Whenever, Wherever: Towards Orchestrating Crowd Simulations with Spatio-Temporal Spawn Dynamics

Thomas Kreutz[1], Max Mühlhäuser[1] and Alejandro Sanchez Guinea[2]

***Abstract*— Realistic crowd simulations are essential for immersive virtual environments, relying on both individual behaviors (microscopic dynamics) and overall crowd patterns (macroscopic characteristics). While recent data-driven methods like deep reinforcement learning improve microscopic realism, they often overlook critical macroscopic features such as crowd density and flow, which are governed by spatio-temporal spawn dynamics, namely, when and where agents enter a scene. Traditional methods, like random spawn rates, stochastic processes, or fixed schedules, are not guaranteed to capture the underlying complexity or lack diversity and realism. To address this issue, we propose a novel approach called nTPP-GMM that models spatio-temporal spawn dynamics using Neural Temporal Point Processes (nTPPs) that are coupled with a spawn-conditional Gaussian Mixture Model (GMM) for agent spawn and goal positions. We evaluate our approach by orchestrating crowd simulations of three diverse real-world datasets with nTPP-GMM. Our experiments demonstrate the orchestration with nTPP-GMM leads to realistic simulations that reflect real-world crowd scenarios and allow crowd analysis.**

I. Introduction

Crowd simulations are a key ingredient for building immersive virtual environments with artificial agents or realistic training environments for mobile robots to be deployed in public places [1]. While mobile robots can benefit society by being used to deliver goods, guide visitors, or serve as mobile trashcans [2], [3], one critical challenge is their safe navigation through a crowded human environment [4]. In this context, realistic crowds are crucial so that mobile robots can leverage them as training environments to learn appropriate and safe navigation policies [1].

To simulate realistic crowds, recent advances in data-driven crowd simulation methods that are based on deep reinforcement learning (e.g., [5], [6]), imitation learning (e.g., [7], [1]), or their combination (e.g., [8]), greatly outperform classical methods (e.g., the Social Force Model [9], or ORCA [10]) in terms of microscopic crowd realism by learning to imitate a large variety of realistic behaviors, such as goal-seeking, collision avoidance, or grouping.

While existing methods for data-driven crowd simulation focus strongly on enhancing the microscopic realism of individual agent behaviors, learning macroscopic features, such as agent spawn dynamics, remain underexplored. Spatio-temporal spawn dynamic of agents, including (i) *where* agents enter and exit a scene, and (ii) *when* agents appear in a scene, directly influence crowd density and flow [11]. In this work, we focus on the temporal spawning of agents, which is traditionally handled by fixed or random spawn rates [12], [13], [14], [15], [6], learned with traditional stochastic process [16], [17], [11], or follows pre-defined schedules [18], [19], [8]. Random or fixed spawn rates as well as fixed-schedules lack diversity and realism, while traditional stochastic processes, such as the Poisson process, are limited in their ability to capture complex temporal inter-dependencies [20]. Furthermore, while the temporal spawn dynamics of agents is an important macroscopic characteristic of realistic crowds, it has not received the necessary attention from previous works to the point of overlooking it when considering crowd simulation components [21].

In this paper, we propose an approach to learn orchestrating the initialization of agents in crowd simulations in space and time. More specifically, our goal is to learn spatio-temporal spawn dynamics from real-world data, i.e., *when* and *where* agents appear in a scene and *where* they will go. Our approach learns spawn timings based on neural Temporal Point Processes (nTPPs) that are combined with a spawn-conditional Gaussian Mixture Model (GMM) for spawns and goals of each agent to an nTPP-GMM. The nTPP-GMM jointly models a spatio-temporal spawn and goal distribution, which can be used to orchestrate a crowd simulation (i.e., to dynamically control the initialization and flow of agents by sampling from the spatio-temporal distribution).

Our approach is evaluated by orchestrating a crowd simulation framework that uses imitation learning to learn and model individual agent behavior. We evaluate our method on three real-world datasets, including Grand Central Station [16], [22], Edinburgh Forum [23], and ETH University [24]. Our results demonstrate that the orchestration of a crowd simulation with nTPP-GMM can effectively replicate realistic crowd scenarios and allows us to implicitly analyze crowd behaviors, such as specific situations or crowd flows, solely by simulation. Our contributions are:

- nTPP-GMM, a novel approach for learning spatio-temporal spawn dynamics from real-world data.
- Orchestration of simulated crowds: Using nTPP-GMM, we can orchestrate crowd simulations, which leads to realistic crowd flows and crowd scenarios.

II. Related Work

A. Temporal Pedestrian Initialization

When to initialize new agents in a crowd simulation is a crucial macroscopic feature, which directly influences crowd density and implicitly leads to common crowd flows [11].

