Model-based Crowd Behaviors in Human-solution Space

Wei Xiang[1] and He Wang[2] and Yuqing Zhang[1] and Milo K. Yip[3] and Xiaogang Jin[†1]

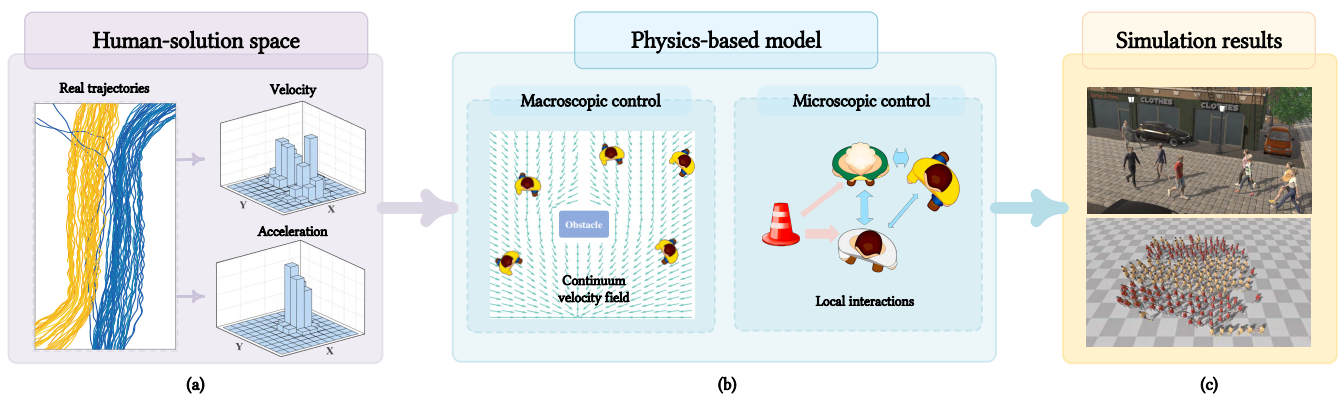[1]State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
[2]School of Computing, University of Leeds, Leeds, UK
[3]MoreFun Studios, Tencent, Shenzhen, China

Figure 1: *Overview of our simulation framework. Taking the real data as input (a), our method employs a physics-inspired energy model which describes path planning with a multi-granularity control (b), and solves the minimizer in human-solution space by leveraging an acceleration-aware data-driven scheme. As a result, we can generate different crowd behaviors in diverse scenes (c).*

Abstract
Realistic crowd simulation has been pursued for decades, but it still necessitates tedious human labor and a lot of trial and error. The majority of currently used crowd modeling is either empirical (model-based) or data-driven (model-free). Model-based methods cannot fit observed data precisely, whereas model-free methods are limited by the availability/quality of data and are uninterpretable. In this paper, we aim at taking advantage of both model-based and data-driven approaches. In order to accomplish this, we propose a new simulation framework built on a physics-based model that is designed to be data-friendly. Both the general prior knowledge about crowds encoded by the physics-based model and the specific real-world crowd data at hand jointly influence the system dynamics. With a multi-granularity physics-based model, the framework combines microscopic and macroscopic motion control. Each simulation step is formulated as an energy optimization problem, where the minimizer is the desired crowd behavior. In contrast to traditional optimization-based methods which seek the theoretical minimizer, we designed an acceleration-aware data-driven scheme to compute the minimizer from real-world data in order to achieve higher realism by parameterizing both velocity and acceleration. Experiments demonstrate that our method can produce crowd animations that are more realistically behaved in a variety of scales and scenarios when compared to the earlier methods.

CCS Concepts
• *Computing methodologies* → *Physical simulation;*

1. Introduction

Human crowds are ubiquitous and have attracted wide research interests, among which replicating naturalistic crowd behaviors has been an important task in computer animation, as well as psychology, transportation research, architectural design, safety and secu-

---

† Xiaogang Jin is the corresponding author. E-mail: jin@cad.zju.edu.cn

rity, etc. However, automated and realistic crowd simulation is still challenging after decades of research and practice.

Existing crowd simulation methods can be conceptually divided into two categories: empirical and data-driven. Empirical methods abstract observed crowd behaviors into explicit mathematical models and deterministic systems. They can model various crowd behaviors at different levels of granularity, e.g. macroscopic pedestrian flows [TCP06; NGCL09], microscopic local interactions [Rey87; HM95; KSG14], and mesoscopic combinations of local behaviors and global navigation [PCQ11]. We refer to these methods as *model-based*, since they are explicitly formulated by observed human behaviors using explicable mathematical terms. However, as pedestrian interactions are complex and subtle, model-based approaches tend to generate crowd motions with limited plausibility and diversity. This is because these models are usually based on simplified and idealized hypotheses with deterministic nature.

On the other hand, data-driven crowd simulation methods tend to rely on real data, which we refer to as *model-free* methods. These methods attempt to establish a model that has sufficient learning capacity so that they can fit complex crowd data in a black-box manner. Most early model-free methods are based on simple strategies: such as rule-based trajectory generation by connecting patches or trajectory segments [LCL07; JCP*10; CC14]. These methods can generate plausible results, but the variety of the simulation entirely depends on that of input data. A recent data-driven method [RXX*21] attempts to solve an optimization-based model in a solution space using real-world velocities. This method mimics the data to make decisions by selecting a velocity from the data that is similar to an agent's current velocity. However, as it ignores natural acceleration, which is critical to realistic movements, the output motion is unrealistic. Furthermore, most data-driven methods consider only simplified local interactions, e.g., local collision avoidance, and ignore macroscopic information. More recently, deep learning techniques have been used to learn more abstract behavioral models [AGR*16; vTGL*20]. Although they have better data-fitting capabilities, their generality is still limited by the training data and the models themselves are not interpretable. We argue that a general simulation framework with better realism and natural trajectories that integrates hierarchical empirical knowledge of crowd behaviors with a model-friendly data-driven scheme is required to improve the plausibility of crowd simulations.

In this paper, we aim for a new methodology that leverages the advantages of both model-based and data-driven approaches. To this end, we propose a new simulation framework grounded on a physics-based model which is designed to be data-friendly, so that the system dynamics is driven simultaneously by the general prior knowledge of crowds encoded by the physics-based model and the specific real-world crowd data at disposal (see Figure 1). The physics-based model is a multi-granularity approach. At the microscopic level, it models a wide range of local individual behaviors as energies, such as individual motions and local interactions. At the macroscopic level, a global control model is utilized for goal-directed guidance for agents, and in this paper, we primarily focus on group motion control and treat the entire crowd as a continuum [TCP06] to enforce directed flow generation. The

whole model is formulated as a dynamical system described by an energy function, where the individual behavior is modeled as the minimizer of an energy minimization problem.

Although employing energy-based formulations to capture crowd motions has been attempted before [GCC*10; KSNG17], our model differs from previous work in that it is data-friendly. Besides, we employ an acceleration-aware data-driven optimization scheme to mimic real-world velocity changes, which can improve the model's realism while maintaining its scalability and generalizability. Instead of seeking a solution that minimizes the energy in the entire solution space, which would make the minimizer only ideal in theory as in existing methods, we seek the minimizer in a human-solution subspace, parameterized by motion features computed from real-world data, so that the simulated behaviors mimic the real ones. The key underlying assumption is that the global optimum in the entire solution space does not necessarily lead to realistic behaviors; it is the human-solution space (a subspace of the entire solution space) where naturalistic human behaviors rise. Consequently, we explicitly construct the human-solution space based on real data and restrict the solution within it. This means we need to explicitly parameterize this space. We investigate the human solution using first- and second-order motion dynamics (i.e., velocity and acceleration), which represent short-term motion decisions and natural velocity changes. To that end, we parameterize the space by extracting these motion dynamics from real-world trajectories and constructing a reference dataset for optimization. We estimate velocities from real trajectories using finite differences of positions and encode the motion dynamics in the dataset as two consecutive timestep velocities in a trajectory. This expression of motion dynamics facilitates data generalization in our simulation framework. During simulation, velocity is regarded as the motion decision. We optimize for the new velocities and update the system with an implicit Euler scheme for numerical stability. Formally, the contributions of the paper include:

- a general simulation framework that introduces real data as human-solution space to enhance a dynamics model and generate realistic crowd animations.
- a generic and data-friendly physics-based model that integrates behavior models at different levels of simulation granularity to generate diversified crowd behaviors.
- an acceleration-aware data-driven optimization scheme that generates plausible trajectories in a natural human-solution space by referring to consecutive velocities in the dataset.

The rest of this paper is organized as follows. After briefly reviewing the related work in Section 2, we give an overview of our approach in Section 3 and elaborate our optimization-based data-driven model and the human-solution space in Section 4 and Section 5, respectively. Then we show simulation results and evaluation in Section 6, and conclude the limitations and future work in Section 7.

2. Related Work

2.1. Empirical crowd models

The empirical methods abstract observed crowd behaviors into mathematical models and deterministic systems, which can be clas-

sified into microscopic and macroscopic models, in terms of the simulation granularity. Microscopic approaches regard individuals as autonomous agents, and primarily focus on modelling the low-level behavioral details of each agent. They can be further divided into three categories: force-based, velocity-based, and vision-based.

Force-based methods use physical forces to model the individual interactions. Early force-based methods include the Boids model where separation, alignment, and cohesion behaviors are modelled by physical forces [Rey87], and the social force model where pedestrian dynamics is modelled by sociological forces [HM95]. Later, several extensions and variants have been proposed to model different pedestrian behaviors, e.g. high-density crowds that combine physical forces with psychological and geometrical rules [PAB07], anticipatory models using the notion of time to collision to predict future collisions and generate smooth trajectories [KHvBO09; ZIK11; KSG14]. Recently, the force-based model has been applied in simulating crowd behaviors in heterogeneous traffic scenarios [CJH*19; HCJ21]. Compared to the traditional force-based models that only consider the interactions among human agents, their methods also consider the interactions between human agents to other types of agents (e.g. cars, bikes). Force-based models can generate a wide range of crowd behaviors by combining different user-defined behavioral models. However, because they ignore prior knowledge hidden behind real-world crowd data, such as real-world motion decisions and natural velocity changes, their results may be unrealistic. Our optimization-based method, on the other hand, is more data-friendly because we use real data for real-time realistic crowd simulation.

While force-based approaches have the advantage of simple formulation, they normally require laborious parameter tuning, and suffer from numerical instability. In parallel, velocity-based models were proposed. They usually use a cost function to compute a new velocity for an agent in a continuous velocity space [vd-BLM08]. There are numerous velocity-based methods, e.g. a predictive pedestrian interaction model that uses a predictive time-varying space area to predict the future collision [PPD07], a pedestrian interaction model using the minimum predicted distance for motion adaptation[POO*09], a biomechanical model based on the principle of least efforts [GCC*10], an extension of the RVO model [vdBLM08] to simulate human-like behavior of agents in crowds [KO10], a cognitive science approach based on behavioral heuristics [MHT11], and combinations of velocity-based and force-based methods to simulate multi-agent interactions [KGM13] and handle the dense crowds [KGH*15].

Vision-based approaches can be regarded as variants of velocity-based methods, while they can better simulate the perception-action of human beings. These approaches include synthetic-vision models [OPOD10; WJDL13; HOD15], perception field based models [KSH*12], gradient based models [DMN*17], and optimal flow based models [LCMP19], etc. Besides, a recent microscopic crowd simulation framework has been proposed to combine existing models by optimization [vTGL*20].

Besides, there are also other microscopic methods that are used in different applications, e.g. the implicit crowd model for large simulation time intervals [KSNG17], position-based methods

[WLJT17], proactive crowd models that select and execute proactive steering strategies [LCM*18], etc.

Microscopic models work well for local navigation/collision avoidance. However, most of them rely on additional control for high-level behaviors. In contrast, macroscopic approaches regard a crowd as continuous flows and focus on modelling the dynamics of the entire crowd, e.g. continuum dynamics [TCP06; JXM*10], aggregate dynamics [NGCL09] and navigation field [PvdBC*11; TWCL18]. There are also hybrid models that extend the continuum dynamics from an agent based perspective [PCQ11]. These macroscopic approaches are hybrid methods that enforce local interactions separately from the macroscopic algorithm, which affects their generalizability.

The aforementioned empirical methods are model-based, as they explicitly model observed human behaviors using explicable terms in their mathematical models. However, as the pedestrian interactions are complex and subtle, the simulation results of empirical models tend to lack plausibility and diversity, because these models are usually based on simplified and idealized hypotheses with deterministic nature, e.g. least-effort. To this end, the proposed method utilizes a data-driven scheme to generate realistic and diversified crowd behaviors.

2.2. Data-driven crowd simulation

With the improvement of data acquisition techniques, data-driven methods are employed to generate realistic crowd animations [LCL07; JCP*10; ZTC13; CC14; SHW*18; HXZW20; XYWJ20]. These methods extract patches or trajectory segments from input datasets and either connect them under pre-defined rules or use them to learn some characteristics of an agent's motion. Those methods can generate plausible crowd behaviors but are limited by the data. Recently, Ren et al. proposed a data-driven method (Heter-Sim) that computes velocity from real datasets to minimizing an energy function [RXX*21]. This method can generate reasonable crowd behaviors, but its optimization scheme ignores second-order real-world motion dynamics, resulting in trajectories with unreasonable accelerations compared to real data and unrealistic simulation results. Furthermore, it only takes into account low-level motions and ignores high-level and macroscopic perspectives. On the contrary, our physics-based model allows for both microscopic and macroscopic motion control in order to generate a variety of crowd behaviors.

