ELHINN: Unifying Dense Crowd Simulation Across Scales via Eulerian–Lagrangian Hydrodynamics

Yanshan Zhou, Pingrui Lai, Jiaqi Yu, Cunyan Li, Hua Yang*, and Xiaoyun Zhang

School of Information Science and Electronic Engineering & School of Integrated Circuits, Shanghai Jiao Tong University, Shanghai, China
{shanshan333,laipingrui,jiaqiyu,licunyan,hyang,xiaoyun.zhang}@sjtu.edu.cn

Abstract. Developing effective dense crowd simulation is challenging due to the scale gap between macroscopic collective coherence and microscopic individual realism. Existing macroscopic (Eulerian) methods capture global motion patterns but lack individual trajectories, whereas microscopic (Lagrangian) methods model individual behaviors yet often fail to preserve systemic consistency in dense scenarios. To address this, we propose the Eulerian–Lagrangian Hydrodynamics-Informed Neural Network (ELHINN), a unified cross-scale framework that couples macroscopic velocity evolution with microscopic trajectory refinement by using evolved Eulerian velocity fields as physical priors to guide Lagrangian trajectories. For velocity evolution, we develop an enhanced Hydrodynamics-Informed Neural Network (HINN++) that incorporates a learnable governing equation with Kolmogorov–Arnold Network (KAN) residual correction and environmental boundary conditions, enabling accurate modeling of complex nonlinear dynamics under varying scenarios. For trajectory refinement, we use a Physics-Informed Neural Network (PINN) augmented with an entrance-aware resampling strategy (EARS) and collision-avoidance constraints to ensure stability and fidelity. Experiments on two real-world crowd datasets demonstrate that ELHINN outperforms existing methods in simulating dense crowds across both scales. Codes are available at `https://github.com/shanshan-zys/ELHINN`.

Keywords: Dense Crowd Simulation · Eulerian–Lagrangian Duality · Physics-Informed Modeling

1 Introduction

Crowd simulation has long been popular within computer science due to its close relevance to daily life. As the process of modeling large-scale crowd motions and interaction dynamics [32], it has found wide applications from urban planning and emergency evacuation to virtual reality and video games [14,30]. Despite its broad utility, developing effective approaches for crowd simulation, particularly in dense scenarios, remains a critical and challenging task.

* Corresponding author.

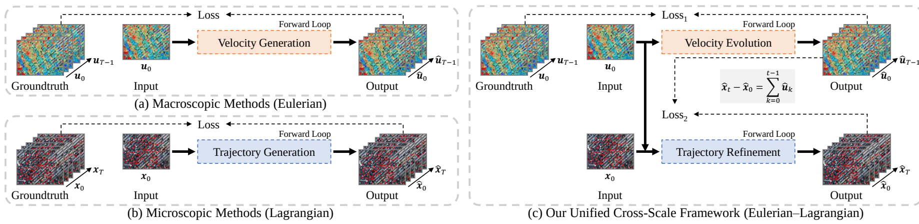

Fig. 1: Comparison of crowd simulation approaches across different scales. Arrows indicate information flow. (a) Macroscopic methods model collective coherence but lack fine-grained individual-level details. (b) Microscopic methods capture individual behaviors but lack systemic consistency and require dense trajectory supervision. (c) Our unified cross-scale framework couples macroscopic velocity evolution with microscopic trajectory refinement, bridging the scale gap to simultaneously ensure global collective coherence and local individual realism.

Current crowd simulation approaches are generally categorized into macroscopic (Eulerian) and microscopic (Lagrangian) scales, as illustrated in Fig. 1. *Macroscopic methods* treat crowds as continuous fluids and model their dynamics using Navier–Stokes equations [4, 16]. These methods efficiently capture global motion patterns in dense scenarios [42], but inherently lack individual-level attributes such as trajectories, limiting their applicability to fine-grained behavior analysis. Conversely, *microscopic methods* rely on either predefined interaction rules [15,33] or neural networks with physical constraints [7,40] to simulate local individual interactions. While these methods produce detailed trajectories, they often struggle in dense crowds due to the scarcity of groundtruth data and the difficulty in preserving collective motion coherence. Between these scales remains a fundamental gap, where the disconnect between global collective coherence and local individual realism leads to the loss of physical information and prevents existing approaches from jointly achieving robustness and fidelity.

However, we observe that Eulerian and Lagrangian descriptions are not isolated domains but complementary representations of the same physical system. While recent fluid dynamics research leverages this duality to reconstruct Eulerian fields from Lagrangian data [35], our core idea is to reverse direction by exploiting Eulerian velocity fields as physical priors to guide Lagrangian trajectories. Based on this insight, we propose the Eulerian–Lagrangian Hydrodynamics-Informed Neural Network (ELHINN), a unified cross-scale framework that explicitly couples Eulerian and Lagrangian representations to bridge the scale gap in dense crowd simulation. Specifically, ELHINN consists of: (1) a *global velocity evolution module* that models macroscopic dynamics via velocity fields in the Eulerian domain, and (2) a *local trajectory refinement module* that derives microscopic Lagrangian trajectories guided by the learned velocity priors.

In this work, we develop the *global velocity evolution module* by extending the Hydrodynamics-Informed Neural Network (HINN) [42] into HINN++, enabling more expressive and adaptive modeling of crowd dynamics. Specifically, we formulate a parameterized and learnable variant of Navier–Stokes equations as the governing equation, allowing adaptive modeling of diverse crowd motion patterns. We integrate a lightweight Kolmogorov–Arnold Network (KAN) [27] for residual correction, leveraging its symbolic expressiveness to capture intricate, heterogeneous nonlinearities beyond universal governing formulations. In addition, we incorporate environmental factors as explicit boundary conditions to improve robustness and generalizability across diverse scenarios. For the *local trajectory refinement module*, we employ a Physics-Informed Neural Network (PINN) [31] to enforce physical consistency, and further introduce an entrance-aware resampling strategy (EARS) together with collision-avoidance constraints to stabilize trajectory generation in dense crowds. With the integrated cross-scale architecture, we conduct extensive experiments on two real-world crowd datasets. The results demonstrate that ELHINN outperforms baseline methods in both macroscopic velocity evolution and microscopic trajectory refinement.

