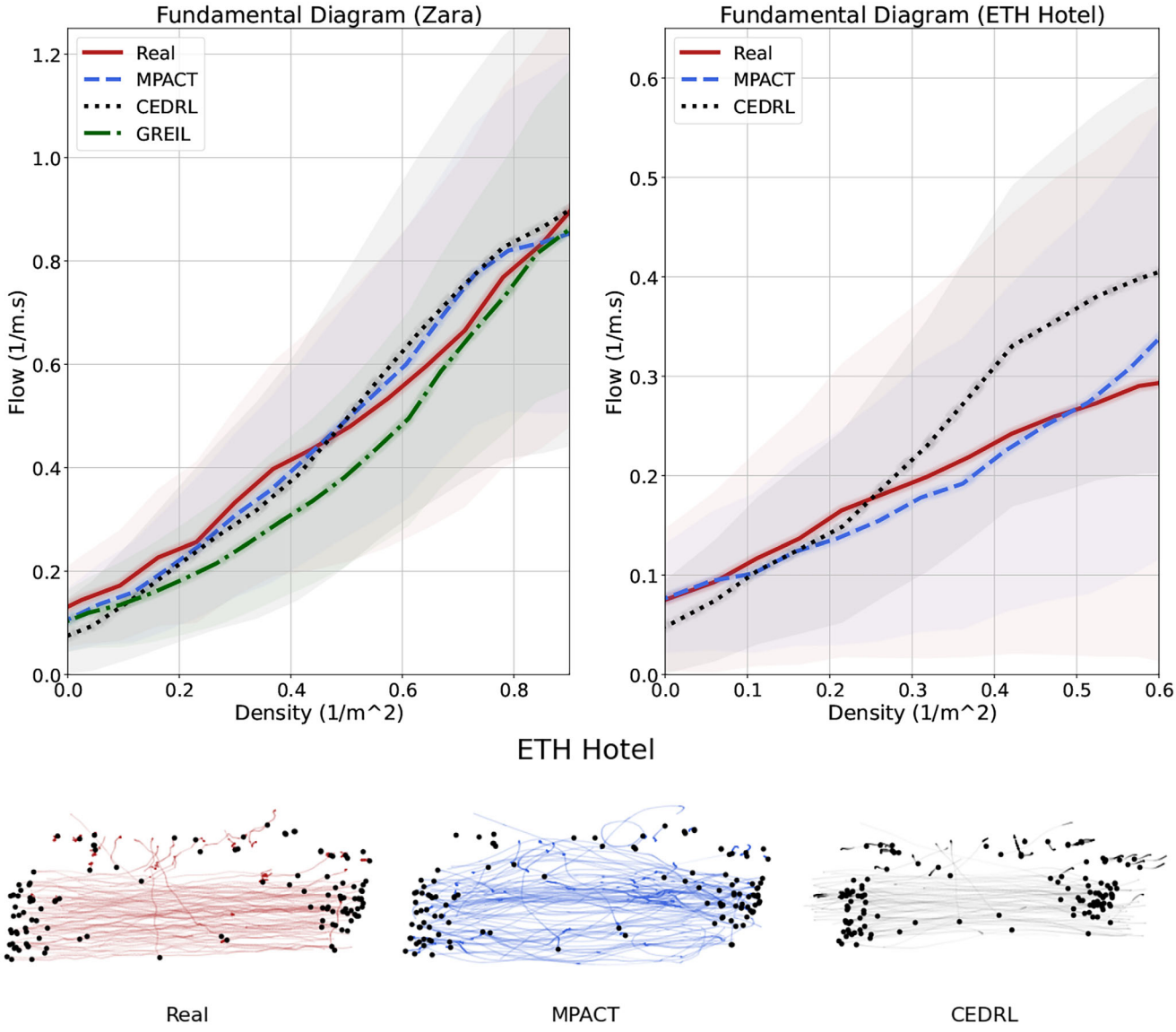

Figure 12: *State-of-the-art comparison. FDs for two data-driven simulators on the Zara and ETH Hotel datasets, along with generated paths; lower colour intensity indicates higher speed.*

controllable latent space. We visualise the MPACT latent space in 2D triangular space. Due to the nature of the profile weights, that is, $w_g + w_{gr} + w_i = 1$, we plot each core behaviour *goal*, *grouping* and *interaction* into the triangle's vertices. Then we use Barycentric coordinates to place each sample in the space; *connectivity* is represented by the size of each marker.

The goal is to achieve a generator-independent method, providing both an explainable insight into crowd behaviours, and high-level user control with authoring capabilities. So, we visualise the MPACT-predicted profiles for the three real world datasets (Figure 14a), along with predictions of real, modCCP simulations and 'experts'—selected profiles (Figure 14b). For the latter, six expert users are shown a 20-s clip of a different crowd video and are asked to replicate the behaviours they observe using our UI; replications refers to capturing similar behaviours and not exact agent paths. They do so by assigning behaviour profiles (of the form $\{w_g, w_{gr}, w_i, w_c\}$) to designated areas for every 5-s window; each user can fine-tune their selections as much as they want. The three shown clips come from the Church, Zara, and Students videos, given to users in this order to correspond to increasing behaviour complexity. We used the real spawn and goal positions from each dataset, making it easier for users to compare generated behaviours to the videos. For each user and scenario, we tracked the time spent, simulated trajectories, and chosen profiles. However, for the analysis of this section, we only use their chosen profiles; more analysis about the time spent and the simulated trajectories can be found later in Section 4.4.2. We then:

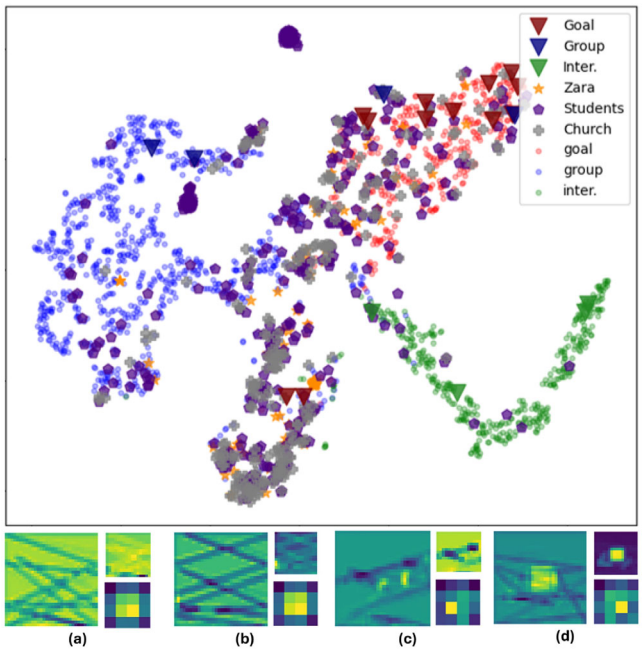

Figure 13: *Image-based encodings manifold. The latent space containing seen and unseen data samples generated by SimCLR (top). SimCLR convolutions: 3-depth feature maps of (a) and (b) goal-scenarios, (c) grouping and (d) interaction (bottom).*

- Observe the distributions in the latent space (Figure 14a), and the spatio-temporal structure (Figure 14b).
- Infer how unknown scenario classify (Figure 15).
- Present how to leverage the learned manifold to intuitively author intended crowds (Figure 16).

Each visualised point represents a specific area at specific timeframe of the respective scenario for example, first 20 s of the top-left grid of the Zara data. Figure 14a shows that Zara scenarios are mostly goal-driven, while Student exhibit stronger grouping. Generally, interaction with POIs is not frequently found in these inference scenarios. These findings reveal that the MPACT latent space gives an insight into the type of crowd (*explainability*) the input data comprise through observing the behaviour distributions.

To verify that the MPACT space is meaningful and structured, first we plot examples of a set of points that correspond to the same scenario (fixed grid location and time-frame) sourced from different generation methods, that is, real captures, modCCP, and experts (Figure 14b); since we have 6 experts, we average similar experts' points for fairer observations. We confirm that these points fall within the same latent areas with slight deviation which can be attributed to generator biases for example, expert familiarisation with different crowd scenarios and their simulation area. We can also explore the smoothness of the traversed path that correspond to timewise-consecutive profile predictions. Rendered simulations showcasing a sample behaviour shift are presented in Figure 17a; for animated results and further discussion, please refer to the supplementary video and Section D of the supplementary material. The structured MPACT space is general since it does not depend on an underlying simulator/generator. For inference, we can use 'unknown' scenarios (ψ) from additional external generators, that is, ETH Hotel [PESVG09], CEDRL [PAC25], Lee [LCHL07] and another version of Zara [LCL07]. We apply the MPACT model and plot the predictions in Figure 15. Performing a K-means clustering on the latent vectors, we find the 5 closest points and classify the unknown ψ to the most probable scenario using majority vote (examples shown in Figure 17b–d).

We can also perform mathematic operations in the latent space, as in Figure 16. For example, given two scenario instances the user can apply linear interpolation between them to author intermediate blended behaviours, controlled by the scenarios that are known to the user and implicitly define the desired crowd tasks.

4.4. Experimental studies

4.4.1. *User study*

Setup:. We conducted a 20-min user study with 44 participants to assess our framework's simulations quality and believability. Our participants were diverse in age (18, 65), sex (25M/17F), and country of work. Users were requested to complete the study on a desktop computer. Overall, the users' crowd simulation knowledge varied with an average of 2.5/5; we believe this to be a relatively high score for a no-prerequisite study. Through this study we aim to test the degree to which the behaviours generated by MPACT are realistic-looking to users. To do that, we show 10 videos of real and MPACT simulations in random order, asking participants to choose the more realistic one. Participants were asked to watch the videos fully and judge based on overall behaviours rather than frame-by-frame paths, and we gather their responses.

Findings:. We measure how confused the users were when having to distinguish real versus MPACT paths and summarise the results in Figure 18. We can deduce that approximately half the time users were unable to differentiate the real paths. Notably, this confusion was not due to averaging nearly correct and nearly incorrect answers across samples. Instead, on a per-sample basis, users were equally confused in around 50% of the cases. For example, in sample 2, approximately 54.5% of users misidentified the path shown as 'real' when it was, in fact, generated by MPACT. This consistent confusion across samples suggests users struggled to differentiate between real and MPACT paths. Additional analysis can be found in Section E of the supplementary material.