In parallel, machine learning approaches have been used to learn human behaviors [WOO17]. Decision trees have been used to build classifiers for pedestrian motion decisions [BKSB15], and support vector machines have been used to simulate dense crowd [MON*16]. These traditional machine learning models do not have enough learning capacity for complex crowd scenarios, as human crowd behaviors are complex and subtle. To this end, deep learning-based models have been proposed to learn the arbitrarily non-linear pedestrian dynamics, to abstract behavioral patterns from motion trajectories for simulation or prediction. Recurrent Neural Networks with social pooling have been used to learn the interactions among pedestrians and predict pedestrians' trajectories [AGR*16]. Generative Adversarial Networks (GANs) with

attention-based pooling have been used to sample plausible predictions of trajectories for the pedestrians in the scene [GJF*18]. Convolutional neural networks have been used to learn agent space heat maps for agent navigation [Osh19]. Residual networks have also been used to learn pedestrian movements [YZLL20]. These deep learning-based methods can generate more plausible or more accurate results than traditional simulation methods. However, the variety of the simulation results is even more strictly adhere to the data and therefore lack diversity. In addition, reinforcement learning has also been employed, with less dependency on data, to simulate the whole crowd [CP15; YZLL19] and agent-based behaviors [LWL18]. These methods are focused on the microscopic-level policies to control agents, which loses the high-level behavior characteristics in the data. Comparatively, our method makes use of both microscopic and macroscopic approaches to consider both the local interactions and the global path planning, which makes the method more general and scalable. Furthermore, when compared to previous data-driven methods, our framework's acceleration-aware data-driven optimization scheme can generate plausible crowd behaviors in a human-solution space while retaining the model's scalability and generalizability.

2.3. Parameter estimation and crowd evaluation

There are also works that focus on improving crowd simulation quality by automatically estimating parameters in their parameterized crowd simulation models [WGO*14; BKHF14; CLH*23] or evaluating simulation quality by developing an evaluation benchmark framework [SKFR09] or assessing the results' similarity with real-world data [LCSC09; CKGC14; WOO17]. It is worth noting that our model can be easily combined with them. Furthermore, Kim et al. [KBB*16] combine parameter estimation with simulation by employing statistical models to estimate dynamics characteristics from real-world data in order to update agents' velocity. The method can generate realistic crowd behavior based on real-world data. However, it only takes into account local motion control and disregards high-level perspectives. On the contrary, our physics-based model uses an acceleration-aware data-driven optimization scheme to generate various crowd behaviors at both the microscopic and macroscopic levels.

3. Methodology Overview

Our method can be conceptually described as a two-phase process: the preprocessing stage and the simulation stage (see Figure 2). In preprocessing, a reference dataset is generated from real crowd data as the human-solution space (see Section 5) and the motion scenario is initialized. The reference dataset consists of several features extracted from real tracklets of human crowds including estimated consecutive velocities. The scene initialization is also performed at the preprocessing stage, with different goals, obstacles, and initial states of individuals. The whole environment is represented by a 2D grid.

Our method makes full use of the motion features extracted from real data for simulation, during which we employ an acceleration-aware data-driven optimization approach to update each individual by mimicking real-world continuous motion decisions (see Section 4.1). For every time step, a continuous collision detection
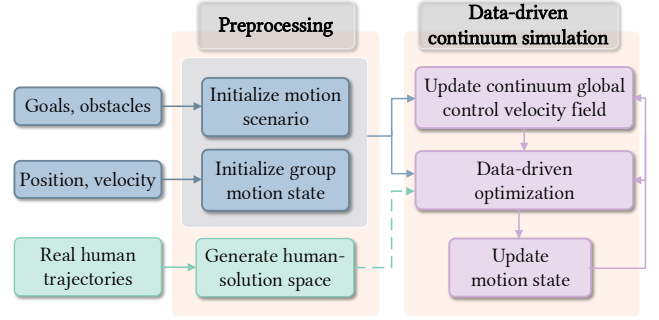

Figure 2: *The pipeline of our method.*

model is employed to generate collision-free trajectories (see Section 4.2), and a continuous macroscopic velocity field that spans the free space in the environment is calculated to generate a preferred velocity for each individual (see Section 4.3). Our model then selects a velocity from the reference dataset that minimizes an objective function which incorporates different energy terms.

4. Data-Driven Continuum Motion Control

The reference dataset is denoted as $D = \{d_v\}$, $d_v = (\mathbf{v}_{\text{arr}}, \mathbf{v}) \in \mathbb{R}^2 \times \mathbb{R}^2$, where $\mathbf{v}$ is a velocity estimated from a real tracklet to update an agent's motion state, and $\mathbf{v}_{\text{arr}}$ is the velocity in the associated real tracklet at the previous timestep of $\mathbf{v}$. We regard each pedestrian as a disk-shaped *agent* with radius r, and the motion state of each agent can be characterized by its position and velocity, i.e. $s = (\mathbf{p}, \mathbf{v}) \in \mathbb{R}^2 \times \mathbb{R}^2$. Given a crowd with N agents, the motion state of an agent i at time t is $s_i^t = (\mathbf{p}_i^t, \mathbf{v}_i^t)$. We further denote the motion state of the whole crowd as $S_t = \{s_i^t | i = 1, 2, ..., N\}$, the whole motion environment as ENV_t, and the macroscopic velocity field is $\mathbf{V}_g$. In detail, ENV_t includes the detailed information of the motion scenario and all the environmental objects such as obstacles and goals. Our stepping scheme computes the agent's new motion state after a time step in an implicit Euler fashion:

$$\mathbf{v}_i^{t+1} = \arg\min_{\mathbf{v} \in d_v \in D} E(i, d_v, S_t, \text{ENV}_t, \mathbf{V}_g),$$
$$\mathbf{p}_i^{t+1} = \mathbf{p}_i^t + \mathbf{v}_i^{t+1} \Delta t, \tag{1}$$

where the new velocity $\mathbf{v}_i^{t+1} \in d_v \in D$ minimizes the objective function E, Δt is a timestep, and $\mathbf{p}_i^{t+1}$ is the position at time $t + 1$. The objective function E is defined as:

$$E(i, d_v, S_t, \text{ENV}_t, \mathbf{V}_g) = E_{\text{df}} + E_{\text{intf}} + E_{\text{mc}}, \tag{2}$$

where E_{df} is the basic drive force, E_{intf} is the agent-agent and agent-environment interaction force, E_{mc} is the macroscopic control force. For each agent, the minimization of E is to calculate a new velocity $\mathbf{v}_i^{t+1}$ by selecting a $\mathbf{v} \in d_v \in D$ whose corresponding $\mathbf{v}_{\text{arr}}$ is similar to the agent's current velocity. For brevity, we use $\hat{\mathbf{v}}$ and v to represent the direction and the magnitude of the velocity $\mathbf{v}$, respectively. The algorithm for updating the motion state of the crowd is illustrated in Algorithm 1.

ALGORITHM 1: Motion State Update at Time t

Input: S_t, ENV_t.

Output: S_{t+1}.

1   $E_{\min} = \infty$; $\mathbf{v}_i^{t+1} = \mathbf{0}$;

2   Update the macroscopic velocity field $\mathbf{V}_g$; // see in Sec. 4.3;

3   **for** each $s_i^t \in S_t$ **do**

4      $E_{\min} = \infty$;

5   **for** each $d_v \in D$ **do**

6         $(\mathbf{v}_{\text{arr}}, \mathbf{v}) = d_v$;

7   Calculate $E(i, d_v, S_t, \text{ENV}_t, \mathbf{V}_g)$; // see in Sec. 4.1 to 4.3;

8   **if** $E(i, d_v, S_t, \text{ENV}_t, \mathbf{V}_g) < E_{min}$ **then**

9             $E_{\min} = E(i, d_v, S_t, \text{ENV}_t, \mathbf{V}_g)$;

10            $\mathbf{v}_i^{t+1} = \mathbf{v}$;

11   **end**

12   **end**

13      $\mathbf{p}_i^{t+1} = \mathbf{p}_i^t + \mathbf{v}_i^{t+1} \Delta t$;

14      $s_i^{t+1} = (\mathbf{p}_i^{t+1}, \mathbf{v}_i^{t+1})$;

15 **end**

4.1. Basic internal drive

We define an energy term E_{df} to model the force that drives the agents to make real-world motion decisions and keep moving. The energy contains a state similarity term E_{ss} for simulating motion decisions in real data, as well as a direction continuity term E_{dc} for trajectory smoothness:

$$E_{\text{df}} = E_{\text{ss}} + E_{\text{dc}}. \tag{3}$$

State similarity. Given a data sample $d_v = (\mathbf{v}_{\text{arr}}, \mathbf{v}) \in D$, we assume that the chosen velocity $\mathbf{v} \in d_v \in D$ is reasonable when an agent's current motion state is similar to that in the real data. The goal of the state similarity energy E_{ss} is to choose a reasonable velocity for each agent in order to mimic the motion decision in real trajectories and generate trajectories that are similar to the real data. For an agent i at time t, E_{ss} calculates the difference between the agent's current velocity $\mathbf{v}_i^t$ and the previous velocity $\mathbf{v}_{\text{arr}} \in d_v \in D$ in the reference dataset. E_{ss} takes into account the similarity of the previous direction E_{dir} and magnitude E_{mag}:

$$E_{\text{ss}} = w_{\text{dir}} E_{\text{dir}} + w_{\text{mag}} E_{\text{mag}},$$
$$E_{\text{dir}} = \|\hat{\mathbf{v}}_{\text{arr}} - \hat{\mathbf{v}}_i^t\|_2, \tag{4}$$
$$E_{\text{mag}} = |v_{\text{arr}} - v_i^t|,$$

where E_{dir} and E_{mag} compute the direction and magnitude difference between $\mathbf{v}_{\text{arr}}$ and $\mathbf{v}_i^t$. $w_{\text{dir}} \geq 0$ and $w_{\text{mag}} \geq 0$ are the weights of E_{dir} and E_{mag}, respectively. The state similarity energy is defined as the ability to generate plausible trajectories that are similar to the real data by using a physics-based model to mimic real-world motion dynamics (continuous velocity changes).

Trajectory smoothness. We also model an extra force for trajectory smoothness as a direction continuity energy E_{dc}, which measures the difference between the direction of the selected velocity $\hat{\mathbf{v}}$ and the current direction of the agent $\hat{\mathbf{v}}_i^t$:

$$E_{\text{dc}} = w_{\text{dc}} \cdot \|\hat{\mathbf{v}} - \hat{\mathbf{v}}_i^t\|_2, \tag{5}$$

where $w_{\text{dc}} \geq 0$ is the weight of the energy. While trajectory smoothness is commonly employed in crowd simulation, state similarity,

which incorporates velocity change to make motion decisions that mimic real data, is less frequently considered. This is to exploit the motion dynamics of the human-solution space in our context.

4.2. Microscopic interaction

Interactions exist between an agent and other agents or the environment. They affect the agent decisions in e.g. collision avoidance. We define an interaction energy term E_{intf}, which includes an agent-agent term E_{aa} to model the interactions among agents, and an agent-environment (agent-env) term E_{ae} to model agents' reactions to the environment:

$$E_{\text{intf}} = E_{\text{aa}} + E_{\text{ae}}. \tag{6}$$

4.2.1. Inter-agent interactions

To generate collision-free trajectories, the agent-agent interaction energy E_{aa} models the instantaneous interactions to avoid possible collisions in a short period (a time step Δt), and the anticipatory interactions to predict possible collisions in a longer term ($T\Delta t$, and $T > 1$). E_{aa} is formulated as:

$$E_{\text{aa}} = w_{\text{insCA}} E_{\text{insCA}} + w_{\text{antiCA}} E_{\text{antiCA}}, \tag{7}$$

where E_{insCA} is the instantaneous interaction energy, E_{antiCA} is the anticipatory interaction energy, with weights $w_{\text{insCA}}, w_{\text{antiCA}} \geq 0$.

Instantaneous interaction. To prevent agent collisions, instantaneous interaction is defined as maintaining inter-agent separations. To that end, the instantaneous interaction energy E_{insCA} is defined to select a velocity for an agent i to separate it from the potential collision neighbors, assuming that the collision neighbors of an agent i are the agents whose distance to this agent is within a range $R_{\text{Ins}} = 2v_{\max}\Delta t$ [GNCL14], where $v_{\max}$ is the maximum velocity in the dataset.

Assuming that agent i chooses a velocity $\mathbf{v} \in d_v \in D$ and a colliding neighbor j holds its current velocity $\mathbf{v}_j^t$, we use a distance-based scheme similar to that used in [RXX*21] to model the instantaneous interaction:

$$E_{\text{insCA}} = \sum_{j \in \text{InsN}} e^{\left(1 - d\left(\Delta t, \mathbf{v}, s_i^t, s_j^t\right)/d_c\right)}, \tag{8}$$

where InsN is the set of instantaneous collision neighbors, $d\left(\Delta t, \mathbf{v}, s_i^t, s_j^t\right)$ is the predicted distance of the two agents at time $t + 1$, and d_c is a constant as the comfort distance between agents.