Our contributions are summarized as follows:

– We propose a unified cross-scale Eulerian–Lagrangian framework for dense crowd simulation, bridging the fundamental scale gap between global collective coherence and local individual realism.
– We develop a learnable governing equation augmented with KAN residual correction and explicit boundary conditions, enhancing adaptability to diverse crowd dynamics and generalization across scenarios.
– Extensive experiments on two real-world crowd datasets demonstrate the effectiveness of our framework in both macroscopic velocity evolution and microscopic trajectory refinement.

2 Related Work

2.1 Macroscopic Methods

Commonly used in dense crowd simulation, macroscopic methods leverage the similarities between crowds and continuous fluids, drawing inspiration from fluid dynamics to model large-scale crowds. These methods guide crowd motions via Navier–Stokes equations and simulate crowd dynamics by solving the governing equations [4, 16, 20, 36]. The main difficulty of these methods lies in calculating numerical solutions to Navier–Stokes equations. One common strategy is to use Smoothed Particle Hydrodynamics (SPH) [9, 37, 39] to approximate solutions, but this usually compromises simulation accuracy and increases computational complexity. Inspired by advances in Partial Differential Equation (PDE)-solving neural networks, such as the PDE-Net [28], PINN [31], Navier–Stokes Flow Net (NSFnet) [21], and PDE-Preserved Neural Network (PPNN) [26], HINN [42] integrates Navier–Stokes equations into its network architecture to efficiently model dense crowd motion patterns. While these methods excel at capturing global collective coherence, they nevertheless fail to provide individual-level attributes such as trajectories, thereby limiting their applicability in fine-grained behavior analysis and comprehensive risk assessment.

2.2 Microscopic Methods

In contrast to macroscopic methods, microscopic methods model crowd behaviors by tracking individual trajectories to capture intricate motion details. Early research, such as the Boids [33] and Social Force Model (SFM) [15], relies on predefined physical rules to simulate individual interactions and responses. However, these methods lack universal principles for modeling complex crowd motions. With advances in deep learning, some research employs neural networks, particularly the Long Short-Term Memory (LSTM) [18], to learn interactions from observed trajectories. Representative methods include the Social-LSTM [1] and Spatial-Temporal Graph Attention network (STGAT) [19]. Follow-up research, such as the Physics-informed Crowd Simulator (PCS) [40], further integrates the data-driven learning with physics-based interaction constraints to improve simulation fidelity. However, these methods are largely limited by the scarcity of reliable groundtruth trajectories and the difficulty of maintaining collective coherence in dense crowd scenarios. Recent research attempts to address this scarcity using material point methods [13], but is confined to narrow settings and struggles to generalize across scenarios.

On the other hand, the uncertainty of crowd behaviors and trajectories introduces the multi-modality of crowd motions [24], motivating generative methods for crowd simulation. Early research uses the Generative Adversarial Network (GAN) [11,12] and Variational Auto-Encoder (VAE) [23,29] to synthesize multimodal crowd motions by learning complex data distributions. Subsequent research, such as the Social Physics Informed Diffusion Model (SPDiff) [7, 17], further incorporates SFM into the diffusion model. However, these methods still suffer from the scarcity of groundtruth trajectory data. Recent interactive platforms, including Sora [5], Kling, and Runway [22], leverage vision-language-motion integration to enable dynamic crowd animation. However, these methods remain limited in simulating dense crowds due to issues of physical consistency and scalability. Crucially, a fundamental scale gap persists between macroscopic collective coherence and microscopic individual realism, as existing approaches largely treat these scales as isolated and decoupled.

3 Methodology

3.1 Problem Formulation

Given a group of N individuals with initial states, including the Eulerian velocity field $\boldsymbol{u}_0$ and Lagrangian positions $\boldsymbol{x}_0 = [\boldsymbol{x}_0^1, \boldsymbol{x}_0^2, \ldots, \boldsymbol{x}_0^N]$, along with the environment state defined by the static walkable area $\boldsymbol{A}$, the goal of crowd simulation is to model the joint evolution of both scales over the subsequent T time-steps. Specifically, our objective is twofold: (1) to evolve the sequence of velocity fields $\hat{\boldsymbol{U}} = [\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_{T-1}]$ with $\hat{\boldsymbol{u}}_0 = \boldsymbol{u}_0$; and (2) to resolve individual trajectories $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_T]$ under the guidance of evolved velocity fields, with $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0$ and $\hat{\boldsymbol{x}}_t = [\hat{\boldsymbol{x}}_t^1, \hat{\boldsymbol{x}}_t^2, \ldots, \hat{\boldsymbol{x}}_t^N]$. We formulate the simulation process as an iterative evolution problem. Starting from the given initial states, our unified cross-scale

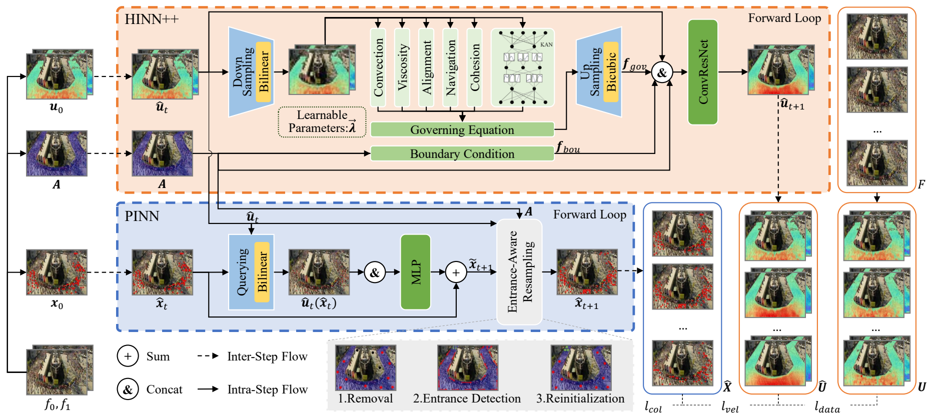

Fig. 2: Overview of the ELHINN framework. The global velocity evolution module HINN++ evolves velocity fields from the initial velocity field and walkable area. Under the guidance of the evolved velocity fields, the local trajectory refinement module PINN resolves individual trajectories from the initial positions and walkable area.

framework progressively maps the current states to the next states. Formally, the coupled process can be formulated as:

$$\hat{\boldsymbol{u}}_{t+1} = \text{HINN}_{++}(\hat{\boldsymbol{u}}_t, \boldsymbol{A}), \tag{1}$$

$$\hat{\boldsymbol{x}}_{t+1} = \text{PINN}(\hat{\boldsymbol{u}}_t, \hat{\boldsymbol{x}}_t, \boldsymbol{A}), \tag{2}$$

where $\hat{\boldsymbol{u}}_t$ and $\hat{\boldsymbol{x}}_t$ denote the evolved velocity field and positions at time-step t, with $\hat{\boldsymbol{u}}_t$ serving as global motion priors in Eq. (2) to guide individual trajectories.