4.4.2. *Expert study*

Setup:. We conduct an additional experiment, by designing a study directly comparing MPACT predictions with CCP baselines, as set by "experts" (six trained users with extensive knowledge of crowd simulation). We ask experts to manually produce a series of behaviour profiles that match input videos. Our attempt is to obtain the manual counterpart of our automated MPACT-predicted behaviours, and compare them in terms of performance and quality. The findings of this study provide insights into the practical usefulness of our model, since we assess how easier this task can be accomplished by bypassing the automated optimisation method, but

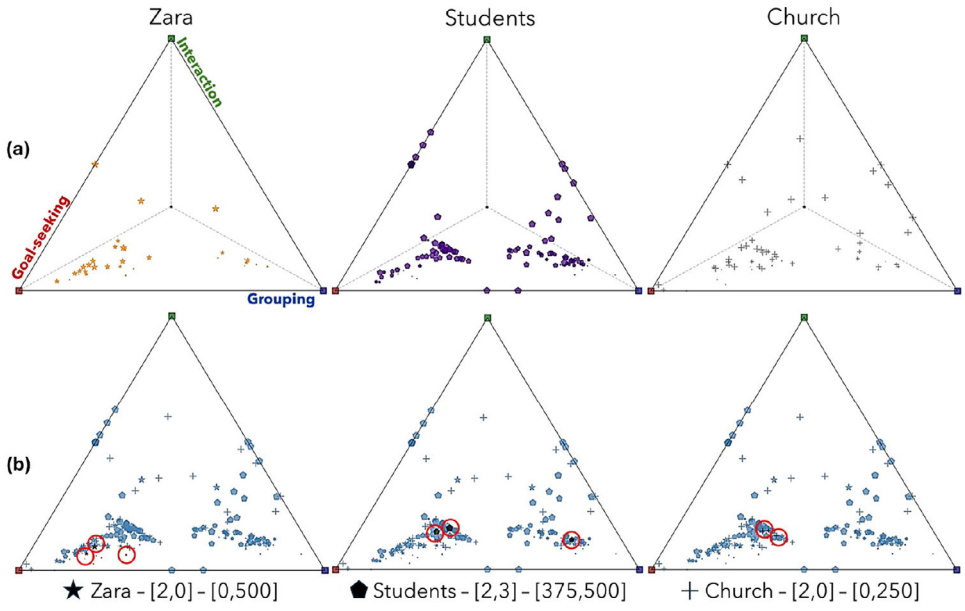

Figure 14: *MPACT latent space. Predicted profiles for real world data (a), and selected real, modCCP, and expert predictions for specific grid cell and frame range (b).*

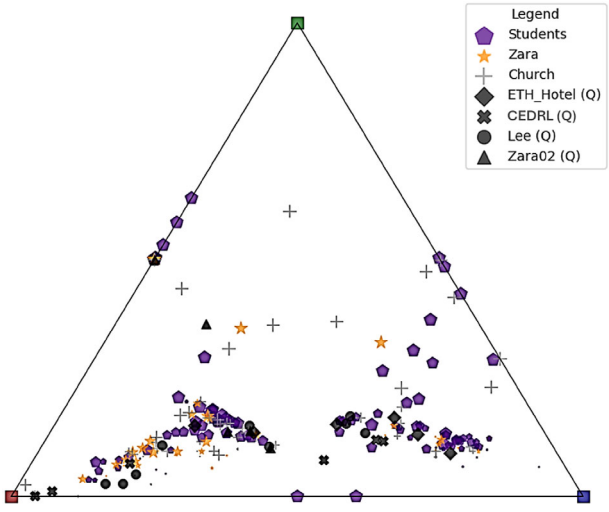

Figure 15: *Unknown scenarios classification. Query (Q) scenarios among inference scenarios comprising the search space.*

instead, having capable users directly define the parameter values, based on experience. During this expert study, users first complete a 'demo' experiment to familiarise themselves with the interface; no data is recorded during this phase. Then, the official study commences that consists of three stages, as described in Section 4.3.

Time-Efficiency Findings:. As our method is automated, the one advantage we expect over manual alternatives is the inference speed. Indeed, having recorded the time spent to settle on the chosen weights, we can verify that experts take much longer than our method, with an average of 9.71 min versus 10.2 s, respectively.

Quality Findings:. To get an idea of how close the expert weights are to the predicted ones (MPACT), we plot the average differences of MPACT with each participant, showing an overall increasing trend (Figure 19). In less complex datasets such as Church, participants could match MPACT easier, however, replication quality declines with increasing behavioural complexity, as seen in the Students dataset. Thus, even expert users exhibit significant deviations when attempting to replicate intended behaviours in complex scenarios. Hence, having an automated, stable method is beneficial, so long as it is closer to the real data that the alternatives. More results can be found in the supplementary material Section F.

To assess similarity with the real data, we build FDs of the paths taken from the real data, experts, and MPACT (Figure 20), for each of the three datasets. We observe that real, MPACT, and most participant runs follow the expected pattern: crowd flow increases with density until high densities reduce agent velocities, implying plausible generations. In determining which is closer to the real data curve, we notice that the MPACT curve is the most consistently similar one. For example, in Zara, we see that P2 also managed to capture the real crowd behaviour, but fails to do so in the other two datasets. Even compared with P5 (most successful participant), MPACT is still closer to the real curve in at least the Zara dataset.