It is worth noting that our method defines short-term collision neighbors differently than [RXX*21], where the collision neighbors are the agents whose predicted distance from the focal agent is within a range d_c. The instantaneous collision avoidance method works well for separating agents, but motion jumps may occur during simulation due to the discontinuity of the energy function at the truncation distance d_c (see the blue curve in Figure 3). Our method defines E_{insCA} in a smoother way to separate agents and keep the distance between agents around the comfort distance d_c (see the orange curve in Figure 3). Besides, we calculate the instantaneous collision avoidance energy in an anisotropic manner by summing each energy. The goal of such a design is to keep the peak energy value from disappearing while maintaining sensitivity in detecting potential collisions.
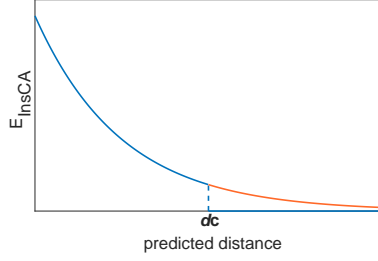
Figure 3: *The visualization of E_{insCA}. The discontinuity of E_{insCA} (blue) in [RXX*21] is smoothed by our method (orange).*

Figure 4: *Simple illustrations for situations in the continuous collision detection method. In each figure, the disks with the deepest color are the current positions of two agents at time t, disks with the lightest color are the predicted positions when the two agents collide with each other at time $t + \alpha\Delta t$, and the other disks are the predicted positions at time $t + \Delta t$.*

Continuous collision detection. Equation 8 is designed to avoid short-horizon collisions and keep agents apart. However, agents may still collide at some point $t' = t + \alpha'\Delta t, \alpha' \in (0, 1]$ during a single simulation step. Therefore, we employ a continuous collision detection method. Denoting $d\left(\Delta t, \mathbf{v}, s_i^t, s_j^t\right)$ as the distance between two agents at time t':

$$d\left(\Delta t, \mathbf{v}, s_i^t, s_j^t\right) = \|\mathbf{p}_i^{t'} - \mathbf{p}_j^{t'}\|_2 - (r_i + r_j), \qquad (9)$$

where $\mathbf{p}_i^{t'} = \mathbf{p}_i^t + (\alpha'\Delta t)\mathbf{v}$ and $\mathbf{p}_j^{t'} = \mathbf{p}_j^t + (\alpha'\Delta t)\mathbf{v}_j^t$ are the predicted positions of the two agents at time t'. Under the assumption of trajectory linearity within a timestep, α' is calculated based on the time coefficient α when the two agents are closest (i.e., the predicted distance at time $t + \alpha\Delta t$ is 0):

$$\alpha' = \begin{cases} \alpha, & 0 < \alpha \leq 1, \\ 1, & \text{others.} \end{cases} \qquad (10)$$
$$\|\mathbf{p}_i^{t+\alpha\Delta t} - \mathbf{p}_j^{t+\alpha\Delta t}\|_2 - (r_i + r_j) = 0.$$

Figure 4 depicts the scenarios in which α will be calculated at various intervals. In Equation 10, $\alpha' \in (0, 1)$ means that, the two agents will collide during a timestep, and the predicted distance is 0 in this situation. Other situations mean that the predicted distance is the distance at time $t + \Delta t$.
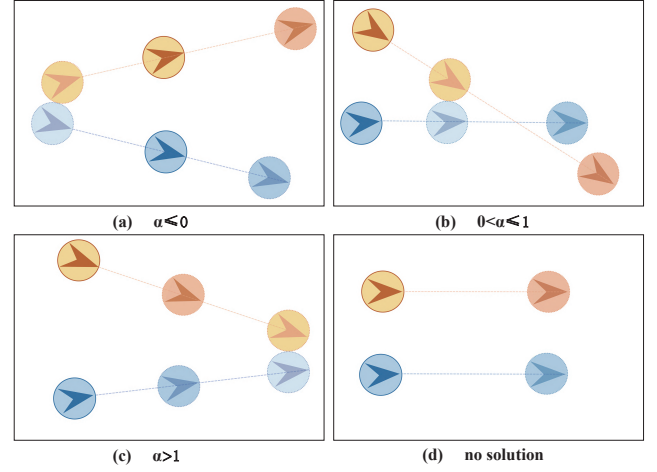
Anticipatory interaction. During navigating through crowds, humans also consider long-horizon collision avoidance, which we define as anticipatory interaction. We model the potential anticipatory collision between two agents using the colliding time α computed in Equation 10 when two agents will collide in the far future:

$$E_{\text{antiCA}} = \sum_{j \in \text{AntiN}} e^{(1-\alpha/\alpha_c)}, \ \alpha \geq 0, \qquad (11)$$

where AntiN is the set of long-horizon collision neighbors, and $\alpha_c \gg 1$ is a truncation time to reduce the influence of potential collisions in the far future. We empirically set $\alpha_c = 3/\Delta t$ in our experiments. For computing performance, the long-range collision neighbors are the agents whose distance from agent i is within a certain range $R_{\text{anti}} = 2v_{\max}\alpha_c\Delta t$ [GNCL14].

4.2.2. Agent-environment interactions

The agent-environment interaction energy E_{ae} models agents' instantaneous reactions to avoid impending environmental obstacles such as walls and buildings. We model potential instantaneous collisions between an agent i and an obstacle k within the short-term collision range R_{Ins} in Section 4.2. To avoid possible collisions within a timestep, we use the same continuous collision detection method as Equation 8 to predict the distance between the agent and an obstacle. Then E_{ae} can be defined as:

$$E_{\text{ae}} = w_{ae} \sum_{k \in \text{InsEnv}} e^{\left(1 - d\left(\Delta t, \mathbf{v}, s_i^t, \mathbf{p}_k^{t+1'}\right)/d_e\right)}, \qquad (12)$$

where $w_{ae} \geq 0$ is the weight, InsEnv is the set of the impending colliding obstacles, $\mathbf{p}_k^{t+1'}$ is the predicted obstacle position, and $d\left(\Delta t, \mathbf{v}, s_i^t, \mathbf{p}_k^{t+1'}\right)$ is the predicted distance.

It should be noted that anticipatory collision avoidance for agent-environment interaction, such as that in Equation 12 for agent-agent collision avoidance, is not considered in this section because it can be handled well by the macroscopic continuum control (see Section 4.3).

To calculate E_{ae}, we use the grid to find neighboring objects, and the cells containing objects are considered obstacles. In Equation 12, each distance calculated is the distance between an agent i and the center of cell k, i.e. $d(i, k) = \|\mathbf{p}_i - \mathbf{p}_k\|_2 - r_i - \frac{\sqrt{2}}{2}l$, where l is the side length of the cell. Because we can discretize both regular and irregularly shaped obstacles into grid cells, our approach can handle obstacles of any shape. When the obstacle is a disk-shaped object, the predicted distance can be calculated as the distance between the agent's predicted position and the predicted position of the center of the obstacle to simplify the calculation and improve performance.

4.3. Macroscopic control

Aside from individual agent local behaviors, we must also drive goal-directed behaviors for individual agents or model global behaviors such as groups, flows, and so on. Not only are they commonly observed in the real world, they can also be utilized for user-defined control in simulation. Therefore, we introduce a continuum control force term E_{mc} to provide macroscopic control that drives each agent to follow a global path. We assume that at every time step, the continuum control model computes a macroscopic control velocity based on each agent's local surroundings and the predicted global motion cost. The energy term E_{mc} computes a velocity that is similar to the desired macroscopic control velocity:

$$E_{mc} = w_{mdir}E_{mdir} + w_{mmag}E_{mmag}, \tag{13}$$

where E_{mdir} is the macroscopic control direction term and E_{mmag} is the control magnitude term. $w_{mdir} \geq 0$ and $w_{mmag} \geq 0$ are their weights.

The macroscopic control can be realized by a global velocity field. Although our framework can employ any methods for the velocity field calculation, we employ a continuum crowd model [TCP06] which has a similar representation of the global motion. In [TCP06], the agents in a group typically have the same goal, and for each group, a macroscopic velocity field $\mathbf{V}_g$ is generated as a position-related function, and each agent in this group is given a preferred velocity by interpolating $\mathbf{V}_g$. In terms of diverse behavior control, in our method, the macroscopic continuum control model can be used not only for generating group behaviors that regard a pedestrian group as a whole with a common goal, but it can also be used for global path planning for individual agents with distinct goals. For an agent i at time t, E_{mdir} and E_{mmag} are calculated as:

$$\begin{aligned} E_{mdir} &= \|\hat{\mathbf{v}} - \hat{\mathbf{V}}_g(\mathbf{p}_i^t)\|_2, \\ E_{mmag} &= |v - V_g(\mathbf{p}_i^t)|, \end{aligned} \tag{14}$$

where $\hat{\mathbf{V}}_g(\mathbf{p}_i^t)$ is the preferred direction and $V_g(\mathbf{p}_i^t) = \|\mathbf{V}_g(\mathbf{p}_i^t)\|$ is the preferred speed.

The macroscopic velocity field $\mathbf{V}_g$ can be represented by the optimal path from each position to a goal [TCP06]. Furthermore, calculating the optimal path is equivalent to minimizing the total cost based on three factors: total path length, total travel time, and total repulsion effect from obstacles, with the cost function defined as:

$$\text{Cost} = w_d \underbrace{\int_P 1 ds}_{\text{path length}} + w_t \underbrace{\int_P \tau ds}_{\text{travel time}} + w_r \underbrace{\int_P R ds}_{\text{repulsion force}} \tag{15}$$
$$= \int_P C ds, \text{ where } C \equiv w_d + w_t \cdot \tau + w_r \cdot R,$$

where w_d, w_t and w_r are weights, and $w_d + w_t + w_r = 1, w_d, w_t, w_r \geq 0$, τ is the travel time. R is environment-repulsion force. All integration is conducted along the whole path. The higher w_r is, the more repulsive the obstacles are.

The grid is used to discretize Equation 15, and a potential function $\Phi : \mathbb{R}^2 \to \mathbb{R}$ is defined over cells. At the goal, we have $\Phi = 0$. For anywhere else, Φ satisfies an Eikonal equation: $\|\nabla\Phi(\mathbf{p})\| = C$. In each time step, the velocity of a crowd is converted to a speed

field based on the maximum permissible speed in any direction in each cell, and then the unit cost field C is updated for each group, followed by updating the potential Φ and its gradient. The velocity field $\mathbf{V}_g$ is finally determined. The velocity of each cell is a two-dimensional vector with the opposite gradient direction, and the magnitude of each dimension is scaled by the corresponding speed at the speed field. Interpolating the velocity field yields the preferred velocity for macroscopic motion planning. We refer readers to [TCP06] for more information.

To generate different scenarios, the macroscopic velocity field can be updated online during simulation to capture dynamic changes in the states of agents or environmental objects, or it can be pre-calculated as a static global motion control map while ignoring agents' motion states. Furthermore, for agent personalization, the preferred speed $V_g(\mathbf{p})$ can be user-defined to impose control.

Continuous environment-repulsion field. To incorporate obstacles in $\mathbf{V}_g$, [TCP06] introduce a 'discomfort' value in obstacle-occupied cells to generate repulsion. However, this causes significant unsmoothness in $\mathbf{V}_g$ near obstacles. Therefore, we propose a new continuous repulsion. For a cell m in the grid, the continuous env-repulsion force $R(\mathbf{p}_m)$ is calculated based on its minimum distance d_{min} to the boundary cells of the environmental obstacles:

$$R(\mathbf{p}_m) = h(d_{min}) =$$
$$\chi \cdot \begin{cases} 1, & d_{min} < 0, \\ 0.5 \cdot \left(\cos(\frac{\pi \cdot d_{min}}{d_{crep}}) + 1 \right), & 0 \leq d_{min} \leq d_{crep}, \\ 0, & d_{min} > d_{crep}, \end{cases} \tag{16}$$

where $\chi > 0$ is a scaling constant, $d_{crep} > 0$ is a predefined distance threshold of the boundary cells. d_{min} is the signed distance determined by whether the cell is inside the obstacle:

$$d = \min\{\|\mathbf{p}_m - \mathbf{p}_k\|_2\}, k \in K,$$
$$d_{min} = \begin{cases} -d, & m \text{ is inside an obstacle}, \\ 0, & m \text{ is at a boundary cell of an obstacle}, \\ d, & m \text{ is outside the obstacles}, \end{cases} \tag{17}$$

where K is the set of the boundary cells of the obstacle.

5. Human-solution Space

Solving Eq. 1 in the entire solution space ensures that our system runs along at the envelop of the minimal energy. However, this is not ideal because individuals might not always follow a minimal-energy trajectory. Existing research assumes certain principles on individual motions, e.g. minimum effort [GCC*10], power-law [KSG14], etc, but they are based on simplified hypotheses. Real-world individual motions are almost always sub-optimal for the physics-based models, which significantly affects the visual realism of crowd animation [WOO16; HXZW20]. Therefore, we propose to advance the system in the human-solution subspace rather in the entire solution space.

5.1. Parameterization

We hope to find reliable information for parameterization by using trajectory segments extracted from real crowd videos. Individual movements in crowds, from a microscopic perspective, form a series of short-term decision-making processes, which are reflected in motion dynamics. As a result, based on the trajectory segments, we investigate high-order information to explore motion dynamics. The 1*st*-order information (velocity) is the change of positions which is a natural reflection of motion decisions. In addition, the 2*nd*-order information (acceleration) shows the trend of decisions which should also be incorporated. Fortunately, the velocity can be estimated relatively reliably from the trajectory segments, and it has been widely used for crowd simulation [RXX*21], crowd activity analysis [WO16], and fidelity evaluation of simulated crowd data [WOO17]. Unlike [RXX*21], which only considers velocity, our method considers higher-order motion dynamics from real data to construct a dataset that includes both first- and second-order real-world information when parameterizing the human-solution space. Furthermore, to reflect real-world continuous motion decisions, we encode the information as velocities in two consecutive timesteps.

Given a trajectory dataset Γ with L trajectory segments, i.e. $\Gamma : \Gamma_1, \Gamma_2, ..., \Gamma_i, ...$, where $i = 1, 2, ..., L$, a reference dataset D is generated based on Γ. In Γ, each trajectory is a discrete time series of positions, i.e. $\Gamma_i : \mathbf{X}_i^1, \mathbf{X}_i^2, ..., \mathbf{X}_i^t, ...$, where $t = 1, 2, ..., T$ refers to the temporal information of a trajectory and T is the total time steps that Γ_i covers. The velocities that imply the motion decision of the pedestrians can be estimated from Γ_i by the first-order forward differencing of the positions in the trajectory, that is, $\mathbf{v}_i^t = \left(\mathbf{X}_i^{t+1} - \mathbf{X}_i^t \right) / \Delta T$, where $t = 1, ..., T-1$ and ΔT is a time step. The accelerations that imply the continuous change of velocities can be estimated by the second-order forward differencing of the positions. The estimated acceleration is a motion dynamics feature used in Section 6 to evaluate the simulation results' performance.