3.2 Framework Overview

As illustrated in Fig. 2, our proposed framework ELHINN performs cross-scale crowd simulation through two sequential modules: (1) HINN++ evolves macroscopic velocity fields by modeling global crowd dynamics conditioned on the initial velocity field $\boldsymbol{u}_0$ and walkable area $\boldsymbol{A}$; and (2) PINN resolves microscopic individual trajectories by modeling position transitions from the evolved velocity fields $\hat{\boldsymbol{U}}$, initial positions $\boldsymbol{x}_0$, and walkable area $\boldsymbol{A}$, and incorporates EARS to handle dynamic inflows and outflows. For clarity of exposition, the complete theoretical foundations supporting our framework are provided in Appendix A.1.

3.3 HINN++: Velocity Evolution

To model complex dense crowd dynamics, we develop HINN++ as the global velocity evolution module with enhanced crowd motion representability and scenario adaptability. HINN++ evolves macroscopic velocity fields in the Eulerian domain with three components: a physics-preserving network embedding the governing equation, a boundary-constrained network enforcing the walkable area, and a predictive network forecasting the next-step velocity field.

Table 1: Physical and social operators proposed by HINN [42].

Category	Operator	Description
Physical	Convection $\boldsymbol{f}_{\text{con}}$	captures nonlinear advection
	Viscosity $\boldsymbol{f}_{\text{vis}}$	models internal friction
Social	Alignment $\boldsymbol{f}_{\text{ali}}$	reflects group conformity
	Navigation $\boldsymbol{f}_{\text{nav}}$	guides goal-directed motions
	Cohesion $\boldsymbol{f}_{\text{coh}}$	maintains crowd boundaries

[a] Note that $\boldsymbol{f}_{\text{nav}}$ models short-term flow-following behavior rather than destination-oriented navigation.

Governing Equation. To capture both the fluid-like and social characteristics of dense crowd dynamics, we integrate two physical and three social operators from HINN [42], as summarized in Tab. 1. These operators are combined via learnable weights $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5]^\top$, enabling adaptive adjustments to diverse crowd motion patterns. To account for complex, heterogeneous crowd behaviors beyond these predefined operators, we leverage the symbolic expressiveness of a lightweight KAN [6, 27] to perform adaptive residual correction. Formally, given the velocity field $\hat{\boldsymbol{u}}_t$ at time-step t, our governing equation extracts acceleration-related features as follows:

$$\boldsymbol{f}_{\text{gov}}(\hat{\boldsymbol{u}}_{\boldsymbol{t}}) = \boldsymbol{\lambda}^\top \boldsymbol{F}(\hat{\boldsymbol{u}}_{\boldsymbol{t}}) + \text{KAN}(\hat{\boldsymbol{u}}_{\boldsymbol{t}}), \tag{3}$$

where $\boldsymbol{F}(\cdot) = [\boldsymbol{f}_{\text{con}}, \boldsymbol{f}_{\text{vis}}, \boldsymbol{f}_{\text{ali}}, \boldsymbol{f}_{\text{nav}}, \boldsymbol{f}_{\text{coh}}]^\top(\cdot)$. Note that for efficiency, we perform downsampling before and upsampling after this feature extraction process.

Boundary Condition. To enhance generalizability across diverse scenarios, we incorporate environmental constraints by encoding the walkable area $\boldsymbol{A}$ as explicit boundary conditions. Specifically, we adopt the no-slip boundary condition, enforcing zero velocity along the boundary $\partial\boldsymbol{A}$ via a corrective negative acceleration that drives velocities toward zero in the next time-step. For the given velocity field $\hat{\boldsymbol{u}}_t$ and walkable area $\boldsymbol{A}$ at time-step t, our boundary conditions compute acceleration-related features as:

$$\boldsymbol{f}_{\text{bou}}(\hat{\boldsymbol{u}}_t, \boldsymbol{A}) = -\mathbf{1}_{\partial\boldsymbol{A}} \odot \hat{\boldsymbol{u}}_t, \tag{4}$$

where $\mathbf{1}_{\partial\boldsymbol{A}}$ is a boundary indicator matrix of the same size as $\hat{\boldsymbol{u}}_t$, with 1 on the boundary and 0 elsewhere, and $\odot$ denotes element-wise multiplication.

Predictive Network. Due to its proven effectiveness in spatiotemporal modeling, we employ a Convolutional Residual Network (ConvResNet) [26] as the predictive network. At time-step t, the input is formed by concatenating the given velocity field $\hat{\boldsymbol{u}}_t$, walkable area $\boldsymbol{A}$, and acceleration-related features from our governing equation $\boldsymbol{f}_{\text{gov}}$ and boundary conditions $\boldsymbol{f}_{\text{bou}}$. Then, our predictive network ConvResNet evolves the next-step velocity field as:

$$\hat{\boldsymbol{u}}_{t+1} = \text{ConvResNet}([\hat{\boldsymbol{u}}_t, \boldsymbol{A}, \boldsymbol{f}_{\text{gov}}, \boldsymbol{f}_{\text{bou}}]), \tag{5}$$

where $[\cdot]$ denotes channel-wise concatenation.

Initialized from the velocity field $\hat{\boldsymbol{u}}_0 = \boldsymbol{u}_0$, HINN++ recursively performs this update over $T-1$ time-steps to evolve the complete velocity field sequence $\hat{\boldsymbol{U}} = [\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_{T-1}]$. For clarity and reproducibility, the full implementation pseudo-code is provided in Appendix A.2.