5. Discussion, Conclusions and Future Directions

In conclusion, this paper presents the *MPACT framework*, an image-based prediction model that extracts optimal weight values from unlabelled input trajectories, for simulator-specific parameters. The MPACT pipeline maps complex, intertwined crowd tasks

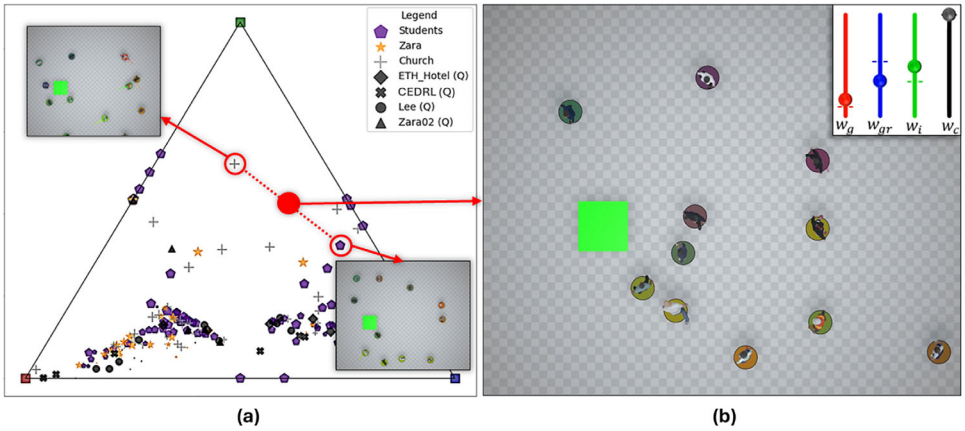

Figure 16: *Latent space linear interpolation. Transitioning from group-dominant behaviour presented in Students, to a more interaction-dominant found in Church.*

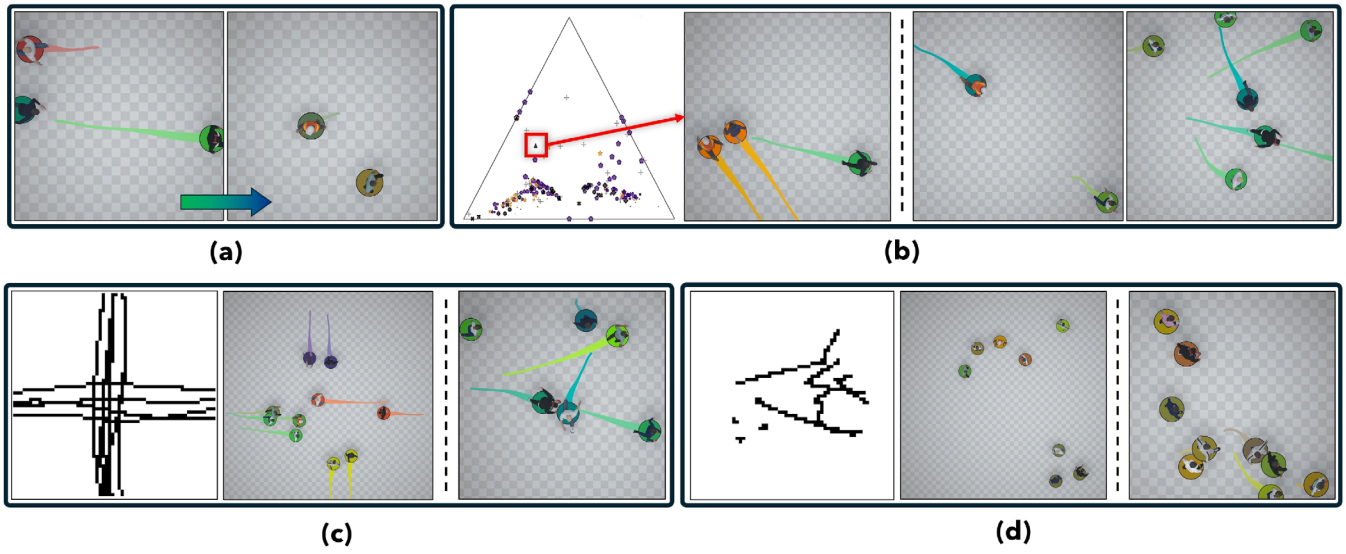

Figure 17: *Rendered ψ examples. (a) Timewise evolution of Zara scenario captured at the first 20 and 80 s time marks. (b) A sampled profile from another version of Zara dataset, its rendering and two additional renderings from the nearest neighbouring profiles. (c) and (d) The trajectories in raw and rendered form, along with a nearest neighbour sample from CEDRL and Lee, respectively.*

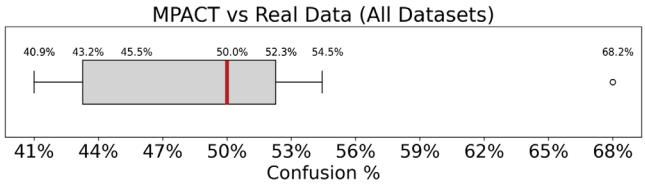

Figure 18: *User study results.*

onto compact, explainable profiles, enabling the parameterisation of in-the-wild behaviours. These profiles capture the behaviours in the input data and can be applied to a crowd simulator ('modCCP') to reflect them in the generated simulation. Additionally, our work is not limited to navigation-type parameters, but rather operates in an elaborate parameter space; it spans a diverse range of behaviours from simple for example, goal-seeking only, to advanced such as mixture of goal, grouping and interaction with POIs.