[1]Telecooperation Lab at the Technical University Darmstadt, Germany, {kreutz, max@informatik}.tu-darmstadt.de,
[2]NTT DATA, Luxembourg, alejandro.guinea@global.ntt

However, learning the spawn dynamics of agents has not received much attention by prior works and is not considered an important aspect for authoring crowds [21]. Common agent spawning techniques include a fixed spawn rate [12], [13], [14] or a random distribution coupled with conditions [15], spawn schedules taken from real data [18], [19], [25], or stochastic processes [16], [17], [11]. While classical stochastic processes, such as the Poisson process as used in [16], [26] or HDP in [11], can capture relatively simple patterns, these approaches can not learn complex dependencies between event occurrences [20]. To address this gap for crowd simulations, we propose learning the stochastic spawn process of agents with neural Temporal Point Processes (nTPPs). Unlike classical (non-neural) TPPs like the Poisson process, nTPPs can learn complex (spatio-)temporal dependencies to generate a large variety of patterns such as global trends, burstiness, or repeating sub-sequences [20], which are essential characteristics needed to realistically simulate the spawn dynamics of agents in crowd simulations.

B. Understanding Crowds with Data-Driven Simulation

Understanding crowds involves capturing both macroscopic aspects, such as agent spawns and destinations, and microscopic behaviors, like common agent paths. Previous works, such as those in [16], [17], [11], use classical statistical methods, like mixture models, Poisson processes, and hierarchical Dirichlet processes (HDP), to model pedestrian flows and patterns. These methods require additional methods for simulation, such as extracting path patterns, incorporating extra collision avoidance measures [17], or relying on separate crowd simulation frameworks that require specific parameters as an input to control each agent [11]. Although effective, prior works are constrained by their specific assumptions about the data distribution. In contrast, our paper leverages deep learning-based methods for crowd simulation which eliminates any assumptions about the data distribution, while allowing implicitly analysis of the underlying crowd data through simulation.

C. Learning Agent Policies for Crowd Simulation

Crowd simulation has progressed from more traditional rule-based approaches such as Reynolds' flocking algorithm [27], Helbing's Social Force Model [9], or Van den Berg's Optimal Reciprocal Collision Avoidance (ORCA) [10] to data-driven approaches that improve the realism of individual agent behavior in a crowd. Recent data-driven methods focus strongly on advancing the realism and controllability of crowd simulations. These methods span imitation learning using on-policy (deep RL, IRL) [5], [6], [7], [8], [1], off-policy methods (e.g., BC) [7], but also generative methods for multi-agent simulation that are mainly driven by trajectory prediction over a limited time horizon [28], [29], [30], [31], [32], [33], [34], [35].

III. Approach

A. Problem Statement and Overview

Our work addresses the challenge of learning spatio-temporal spawn dynamics from real-world crowd trajectory data. Specifically, we aim to determine *when* and *where* agents spawn in a scene, as well as *where* their goal positions are. Figure 1 outlines our approach. First, we cluster the start and end positions of each trajectory to estimate common spawn and goal areas (S^* and E^*). We also compute a co-occurrence distribution between these clusters, leading to a spawn-conditional Gaussian Mixture Model (GMM), which jointly defines *where* agents spawn and *where* their goal is. To model *when* agents spawn, we combine this GMM with a neural Temporal Point Process (nTPP), forming an nTPP-GMM that learns the timing of spawns for each spawn area. This allows us to generate a sequence of spawn times, where each time generates a new agent with corresponding spawn and goal positions. Finally, a set of nTPP-GMMs orchestrates a crowd simulation by generating agents over time while the agent policy of the simulation moves them toward their goals.

B. Where: Spawn and Goal Areas of Agents

Agents that traverse public places usually have common spawn and goal areas [16], which are fundamental for a realistic crowd simulation. We are given a dataset of trajectories $D := \{tr_1, ...tr_N\}$ obtained from, e.g., object tracks in a real-world scene, where each trajectory $tr_i \in D$ is a sequence of $(x, y) \in \mathbb{R}^2$ positions of arbitrary length. Our goal is to obtain two sets of clusters for common spawn and goal areas of D. To this end, we first split D into a set $S := \{s_1, ..., s_N\} \subseteq \mathbb{R}^2$ and $E := \{e_1, ..., e_N\} \subseteq \mathbb{R}^2$, where each $s_i \in S, e_i \in E$ are the respective start and end positions of a trajectory tr_i. Next, we apply a clustering algorithm (DBSCAN [36]) on S and E separately, to obtain a set of spawn areas S^* and goal areas E^*. For each area in S^*, E^*, we assume them to be normally distributed and compute the respective parameters μ_k, σ_k so that we can sample positions from each area.