3.4 PINN: Trajectory Refinement

For the local trajectory refinement module, we employ a basic Multi-Layer Perceptron (MLP)-based PINN [31, 34] to demonstrate how fine-grained behaviors are resolved under the guidance of the evolved velocity priors, laying a foundation upon which more sophisticated architectures can be readily integrated. PINN resolves microscopic individual trajectories in the Lagrangian domain with four components: velocity querying, a predictive network, a position update mechanism, and an entrance-aware resampling strategy (EARS).

Velocity Querying. At time-step t, local velocities are queried from the global velocity field $\hat{\boldsymbol{u}}_t$ at current individual positions $\hat{\boldsymbol{x}}_t = [\hat{\boldsymbol{x}}_t^1, \ldots, \hat{\boldsymbol{x}}_t^N]$, resulting in a set of individual-specific velocity vectors $\hat{\boldsymbol{u}}_t(\hat{\boldsymbol{x}}_t)$.

Predictive Network. We implement the predictive network as a lightweight yet effective MLP that models the mapping from velocity–position pairs to motion vectors. For the given queried velocity vectors $\hat{\boldsymbol{u}}_t(\hat{\boldsymbol{x}}_t)$ and individual positions $\hat{\boldsymbol{x}}_t$ at time-step t, our predictive network MLP derives the motion vectors as:

$$\hat{\boldsymbol{v}}_t = \mathrm{MLP}([\hat{\boldsymbol{u}}_t(\hat{\boldsymbol{x}}_t), \hat{\boldsymbol{x}}_t]). \tag{6}$$

Position Update Mechanism. At time-step t, individual positions are preliminarily updated with the estimated motion vectors $\hat{\boldsymbol{v}}_t$, following the discrete-time kinematic equation [2] that governs individual motions:

$$\widetilde{\boldsymbol{x}}_{t+1} = \hat{\boldsymbol{x}}_t + \Delta t \cdot \hat{\boldsymbol{v}}_t, \tag{7}$$

where $\Delta t = 1$ denotes the time-step length.

Entrance-Aware Resampling. To maintain population consistency and constrain individuals within the walkable area, we introduce EARS, which consists of three steps: removal, entrance detection, and reinitialization. At time-step t, individuals whose preliminary updated positions fall outside the static walkable area, *i.e.*, $\widetilde{\boldsymbol{x}}_{t+1}^i \notin \boldsymbol{A}$, are removed. Potential entrances are then identified along the boundary $\partial \boldsymbol{A}$, where local velocities satisfy the inward-flow condition $\hat{\boldsymbol{u}}_t(\boldsymbol{x}^b) \cdot \boldsymbol{n}(\boldsymbol{x}^b) > 0$, with $\boldsymbol{x}^b$ denoting boundary points and $\boldsymbol{n}(\boldsymbol{x}^b)$ their inward normal vectors. Lastly, new individuals are randomly initialized near the entrances to replace the removed ones, forming the next-step positions $\hat{\boldsymbol{x}}_{t+1}$.

Initialized from individual positions $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0$, PINN iteratively performs this update over T time-steps to resolve the complete individual trajectory sequence $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_T]$. For clarity and reproducibility, the full implementation pseudo-code is provided in Appendix A.3.

3.5 Training Strategy

Scheduled Sampling. To mitigate exposure bias and reduce error accumulation in long-term sequence evolution, we employ scheduled sampling [3] during HINN++ training. Specifically, at time-step t, HINN++ takes the groundtruth velocity field $\boldsymbol{u}_t$ as input with a probability that gradually decreases over training, thereby shifting from groundtruth supervision to self-generated predictions and enabling stable autoregressive dynamics during inference.

Loss Functions. To supervise the global velocity evolution, we adopt the Smooth L1 loss [10]:

$$L_{\text{data}}(\boldsymbol{\lambda}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \text{SmoothL1}(\boldsymbol{U}, \hat{\boldsymbol{U}}), \tag{8}$$

where $\boldsymbol{U}$ and $\hat{\boldsymbol{U}}$ denote the groundtruth and evolved velocity fields, respectively; $\boldsymbol{\lambda}$ are the learnable weights of the governing equation, while $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ correspond to the parameters of the KAN-based residual correction and the predictive network ConvResNet. HINN++ is optimized by minimizing L_{data}, and these parameters remain fixed thereafter.

For the local trajectory refinement, we introduce a composite loss comprising two constraints: velocity consistency and collision avoidance. To enforce alignment with velocity priors, we define the velocity-consistency loss as:

$$\ell_{\text{vel}}(i,t) = \left\| (\hat{\boldsymbol{x}}^i_{t+1} - \hat{\boldsymbol{x}}^i_t) - \hat{\boldsymbol{u}}_t(\hat{\boldsymbol{x}}^i_t) \right\|^2, \tag{9}$$

where $\hat{\boldsymbol{x}}^i_t$ denotes the position of the i-th individual at time-step t, $\hat{\boldsymbol{u}}_t(\hat{\boldsymbol{x}}^i_t)$ is the corresponding queried velocity, and $\| \cdot \|$ represents the Euclidean norm. Note that ℓ_{vel} is not applied at resampling steps where positions are reset. To recover interaction details and reinforce fine-grained individual realism, we define the collision-avoidance loss as:

$$\ell_{\text{col}}(i,t) = \sum_{j \neq i} [\max(d_{\min} - \left\| \hat{\boldsymbol{x}}^i_t - \hat{\boldsymbol{x}}^j_t \right\|, 0)]^2, \tag{10}$$

where $d_{\min}$ denotes the minimum safety distance threshold. The total trajectory loss is then formulated as:

$$L_{\text{trajectory}}(\boldsymbol{\theta}) = \mathbb{E}_{i,t}\left[\ell_{\text{vel}}(i,t) + \alpha \cdot \ell_{\text{col}}(i,t)\right], \tag{11}$$

where $\boldsymbol{\theta}$ are the parameters of PINN, α is a weighting factor, and $\mathbb{E}_{i,t}$ denotes the expectation computed over all individuals and time-steps. PINN is optimized by minimizing $L_{\text{trajectory}}$.