The framework's usability is unlocked via the *MPACT UI* where users assume authoring and control by leveraging the MPACT predictions. Users implicitly define the desired behavioural pattern by selecting an appropriate input crowd scenario, then realise these reference behaviours by applying the MPACT-predicted profiles. These profiles can be distributed in space and time according to their wishes, in custom virtual environments, without compromising simulation plausibility. Some familiarisation with the interface and how parameters correlate to agent movements is necessary for the proper utilisation of the framework for example, what a half-grouping, half-goal profile looks like.

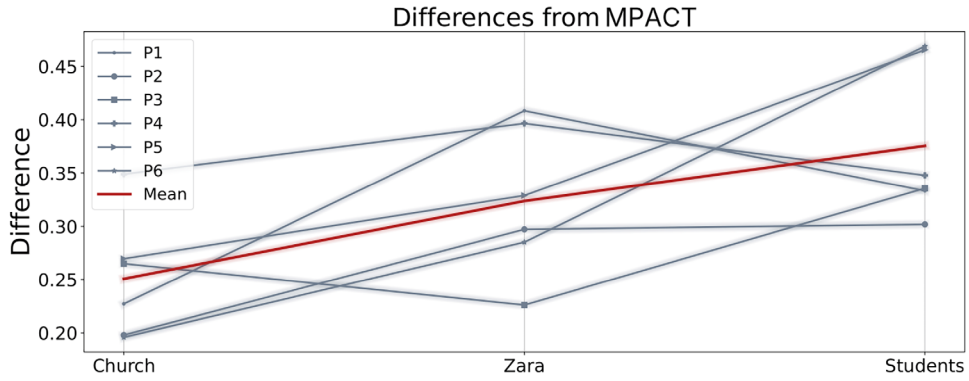

Figure 19: *Average overall profile difference between all participants and MPACT predicted profiles.*

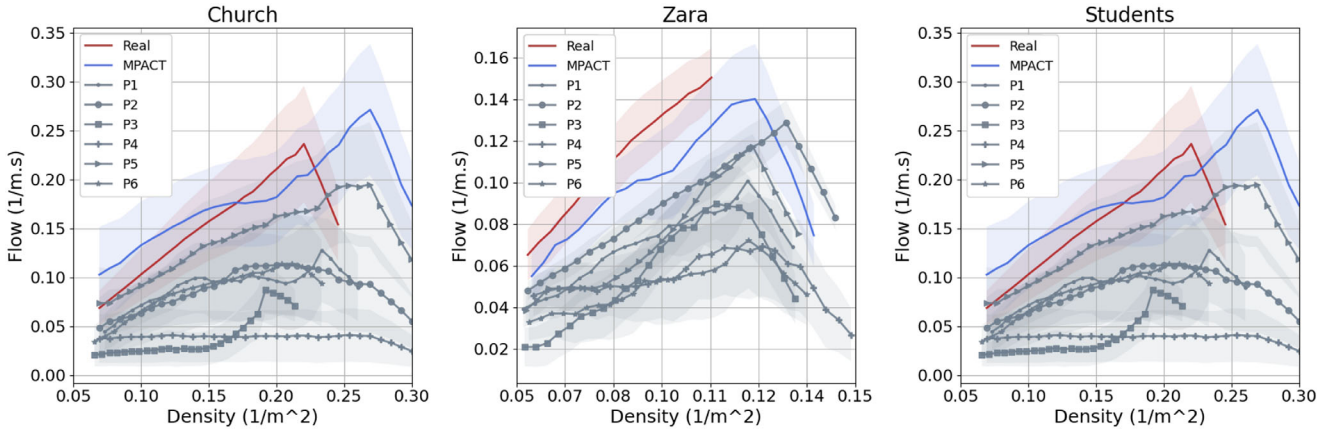

Figure 20: *Fundamental diagrams for 20 s snippets of all three datasets (Church, Zara, Students). For each, we depict the trends for Real (red), MPACT (blue) and all expert study participant runs (grey variations).*

We conduct a series of experiments with attention to our input representation (image-based encoding) and the MPACT latent space. Our findings show that MPACT effectively abstracts crowd data into a compact representation that encodes behaviour similarity, which is also compatible with a profiling model that assigns explainable and controllable behaviour profiles to the otherwise unlabelled crowd data. We qualitatively and quantitatively **evaluate** our model's generation quality, establishing MPACT as a fast and reliable framework for generating plausible simulations. The quantitative comparisons of crowd-related and behaviour-characterising metric distributions (e.g., densities) against existing methods and real data, subsidises MPACT's contribution.

Despite the advantages and effectiveness of our framework, it has certain limitations. Currently, we do not integrate a video-to-trajectory method, and so only tracked crowd data can be used without preprocessing. Also, we acknowledge that the underlying crowd model we employ, has its own limitations that may accumulate in our framework for example, CCP has not been trained on real world data. The MPACT paradigm has the potential to be extended to other simulators, requiring a similar framework setup but retraining with different configurations and data. In general, training with real data is particularly challenging due to the lack of trajectory-parameters pairs. A possible direction would be getting experts to label the dominant behaviours, enabling fine-tuning of the proposed pipeline. Finally, we recognise that further improvements on UI can be made for maximum effectiveness such as allowing for the propagation of selected profiles in future time windows, and seamlessly integrating trajectory prediction.