Each data term in the referenced dataset D consists of two consecutive timestep velocities in a trajectory, $[\mathbf{v}_i^t, \mathbf{v}_i^{t+1}]$, where $\mathbf{v}_i^t$ is the velocity of a pedestrian during the previous timestep to get to $\mathbf{X}_i^t$, and $\mathbf{v}_i^{t+1}$ is the new velocity to get to the next position $\mathbf{X}_i^{t+1}$.

Notably, the human-solution space contains velocities that are not constrained by the context state, implying that we do not presume to solve a context-solution matching problem. This is due to the fact that the context state and the solution (i.e., velocity and acceleration) have a many-to-many mapping in general. We can observe sufficient solutions across all possible context states if we consider the solution as a distribution conditioned on the context state. As a result, we do not match the solutions to their context states. Our method, on the other hand, chooses the best velocity from the human-solution space to update an agent's motion by matching the previous velocity in the dataset to the agent's current one, allowing it to mimic real-world continuous motion decisions. Although it is theoretically possible that a mismatch between the chosen solution and its context state could result in unnatural motions, we did not observe this in practice. Furthermore, because velocity and acceleration are fundamental features used in most physics-based methods to describe crowd motion dynamics, those obtained from real trajectories are friendly to the optimization-based model for improving realism while maintaining scalability and generalizability.

5.2. Data generalization to different scenarios

Similar to other data-driven methods, our method relies on the availability of data. To reduce the data dependency and improve the generalizability for different scenarios, we use a direction alignment method to simulate scenarios where the desired agents' movements differ greatly from those of the reference dataset obtained in Section 5.1, for example, simulating the adversarial movements of pedestrians by referencing a dataset that only includes unidirectional movements (see Section 6.3.4). Furthermore, an augmented dataset can be generated by blending datasets from different real-world scenarios to generate different pedestrian behaviors in a complex scenario.

Direction alignment. If we directly search for the optimal new velocity from the dataset obtained in Section 5.1, the synthesized scenario will be limited to generating movements similar to the real data, according to Equation 2. The direction adaption method introduced in [RXX*21] is one method for removing the constraint, which maps the local coordinates of the velocities in the dataset to those of the simulation scenarios by aligning their control directions, where the estimation error for estimating the control directions from real data may reduce simulation plausibility. In contrast, because the velocity pairs indicate that each data term is a trajectory segment of two consecutive time steps with three sequential positions, we use a direction alignment method to align the previous direction of a data term with an agent's current direction and convert the data term's selected velocity to the agent's local coordinate.

Given a data term $d_v = (\mathbf{v}_{arr}, \mathbf{v}) \in D$, the chosen direction of the agent and the data term shares the same rotation $\mathcal{M} \in \mathbb{R}^{2 \times 2}$ with the current direction, i.e.:

$$\hat{\mathbf{v}}_i^t = \mathcal{M} \cdot \hat{\mathbf{v}}_{arr}, \tag{18}$$

$$\hat{\mathbf{v}}' = \mathcal{M} \cdot \hat{\mathbf{v}}, \tag{19}$$

where $\hat{\mathbf{v}}'$ is the predicted new direction of the agent calculated from a data term d_v, thus the aligned new velocity is $\mathbf{v}' = v \cdot \hat{\mathbf{v}}'$. $\mathcal{M} = \begin{bmatrix} m_1, & -m_2 \\ m_2, & m_1 \end{bmatrix}$ is the standard rotation matrix in Euclidean space in the counterclockwise direction, where $m_1^2 + m_2^2 = 1$. Given $\hat{\mathbf{v}}_i^t$ and $\hat{\mathbf{v}}_{arr}$, m_1 and m_2 can be calculated by solving the quadratic Equation 18.

When utilizing the direction alignment method, the energy term of the similarity of previous direction E_{dir} for state similarity is constantly 0 during simulation, thus saves the computing cost.

6. Experimental Results and Evaluations

The implementation is in C++ and the experiments were run on a PC with an Intel (R) Core (TM) i7 4.00 GHz CPU, 32 GB RAM, and an NVIDIA Geforce GTX 1060 GPU. We provide both qualitative and quantitative evaluations to demonstrate the performance of our method. Due to the space limit, we only show representative results and refer the readers to the supplementary materials for

Table 1: *The weights used in different simulation scenarios of our result.*

Scenario		E_{ss}		E_{dc}	E_{aa}		E_{ae}	E_{mc}	
		w_{dir}	w_{mag}	w_{dc}	w_{insCA}	w_{antiCA}	w_{insE}	w_{mdir}	w_{mmag}
Hallway	no obstacle (100/200 agents)	1.0	1.0	0.25	0.5	1.0	0	0.75	0.75
	with an obstacle	1.0	1.0	0.25	0.5	1.0	1.0	0.75	0.75
Dynamic Envs	Obstacle	0.75	0.75	0.25	0.5	1.0	1.0	1.0	1.0
	Goal	0.75	0.75	0.25	0.5	1.0	0	1.0	1.0
Crowd crossing		1.0	1.0	0.25	0.5	1.0	0	0.75	0.75
Crowd wandering		1.0	1.0	0.5	0.5	1.0	1.0	1.0	1.0
Train station		1.0	1.0	0.5	0.5	1.0	1.0	0.5	0.5
Intersection		1.0	1.0	0.25	0.5	1.0	1.0	0.5	0.5
Bottleneck		1.0	1.0	0.25	0.25	1.0	1.0	1.0	2.5

more details. In all our experiments, we set $K = 1$, and the weights of the energy terms for the test scenarios are shown in Table 1. Additionally, the weights for the street scenario are shown in Table 3.

6.1. Qualitative results

We first show several results generated by referencing the dataset from [ZKSS12]. The simulation results show that, based on a single reference dataset, our method can generate various crowd scenarios, not merely restricted to that of the dataset. In the following simulations, each agent is initialized with a random velocity chosen from the reference dataset, and the desired speed is the maximum speed in the dataset.

Hallway scenario. We show the results of the adversarial movements of different scales of crowds in a narrow hallway (see Figure 5), similar to the scenario of the reference dataset from [ZKSS12]. Note that a static obstacle is placed in the center of the hallway in Figure 5(c). In addition, the macroscopic velocity field is updated online to capture dynamic changes in agent group states and guide agents to avoid dense congestion.

Dynamic environments. To demonstrate our method's ability to adapt to changing environments, a scenario with a moving obstacle and a scenario with a moving goal is simulated (see Figure 6), including 50 and 50 agents respectively. As the obstacle/goal is moving, the macroscopic velocity field is updated online to reflect dynamic changes in the environment.

Crowd crossing. In Figure 7, four groups of agents are moving in an adversarial way to reach the opposite corner and each group includes 50 agents. Similar to the hallway scenario, the macroscopic velocity field is updated online.

Crowd wandering. To demonstrate the scalability of our method, we simulate a high-density scenario with a crowd of 500 agents in Figure 8. There are several static obstacles in the area. In this experiment, we set several goals, and each agent is assigned a goal. When the agent reaches its current goal, it will change to a random goal. Because agents in the scenario are separated and have a variety of motion options, we use a pre-calculated static macroscopic velocity field that considers static environmental objects while ignoring agents' motion states to globally guide individual motion. We compute the static macroscopic velocity field for each goal (i.e., the end position of an agent or several agents) using a speed field

in which the maximum permissible speed in any direction is the preferred speed of agents toward the goal. Using a pre-calculated velocity map can reduce the computational cost of updating the macroscopic velocity field during the simulation.

In Figure 9, we show more simulation results that are generated by referencing other datasets. The simulation results show that our method is adaptive to different data.
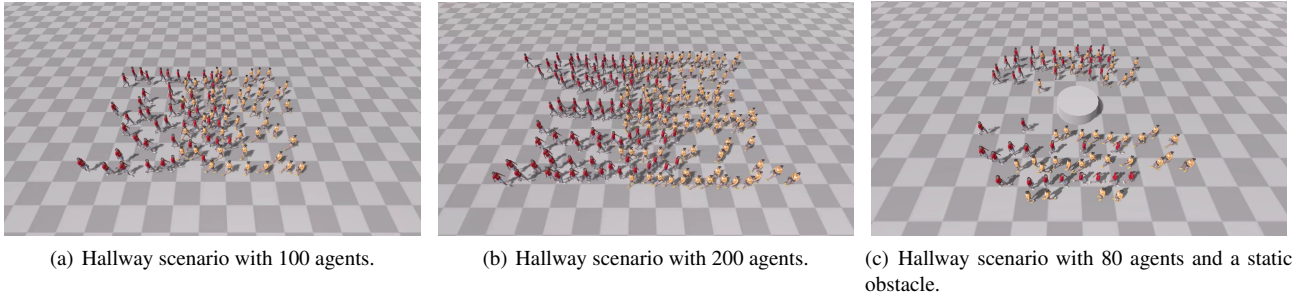
Street scenario. In Figure 9(a), we use the reference dataset from [LCL07] to generate a similar scenario, which is a bidirectional street with sparse pedestrians. The appearance time, the initial motion state, and the desired speed of each agent is similar to that in the dataset. Similar to the crowd wandering scenario, to guide individual motion, we use a pre-calculated static macroscopic velocity field for each goal.

Station scenario. In Figure 9(b), we use the reference dataset from [ZWT12], which is a multi-gate (multi-goal) station with dense pedestrians. The motion state of each agent and the goal gate are initialized in a random way. We use a pre-calculated static macroscopic velocity field for each goal to guide individual motion, similar to the crowd wandering scenario.
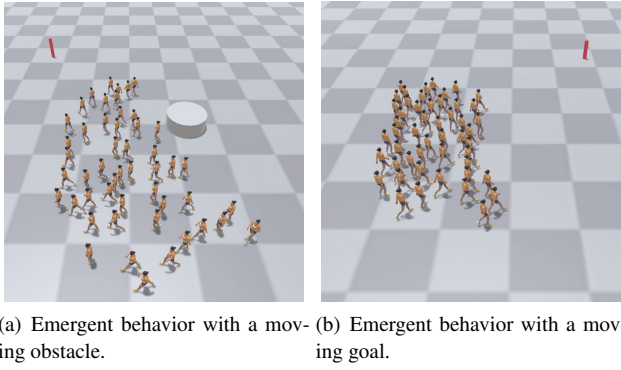
Intersection scenario. In Figure 9(c), we also simulate an intersection scenario by referencing the dataset from [YLRÖ19], where several pedestrians avoid a passing car to cross the road in an intersection. During simulation, the motion state of the agents are initialized in a random way, and the cars are initialized as dynamic obstacles with pre-defined paths. The macroscopic velocity field is updated in real-time to globally guide the agents in responding to dynamic changes in the obstacles.

Evacuation through a bottleneck. In Figure 9(d), we use the dataset from [SPS*09] to generate crowd evacuation behaviors in a bottleneck scenario similar to the referenced dataset. During simulation, the motion state of the agents are initialized in a random way. The agents in this scenario are all walking toward the same goal. To reduce congestion in the confined space, we treat the agents as a continuum and update the macroscopic velocity field at each timestep.

Parameter tuning. By varying the weights of the energy terms, crowd behavior can be intuitively adjusted. For example, having the default value of each weight set to 1 means that all behavior models have equal control. To reduce sudden motion changes, the weight of instantaneous interaction energy (w_{insCA}) is reduced in

(a) Hallway scenario with 100 agents.

(b) Hallway scenario with 200 agents.

(c) Hallway scenario with 80 agents and a static obstacle.

Figure 5: *The hallway scenario with different scales of crowds.*

(a) Emergent behavior with a moving obstacle.

(b) Emergent behavior with a moving goal.

Figure 6: *Emergent behaviors with different moving environmental objects in the scenario. In both (a) and (b), the red cylinder is the goal of a crowd and the white disk-shaped object is the obstacle.*
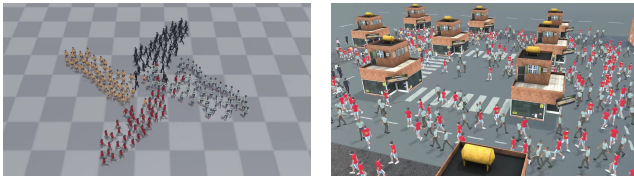
Figure 7: *Four groups of pedestrians walking through each other.*

Figure 8: *Pedestrians wandering in a virtual city block.*

all simulated scenarios. The weight of the direction control model (w_{dc}) is smaller than that of state similarity and macroscopic control in all scenarios to encourage agents to focus on natural velocity changes and global guidance. The weights of the macroscopic control model (w_{mdir} and w_{mmag}) in the hallway and crossing, for example, are smaller than those of the state similarity energy to encourage agents to focus on real-world continuous motion decisions. In the dynamic environment scenario, the weights of the state similarity energy (w_{dir} and w_{mag}) are smaller than that of the macroscopic control energy to improve the efficiency of responding to environmental change. The weight of the energy for macroscopic magnitude similarity (w_{mmag}) is increased in the bottleneck scenario to improve the efficiency of changing agent speeds so that
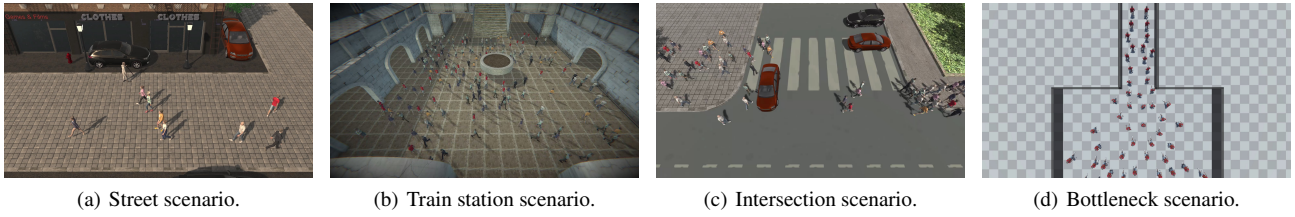
agents can avoid congestion when entering a crowded environment from a sparse one and vice versa. Furthermore, $w_{dir} = w_{mag}$ and $w_{mdir} = w_{mmag}$ apply to the majority of scenarios in practice, reducing the need for manual adjustments.
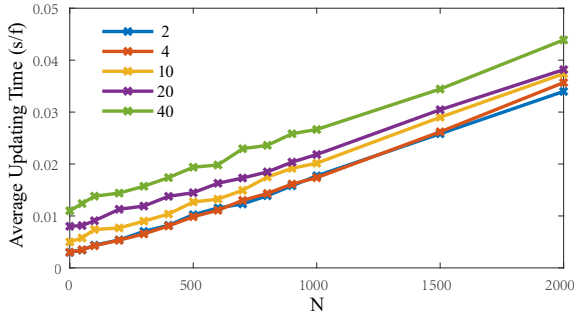
6.2. Evaluations

6.2.1. Time performance

The time complexity of our motion updating algorithm (see Algorithm 1) is $O(M \times (WH)) + O(kN)$, where $O(M \times (WH))$ is the time complexity to calculate the macroscopic velocity field, M is the group (or goal) number, W and H are the width and height of the grid map; $O(kN)$ is the time complexity to update the motion of the agents in a time step, N is the number of agents, and k is the data sample size in the reference data. In our experiments, for fast indexing, we only traverse data terms whose v_{arr} is close to the agent's current speed, and $k = 300$ is enough to generate all scenarios. The time complexity is nearly linear ($O(kN)$) if the macroscopic velocity field is pre-calculated.