4 Experiments

4.1 Experiment Setup

Datasets. We evaluate our unified cross-scale framework for dense crowd simulation on two public real-world crowd datasets. Following established practice, the primary evaluation is conducted on the Dense Crowd Flow Dataset (DCFD) [42], which comprises 457 dense crowd video clips across six representative crowd motion patterns. To enable trajectory-based assessment, we additionally test our framework on the MOT20 dataset [8], which contains four high-density surveillance sequences with 2,332 annotated individual trajectories. More details on datasets are provided in Appendix B.1.

Baseline Methods. We compare our framework with thirteen representative methods spanning both macroscopic and microscopic scales. For velocity evolution, we consider four macroscopic methods that model crowd dynamics with different PDE-solving neural networks, including data-driven (PDE-Net [28]), physics-informed (NSFnet [21]), and physics-preserving (PPNN [26], HINN [42]) methods. For trajectory refinement, we evaluate against nine representative microscopic methods, including physics-based models (Boids [33], SFM [15]), data-physics networks (Social-LSTM [1], STGAT [19], PCS [40], SPDiff [7]), and generative platforms (Sora [5], Kling, Runway [22]). More details on baseline methods are provided in Appendix B.2.

Evaluation Metrics. We evaluate the simulation performance using eleven metrics. To measure the diversity and fidelity of macroscopic velocity fields, we employ the Inception Score (IS), Fréchet Inception Distance (FID), and Structural Similarity Index Measure (SSIM) [42]. For microscopic individual trajectories, we assess the dynamic smoothness (perpendicular deviation D_i, speed change $\Delta V_{i,t}$, angle change $\Delta A_{i,t}$), motion intensity (travel distance $L_{i,t}$, energy $E_{i,t}$, collision-avoidance energy $steerE_{i,t}$) [41], and trajectory accuracy using the Joint Average Displacement Error (JADE) and Joint Final Displacement Error (JFDE) [38]. More details on evaluation metrics are provided in Appendix B.3.

Experiment Settings. Both DCFD and MOT20 datasets are split into training and testing sets with a 4:1 ratio. PDE-Net, NSFnet, PPNN, and HINN are trained from scratch under the same configurations as ours. Boids and SFM are executed directly on the testing set without requiring any training. Social-LSTM, STGAT, PCS, and SPDiff are trained following their official settings and evaluated on the testing set. Sora, Kling, and Runway generate videos from standardized prompts on the testing set, after which trajectories are extracted using the zero-shot tracker Match Anything by Segment Anything (MASA) [25]. More details on experiment settings are provided in Appendix B.4.

4.2 Results on Velocity Evolution

The comparison between HINN++ and macroscopic methods on velocity evolution is presented in Tab. 2. HINN++ outperforms all baseline methods, achieving the best overall performance across IS, FID, and SSIM metrics on the DCFD dataset. Specifically, we summarize the key observations as follows. First, compared to the data-driven PDE-Net, our learnable governing equation with the symbolic expressiveness of KAN residual correction enables HINN++ to capture

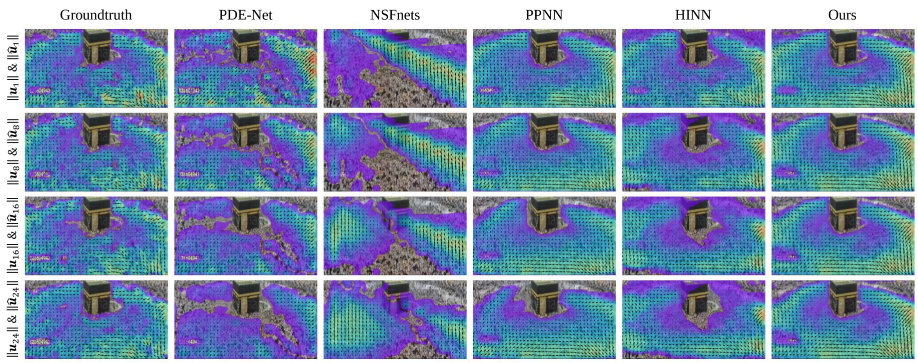

Fig. 3: Qualitative comparison under the "circle" pattern, where $\|\boldsymbol{u}\|$ and $\|\hat{\boldsymbol{u}}_t\|$ represent the magnitudes of the groundtruth and evolved velocity fields.

Table 2: Performance comparison on macroscopic velocity evolution, where **red** and **blue** represent the best and second-best results.

Method	IS ↑	FID ↓	SSIM ↑
PDE-Net [28]	1.7016	0.2924	**0.5237**
NSFnet [21]	1.0369	0.7326	0.3526
PPNN [26]	**1.7440**	**0.0275**	0.4893
HINN [42]	1.6789	0.0638	0.4782
Ours	**1.7469**	**0.0278**	**0.6030**

complex and varying crowd dynamics adaptively across diverse motion patterns, leading to higher diversity and fidelity. Second, NSFnet performs the worst across all metrics, likely due to its non-convolutional architecture, which limits its ability to capture intricate spatial dependencies essential for modeling crowd dynamics in the Eulerian domain. Third, while PPNN and HINN achieve relatively competitive results on IS and FID, both of them exhibit noticeable deficiencies on SSIM. In contrast, HINN++ benefits from incorporating environmental factors as boundary conditions, enhancing its ability to preserve structural motion patterns and spatial consistency. Overall, the superior performance of HINN++ across all metrics demonstrates its macroscopic evolution capability in capturing realistic global dynamics.

For qualitative comparison with these baseline methods, we visualize the groundtruth and evolved velocity fields using heatmaps and quiver plots overlaid on video frames. Fig. 3 illustrates dense crowd motions under the "circle" pattern, where both the groundtruth and evolved velocity magnitudes, $\|\boldsymbol{u}_t\|$ and $\|\hat{\boldsymbol{u}}_t\|$, are presented at four time-steps $t = 1, 8, 16, 24$. Notably, PDE-Net exhibits insufficient spatial coverage in simulating large-scale crowd dynamics, neglecting crowd motions in substantial areas. Even worse, NSFnet largely fails to represent the fundamental structure of the crowd motion pattern. While PPNN and HINN generate reasonable crowd motion structures at early time-steps, their spatial fidelity gradually degrades as the simulation progresses. Conversely, HINN++ reproduces complete crowd motions and preserves structural integrity across all time-steps. Such evolutionary stability and robustness underscore the superiority of our framework for long-term dense crowd simulation, providing the essential macroscopic velocity prior for subsequent high-fidelity microscopic refinement of individual trajectories. Additional results are provided in Appendix C.1.