To mimic the distribution of agents that move from a spawn area $s \in S^*$ to some goal area $e \in E^*$, we compute the frequency of co-occurences for all $(s, e) \in S^* \times E^*$ with a function $freq : S^* \times E^* \to \mathbb{N}$. Afterward, given a spawn area s, we obtain a probability distribution $P(E|s)$ over the possible goal areas of an agent by computing the relative frequency of each co-occurrence[1]. For any $(s, e) \in S^* \times E^*$, $P(E^* = e|s)$ is defined as follows:

$$P(E^* = e|s) = \frac{freq(s, e)}{\sum_{e' \in E^*} freq(s, e')} \tag{1}$$

Under $P(E^* = e|s)$, for each s, there is a set of possible goal areas $\mathcal{E}(s) = \{e \mid P(E^* = e|s) > 0 \wedge e \in E^*\}$, where s and e co-occur at least once (otherwise $P(e|s) = 0$). As a result, a goal area $e \sim P(E^*|s)$ can be sampled with weighted uniform sampling for any $s \in S^*$. Given a spawn area $s \in S^*$, the sampling procedure for an agent's spawn position x_s and a goal area e with goal position x_e is:

$$x_s \sim \mathcal{N}(\mu_s, \sigma_s),\ x_e \sim \mathcal{N}(\mu_e, \sigma_e) \text{ with } e \sim P(E^*|s) \tag{2}$$

where $\mu_s, \sigma_s, \mu_e, \sigma_e$ are spawn and goal area parameters.

[1]This is a simplifying assumption since we assume no information about the real intentions to be available.

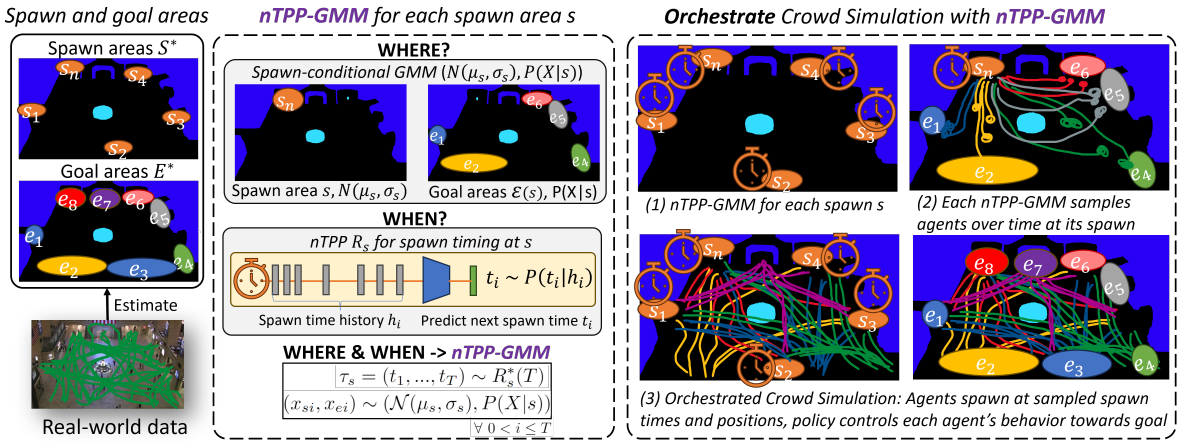


Fig. 1: We propose orchestrating a crowd simulation with spatio-temporal spawn dynamics learned by an nTPP-GMM from real-world data. First, spawn and goal areas S^*, E^* are estimated from real-world data defining *where* they spawn and *where* their goal is. Second, for each spawn $s \in S^*$, we train an nTPP-GMM to learn *when* agents will spawn, which allows sampling a spawn time sequence τ_s for simulation. Each spawn time $t_i \in \tau_s$ yields a new agent with sampled spawn and goal positions (x_{si}, x_{ei}). Third, to orchestrate a crowd simulation, the nTPP-GMM of each spawn independently generates agents (1,2), and a policy controls each agent while they are present in the scene to reach their goal (3).

Any goal area $e \in E^*$ with $freq(s, e) > 0$ is a Gaussian with μ_e, σ_e and $e \in \mathcal{E}(s)$. Let $X = \mathbb{R}^2$. In this case, the distribution over possible goal positions $P(X = x_e|s)$ conditioned on a spawn area s (as described in Equation 2) can more generally be defined as a GMM, which defines the following marginal probability of a goal position x_e:

$$P(X = x_e \mid s) = \sum_{k \in \mathcal{E}(s)} \mathcal{N}(X = x_k \mid \mu_k, \sigma_k) * \pi_k^s \quad (3)$$

where the number of components depends on s and the set of co-occurring goal areas $\mathcal{E}(s)$, π_k^s are the mixture weights, and μ_k, σ_k are the mixture components, with the interpretation that $\pi_k^s = P(k|s)$. With Equation 3, we redefine the sampling procedure in Equation 2 as a *spawn-conditional* GMM:

$$(x_s, x_e) \sim (\mathcal{N}(\mu_s, \sigma_s), P(X|s)) \quad (4)$$

C. When: Learning spawn timings of agents

We assume that spawn dynamics of all different spawn areas $s_i, s_j \in S^*, i \neq j$ are statistically independent of each other. This assumption allows modeling each spawn area as an independent spawn process. In this context, we propose learning the spawn process at a spawn area $s \in S^*$ with a neural Temporal Point Process (nTPP) from real data. Neural TPPs can learn different patterns, such as global trends, burstiness, or repeating subsequences [20], which are essential characteristics of agent spawn timings in a crowd.