Our approach opens new possibilities for refining and expanding the MPACT framework. An interesting improvement to our method would be predicting a profile distribution for each area, instead of a single one, an addition that would also highly increase the complexity. Additionally, our current system treats behaviour areas as fixed-sized rectangular segments and the discretisation of the area is not universal across different input data. In the future, we aim to explore integrating more complex area shapes and automating their establishment based on real, observed environmental information. A promising future study would be to broaden the range of explainable parameters by introducing an enhanced version of modCCP; however, the more weights (behaviours) we add, the more challenging it becomes to balance them. To keep a small number of weights per policy, an option is to have separate policies for different sets of weights.

The generalisability of our method, as well as its applicability to real world scenarios, can be enhanced. Transfer learning could be explored to translate modCCP profiles into parameters of other crowd simulators. More importantly, looking at the distribution of real scenarios in the MPACT latent space (Figure 15), we notice a smaller manifold in the real data rather than the training space. This challenges the presence of the current definition of the interaction weight in real behaviours. We aim to further investigate formalising several behaviour types and adjust our training data to better align with the 'real-data space'. This creates the opportunity for creative and informed crowd synthesis. We emphasize, however, that this is challenging since it requires extensive research and careful analysis to achieve unbiased results.

Acknowledgements

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). This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 739578, the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy and Department of Research and Universities of the Government of Catalonia. Grant Number: 2021 SGR 01035. This work has received funding from the European Union under grant agreement No 10192889. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Conflict of Interest

The authors declare no conflicts of interest for this work.

References

[All10] Allbeck J. M.: CAROSA: A tool for authoring NPCs. In *Motion in Games: Third International Conference, MIG 2010, Utrecht, The Netherlands, November 14-16, 2010. Proceedings 3* (2010), Springer, pp. 182–193.

[AZC*21] Amirian J., Zhang B., Castro F. V., Baldelomar J. J., Hayet J.-B., Pettré J.: OpenTraj: Assessing prediction complexity in human trajectories datasets. In *Computer Vision —ACCV 2020* (Cham, 2021), Ishikawa H., Liu C.-L., Pajdla T., Shi J., (Eds.), Springer International Publishing, pp. 566–582.

[BKHF16] Berseth G., Kapadia M., Haworth B., Faloutsos P.: Steerfit: Automated parameter fitting for steering algorithms. In *Simulating Heterogeneous Crowds with Interactive Behaviors*. AK Peters/CRC Press, 2016, pp. 229–246.

[BSK16] Barnett A., Shum H. P. H., Komura T.: Coordinated crowd simulation with topological scene analysis. *Computer Graphics Forum 35*, 6 (2016), 120–132.

[CC14] Charalambous P., Chrysanthou Y.: The PAG crowd: A graph based approach for efficient data-driven crowd simulation. *Computer Graphics Forum 33*, 8 (2014), 95–108.

[CKGC14] Charalambous P., Karamouzas I., Guy S. J., Chrysanthou Y.: A data-driven framework for visual crowd analysis. *Computer Graphics Forum 33*, 7 (2014), 41–50.

[CKNH20] Chen T., Kornblith S., Norouzi M., Hinton G.: A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)* (2020), PMLR.

[CPV*23] Charalambous P., Pettre J., Vassiliades V., Chrysanthou Y., Pelechano N.: GREIL-crowds: Crowd simulation with deep reinforcement learning and examples. *ACM Transactions on Graphics 42*, 4 (July 2023), 1–15.

[CRH*24] Cao H., Rajan S., Hahn B., Kocak E., Durstewitz D., Schwarz E., Schneider-Lindner V.: Mtlcomb: multi-task learning combining regression and classification tasks for joint feature selection. *arXiv preprint arXiv:2405.09886* (2024).

[CvTH*20] Colas A., van Toll W., Hoyet L., Pacchierotti C., Christie M., Zibrek K., Olivier A.-H., Pettré J.: Interaction fields: Sketching collective behaviours. In *MIG 2020: Motion, Interaction, and Games* (2020).

[DGAB16] Durupinar F., Gudukbay U., Aman A., Badler N. I.: Psychological parameters for crowd simulation: From audiences to mobs. *IEEE Transactions on Visualization and Computer Graphics 22*, 9 (2016), 2145–2159.

[DMCN*17] Dutra T. B., Marques R., Cavalcante-Neto J. B., Vidal C. A., Pettré J.: Gradient-based steering for vision-based crowd simulation algorithms. *Computer Graphics Forum 36*, (2017), 337–348.

[DPA*09] Durupinar F., Pelechano N., Allbeck J., Güdükbay U., Badler N. I.: How the ocean personality model affects the perception of crowds. *IEEE Computer Graphics and Applications 31*, 3 (2009), 22–31.

[EKS*96] Ester M., Kriegel H.-P., Sander J., Xu X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining 96* (1996), 226–231.

[FMN18] Feliciani C., Murakami H., Nishinari K.: A universal function for capacity of bidirectional pedestrian streams: Filling the gaps in the literature. *PLoS One 13*, 12 (2018), e0208496.

[Fru71] Fruin J. J.: *Pedestrian Planning and Design*. Metropolitan Association of Urban Designers and Environmental Planners, New York, 1971.

[GAK62] Gnedenko B. V., Aleksandr I., Khinchin A.: *An Elementary Introduction to the Theory of Probability*, vol. *155*. Courier Corporation, 1962.