Table 3: Performance comparison on microscopic trajectory refinement, where metrics are reported as deviations from groundtruth, and smaller absolute deviations indicate higher consistency. **Red**, **blue**, and **green** represent the top three results.

Method	D_i	$\Delta V_{i,t}$	$\Delta A_{i,t}$	$L_{i,t}$	$E_{i,t}$	$steerE_{i,t}$	JADE ↓	JFDE ↓
GT	136.963	0.124	24.072	0.432	184.208	162.003	-	-
Boids [33]	+1285.546	+0.295	**−13.807**	+2.728	+1848.020	+51.735	63.013	75.423
SFM [15]	+415.432	**−0.080**	−20.552	+0.506	+75.114	**−4.517**	13.247	19.607
Social-LSTM [1]	−119.141	−0.123	−22.024	−0.405	−27.842	−5.828	11.571	17.189
STGAT [19]	−47.650	−0.121	−23.096	−0.282	**−25.190**	**−5.791**	12.431	17.935
PCS [40]	−125.164	**−0.119**	−23.545	**−0.023**	**−1.852**	**−5.475**	**9.231**	**14.897**
SPDiff [7]	**−22.083**	**+0.097**	−22.981	**+0.015**	+292.087	+270.146	**9.850**	**14.715**
Sora [5]	−43.776	+0.994	+24.311	+1.357	+5231.787	+8066.645	19.186	21.535
Kling	**−39.108**	+0.408	**+5.731**	+0.425	+254.418	+417.607	28.672	33.744
Runway [22]	−53.206	+0.473	**+6.608**	+0.469	+268.137	+426.389	22.199	23.856
Ours	**−42.895**	+0.314	+14.885	**−0.125**	**+0.979**	+47.424	**8.274**	**12.836**

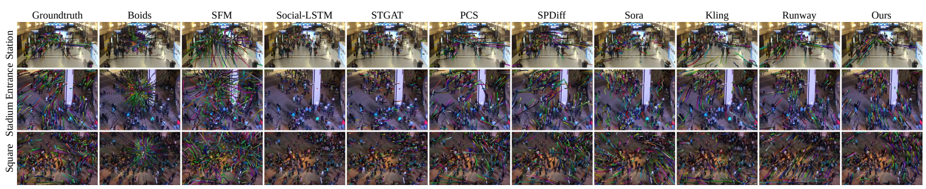

Fig. 4: Qualitative comparison across three scenarios, where unique base colors represent individual trajectories and light-to-dark gradients indicate the temporal progression of motion, from initial to subsequent states.

4.3 Results on Trajectory Refinement

The comparison between ELHINN and microscopic methods on trajectory refinement is presented in Tab. 3. To intuitively illustrate the performance differences across methods and metrics, Fig. 4 visualizes the groundtruth and resolved trajectories across three scenarios from the MOT20 dataset, overlaid on the initial video frame. Here, we summarize the key observations as follows.

First, *physics-based models struggle to capture structured collective motion patterns.* Due to its cohesion–separation mechanism, Boids oscillates between excessive clustering and sudden dispersion, causing discontinuities in crowd flow. Lacking explicit goals or attractive forces, SFM relies mainly on repulsive interactions to adjust velocities for collision avoidance, but fails to reproduce the intentional steering and coordinated dense crowd motion. Second, *data-physics networks suffer from scale-missing, leading to fragmented simulations.* These networks depend heavily on historical trajectories to infer latent intentions. Without the high-fidelity macroscopic velocity prior provided by HINN++, data-driven networks (Social-LSTM, STGAT) generate short, near-static trajectories with minimal speed change $\Delta V_{i,t}$, angle change $\Delta A_{i,t}$, and travel distance $L_{i,t}$. Although physics-infused networks (PCS, SPDiff) enforce basic motion continuity with physical modules, the absence of systemic consistency results in almost linear trajectories, failing to resolve the complex, nonlinear angle change $\Delta A_{i,t}$ necessary for realistic crowd motions. Third, *generative platforms produce unrealistically intense and biased crowd dynamics.* Without explicit physical priors or constraints, these platforms often synthesize excessively rapid crowd motions, yielding inflated energy $E_{i,t}$ and collision-avoidance energy $steerE_{i,t}$. Despite fixed-view prompts, Sora introduces unintended camera shifts that exaggerate speed change $\Delta V_{i,t}$ and angle change $\Delta A_{i,t}$. Kling and Runway occasionally produce physically inconsistent crowd motions in open-space scenarios, such as backward flow, thereby degrading their performance on JADE and JFDE.

In contrast, ELHINN achieves dominant performance on trajectory accuracy metrics such as JADE and JFDE, mainly attributed to its stable and robust macroscopic velocity evolution, which provides a high-fidelity velocity prior that enables precise and consistent microscopic trajectory refinement. Besides, ELHINN attains strong results on dynamic smoothness and motion intensity metrics, including perpendicular deviation D_i, travel distance $L_{i,t}$, and energy $E_{i,t}$, indicating stable and physically plausible crowd motion patterns. While ELHINN exhibits slightly lower performance on speed change $\Delta V_{i,t}$, angle change $\Delta A_{i,t}$, and collision-avoidance energy $steerE_{i,t}$, these metrics are generally larger than the groundtruth values. This suggests that ELHINN captures complex, nonlinear behaviors and tends to produce more dynamic and active crowd motions rather than overly conservative ones, which constitutes a more desirable yet challenging objective in dense crowd simulation. Unlike baseline methods that often collapse on particular metrics, ELHINN achieves reliable and well-balanced performance across all metrics. This superiority highlights the effectiveness of our framework in simulating realistic individual trajectories while maintaining global collective coherence. Additional results are provided in Appendix C.2.