To quantitatively test the performance of the proposed algorithm, we simulate a crowd in a 200*200 grid with no obstacle. The length of each cell is $1m$. During the initialization, we randomly position N agents, and divide them into M groups. The initial velocities of the agents are randomly selected from a dataset [ZKSS12]. Figure 10 shows a comparison of the averaging updating time for different agent groups with online calculation of the macroscopic velocity field during simulation. In Figure 10, the average updating time with 0 agents is the time performance of updating the macroscopic velocity field. The computation cost increases with the group number. However, as we only perform the online updating of the macroscopic velocity field in dense and flow-like crowd scenarios with few agent groups, this cost is negligible. In the sparse crowd scenarios, calculating the macroscopic velocity field is part of initialization, so the computation does not affect the online performance. Table 2 also shows the time performance in different simulations, demonstrating that all of the tested scenarios can be simulated in real time. Furthermore, in more complex scenarios, such as the crossing and wandering scenario, where dense agents are assigned various goals, frequent interactions with the environment or other types of agents affect time performance.

| (a) Street scenario. | (b) Train station scenario. | (c) Intersection scenario. | (d) Bottleneck scenario. |

Figure 9: *The simulation results referencing different datasets.*

Figure 10: *Time performance of our approach. The computation cost is linear w.r.t the number of agents. The legends are different group (goal) numbers set in the simulation.*

Table 2: *The time performance of different simulation scenarios. N is the number of agents present at any given time.*

Scenario		N	Time (s/f)
Hallway		100	0.0035
		200	0.0110
Hallway with an obstacle		80	0.0026
Dynamic Envs	Obstacle	50	0.0020
	Goal	50	0.0028
Crowd crossing		200	0.0147
Crowd wandering		500	0.0187
Street		1-20	0.0004
Train station		138-193	0.0069
Intersection		54-89	0.0058
Bottleneck		1-63	0.0068

6.2.2. Comparisons

To demonstrate the plausibility of our method, we compare simulation results in a narrow hallway scenario with real data from [ZKSS12] and in a sparse street scenario with real data from [LCL07] (i.e., Fig. 9(a)). During the initialization phase of each experiment, the initial motion state, appearance time, and motion preferences (e.g., the minimal and maximal velocities, the minimal distance, and the maximum acceleration) of each agent are copied from the real data. For the compared methods, we use a genetic algorithm to estimate the optimal key parameters from the simulated scenario's real data, and the learned parameters are shown in Table 3. The objective function of the genetic algorithm is based on the absolute difference metric (ADM) proposed by [WGO*14].
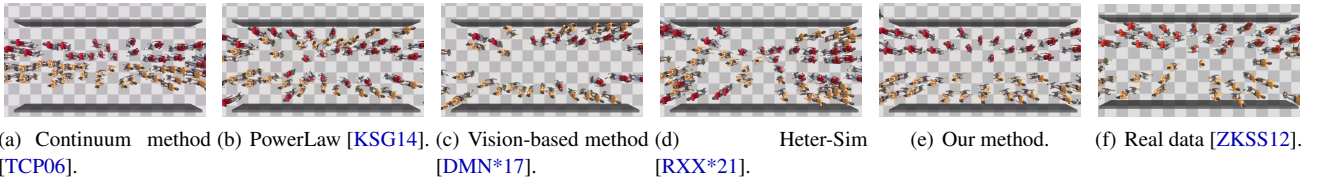
Trajectory and statistical comparisons in a hallway scenario. We simulate bidirectional pedestrians in a narrow corridor, similar to [ZKSS12], to compare our method with a continuum method [TCP06], a force-based method (PowerLaw) [KSG14], a vision-based method [DMN*17] and a state-of-art data-driven method (Heter-Sim) [RXX*21] both qualitatively and quantitatively. It is worth noting that the key parameters of the compared continuum method are predefined and identical to those in our method ($w_d, w_t = 0.2, w_r = 0.6$). The compared Heter-Sim method [RXX*21] also uses a dataset obtained from [ZKSS12]. The frame rate is 16 frames per second as in the data.

In this scenario, each agent aims to reach the opposite exit, and we regard the agent as inside the corridor before it arrives at the goal exit. The goal of each agent in all of the compared experiments is the closest boundary point of its goal exit in the corridor. Because the real data includes the trajectories of agents outside the corridor, which influence the local environment of the nearby agents moving into the corridor, for all simulation methods, the agents will move toward the nearest boundary points of the scenario as soon as they complete their travel in the corridor.
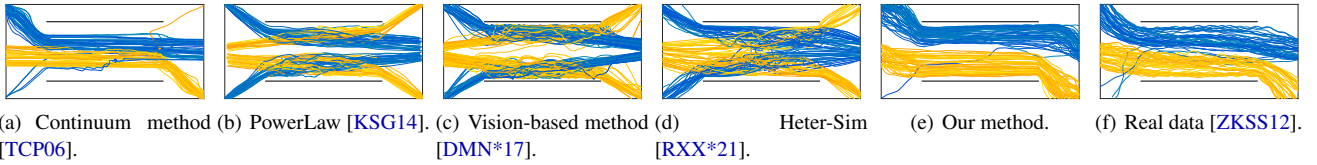
Figure 11 shows the visual simulation results compared with the real data. The simulation results show that the continuum model from [TCP06] and our method mimics real crowds well in separate pedestrian flows to reduce congestion, while the other approaches fail to achieve this because they do not take high-level motion control into account in the same manner that we do. Figure 12 presents visual comparisons on the generated trajectories, where our method mimics data more closely.

Quantitatively, we quantify the trajectory similarity. We employ the absolute difference metric (ADM) and the path length metric (PLM) proposed by Wolinski et al. [WGO*14]. Table 4 shows the results. Our method achieves better scores compared with the baseline methods [TCP06; KSG14; DMN*17; RXX*21], showing that our trajectories are more similar to the real data. Although the trajectories of the continuum method in [TCP06] are visually similar to the real ones (see Figure 12), the speed of crowd flows is slower during simulation (Figure 13(a)), resulting in higher scores for trajectory similarity.

We also compare distribution similarity, which includes velocity, minimum distance (the distance to the nearest agent), and acceleration distributions, as these are descriptors that capture both the state and the motion dynamics of agents. The velocity denotes first-order motion dynamics, which denotes an instantaneous motion decision, and the acceleration denotes second-order motion dy-

(a) Continuum method [TCP06]. (b) PowerLaw [KSG14]. (c) Vision-based method [DMN*17]. (d) Heter-Sim [RXX*21]. (e) Our method. (f) Real data [ZKSS12].

Figure 11: *Qualitative comparisons in a bidirectional scene with real data.*

(a) Continuum method [TCP06]. (b) PowerLaw [KSG14]. (c) Vision-based method [DMN*17]. (d) Heter-Sim [RXX*21]. (e) Our method. (f) Real data [ZKSS12].

Figure 12: *Comparisons of the generated trajectories. The corridor is in the scene's center and is surrounded by black walls. The blue curves show the trajectories of the group that starts on the right side of the corridor, while the yellow curves show the trajectories of the group that starts on the left. For each example, 100 trajectories are sampled from the simulation results/real trajectories.*
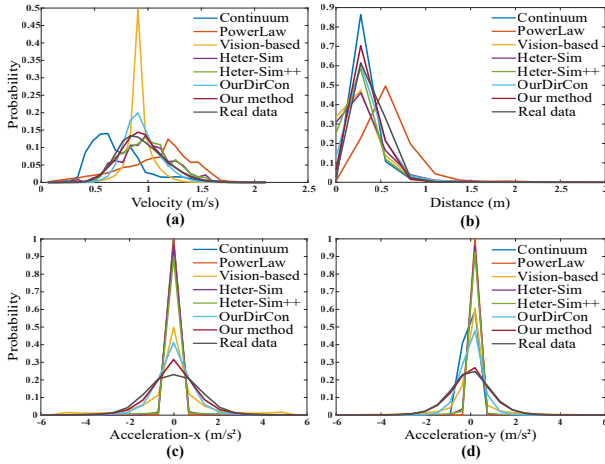
Figure 13: *Comparisons of the distributions of velocity (a), minimal distance (b), and acceleration (c for x (lateral of the corridor) direction and d for y (longitudinal of the corridor) direction).*

namics, which denotes velocity change and continuity. We use a widely accepted metric, Kullback–Leibler divergence (KL divergence) [KL51], $D_{\mathrm{KL}}(P_s||P_r) = \sum_i P_s(i) \cdot \log \frac{P_s(i)}{P_r(i)}$, to measure the similarity of the empirical distributions shown in Figure 13. The KL-divergence scores are given in Table 4. Our method has significantly smaller scores, which demonstrates that it generates motions that are statistically more similar to real data than the baselines. This indicates model-based simulation in human-solution space is superior to pure model-based methods, which is true for both motion dynamics (velocity and acceleration) and states (density). Therefore, our simulations are statistically more similar to data by a large margin.

It should be noted that Heter-Sim [RXX*21] investigated the

human-solution space as well. Our method differs significantly from it in both the data-driven scheme and the physics-based controls, resulting in more realistic crowd behaviors. From a data-driven standpoint, our method uses state similarity to simulate real-world continuous velocity changes by referencing second-order motion dynamics (acceleration) from data. Heter-Sim, in particular, tends to choose a new velocity that is similar to an agent's current one in order to maintain velocity continuity. As a result, the majority of the accelerations calculated from it tend to be close to zero (see Figures 13(c)-(d)), resulting in unnatural straight trajectories with nearly constant velocities (see Figure 12(d)). Quantitatively, our method's lower motion dynamics scores than Heter-Sim (see Table 4) show that it outperforms in generating statistically more similar motions to the real data. In terms of physics-based control, our method employs continuous instantaneous collision detection and collision time-based anticipatory collision avoidance to reduce possible congestion while generating smooth velocity changes in narrow scenarios such as those found in real data [ZKSS12]. However, in narrow scenarios, the simple distance-based collision avoidance model used in Heter-Sim may ignore anticipatory collision neighbors who will pass through the agent during the long-term time threshold, increasing the risk of congestion. Due to limited dataset for collision avoidance, in Heter-Sim, unnatural turning and velocity jittering may arise frequently in the narrow hallway scenario, resulting in unsmooth trajectories (see Figure 12(d)). The smoother trajectories in Figure 12(e) and lower score for the minimum distance in Table 4 of our result show that our interaction model outperforms Heter-Sim both qualitatively and quantitatively.

To further validate our acceleration-aware data-driven optimization scheme's superiority in generating more plausible crowd behaviors than Heter-Sim [RXX*21], we conduct an ablation study by introducing the macroscopic control model used in our method to Heter-Sim (we call it Heter-Sim++) and comparing Heter-Sim++ with both Heter-Sim and our method. The macroscopic continuum model calculates the control direction and desired speed for each

Table 3: *The parameters of each compared models learned from real-world data [ZKSS12]. N.D. stands for neighbor distance.*

Scenario	Our method	PowerLaw [KSG14]	Vision-based method [DMN*17]	Heter-Sim [RXX*21]	Heter-Sim++
Hallway	$w_{dir}, w_{mag} = 0.83$ $w_{dc} = 0.09$ $w_{insCA} = 0.37$ $w_{antiCA} = 0.60$ $w_{mdir}, w_{mmag} = 0.62$	$m = 1.99, k = 1.47$ $N.D. = 9.92$ $\tau_0 = 3.05, ksi = 0.03$ $s_{comf} = 1.20$	$\sigma_{\alpha_g} = 1.86$ $\sigma_s = 3.07$ $\sigma_{ttca} = 2.16$ $\sigma_{dca} = 0.06$ $s_{comf} = 0.89$	$w_{m1} = 1.01, w_{m2} = 1.29$ $w_{c1} = 1.06, w_{c2} = 0.70$ $w_d = 0.53$ $w_{sc} = 1.44$	$w_{m1} = 0.85, w_{m2} = 1.14$ $w_{c1} = 0.52, w_{c2} = 0.97$ $w_d = 1.08$ $w_{sc} = 1.02$
Street	$w_{dir}, w_{mag} = 0.75$ $w_{dc} = 0.21$ $w_{insCA} = 0.50$ $w_{antiCA} = 0.51$ $w_{mdir}, w_{mmag} = 0.93$	$m = 2.10, k = 1.40$ $N.D. = 10.16$ $\tau_0 = 3.01, ksi = 0.06$	$\sigma_{\alpha_g} = 1.79$ $\sigma_s = 1.99$ $\sigma_{ttca} = 1.43$ $\sigma_{dca} = 0.14$	$w_{m1} = 0.67, w_{m2} = 0.58$ $w_{c1} = 1.07, w_{c2} = 0.88$ $w_d = 1.60$ $w_{sc} = 1.47$	-

Table 4: *Benchmark scores on the trajectory and statistical similarity to the real data [ZKSS12], the lower is the better.*

For simplicity, the values of the ADM and PLM are normalized by sample size. The best results are highlighted in **bold** font.