We build on the implementation[2] of an nTPP similar to [37], which models the nTPP as a recurrent neural network (RNN), which we denote as R. Given a sequence of spawn timings $\tau = \{t_1, t_2, \ldots, t_N\}$, the model predicts the probability distribution of inter-event times $\Delta t_i = t_{i+1} - t_i$. To this end, R encodes past inter-event times into a hidden state h_i, which parameterizes the next inter-event time distribution $f(\Delta t_i \mid h_i)$. $f(\Delta t_i \mid h_i)$ is parameterized as a Weibull distribution, and it represents the likelihood of observing the next inter-event time Δt_i given observed past inter-event times $\{\Delta t_{i-1}, ...\}$. To train R on real data, we split the full sequence τ into overlapping sliding windows of length w, where each subsequence starts at t_k and ends at t_{k+w}. Let $S(T_{end} - t_{k+w})$ be a survival function that accounts for the probability that no events occur after the last observed event t_{k+w} in the subsequence, where T_{end} is the maximum time in the window. The training objective of R is to minimize the negative log-likelihood (NLL) over all sliding windows, which for each subsequence leads to the following loss:

$$\mathcal{L}_{\text{NLL}} = -\left(\sum_{i=k}^{k+w-1} \log f(\Delta t_i \mid h_i) + \log S(T_{end} - t_{k+w}) \right) \quad (5)$$

where $f(\Delta t_i \mid h_i)$ is the predicted PDF of inter-event times, and $S(T_{end} - t_{k+w})$ represents the survival function.

D. Where+When: nTPP-GMM

After training the nTPP, we can autoregressively sample agent spawn sequences up to time T from each trained R_s, which we denote as $R_s^*(T)$. The sampling is unconditional and starts with a random initial hidden state. By defining each individual spawn t_i as a random variable and including the distribution of agent spawn and goal positions from Equation 4, the overall spatio-temporal spawn dynamics of a scene lead to the definition of a *neural Temporal Point Process Gaussian Mixture Model* (nTPP-GMM):

$$\tau_s = (t_1, ..., t_T) \sim R_s^*(T) \quad (6)$$

$$(x_{si}, x_{ei}) \sim (\mathcal{N}(\mu_s, \sigma_s), P(X|s)),\ \forall\ 0 < i \leq T \quad (7)$$

where (i) a sequence of spawn timings $\tau_s = (t_1, ..., t_T)$ is sampled from the nTPP $R_s^*(T)$, and (ii) for each spawn time $t_i \in \tau_s$, a spawn, goal position pair (x_{si}, x_{ei}) is sampled from its spawn-conditional GMM defined in Equation 4.

[2]https://shchur.github.io/blog/2021/tpp2-neural-tpps/

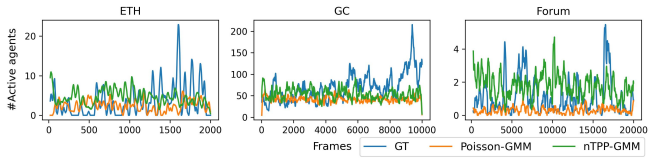


Fig. 2: Number of agents in the scene (y-axis) at each frame. Our approach approximates the real data better.

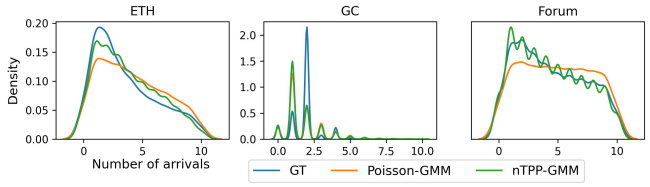


Fig. 3: Inter-spawn times within 10 frames

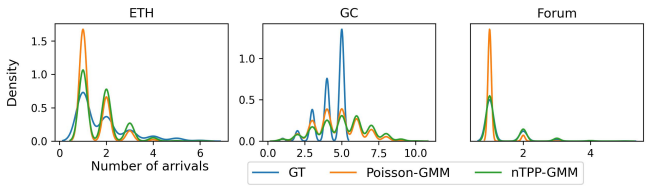
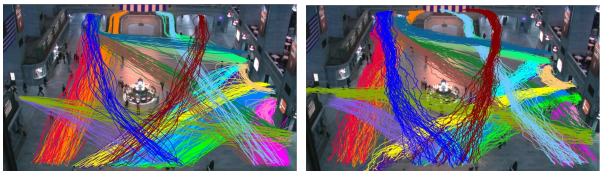


Fig. 4: Number of spawns within 10 frames



(a) Ground Truth (b) Behavior Cloning

Fig. 5: Trajectories generated with a behaviour cloning policy (right) effectively approximate the underlying trajectory distribution of the ground truth dataset (left). Each spawn-destination pair is assigned a unique color.

E. Orchestrating Crowds with nTPP-GMM

To orchestrate a crowd simulation, nTPP-GMM is integrated into the initialization process of agents. For each spawn $s \in S^*$, the nTPP-GMM samples a set of agents with respective spawn timing, spawn, and goal position according to Equation 6 and 7. With a global clock keeping track of time, agents are spawned at their designated spawn position whenever their spawn time is reached, while the policy of the crowd simulation controls each agent. Finally, agents are removed from the simulation upon reaching the goal position.