4.4 Ablation Study on Global Velocity Evolution Module

To investigate the impact of our designs on macroscopic velocity evolution, we analyze two essential components of HINN++, including the boundary conditions (BC) and KAN residual correction. Specifically, we evaluate three variants: (1) HINN++ w/o BC, which removes walkable-area constraints; (2) HINN++ w/o KAN, which excludes the residual correction term; (3) and HINN++ with MLP, which replaces the KAN residual correction with a standard MLP of equivalent layers. We follow the same experimental settings as in Sec. 4.2.

As presented in Tab. 4, removing either component leads to a noticeable decrease in simulation fidelity. Specifically, excluding boundary conditions causes a more pronounced degradation in both FID and SSIM, underscoring the importance of environmental constraints for preserving structural motion patterns and modeling global crowd dynamics. In contrast, removing KAN results in a milder decline, suggesting its specialized role in capturing intricate, nonlinear motion details and enhancing fine-grained realism. Besides, while MLP residual correction slightly mitigates the decline on SSIM, it struggles to resolve complex,

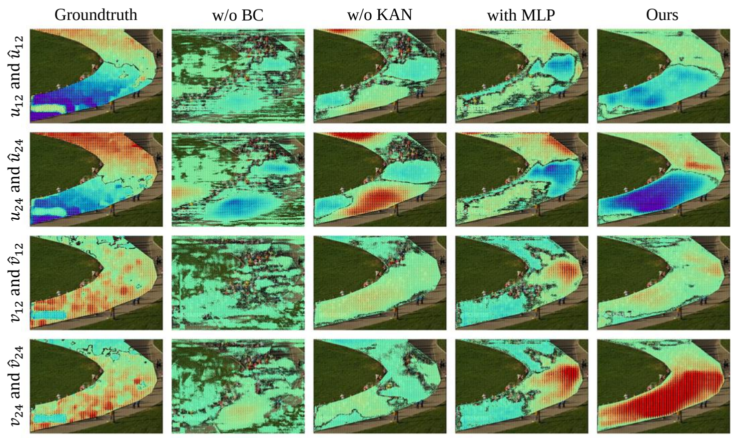

Fig. 5: Qualitative comparison of ablated variants under the "curve" pattern, where u and v represent the horizontal and vertical velocity fields respectively.

Table 4: Ablation study on the global velocity evolution module HINN++, where **bold** represents the best results.

Method	IS ↑	FID ↓	SSIM ↑
w/o BC	1.7883	0.0395	0.4837
w/o KAN	1.7497	0.0350	0.6019
with MLP	**1.7937**	0.0382	0.6026
Ours	1.7469	**0.0278**	**0.6030**

nonlinear dynamics, failing to prevent the substantial degradation on FID. Interestingly, all ablated variants achieve higher IS than the full HINN++. However, we argue that this increase does not reflect genuine motion diversity, but rather results from weakened constraints, leading to more erratic and less physically consistent crowd motions.

To validate our claims and enable intuitive comparison, Fig. 5 visualizes dense crowd motions under the "curve" pattern, showing the horizontal and vertical components of both the groundtruth and evolved velocity fields at time-steps $t = 12, 24$. The w/o BC variant fails to preserve the global motion structure, while the w/o KAN and with MLP variants exhibit limited simulation coverage, supporting our claim that their IS scores stem from weakened constraints and less realistic motions. Besides, compared to the w/o KAN and with MLP variants, the full HINN++ better captures velocity transitions around the road bend, demonstrating KAN's adaptivity in modeling complex, nonlinear crowd behaviors beyond conventional operators. These results highlight the complementary roles of both components, where the boundary conditions govern global motion structure, while KAN adaptively captures intricate, heterogeneous nonlinearities. Additional results are provided in Appendix C.3.

4.5 Ablation Study on Local Trajectory Refinement Module

To explore the contribution of our designs to microscopic trajectory refinement, we examine three critical components of PINN, including the collision-avoidance loss ℓ_{col}, predictive network MLP, and entrance-aware resampling

Table 5: Ablation study on the local trajectory refinement module PINN, where metrics are reported as deviations from groundtruth, and smaller absolute deviations indicate higher consistency. **Bold** represents the best results.

Method	D_i	$\Delta V_{i,t}$	$\Delta A_{i,t}$	$L_{i,t}$	$E_{i,t}$	$steerE_{i,t}$	JADE ↓	JFDE ↓
GT	136.963	0.124	24.072	0.432	184.208	162.003	-	-
w/o ℓ_{col}	**−39.414**	+0.317	+15.087	−0.124	+1.421	+47.997	8.293	12.962
w/o MLP	−39.638	+0.317	+15.095	−0.124	+1.270	+47.743	8.307	12.969
w/o EARS	−92.478	+0.537	+19.693	**+0.012**	+14.281	+78.183	13.257	17.438
Ours	−42.895	**+0.314**	**+14.885**	−0.125	**+0.979**	**+47.424**	**8.274**	**12.836**

strategy (EARS). Specifically, we investigate three variants: (1) PINN w/o ℓ_{col}, which removes collision-avoidance constraints; (2) PINN w/o MLP, which replaces MLP with direct integration; (3) and PINN w/o EARS, which disables population resampling. We follow the same experimental settings as in Sec. 4.3.

As presented in Tab. 5, removing any component consistently degrades performance across dynamic smoothness, motion intensity, and trajectory accuracy metrics, confirming the necessity of all three designs. Among these variants, removing EARS leads to the most pronounced degradation. One notable observation is that, compared to the full PINN, the w/o EARS variant exhibits smaller perpendicular deviation D_i but larger speed change $\Delta V_{i,t}$ and angle change $\Delta A_{i,t}$. This behavior arises because disabling resampling breaks population balance, causing trajectories to collapse toward the dominant flow direction and exhibit intensified short-term fluctuations, thereby highlighting the critical role of EARS in stabilizing trajectory refinement. Additional results are provided in Appendix C.4. We also further provide analyses on computational complexity and system robustness in Appendix C.5.