[GNL14] Golas A., Narain R., Lin M.: A continuum model for simulating crowd turbulence. In *ACM SIGGRAPH 2014 Talks* (New York, NY, USA, 2014), SIGGRAPH '14, Association for Computing Machinery.

[GVDBL*12] Guy S. J., Van Den Berg J., Liu W., Lau R., Lin M. C., Manocha D.: A statistical similarity measure for aggregate crowd dynamics. *ACM Transactions on Graphics 31*, 6 (2012), 11.

[Hal63] Hall E. T.: A system for the notation of proxemic behavior. *American Anthropologist 65*, 5 (1963), 1003–1026.

[HHB*21] Hu K., Haworth B., Berseth G., Pavlovic V., Faloutsos P., Kapadia M.: Heterogeneous crowd simulation using parametric reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics 29*, 4 (2021), 2036–2052.

[HJAA07] Helbing D., Johansson A., Al-Abideen H. Z.: Dynamics of crowd disasters: An empirical study. *Physical Review E 75*, 4 (2007), 046109.

[HPNM16] He L., Pan J., Narang S., Manocha D.: Dynamic group behaviors for interactive crowd simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2016), SCA '16, Eurographics Association, pp. 139–147.

[HXZW20] He F., Xiang Y., Zhao X., Wang H.: Informative scene decomposition for crowd analysis, comparison and simulation guidance. *ACM Transactions on Graphics 39*, 4 (August 2020), 50:1–50:13.

[JBT*20] Juliani A., Berges V.-P., Teng E., Cohen A., Harper J., Elion C., Goy C., Gao Y., Henry H., Mattar M., Lange D.: Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627* (2020).

[JCP*10] Ju E., Choi M. G., Park M., Lee J., Lee K. H., Takahashi S.: Morphable crowds. *ACM Transactions on Graphics 29*, 6 (December 2010), 140.

[JPCC14] Jordao K., Pettré J., Christie M., Cani M.-P.: Crowd sculpting: A space-time sculpting method for populating virtual environments. *Computer Graphics Forum 33* (2014), 351–360.

[JXM*10] Jiang H., Xu W., Mao T., Li C., Xia S., Wang Z.: Continuum crowd simulation in complex environments. *Computers 'I&' Graphics 34*, 5 (2010), 537–544.

[KBK16] Krontiris A., Bekris K. E., Kapadia M.: ACUMEN: Activity-centric crowd authoring using influence maps. In *Proceedings of the 29th International Conference on Computer Animation and Social Agents* (2016), pp. 61–69.

[KFS*16] Kapadia M., Frey S., Shoulson A., Sumner R. W., Gross M. H.: CANVAS: Computer-assisted narrative animation synthesis. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2016), pp. 199–209.

[KJBS24] Kini A., Jansche A., Bernthaler T., Schneider G.: Fastcar: Fast classification and regression multi-task learning via task consolidation for modelling a continuous property variable of object classes. *arXiv preprint arXiv:2403.17926* (2024).

[KKPC23] Kwiatkowski A., Kalogeiton V., Pettré J., Cani M.-P.: Reward function design for crowd simulation via reinforcement learning. In *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games* (2023), pp. 1–7.

[KLLT08] Kwon T., Lee K. H., Lee J., Takahashi S.: Group motion editing. *ACM Transactions on Graphics 27*, 3 (August 2008), 1–8.

[KO12] Karamouzas I., Overmars M.: Simulating and evaluating the local behavior of small pedestrian groups. *IEEE Transactions on Visualization and Computer Graphics 18*, 3 (2012), 394–406.

[KSHG18] Karamouzas I., Sohre N., Hu R., Guy S. J.: Crowd space: A predictive crowd analysis technique. *ACM Transactions on Graphics 37*, 6 (December 2018), 1–14.

[KSNG17] Karamouzas I., Sohre N., Narain R., Guy S. J.: Implicit crowds: Optimization integrator for robust crowd simulation. *ACM Transactions on Graphics 36*, 4 (July 2017), 1–13.

[LBC*22] Lemonari M., Blanco R., Charalambous P., Pelechano N., Avraamides M., Pettré J., Chrysanthou Y.: Authoring virtual crowds: A survey. *Computer Graphics Forum 41*, 2 (2022), 677–701.

[LCF05] Lai Y.-C., Chenney S., Fan S.: Group motion graphs. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 281–290.

[LCHL07] Lee K. H., Choi M. G., Hong Q., Lee J.: Group behavior from video: a data-driven approach to crowd simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 109–118.

[LCL07] Lerner A., Chrysanthou Y., Lischinski D.: Crowds by example. *Computer Graphics Forum 26* (2007), 655–664.

[LCSCO10] Lerner A., Chrysanthou Y., Shamir A., Cohen-Or D.: Context-dependent crowd evaluation. *In Computer Graphics Forum 29* (2010), 2197–2206.

[Lin91] Lin J.: Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory 37*, 1 (1991), 145–151.

[MBA22] Mathew C. T., Benes B., Aliaga D. G.: Sketching vocabulary for crowd motion. *Computer Graphics Forum 41* (2022), 119–130.

[MM17] Montana L. R., Maddock S.: Sketching for real-time control of crowd simulations. In *Proceedings of the Conference on Computer Graphics & Visual Computing* (2017), pp. 81–88.

[Moo09] Moore D. S.: *Introduction to the Practice of Statistics*. WH Freeman and Company, New York, 2009.