IV. Experiments

To evaluate our approach, we implemented[3] a Gym-based crowd simulation environment [38] that can leverage real-world datasets for training of agent policies, and integrate the nTPP-GMM into it. We compare nTPP-GMM against the ground truth crowd data (GT) and a Poisson process used in [16], [17] as a baseline[4] (referred to as Poisson-GMM) for learning the spatio-temporal spawn dynamics in three real world scenes. Afterward, we show that our crowd simulation framework combined with nTPP-GMM can realistically simulate real-world scenarios and can be used for crowd analysis.

A. Datasets

We evaluate our crowd orchestration approach in 3 different real-world scenes: Grand Central Station (GC) [16], [22], Edinburgh Forum (Forum) [23], and ETH University (ETH) [24]. In Forum, several full days of recordings are available and we chose the recording on July 14, which includes a good amount of agents throughout the whole day as the number of agents in Forum is usually very low. The number of frames and agents in each scene varies significantly between each dataset, which results in differences in terms of agent density, variety of behavior, number of spawns and destinations as well as the spawn dynamics. There are 1929 frames with 258 agents in ETH, 11999 frames with 12255 agents in GC, and 58480 frames with 1331 agents in Forum July 14.

B. Implementation Details

We use DBSCAN [36] to obtain start and goal areas S^*, E^*. The hyperparameters for DBSCAN are optimized for each test scene with $\epsilon = 0.2$ and $minsamples = 20$ for GC, $\epsilon = 2$ and $minsamples = 5$ for Forum, and $\epsilon = 0.8$ and $minsamples = 3$ for ETH. The agent policy is an MLP with two hidden layers of 32 units for all datasets, which matches the policy network used in [39]. We use behavior cloning to train the agent policy for 1000 epochs in our Gym environment with a learning rate of 1e-4 and the Adam optimizer [40] using the imitation learning library introduced in [41]. For the nTPP, we follow the implementation[5] based on [37]. The nTPP consists of a GRU with 32 hidden units and an MLP head with 32 units, which is trained for a maximum number of 500 epochs with a learning rate of 1e-4 and the Adam optimizer [40].

C. Number of Agents over Time

We compare the number of agents over time of nTPP-GMM against Poisson-GMM and GT in Figure 2. Compared to the Poisson-GMM, nTPP-GMM achieves more variety in behavior that is more similar to the ground truth. Our approach can capture different spiking and bursting behavior, which leads to an overall more realistic number of agents and spawn patterns when comparing it to the ground truth. We find that a current limitation of our approach is that strong bursts are not learned in all three datasets (e.g., only in Forum), which we hypothesize is due to the limited amount of training data in GC and ETH compared to Forum.

[3]Our code and implementations are available on github: `https://github.com/thkreutz/crowdorchestrationsim`

[4]To the best of our knowledge, other learning-based methods that learn spawn dynamics from real-world data for crowd simulations not exist in the literature or don't share code (e.g.,[11]).

[5]https://shchur.github.io/blog/2021/tpp2-neural-tpps/

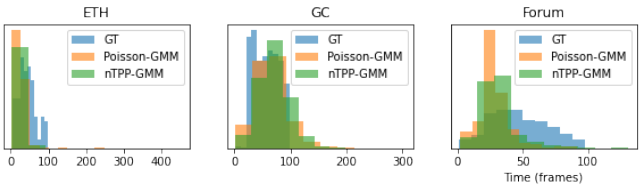


Fig. 6: Time that agents controlled by our policy spend in the scene in comparison to the real crowd.

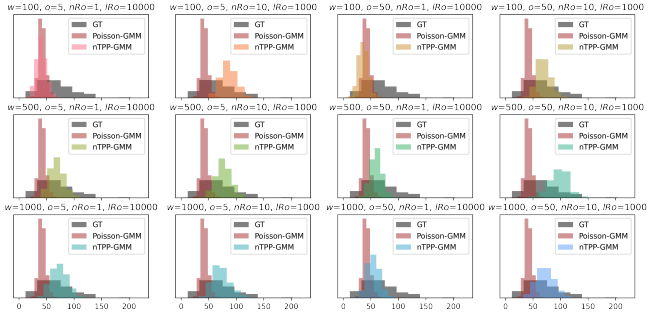


Fig. 7: In GC, we compare 5 samples of different hyperparameter combinations for nTPP-GMM against 10000 GT frames and Poisson-GMM to test the influence of window size (w), sliding window overlap (o), number (nRo) of rollouts, and length (lRo) of rollouts. We show the distribution of the number of agents (x-axis) measured at any point in time.

Hyperparameter	Values
Window Size (w)	$[100, 500, 1000]$
Sliding Window Overlap (o)	$[5, 50]$
Number Rollouts (nRo)	$[1, 10]$
Length Rollout (lRo)	$[1000, 10000]$
Total Length	10000

TABLE I: Summary of hyperparameter evaluation: Window size (w), sliding window overlap (o) during training window creation, number (nRo) and length (lRo) of rollouts.