5 Conclusion

In this work, we propose a unified cross-scale framework ELHINN for dense crowd simulation, which bridges the fundamental scale gap by explicitly coupling macroscopic velocity evolution with microscopic trajectory refinement. To enhance crowd motion representability and scenario adaptability, we develop HINN++, which integrates a learnable governing equation with KAN-based residual correction and environmental boundary conditions. To ensure trajectory stability and fidelity, we introduce PINN with EARS and collision-avoidance constraints. Extensive experiments on two public real-world crowd datasets demonstrate the superiority of our framework in maintaining global collective coherence while achieving local individual realism. Overall, our framework paves the way for more data-efficient and physics-preserving crowd simulation, as well as fine-grained crowd behavior analysis, bridging the gap between observation, simulation, understanding, and machine learning in crowd motion. Future works include extending our framework to variable-density crowds, emergency scenarios, and cross-view or multi-modal data for improved simulation.

Acknowledgements

This research was partly supported by the National Natural Science Foundation of China (NSFC, Grant No. 62171281) and the Science and Technology Commission of Shanghai Municipality (STCSM, Grant No. 25GA32001003).

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 961–971 (2016)
2. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with applications to tracking and navigation: theory algorithms and software. John Wiley & Sons (2001)
3. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. Advances in neural information processing systems **28** (2015)
4. Bradley, G.: A proposed mathematical model for computer prediction of crowd movements and their associated risks. In: Proceedings of the International Conference on Engineering for Crowd Safety. pp. 303–311. Elsevier Publishing Company London (1993)
5. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al.: Video generation models as world simulators. OpenAI Blog **1**(8), 1 (2024)
6. Cai, X., Pan, R., Yang, H.: Loki: Low-dimensional kan for efficient fine-tuning image models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14869–14880 (2025)
7. Chen, H., Ding, J., Li, Y., Wang, Y., Zhang, X.P.: Social physics informed diffusion model for crowd simulation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 474–482 (2024)
8. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 (2020)
9. Gingold, R.A., Monaghan, J.J.: Smoothed particle hydrodynamics: theory and application to non-spherical stars. Monthly notices of the royal astronomical society **181**(3), 375–389 (1977)
10. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
11. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
12. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2255–2264 (2018)
13. He, F., Yue, J., Zhu, J., Seyfried, A., Casas, D., Pettré, J., Wang, H.: Learning extremely high density crowds as active matters. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 540–550 (2025)
14. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407**(6803), 487–490 (2000)

15. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E **51**(5), 4282 (1995)
16. Henderson, L.: The statistics of crowd fluids. nature **229**(5284), 381–383 (1971)
17. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
19. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6272–6281 (2019)
20. Hughes, R.L.: A continuum theory for the flow of pedestrians. Transportation Research Part B: Methodological **36**(6), 507–535 (2002)
21. Jin, X., Cai, S., Li, H., Karniadakis, G.E.: Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. Journal of Computational Physics **426**, 109951 (2021)
22. Kheradmand, S., Vicini, D., Kopanas, G., Lagun, D., Yi, K.M., Matthews, M., Tagliasacchi, A.: Stochasticsplats: Stochastic rasterization for sorting-free 3d gaussian splatting. arXiv preprint arXiv:2503.24366 (2025)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
24. Korbmacher, R., Tordeux, A.: Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches. IEEE Transactions on Intelligent Transportation Systems **23**(12), 24126–24144 (2022)
25. Li, S., Ke, L., Danelljan, M., Piccinelli, L., Segu, M., Van Gool, L., Yu, F.: Matching anything by segmenting anything. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18963–18973 (2024)
26. Liu, X.Y., Zhu, M., Lu, L., Sun, H., Wang, J.X.: Multi-resolution partial differential equations preserved learning framework for spatiotemporal dynamics. Communications Physics **7**(1), 31 (2024)
27. Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y., Tegmark, M.: Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756 (2025)
28. Long, Z., Lu, Y., Ma, X., Dong, B.: Pde-net: Learning pdes from data. In: International conference on machine learning. pp. 3208–3216. PMLR (2018)
29. Mangalam, K., Girase, H., Agarwal, S., Lee, K.H., Adeli, E., Malik, J., Gaidon, A.: It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: European conference on computer vision. pp. 759–776. Springer (2020)
30. Moussaïd, M., Kapadia, M., Thrash, T., Sumner, R.W., Gross, M., Helbing, D., Hölscher, C.: Crowd behaviour during high-stress evacuations in an immersive virtual environment. Journal of The Royal Society Interface **13**(122), 20160414 (2016)
31. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics **378**, 686–707 (2019)
32. Rasouli, A.: Pedestrian simulation: A review. arXiv preprint arXiv:2102.03289 (2021)
33. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques. pp. 25–34 (1987)

34. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. nature **323**(6088), 533–536 (1986)
35. Thakur, S., Raissi, M.: Elpinn: Eulerian lagrangian physics-informed neural network. arXiv preprint arXiv:2504.09053 (2025)
36. Treuille, A., Cooper, S., Popović, Z.: Continuum crowds. ACM transactions on graphics (TOG) **25**(3), 1160–1168 (2006)
37. Van Toll, W., Chatagnon, T., Braga, C., Solenthaler, B., Pettré, J.: Sph crowds: Agent-based crowd simulation up to extreme densities using fluid dynamics. Computers & Graphics **98**, 306–321 (2021)
38. Weng, E., Hoshino, H., Ramanan, D., Kitani, K.: Joint metrics matter: A better standard for trajectory forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20315–20326 (2023)
39. Yuan, Y., Goñi-Ros, B., Bui, H.H., Daamen, W., Vu, H.L., Hoogendoorn, S.P.: Macroscopic pedestrian flow simulation using smoothed particle hydrodynamics (sph). Transportation research part C: emerging technologies **111**, 334–351 (2020)
40. Zhang, G., Yu, Z., Jin, D., Li, Y.: Physics-infused machine learning for crowd simulation. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. pp. 2439–2449 (2022)
41. Zhao, M., Cai, W., Turner, S.J.: Clust: Simulating realistic crowd behaviour by mining pattern from crowd videos. In: Computer graphics forum. vol. 37, pp. 184–201. Wiley Online Library (2018)
42. Zhou, Y., Lai, P., Yu, J., Xiong, Y., Yang, H.: Hydrodynamics-informed neural network for simulating dense crowd motion patterns. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 4553–4561 (2024)