	Trajectory similarity		KL divergence			
	ADM	PLM	Velocity	Distance	Acceleration x	Acceleration y
Continuum [TCP06]	2.1903	0.7499	0.9511	0.2379	1.4695	0.7494
PowerLaw [KSG14]	1.5765	1.5932	0.6771	0.9169	1.4695	1.3958
Vision-based [DMN*17]	1.3304	0.9679	0.5134	0.7424	0.3740	0.3701
Heter-Sim [RXX*21]	1.2373	0.8719	0.0847	0.8338	1.2715	1.1917
Heter-Sim++	1.1032	0.6576	0.0482	0.4152	1.2741	1.0929
OurDirCon	0.9447	0.5405	0.0779	0.1705	0.1346	0.2206
Our method	**0.8763**	**0.5176**	**0.0089**	**0.0806**	**0.0598**	**0.0130**

agent in Heter-Sim++ (refer to the preferred direction and preferred speed in Equation 15, respectively). Heter-Sim++ shares the same parameters as the previous experiments, and the weights of the energy terms in Heter-Sim++ are also learned from real-world data [ZKSS12] by the genetic algorithm (see Table 3). Figure 14(a) shows a snapshot of the visual result. Figure 14(b) depicts the generated trajectories and shows how incorporating the macroscopic control model produces better visual results that are close to the real data. However, as shown in Figure 14(b), unnatural straight trajectories or velocity jittering may still occur, resulting in unsmooth behavior.

Quantitatively, as shown in Table 4, our method achieves lower trajectory similarity scores, indicating that it outperforms Heter-Sim++ in terms of generating trajectories that are more similar to the real data. We also compare the distributions of our method and Heter-sim++ (see Figure 13). According to the KL-divergence scores shown in Table 4, the scores of our result's motion dynamics and state are both significantly lower than those of Heter-Sim++, demonstrating our method's superior ability to generate more realistic crowd behaviors. We also demonstrate that the hierarchical nature of crowd motions can improve the visual quality of a simulation method by comparing Heter-Sim++ to the original Heter-Sim method [RXX*21]. Heter-Sim++ achieves more accurate visual results and lower quantitative scores for performance about states (ADM, PLM, and distance distribution) than the original Heter-Sim method, according to the statistical results shown in Table 4.

To further assess the adaptability and plausibility of our acceleration-aware data-driven optimization scheme when leveraging different goal-directed control models, we first build *OurDirCon*, a model that replaces the continuum model in our approach with the local direction control used in the baseline methods [KSG14; DMN*17; RXX*21]. Then, we compare OurDirCon to these baseline methods. OurDirCon's initialization method and parameters are the same as that of our method. Figure 15 shows the visual results of OurDirCon. When compared to the baselines (see Figures 12(b)-12(d)), OurDirCon produces smoother trajectories (see Figure 15(b)). This is because, under the control of our agent-agent interaction model, the agents tend to move in parallel lines to avoid potential collisions (see Figure 15(a)). We also show the distributions in Figure 13 and the quantitative metrics in Table 4. According to Table 4, OurDirCon outperforms the baseline methods in terms of both motion dynamics and states. This demonstrates that when using different goal-directed control models, our acceleration-aware data-driven optimization scheme can generate motions statistically more similar to real-world motion decision.

Comparisons of trajectory in a street scenario. Because of the confined space, the agents tend to have similar velocities in the previous hallway scenario. In this experiment, we use the PowerLaw method [KSG14], the vision-based method [DMN*17], the Heter-Sim method [RXX*21], and our method to simulate a sparse street scenario similar to [LCL07], where agents' velocity choices are more diverse. The compared Heter-Sim method [RXX*21] makes
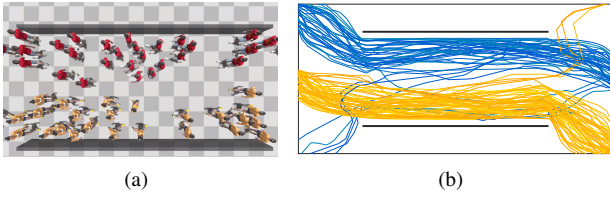
Figure 14: *The visual results of Heter-Sim++: (a) a snapshot of the simulation result, and (b) the generated trajectories.*
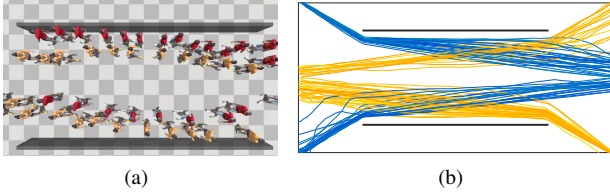
Figure 15: *The visual results of OurDirCon: (a) a snapshot of the simulation result, and (b) the generated trajectories.*

use of a dataset from [LCL07]. We assess the plausibility of the generated trajectories by comparing them to the real ones. 148 agents are simulated in about 6 minutes at 25 fps, just like in the data. During simulation, each agent follows a different short-term trajectory based on its corresponding real-world trajectory. Each agent's comfort speed is its average speed when moving toward its current goal in real data. In our experiment, we set the length of each cell in the grid map to 0.1m in order to accurately locate each agent's goal.

The ADM and PLM metrics proposed by [WGO*14] are used to quantitatively evaluate the generated trajectories, and the results are shown in Table 5. When compared to the baseline methods [KSG14; DMN*17; RXX*21], the quantitative results show that our method has the lowest score. This indicates that the trajectories produced by our method are more plausible than those produced by the other methods. In comparison to the non-data-driven PowerLaw method [KSG14] and the vision-based method [DMN*17], our method can generate more realistic motion decisions by referencing the real dataset. The Heter-Sim method [RXX*21] tends to enforce continuous motion by selecting a new velocity similar to an agent's current one, which may take longer time to steer towards the goal when the goal is changed dynamically. Our acceleration-aware data-driven optimization scheme, on the other hand, can mimic natural velocity changes from real data by employing the state similarity energy, which leads the agent to quickly find a new velocity to move toward its new goal.

6.3. Method Analysis

6.3.1. Human-solution prior knowledge

In the following, we analyze our method using the narrow hallway scenario in Section 6.2.2 to demonstrate the significance of utilizing the human-solution space in two ways: comparing our method

Table 5: *Benchmark scores on the trajectory similarity to the real data [LCL07], the lower is the better. The ADM and PLM methods' values are normalized by sample size. The best results are highlighted in **bold** font.*

	PowerLaw [KSG14]	Vision-based [DMN*17]	Heter-Sim [RXX*21]	Our method
ADM	0.0946	0.3139	0.0977	**0.0440**
PLM	0.2828	0.4825	0.2494	**0.0685**
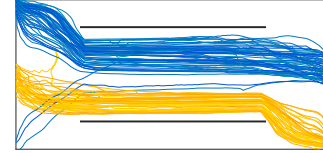
Figure 16: *The trajectories generated by the least-human-solution model.*

with a *least-human-solution model* that solves Equation 2 without referencing a dataset and a *reduced-human-solution model* that references an incomplete dataset to demonstrate the significance of referencing the human-solution space and illustrate the trade off between quality and size of dataset.

Least-human-solution model. The key assumption of our research is that human-solution is a just a subspace of the entire solution space, and solving the optimization in human-solution space gives more realistic motions than in the whole solution space. To investigate this question, we build a least-human-solution model that uses the same parameters and initialization method as those in Section 6.2.2. The comparison is conducted by solving Equation 2 with different amounts of human-solution knowledge. As a baseline, we set the energy terms for state similarity to 0 and constrain the magnitudes of velocities within $[v_{dmin}, v_{dmax}]$, where v_{dmin} and v_{dmax} are the minimal and maximal speeds in the real data. This way, we induce the minimal amount of knowledge from the data, i.e. only knowing the speed range but no motion dynamics.

The results are shown in Figure 16. Compared with other methods/settings shown in Figure 12, the least-human-solution model can simulate reasonable trajectories in the sense that flows are also relatively separate as in real data in Figure 12(f). We also conduct numerical comparisons. The comparisons of distributions are shown in Figure 17, and the KL divergence are shown in Table 6. Although the least-human-solution model visually generates similar results, its distributions on crowd state and dynamics are vastly different from the real data (Figure 17). This shows how vital it is to explicitly incorporate the relevant energy terms in our model.

Reduced-human-solution model. Although distinguishing between the human-solution space and the entire solution space is simple, the question of how much prior knowledge is required remains. To investigate this question, a reduced-human-solution model is built. The reduced-human-solution space is generated from the entire human-solution space in this experiment by randomly removing a portion of the entire dataset. Experiments with four different dataset sizes were tested by randomly selecting 20%,

Table 6: *KL divergence of our result by referencing different datasets on the statistical similarity to the ground-truth data from [ZKSS12].*

|	Least-human-solution	Reduced-human-solution			
		$D_{20\%}$	$D_{40\%}$	$D_{60\%}$	$D_{80\%}$
Velocity	0.2013	0.0310	0.0215	0.0203	0.0202
Distance	0.4484	0.1943	0.1785	0.1639	0.1817
Acceleration x	0.3509	0.1233	0.1070	0.0999	0.0974
Acceleration y	0.3657	0.0155	0.0093	0.0069	0.0058
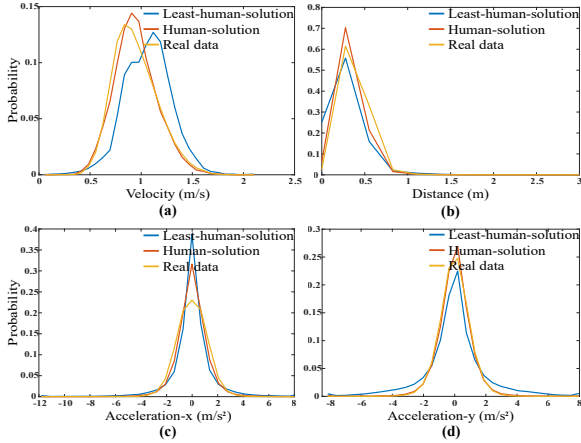
Figure 17: *The distribution comparisons of the least-human-solution results, our full model and the real data [ZKSS12].*
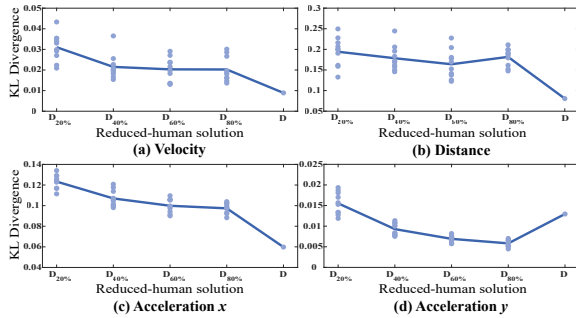
Figure 18: *Statistical figures for comparing the results of the reduced-human solution model with different dataset sizes to our full model. The KL divergence of each experiment is depicted as light blue scatter points.*

To demonstrate the trend of the result with increasing dataset size, the average KL divergence of each reduced-human solution model is concatenated into a dark blue polyline.

40%, 60% and 80% of the total dataset and naming them $D_{20\%}$, $D_{40\%}$, $D_{60\%}$, $D_{80\%}$, and each experiment was iterated ten times. In Figure 18, we compare the distributions of the reduced-human-solution model with our full model (D) in terms of KL-divergence, and show intuitively the curve of the average KL-divergence of each experiment. Table 6 shows the corresponding numerical metrics, indicating that, while reducing the size of the dataset reduces

Table 7: *KL divergence of the ablation studies about the energy terms on the statistical similarity to the real data [ZKSS12].*

	Ablation on *state similarity*	Ablation on *macroscopic control*
Velocity	0.1365	0.0212
Distance	0.2575	0.0685
Acceleration x	0.3997	0.1565
Acceleration y	0.1208	0.0098

(a) Ablation study on the state similarity energy term. (b) Ablation study on the global control energy term.
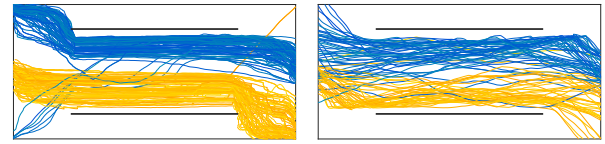
Figure 19: *The trajectories generated by the ablation experiments.*

the plausibility of our method, it still outperforms the compared methods, implying that our method has the potential to be used to recover incomplete trajectory datasets while maintaining accuracy.

6.3.2. Ablations on energy terms

To further demonstrate the importance of the behavioral models in our method, using the narrow hallway scenario in Section 6.2.2, we show the results of several ablation studies. The ablation studies mainly show how the result changes with pruning an energy term in Equation 2.

State similarity. In this experiment, the weight of the state similarity energy term is 0 while the others are set the same as those in Section 6.2.2. The generated trajectories of this experiment are qualitatively shown in Figure 19(a). We compare the distribution similarity in Figure 20, and the KL divergence in Table 7. Comparing with our quantitative result in Table 4, removing the state similarity term leads to severe performance degradation on motion dynamics.