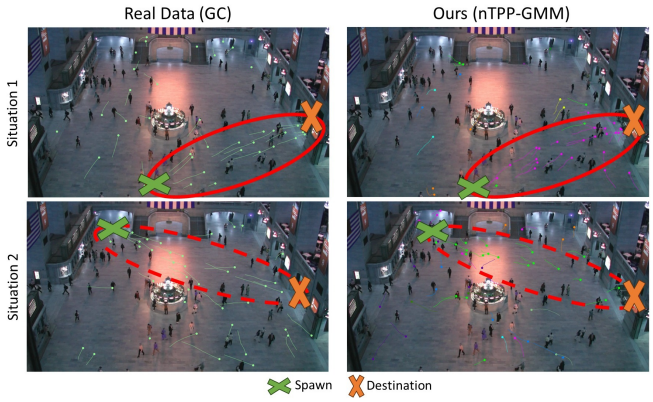


Fig. 8: Comparison of two crowd flow situations present in the real data in GC that are reproduced by nTPP-GMM.

D. Inter Spawn Times of Agents

Figure 3 and Figure 4 compare the overall spawn distribution in the scene by comparing inter-spawn times and number of spawns in short 10 frame durations. For the inter-spawn times in Figure 3, we can see that nTPP-GMM captures an inter-spawn time distribution close to the ground truth that is more accurate than the Poisson-GMM in ETH and Forum. In GC, both nTPP-GMM and Poisson-GMM have a slightly shorter inter-spawn time than the ground truth. For the number of spawns in Figure 4, nTPP-GMM outperforms Poisson-GMM in terms of realism in both ETH and Forum. In GC, there is a difference in burst behavior given that both Poisson-GMM and nTPP-GMM do not replicate the spike at 5 or 4 agents. The spawn behavior in GC is much more complex compared to ETH and Forum, so some spawns might be more difficult to imitate than others. The independence assumption between spawns can also limit the realism in some spawns, which in the real world may have a dependency.

E. Ablation Study on Hyperparameters

We evaluate the impact of hyperparameters on nTPP-GMM in GC, which are summarized in Table I. We vary between training the nTPP-GMM on sliding windows of short subsequences (100 frames), mid-length subsequences (500 frames), and long subsequences (1000 frames). It is trained on all frames of the GC dataset. We also vary the overlap between the sliding windows during training as well as the length of the final autoregressive rollout. The final rollout is obtained either by doing a rollout with the nTPP-GMM of length 10000 or 10 rollouts of length 1000 that are concatenated to obtain 10000 timesteps. Each rollout strategy is repeated 5 times to get 5 spawn sequence samples for each hyperparameter combination.

With a short window size of 100, we approximate a Poisson process when sampling rollouts of 10000 frames. Overall, a larger window size of 500 or 1000 is desirable, while a rollout of 1000 or 10000 seems to lead to more similar agent distributions over time.

F. Verifying Realism of Crowd Simulation Agent Policy

When integrating nTPP-GMM into a crowd simulation framework, the simulation requires a policy for agents that effectively mimics how long they usually stay inside the scene so that the number of agents remains realistic over time. Otherwise, if agents stay too long, nTPP-GMM will simply continue spawning more agents and the crowd density can (theoretically) grow infinitely. For a proof of concept, we chose behaviour cloning (BC) as an imitation learning policy, which learns these timings as well as a distribution over common paths in the data. The time agents usually spend in the scene is shown in Figure 6, where we can see that the distribution is close to the real data. That our policy learns to mimic the distribution over trajectories that connect the learned spawn and goal positions is shown in a simple experiment in Figure 5, where we show several rollouts of the BC policy in GC and compare generated trajectories between common spawn and goal positions (right) with the real trajectories of the dataset (left).

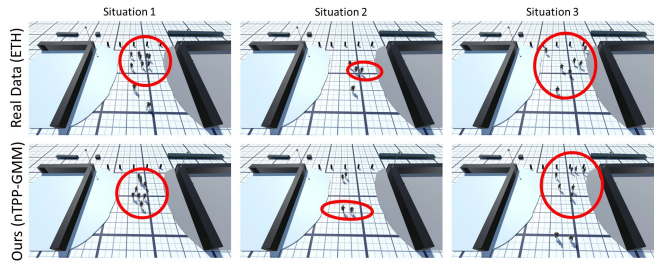


Fig. 9: Comparison of three scenarios in ETH that differ in crowd density and group sizes reproduced by nTPP-GMM.