[PAB07] Pelechano N., Allbeck J. M., Badler N. I.: Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2007), SCA '07, Eurographics Association, pp. 99–108.

[PAC25] Panayiotou A., Aristidou A., Charalambous P.: CE-DRL: Simulating diverse crowds with example-driven deep reinforcement learning. *Computer Graphics Forum 44*, 2 (2025), e70015.

[PESVG09] Pellegrini S., Ess A., Schindler K., Van Gool L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision* (2009), IEEE, pp. 261–268.

[PKL*22] Panayiotou A., Kyriakou T., Lemonari M., Chrysanthou Y., Charalambous P.: CCP: Configurable crowd profiles. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery.

[PKM*18] Peng X. B., Kanazawa A., Malik J., Abbeel P., Levine S.: SFV: Reinforcement learning of physical skills from videos. *ACM Transactions on Graphics 37*, 6 (December 2018), 1–14.

[POO*09] Pettré J., Ondřej J., Olivier A.-H., Cretual A., Donikian S.: Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), pp. 189–198.

[PPD07] Paris S., Pettré J., Donikian S.: Pedestrian reactive navigation for crowd simulation: a predictive approach. *Computer Graphics Forum 26* (2007), 665–674.

[PvdBC*11] Patil S., van den Berg J., Curtis S., Lin M. C., Manocha D.: Directing crowd simulations using navigation fields. *IEEE Transactions on Visualization and Computer Graphics 17*, 2 (2011), 244–254.

[R*99] Reynolds C. W., et al.: Steering behaviors for autonomous characters. *Proceedings of the Game Developers Conference 1999* (1999), 763–782.

[RCB*17] Ren Z., Charalambous P., Bruneau J., Peng Q., Pettré J.: Group modeling: A unified velocity-based approach. *Computer Graphics Forum 36*, 8 (2017), 45–56.

[RPP21] Rogla O., Patow G. A., Pelechano N.: Procedural crowd generation for semantically augmented virtual cities. *Computers & Graphics 99* (2021), 83–99.

[TYK*09] Takahashi S., Yoshida K., Kwon T., Lee K. H., Lee J., Shin S. Y.: Spectral-based group formation control. *Computer Graphics Forum 28* (2009), 639–648.

[TZW24] Talukdar B., Zhang Y., Weiss T.: Learning crowd motion dynamics with crowds. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 7*, 1 (2024), 1–17.

[UCT04] Ulicny B., Ciechomski P. d. H., Thalmann D.: Crowdbrush: interactive authoring of real-time crowd scenes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 243–252.

[VBJK09] Van Basten B. J., Jansen S. E., Karamouzas I.: Exploiting motion capture to enhance avoidance behaviour in games. In *Motion in Games: Second International Workshop, MIG 2009, Zeist, The Netherlands, November 21-24, 2009. Proceedings 2* (2009), Springer, pp. 29–40.

[VdBLM08] Van den Berg J., Lin M., Manocha D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation* (2008), IEEE, pp. 1928–1935.

[vdBSGM11] van den Berg J., Snape J., Guy S. J., Manocha D.: Reciprocal collision avoidance with acceleration-velocity obstacles. In *2011 IEEE International Conference on Robotics and Automation* (2011), pp. 3475–3482.

[VTP21] Van Toll W., Pettré J.: Algorithms for microscopic crowd simulation: Advancements in the 2010s. *Computer Graphics Forum 40* (2021), 731–754.

[Wei93] Weidmann U.: *Transporttechnik der Fußgänger*. Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau (IVT), ETH Zürich, Zurich, Switzerland, 1993.

[WJGO*14] Wolinski D., J. Guy S., Olivier A.-H., Lin M., Manocha D., Pettré J.: Parameter estimation and comparative evaluation of crowd simulations. *Computer Graphics Forum 33* (2014), 303–312.

[WOO16] Wang H., Ondřej J., O'Sullivan C.: Trending paths: A new semantic-level metric for comparing simulated and real crowd data. *IEEE Transactions on Visualization and Computer Graphics 23*, 5 (2016), 1454–1464.

[XLL*19] Xu M., Li C., Lv P., Chen W., Deng Z., Zhou B., Manocha D.: Emotion-based crowd simulation model based on physical strength consumption for emergency scenarios. *arXiv preprint arXiv:1801.00216* (2019).

[XWZ*23] Xiang W., Wang H., Zhang Y., Yip M. K., Jin X.: Model-based crowd behaviours in human-solution space. *Computer Graphics Forum 42* (2023), e14919.

[YMPT09] Yersin B., Maïm J., Pettré J., Thalmann D.: Crowd patches: populating large-scale virtual environments for real-time applications. In *Proceedings of the Symposium on Interactive 3D Graphics and Games* (2009), pp. 207–214.

[ZCT18] Zhao M., Cai W., Turner S. J.: Clust: simulating realistic crowd behaviour by mining pattern from crowd videos. *Computer Graphics Forum 37* (2018), 184–201.

[ZY10] Zhang Y., Yeung D.-Y.: A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intel-*

ligence (Arlington, Virginia, USA, 2010), UAI'10, AUAI Press, pp. 733–742.

[ZYJL22] Zhang G., Yu Z., Jin D., Li Y.: Physics-infused machine learning for crowd simulation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), pp. 2439–2449.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information

Video S1