Macroscopic control. In this experiment, the weight of the macroscopic motion control energy term is 0 while the others are set the same as those in Section 6.2.2. The generated trajectories are shown in Figure 19(b). We compare the distribution similarity in Figure 20, and the KL divergence in Table 7. Compared with our quantitative result shown in Table 4, without macroscopic control in our model, the generated trajectories are greatly different from
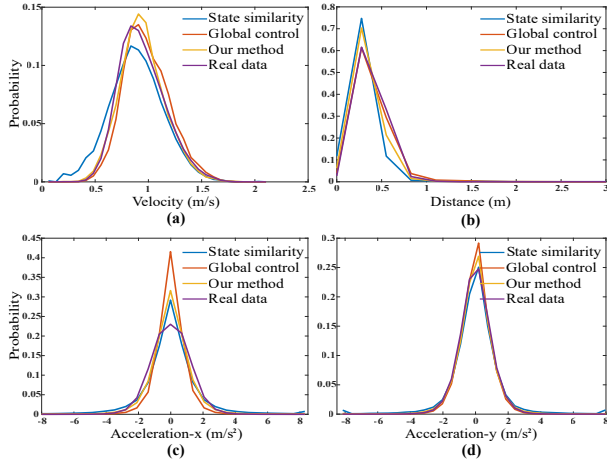
Figure 20: *The distribution comparisons of the ablation experiments on the state similarity energy term and the global control energy term, our full model, and the real data [ZKSS12].*

the real ones (see Figure 12(e)), as there is no global control during simulation, interactions among agents may change the motion tendency of an agent and result in a relatively different trajectory compared with the ground truth in the real data. Note that ablating the macroscopic control term has a small influence on the motion dynamics, as the state similarity term tries to imitate the motion decision mechanism from the real data.

6.3.3. Local navigation

Because collision avoidance is an important component of local navigation in crowd simulation, we use an antipodal scenario with 20 agents to compare our method to the RVO method [vdBLM08] and Heter-Sim [RXX*21]. In the antipodal scenario, the agents are initialized on a circle with the same spacing, and each agent attempts to move to the circle's antipodal position. The datasets used in Heter-Sim [RXX*21] and our method are both from [LCL07]. All shared parameters (e.g., agent radius, initial velocity and preferred speed, simulation timestep) are initialized in the same way. For the remaining parameters, we just use the ones of each compared model in their papers [vdBLM08] [RXX*21]. Our model's weights are $[1.0, 1.0, 1.0, 0.5, 0.5, 0, 1.0, 1.0]$. To avoid sudden velocity changes when potential collisions are detected, we empirically set both collision avoidance weights to 0.5 in our method.

Figures 21(a), 21(b) and 21(c) show the trajectories generated by each method with the simulation $\Delta t = 0.04s$. The results show that our method produces smoother trajectories than RVO [vdBLM08] and Heter-Sim [RXX*21]. Because the RVO model is limited to local planning in a small neighborhood for better performance, there are sudden motion changes around the center of the circle in Figure 21(a). The results of Heter-Sim (Figure 21(b)) and our method (Figure 21(c)) show that using a long-range collision avoidance technique can prevent intense local interactions near the circle's center. In Figure 21(b), agents may detour to reach their goals, as in Heter-Sim, when the distance between two agents is less than a threshold, the discontinuous local interaction method causes agents

(a) RVO [vdBLM08]. (b) Heter-Sim [RXX*21].

(c) Our method with $\Delta t = 0.04s$. (d) Our method with $\Delta t = 0.1s$.
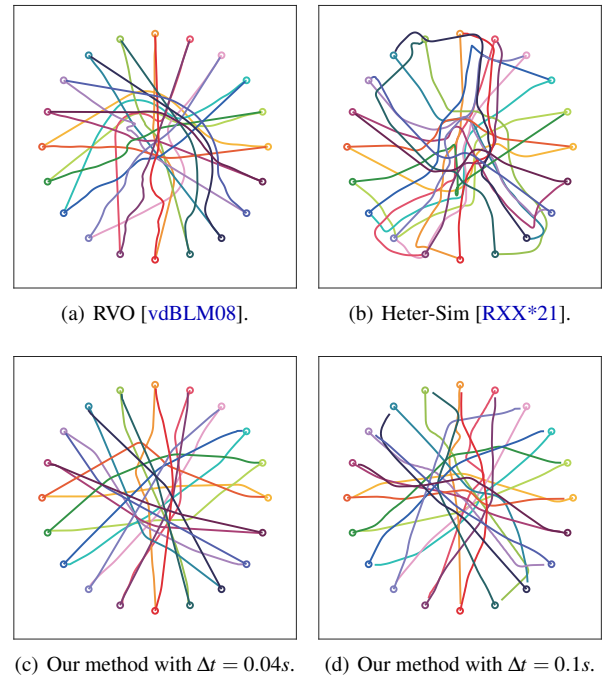
Figure 21: *Local navigation comparisons between RVO [vdBLM08] (a), Heter-Sim [RXX*21] (b), and our method (c, d). Each agent's starting position is represented by the corresponding disk on the trajectory.*

with nearby goals to move at the same velocity to avoid collisions, causing some agents to deviate from their goals.

As our method employs a continuous collision detection technique, we also present a result with $\Delta t = 0.1s$ to demonstrate our method's ability to generate collision-free continuous trajectories for agents with large timesteps. Figure 21(d) depicts how the agents avoid potentially damaging collisions near the circle's center.

6.3.4. Data generalization to different scenarios

Since the importance of human-solution space and each behavioral model has been demonstrated in the preceding sections, the remaining question is how general our method is in using a dataset to generate other scenarios that differ from the dataset's scenario. The visual results shown in Section 6 have answered this question (please refer to Figures 5 - 8). Furthermore, to demonstrate the generalization of our method, we generate a one-way bottleneck scenario similar to [SPS*09] by referencing the bi-directional hallway dataset from [ZKSS12] and generate a bi-directional hallway scenario the other way around, where the generated crowds are quite different from that of their referenced datasets. Both experiments use the same initialization method and parameters as their corresponding similar simulation experiments in Section 6.1 for the bottleneck scenario and Section 6.2.2 for the hallway scenario. Figure 22 shows snapshots of simulation results compared to ground-truth real data. In Figure 23, we compare the generated trajectories of our method with ground-truth real data by referencing differ-
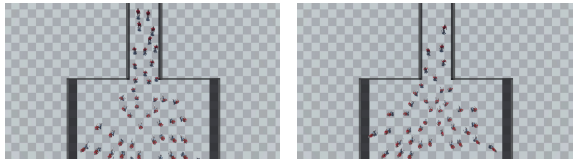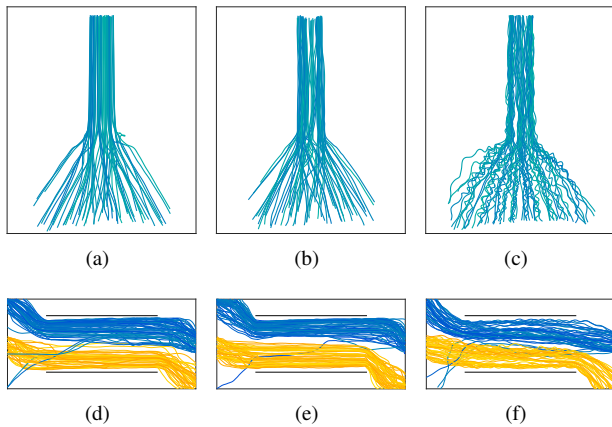
(a) One-way bottleneck result referencing the bidirectional pedestrian dataset from [ZKSS12].

(b) Real one-way bottleneck data from [SPS*09].
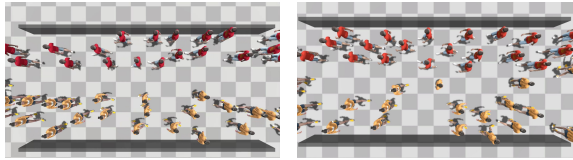
(c) Bi-directional hallway result referencing the one-way bottleneck dataset from [SPS*09].

(d) Real bi-directional pedestrian data from [ZKSS12].

Figure 22: *Results by referencing datasets from various scenarios.*

(a) (b) (c)

(d) (e) (f)

Figure 23: *Comparisons of the trajectories of our results referencing various datasets from various scenarios and real data. The trajectories of different crowds in a one-way bottleneck are shown in (a)-(c): (a) referencing the bidirectional dataset [ZKSS12], (b) referencing the bottleneck evacuation dataset [SPS*09], and (c) real trajectories [SPS*09]. (d)-(f) are crowd trajectories in the bidirectional hallway scenario: (d) referencing [SPS*09], (e) referencing [ZKSS12], and (f) real trajectories [ZKSS12].*

ent datasets from different real scenarios. Given reasonable motion preferences from the ground-truth real data (e.g., initial speed, maximum speed), our method can find the best velocity from the reference data to match the target behavior observed from the ground-truth real data during simulation. The results show that our method can generate visually similar crowd scenarios even when the referenced dataset's scenario is not similar to the ground-truth scenario.

7. Conclusion

In this paper, we present a generic model-based data-driven continuum method that can generate plausible and scalable crowd animations in different scenarios. Our method is adaptive to different datasets to generate similar scenarios as the real data (see Figures 9(a) - 9(d)). Our method is scalable in generating different scales of crowds in different scenarios in similar environments to real data (see Figures 5(a) and 5(b)). Our method is also able to generate diversified crowds that may differ from the reference data (see Figures 5(c), 7, 8, 22(a), 22(c)). Compared with the state-of-art methods, our method can generate results that are significantly closer to the real data. In addition, our model is fast and can be used for interactive simulation (see Table 2).

Limitations and future work. Similar to other data-driven methods, the quality of the simulation result relies on the quality of the reference data. If the reference data deviates significantly from normal crowds, e.g. only containing large velocities, the simulator might not be able to resolve congestion as the algorithm cannot find a proper velocity to slow down the agents. However, we argue that data with a good variety of velocities can be easily acquired. Even in data with noise, the motion dynamics based on velocity and acceleration can still be reliably extracted. Furthermore, combining data from various crowd scenarios can help to avoid data limitations.

The second limitation is that our approach is 'model-based' compared with pure data-driven models. This suggests that it requires certain expertise to design the underlying physics-based model and our method cannot just 'plug and play' on data. However, we argue this effort provides model explicability which leads to insights of crowd behaviors, rather than fitting data in a black-box manner. Furthermore, we represent the agents as discs with fixed radii. However, human beings in the real world have more complicated personal spaces. We will use more precise geometries for better collision avoidance in future. Furthermore, we only consider explicit motion features, i.e., velocity and acceleration. We intend to investigate more trajectories' characteristics in the future by incorporating deep neural networks into our model to capture arbitrary non-linearity in motion dynamics. Because we use 2D captured data as input, our method can only simulate crowd motions in 2D space. Our method will be expanded to include uneven terrain and complex environments with stairs. Besides, the macroscopic control model in our simulation framework are mainly focused on pedestrian groups with group-desired goals, which may not efficient for agent-desired motion control. Although we can separately model the macroscopic control map for agents with different goals, it involves trade-offs between memory usage and computing performance, especially for large crowds. A feasible solution is to replace the continuum model with other goal-directed motion control methods, e.g., local direction control model used in [RXX*21], A* algorithm for global path planning.

References

[AGR*16] ALAHI, ALEXANDRE, GOEL, KRATARTH, RAMANATHAN, VIGNESH, et al. "Social LSTM: Human Trajectory Prediction in Crowded Spaces". *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*. IEEE Computer Society, 2016, 961–971 2, 3.

[BKHF14] BERSETH, GLEN, KAPADIA, MUBBASIR, HAWORTH, BRANDON, and FALOUTSOS, PETROS. "SteerFit: Automated Parameter Fitting for Steering Algorithms". *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '14. 2014, 113–122 4.

[BKSB15] BOATRIGHT, CORY D., KAPADIA, MUBBASIR, SHAPIRA, JENNIE M., and BADLER, NORMAN I. "Generating a multiplicity of policies for agent steering in crowd simulation". *Comput. Animat. Virtual Worlds* 26.5 (2015), 483–494 3.

[CC14] CHARALAMBOUS, PANAYIOTIS and CHRYSANTHOU, YIORGOS. "The PAG Crowd: A Graph Based Approach for Efficient Data-Driven Crowd Simulation". *Comput. Graph. Forum* 33.8 (2014), 95–108 2, 3.

[CJH*19] CHAO, QIANWEN, JIN, XIAOGANG, HUANG, HEN-WEI, et al. "Force-based Heterogeneous Traffic Simulation for Autonomous Vehicle Testing". *International Conference on Robotics and Automation, ICRA 2019*. IEEE, 2019, 8298–8304 3.

[CKGC14] CHARALAMBOUS, PANAYIOTIS, KARAMOUZAS, IOANNIS, GUY, STEPHEN J, and CHRYSANTHOU, YIORGOS. "A data-driven framework for visual crowd analysis". *Computer Graphics Forum*. Vol. 33. 7. Wiley Online Library. 2014, 41–50 4.

[CLH*23] CHAO, QIANWEN, LIU, PENGFEI, HAN, YI, et al. "A calibrated force-based model for mixed traffic simulation". *IEEE transactions on visualization and computer graphics* 29.3 (2023), 1664–1677 4.

[CP15] CASADIEGO, LUISELENA and PELECHANO, NURIA. "From One to Many: Simulating Groups of Agents with Reinforcement Learning Controllers". *Intelligent Virtual Agents - 15th International Conference, IVA 2015*. Vol. 9238. Lecture Notes in Computer Science. Springer, 2015, 119–123 4.

[DMN*17] DUTRA, TEOFILO BEZERRA, MARQUES, RICARDO, NETO, JOAQUIM B. CAVALCANTE, et al. "Gradient-based steering for vision-based crowd simulation algorithms". *Comput. Graph. Forum* 36.2 (2017), 337–348 3, 11–14.

[GCC*10] GUY, STEPHEN J., CHHUGANI, JATIN, CURTIS, SEAN, et al. "PLEdestrians: A Least-Effort Approach to Crowd Simulation". *Proceedings of the 2010 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2010*. Eurographics Association, 2010, 119–128 2, 3, 7.

[GJF*18] GUPTA, AGRIM, JOHNSON, JUSTIN, FEI-FEI, LI, et al. "Social GAN: Socially Acceptable Trajectories With Generative Adversarial Networks". *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*. IEEE Computer Society, 2018, 2255–2264 4.