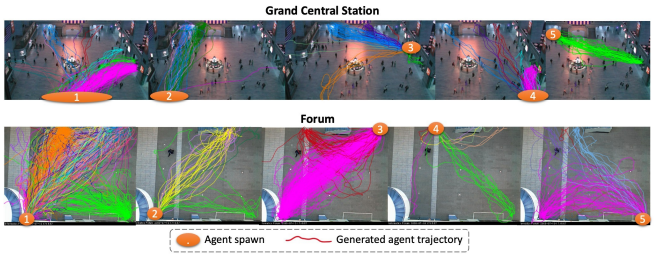


Fig. 10: Running the simulation in GC and Forum and separating by spawns and destinations implicitly leads to common crowd flows. The crowd density and destination distribution are controlled by the learned nTPP-GMM, while each individual agent is controlled by the policy.

G. Generation of realistic scenarios

We show that our approach can reproduce real spatio-temporal spawn dynamics that govern different crowd flows without human interference. We simulate crowds for ETH and GC and qualitatively compare it to different scenarios found in the ground truth.

We show that nTPP-GMM can reproduce realistic strong crowd flows of GC in Figure 8. Our approach learns two distinct temporal spawn patterns where many agents walk from either the bottom or the top spawn to the destination on the right-hand side. We further show three different situations that nTPP-GMM can learn to reproduce in ETH in Figure 9. Agents often walk in groups in ETH. In the first column (Situation 1), we show that our approach replicates a large group of pedestrians. In the second column (Situation 2), a smaller number of agents is generated. Finally, in the third column (Situation 3), many pedestrians who do not belong to a single big group are simulated.

H. Application: Implicit Crowd Analysis Through Simulation

In this experiment, we show that combining nTPP-GMM with imitation learning allows the analysis of the common paths that agents usually take to reach their goal *without explicitly clustering the trajectories of the real dataset*. Instead, we implicitly generate the trajectories of each spawned agent. First, in comparison to the real dataset, we can see in Figure 5 that we learn the overall distribution of the trajectories in GC. Second, our approach allows us to analyze a crowd's common crowd flows from each spawn. We visualize the respective analytical results of a simulation in GC and Forum in Figure 10, where we can see a subset of common crowd flows of both datasets without clustering the trajectories, but implicitly generating them using nTPP-GMM and simulating the agents with imitation learning.

V. Discussion and Future Work

With nTPP-GMM, we propose a novel approach for learning the spatio-temporal spawn dynamics of crowd scenes. Our experiments demonstrate its potential for realistic simulations and crowd analysis. Furthermore, we want to emphasize that our approach is flexible and agnostic to the specific methods used for each component. For example, agent behavior can be modeled using any kind of imitation learning algorithm or agent policy. Similarly, while we propose using nTPPs to model the spawn dynamics or DBSCAN to identify the spawn and goal areas, our framework is not limited to a particular nTPP or clustering model.

Future work should focus on integrating nTPP-GMM into existing crowd simulation frameworks or robot training environments for social navigation. Since nTPP-GMM is independent of the agent policy, it can easily enhance the macroscopic realism in existing simulation systems. Additionally, while our independence assumption leads to the use of separate nTPPs for each spawn, it also leads to the limitation that nTPP-GMM can not explicitly model precise group spawn dynamics. To overcome this limitation, future improvements should explore marked or spatio-temporal nTPPs, which can be trained end-to-end on aspects such as spawns, goals, and the spawn timings in a unified way, which likely captures more complex interdependencies and leads to improved realism.

VI. Conclusions

We demonstrate that realistic crowd simulations can be orchestrated by spatio-temporal spawn dynamics, which include *where* agents spawn, *where* they will go, and *when* they appear. More specifically, we propose nTPP-GMM to learn spatio-temporal spawn dynamics from real-world data, which allows to orchestrate crowd simulations where arbitrary policies can control agents. With nTPP-GMM, we propose a new learnable layer for realistic crowd simulations, which can be seamlessly integrated into existing crowd simulation frameworks. Without the need for domain expert knowledge or any human interference, orchestrating a crowd with nTPP-GMM leads to the replication of real crowd scenarios, including diverse crowd flows and crowd densities, which can benefit several applications in urban planning, social robot navigation, and public safety in the future. We believe that this work paves the way for future research in realistic crowd simulations, and we hope that we can inspire more research focusing on the orchestration of crowds based on their spatio-temporal spawn dynamics.

Acknowledgment

This work has been partially funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY centre and by the Federal Ministry of Education and Research (BMBF) grant 01|S17050.

REFERENCES

[1] B. Ling, Y. Lyu, D. Li, G. Gao, Y. Shi, X. Xu, and W. Wu, "Socialgail: Faithful crowd simulation for social robot navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 873–16 880.

[2] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *Ieee Access*, vol. 5, pp. 26 754–26 773, 2017.

[3] D. Mukherjee, K. Gupta, L. H. Chang, and H. Najjaran, "A survey of robot learning strategies for human-robot collaboration in industrial settings," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102231, 2022.

[4] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[5] J. Lee, J. Won, and J. Lee, "Crowd simulation by deep reinforcement learning," in *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2018, pp. 1–7.

[6] A. Panayiotou, T. Kyriakou, M. Lemonari, Y. Chrysanthou, and P. Charalambous, "Ccp: Configurable crowd profiles," in *ACM SIGGRAPH 2022 conference proceedings*, 2022, pp. 1–10.