[GNCL14] GOLAS, ABHINAV, NARAIN, RAHUL, CURTIS, SEAN, and LIN, MING C. "Hybrid Long-Range Collision Avoidance for Crowd Simulation". *IEEE Transactions on Visualization & Computer Graphics* 20.07 (2014), 1022–1034 5, 6.

[HCJ21] HAN, YI, CHAO, QIANWEN, and JIN, XIAOGANG. "A simplified force model for mixed traffic simulation". *Comput. Animat. Virtual Worlds* 32.1 (2021), e1974 3.

[HM95] HELBING, DIRK and MOLNÁR, PÉTER. "Social force model for pedestrian dynamics". *Phys. Rev. E* 51 (5 1995), 4282–4286 2, 3.

[HOD15] HUGHES, ROWAN T., ONDREJ, JAN, and DINGLIANA, JOHN. "DAVIS: density-adaptive synthetic-vision based steering for virtual crowds". *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, MIG 2015*. ACM, 2015, 79–84 3.

[HXZW20] HE, FEIXIANG, XIANG, YUANHANG, ZHAO, XI, and WANG, HE. "Informative scene decomposition for crowd analysis, comparison and simulation guidance". *ACM Trans. Graph.* 39.4 (2020), 50 3, 7.

[JCP*10] JU, EUNJUNG, CHOI, MYUNG GEOL, PARK, MINJI, et al. "Morphable crowds". *ACM Trans. Graph.* 29.6 (2010), 140 2, 3.

[JXM*10] JIANG, HAO, XU, WENBIN, MAO, TIANLU, et al. "Continuum crowd simulation in complex environments". *Comput. Graph.* 34.5 (2010), 537–544 3.

[KBB*16] KIM, SUJEONG, BERA, ANIKET, BEST, ANDREW, et al. "Interactive and adaptive data-driven crowd simulation". *2016 IEEE Virtual Reality (VR)*. IEEE. 2016, 29–38 4.

[KGH*15] KIM, SUJEONG, GUY, STEPHEN J, HILLESLAND, KARL, et al. "Velocity-based modeling of physical interactions in dense crowds". *The Visual Computer* 31.5 (2015), 541–555 3.

[KGM13] KIM, SUJEONG, GUY, STEPHEN J., and MANOCHA, DINESH. "Velocity-based modeling of physical interactions in multi-agent simulations". *The ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '13*. ACM, 2013, 125–133 3.

[KHvBO09] KARAMOUZAS, IOANNIS, HEIL, PETER, van BEEK, PASCAL, and OVERMARS, MARK H. "A Predictive Collision Avoidance Model for Pedestrian Simulation". *Motion in Games, Second International Workshop, MIG 2009, Zeist, The Netherlands, November 21-24, 2009. Proceedings*. Vol. 5884. Lecture Notes in Computer Science. Springer, 2009, 41–52 3.

[KL51] KULLBACK, S. and LEIBLER, R. A. "On Information and Sufficiency". *The Annals of Mathematical Statistics* 22.1 (1951), 79–86. ISSN: 00034851 12.

[KO10] KARAMOUZAS, IOANNIS and OVERMARS, MARK H. "A Velocity-Based Approach for Simulating Human Collision Avoidance". *Intelligent Virtual Agents, 10th International Conference, IVA 2010, Philadelphia, PA, USA, September 20-22, 2010. Proceedings*. Vol. 6356. Lecture Notes in Computer Science. Springer, 2010, 180–186 3.

[KSG14] KARAMOUZAS, IOANNIS, SKINNER, BRIAN, and GUY, STEPHEN J. "Universal Power Law Governing Pedestrian Interactions". *Phys. Rev. Lett.* 113 (23 2014), 238701 2, 3, 7, 11–14.

[KSH*12] KAPADIA, MUBBASIR, SINGH, SHAWN, HEWLETT, WILLIAM, et al. "Parallelized egocentric fields for autonomous navigation". *Vis. Comput.* 28.12 (2012), 1209–1227 3.

[KSNG17] KARAMOUZAS, IOANNIS, SOHRE, NICK, NARAIN, RAHUL, and GUY, STEPHEN J. "Implicit crowds: optimization integrator for robust crowd simulation". *ACM Trans. Graph.* 36.4 (2017), 136:1–136:13 2, 3.

[LCL07] LERNER, ALON, CHRYSANTHOU, YIORGOS, and LISCHINSKI, DANI. "Crowds by Example". *Comput. Graph. Forum* 26.3 (2007), 655–664 2, 3, 9, 11, 13, 14, 16.

[LCM*18] LUO, LINBO, CHAI, CHENG, MA, JIANFENG, et al. "ProactiveCrowd: Modelling Proactive Steering Behaviours for Agent-Based Crowd Simulation". *Comput. Graph. Forum* 37.1 (2018), 375–388 3.

[LCMP19] LÓPEZ, AXEL, CHAUMETTE, FRANÇOIS, MARCHAND, ÉRIC, and PETTRÉ, JULIEN. "Character navigation in dynamic environments based on optical flow". *Comput. Graph. Forum* 38.2 (2019), 181–192 3.

[LCSC09] LERNER, ALON, CHRYSANTHOU, YIORGOS, SHAMIR, ARIEL, and COHEN-OR, DANIEL. "Data driven evaluation of crowds". *International Workshop on Motion in Games*. Springer. 2009, 75–83 4.

[LWL18] LEE, JAEDONG, WON, JUNGDAM, and LEE, JEHEE. "Crowd simulation by deep reinforcement learning". *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games, MIG 2018*. ACM, 2018, 2:1–2:7 4.

[MHT11] MOUSSAÏÁD, MEHDI, HELBING, DIRK, and THERAULAZ, GUY. "How simple rules determine pedestrian behavior and crowd disasters". *Proceedings of the National Academy of Sciences* 108.17 (2011), 6884–6888 3.

[MON*16] MOHAMAD, SHAMSUL, OSHITA, MASAKI, NOMA, TSUKASA, et al. "Making decision for the next step in dense crowd simulation using support vector machines". *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry, VRCAI 2016*. ACM, 2016, 281–287 3.

[NGCL09] NARAIN, RAHUL, GOLAS, ABHINAV, CURTIS, SEAN, and LIN, MING C. "Aggregate dynamics for dense crowd simulation". *ACM Trans. Graph.* 28.5 (2009), 122 2, 3.

[OPOD10] ONDREJ, JAN, PETTRÉ, JULIEN, OLIVIER, ANNE-HÉLÈNE, and DONIKIAN, STÉPHANE. "A synthetic-vision based steering approach for crowd simulation". *ACM Trans. Graph.* 29.4 (2010), 123:1–123:9 3.

[Osh19] OSHITA, MASAKI. "Agent navigation using deep learning with agent space heat map for crowd simulation". *Comput. Animat. Virtual Worlds* 30.3-4 (2019), e1878 4.

[PAB07] PELECHANO, NURIA, ALLBECK, JAN M., and BADLER, NORMAN I. "Controlling individual agents in high-density crowd simulation". *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2007*. Eurographics Association, 2007, 99–108 3.

[PCQ11] PARK, SEUNG IN, CAO, YONG, and QUEK, FRANCIS. "Large scale crowd simulation using a hybrid agent model". *Motion in Games* (2011), 2, 3.

[POO*09] PETTRÉ, JULIEN, ONDREJ, JAN, OLIVIER, ANNE-HÉLÈNE, et al. "Experiment-based modeling, simulation and validation of interactions between virtual walkers". *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2009*. ACM, 2009, 189–198 3.

[PPD07] PARIS, SÉBASTIEN, PETTRÉ, JULIEN, and DONIKIAN, STÉPHANE. "Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach". *Comput. Graph. Forum* 26.3 (2007), 665–674 3.

[PvdBC*11] PATIL, SACHIN, van den BERG, JUR P., CURTIS, SEAN, et al. "Directing Crowd Simulations Using Navigation Fields". *IEEE Trans. Vis. Comput. Graph.* 17.2 (2011), 244–254 3.

[Rey87] REYNOLDS, CRAIG W. "Flocks, herds and schools: A distributed behavioral model". *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*. ACM, 1987, 25–34 2, 3.

[RXX*21] REN, JIAPING, XIANG, WEI, XIAO, YANGXI, et al. "HeterSim: Heterogeneous Multi-Agent Systems Simulation by Interactive Data-Driven Optimization". *IEEE Trans. Vis. Comput. Graph.* 27.3 (2021), 1953–1966 2, 3, 5, 6, 8, 11–14, 16, 17.

[SHW*18] SHEN, YIJUN, HENRY, JOSEPH, WANG, HE, et al. "Data-Driven Crowd Motion Control With Multi-Touch Gestures". *Computer Graphics Forum* 37.6 (2018), 382–394 3.

[SKFR09] SINGH, SHAWN, KAPADIA, MUBBASIR, FALOUTSOS, PETROS, and REINMAN, GLENN. "Steerbench: a benchmark suite for evaluating steering behaviors". *Computer Animation and Virtual Worlds* 20.5-6 (2009), 533–548 4.

[SPS*09] SEYFRIED, ARMIN, PASSON, OLIVER, STEFFEN, BERNHARD, et al. "New insights into pedestrian flow through bottlenecks". *Transportation Science* 43.3 (2009), 395–406 9, 16, 17.

[TCP06] TREUILLE, ADRIEN, COOPER, SETH, and POPOVIC, ZORAN. "Continuum crowds". *ACM Trans. Graph.* 25.3 (2006), 1160–1168 2, 3, 7, 11–13.

[TWCL18] TSAI, TSUNG-YU, WONG, SAI-KEUNG, CHOU, YI-HUNG, and LIN, GUAN-WEN. "Directing virtual crowds based on dynamic adjustment of navigation fields". *Comput. Animat. Virtual Worlds* 29.1 (2018), e1765 3.

[vdBLM08] Van den BERG, JUR P., LIN, MING C., and MANOCHA, DINESH. "Reciprocal Velocity Obstacles for real-time multi-agent navigation". *2008 IEEE International Conference on Robotics and Automation, ICRA 2008*. IEEE, 2008, 1928–1935 3, 16.

[vTGL*20] Van TOLL, WOUTER, GRZESKOWIAK, FABIEN, LÓPEZ-GANDIÉA, AXEL, et al. "Generalized Microscropic Crowd Simulation using Costs in Velocity Space". *I3D '20: Symposium on Interactive 3D Graphics and Games*. ACM, 2020, 6:1–6:9 2, 3.

[WGO*14] WOLINSKI, DAVID, GUY, STEPHEN J., OLIVIER, ANNE-HÉLÈNE, et al. "Parameter estimation and comparative evaluation of crowd simulations". *Comput. Graph. Forum* 33.2 (2014), 303–312 4, 11, 14.

[WJDL13] WU, QIANQIAN, JI, QINGGE, DU, JINGHONG, and LI, XIAOLIAN. "Simulating the local behavior of small pedestrian groups using synthetic-vision based steering approach". *12th ACM International Conference on Virtual Reality Continuum and Its Applications in Industry, VRCAI 2013*. ACM, 2013, 41–50 3.

[WLJT17] WEISS, TOMER, LITTENEKER, ALAN, JIANG, CHENFANFU, and TERZOPOULOS, DEMETRI. "Position-based multi-agent dynamics for real-time crowd simulation". *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 2017*. Eurographics Association / ACM, 2017, 27:1–27:2 3.

[WO16] WANG, HE and O'SULLIVAN, CAROL. "Globally Continuous and Non-Markovian Crowd Activity Analysis from Videos". *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*. Vol. 9909. Lecture Notes in Computer Science. Springer, 2016, 527–544 8.

[WOO16] WANG, HE, ONDREJ, JAN, and O'SULLIVAN, CAROL. "Path patterns: analyzing and comparing real and simulated crowds". *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, 2016*. ACM, 2016, 49–57 7.

[WOO17] WANG, HE, ONDREJ, JAN, and O'SULLIVAN, CAROL. "Trending Paths: A New Semantic-Level Metric for Comparing Simulated and Real Crowd Data". *IEEE Trans. Vis. Comput. Graph.* 23.5 (2017), 1454–1464 3, 4, 8.

[XYWJ20] XIANG, WEI, YAO, XINRAN, WANG, HE, and JIN, XIAOGANG. "FASTSWARM: A data-driven framework for real-time flying insect swarm simulation". *Computer Animation and Virtual Worlds* 31.4-5 (2020), e1957 3.

[YLRÖ19] YANG, DONGFANG, LI, LINHUI, REDMILL, KEITH, and ÖZGÜNER, ÜMIT. "Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus". *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, 899–904 9.

[YZLL19] YAO, ZHENZHEN, ZHANG, GUIJUAN, LU, DIANJIE, and LIU, HONG. "Data-driven crowd evacuation: A reinforcement learning method". *Neurocomputing* 366 (2019), 314–327 4.

[YZLL20] YAO, ZHENZHEN, ZHANG, GUIJUAN, LU, DIANJIE, and LIU, HONG. "Learning crowd behavior from real data: A residual network method for crowd simulation". *Neurocomputing* 404 (2020), 173–185 4.

[ZIK11] ZANLUNGO, FRANCESCO, IKEDA, TETSUSHI, and KANDA, TAKAYUKI. "Social force model with explicit collision prediction". *EPL (Europhysics Letters)* 93.6 (2011), 68005 3.

[ZKSS12] ZHANG, JUN, KLINGSCH, WOLFRAM, SCHADSCHNEIDER, ANDREAS, and SEYFRIED, ARMIN. "Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram". *Journal of Statistical Mechanics: Theory and Experiment* 2012.02 (2012), P02002 9–13, 15–17.

[ZTC13] ZHAO, MINGBI, TURNER, STEPHEN JOHN, and CAI, WENTONG. "A Data-Driven Crowd Simulation Model Based on Clustering and Classification". *17th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2013*. IEEE Computer Society, 2013, 125–134 3.

[ZWT12] ZHOU, BOLEI, WANG, XIAOGANG, and TANG, XIAOOU. "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents". *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, 2871–2878 9.