[7] G. Qiao, H. Zhou, M. Kapadia, S. Yoon, and V. Pavlovic, "Scenario generalization of data-driven imitation models in crowd simulation," in *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2019, pp. 1–11.

[8] P. Charalambous, J. Pettre, V. Vassiliades, Y. Chrysanthou, and N. Pelechano, "Greil-crowds: Crowd simulation with deep reinforcement learning and examples," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–15, 2023.

[9] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[10] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*. Springer, 2011, pp. 3–19.

[11] F. He, Y. Xiang, X. Zhao, and H. Wang, "Informative scene decomposition for crowd analysis, comparison and simulation guidance," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 50–1, 2020.

[12] P. Dickinson, K. Gerling, K. Hicks, J. Murray, J. Shearer, and J. Greenwood, "Virtual reality crowd simulation: effects of agent density on user experience and behaviour," *Virtual Reality*, vol. 23, pp. 19–32, 2019.

[13] T. G. V. de Mello, M. S. H. da Silva, G. F. Silva, and S. R. Musse, "Exploring crowd dynamics: Simulating structured behaviors through crowd simulation models," *arXiv preprint arXiv:2312.06549*, 2023.

[14] I. Karamouzas, J. Bakker, and M. H. Overmars, "Density constraints for crowd simulation," in *2009 International IEEE Consumer Electronics Society's Games Innovations Conference*. IEEE, 2009, pp. 160–168.

[15] N. Kraayenbrink, J. Kessing, T. Tutenel, G. de Haan, and R. Bidarra, "Semantic crowds," *Entertainment Computing*, vol. 5, no. 4, pp. 297–312, 2014.

[16] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2871–2878.

[17] J. Zhong, W. Cai, L. Luo, and H. Yin, "Learning behavior patterns from video: A data-driven framework for agent-based crowd modeling," in *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, 2015, pp. 801–809.

[18] B. Liu, H. Liu, H. Zhang, and X. Qin, "A social force evacuation model driven by video data," *Simulation Modelling Practice and Theory*, vol. 84, pp. 190–203, 2018.

[19] L. Bein Fahlander and M. Mossberg, "Simulating people flow at an airport: Case study: Arlanda airport," 2020.

[20] O. Shchur, A. C. Türkmen, T. Januschowski, and S. Günnemann, "Neural temporal point processes: A review," *arXiv preprint arXiv:2104.03528*, 2021.

[21] M. Lemonari, R. Blanco, P. Charalambous, N. Pelechano, M. Avraamides, J. Pettré, and Y. Chrysanthou, "Authoring virtual crowds: A survey," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 677–701.

[22] S. Yi, H. Li, and X. Wang, "Understanding pedestrian behaviors from stationary crowd groups," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3488–3496.

[23] B. Majecka, "Statistical models of pedestrian behaviour in the forum," *Master's thesis, School of Informatics, University of Edinburgh*, 2009.

[24] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*. Springer, 2010, pp. 452–465.

[25] R. Nishida, M. Onishi, and K. Hashimoto, "Crowd simulation incorporating a route choice model and similarity evaluation using real large-scale data," *arXiv preprint arXiv:2302.10421*, 2023.

[26] B. Zhou, X. Tang, and X. Wang, "Learning collective crowd behaviors with dynamic pedestrian-agents," *International Journal of Computer Vision*, vol. 111, pp. 50–68, 2015.

[27] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.

[28] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1349–1358.

[29] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 233–15 242.

[31] J. Yue, D. Manocha, and H. Wang, "Human trajectory prediction via neural social physics," in *European Conference on Computer Vision*. Springer, 2022, pp. 376–394.

[32] D. Rempe, J. Philion, L. J. Guibas, S. Fidler, and O. Litany, "Generating useful accident-prone driving scenarios via a learned traffic prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 305–17 315.

[33] L. Shi, L. Wang, C. Long, S. Zhou, F. Zheng, N. Zheng, and G. Hua, "Social interpretable tree for pedestrian trajectory prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2235–2243.

[34] D. Rempe, Z. Luo, X. Bin Peng, Y. Yuan, K. Kitani, K. Kreis, S. Fidler, and O. Litany, "Trace and pace: Controllable pedestrian animation via guided trajectory diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 756–13 766.

[35] L. F. Chiara, P. Coscia, S. Das, S. Calderara, R. Cucchiara, and L. Ballan, "Goal-driven self-attentive recurrent networks for trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2518–2527.

[36] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[37] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1555–1564.

[38] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," Mar. 2023. [Online]. Available: https://zenodo.org/record/8127025

[39] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *arXiv preprint arXiv:1710.11248*, 2017.

[40] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] A. Gleave, M. Taufeeque, J. Rocamonde, E. Jenner, S. H. Wang, S. Toyer, M. Ernestus, N. Belrose, S. Emmons, and S. Russell, "imitation: Clean imitation learning implementations," arXiv:2211.11972v1 [cs.LG], 2022. [Online]. Available: https://arxiv.org/abs/2211.11972