Physics-infused Machine Learning for Crowd Simulation

Guozhen Zhang
Department of Electronic Engineering, Tsinghua University & TsingRoc
Beijing, China
zgz18@mails.tsinghua.edu.cn

Zihan Yu
Department of Electronic Engineering, Tsinghua University & TsingRoc
Beijing, China
yuzh19@mails.tsinghua.edu.cn

Depeng Jin
Department of Electronic Engineering, Tsinghua University
Beijing, China
jindp@tsinghua.edu.cn

Yong Li
Department of Electronic Engineering, Tsinghua University
Beijing, China
liyong07@tsinghua.edu.cn

ABSTRACT

Crowd simulation acts as the basic component in traffic management, urban planning, and emergency management. Most existing approaches use physics-based models due to their robustness and strong generalizability, yet they fall short in fidelity since human behaviors are too complex and heterogeneous for a universal physical model to describe. Recent research tries to solve this problem by deep learning methods. However, they are still unable to generalize well beyond training distributions. In this work, we propose to jointly leverage the strength of the physical and neural network models for crowd simulation by a Physics-Infused Machine Learning (PIML) framework. The key idea is to let the two models learn from each other by iteratively going through a physics-informed machine learning process and a machine-learning-aided physics discovery process. We present our realization of the framework with a novel neural network model, Physics-informed Crowd Simulator (PCS), and tailored interaction mechanisms enabling the two models to facilitate each other. Specifically, our designs enable the neural network model to identify generalizable signals from real-world data better and yield physically consistent simulations with the physical model's form and simulation results as a prior. Further, by performing symbolic regression on the well-trained neural network, we obtain improved physical models that better describe crowd dynamics. Extensive experiments on two publicly available large-scale real-world datasets show that, with the framework, we successfully obtain a neural network model with strong generalizability and a new physical model with valid physical meanings at the same time. Both models outperform existing state-of-the-art simulation methods in accuracy, fidelity, and generalizability, which demonstrates the effectiveness of the PIML framework for improving simulation performance and its capability for facilitating scientific discovery and deepening our understandings of crowd dynamics. We release the codes at https://github.com/tsinghua-fib-lab/PIML.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08.
https://doi.org/10.1145/3534678.3539440

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Agent / discrete models*; • **Applied computing** → *Sociology*.

KEYWORDS

Physics-infused machine learning, crowd simulation, symbolic regression.

ACM Reference Format:
Guozhen Zhang, Zihan Yu, Depeng Jin, and Yong Li. 2022. Physics-infused Machine Learning for Crowd Simulation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539440

1 INTRODUCTION

Crowd simulation, the process of simulating the movements and interaction dynamics of a large number of people [28], has long been one of the most significant topics in traffic management, urban planning, and emergency management [36]. It serves as the foundation of various important applications, ranging from traffic signal management [23], building architectural design [8], and crowd evacuation [34]. For example, simulating how crowds move in public transport interchanges, such as airports and railway stations, helps us analyze the efficiency and safety of the interchange when facing large-capacity passenger flows and further facilitates the optimization of the architecture design. Recently, with the development of data science and deep learning, crowd simulation has become increasingly important because a good simulator is considered as one of the keys to applying reinforcement learning in real-world systems [6, 31].

Current crowd simulation methods can be mainly divided into two categories, including *physics-based methods* and *deep learning methods* [28]. *Physics-based methods* typically use calibrated physical models and rules derived from expert knowledge to model crowd dynamics. For example, the Social Force Model (SFM) [13], one of the most widely used physical methods, characterizes the interactions among pedestrians, obstacles, and destinations as different forces and simulates crowd dynamics based on Newtonian mechanics. This type of method generally achieves robust performance across different scenarios, which is why it is widely adopted

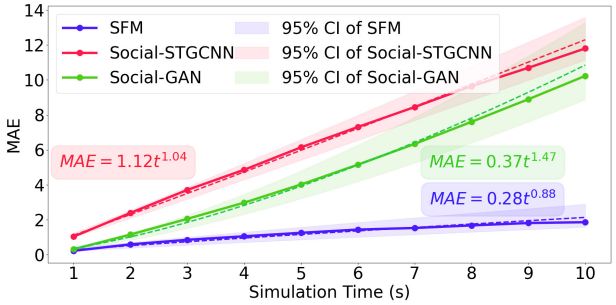

Figure 1: Demonstration of the poor generalizability of deep learning models. The Mean Absolute Error (MAE) of two state-of-the-art neural network models' simulations increases super-linearly, while the social force model increases sub-linearly.

in state-of-the-art commercial crowd simulators, such as VISSIM[1] and MassMotion[2]. Nevertheless, it is less realistic in characterizing micro pedestrian behaviors [24] because there is no universal physical model that can accurately describe human behaviors, given their complexity and heterogeneity. On the contrary, benefiting from the strong representation power, *deep learning methods* are able to learn the complex and heterogeneous micro crowd dynamics better, such as collision avoidance. However, methods of this type are typically hard to generalize to different scenarios [37]. As we show in our experiments later, many even cannot generalize to simulations with a longer time than in training settings. For instance, as shown in Figure 1, as the simulation time grows, the simulation errors of two state-of-the-art neural network models increase super-linearly, while that of the social force model increases sub-linearly.

As we demonstrated above, the strength of physics-based methods and deep learning methods are complementary. Thus, the core idea of this paper is that if we can propose a framework to leverage the strengths of both methods, it is possible to achieve realistic and robust crowd simulation.

Following this novel idea, we propose to use a Physics-Infused Machine Learning (PIML) framework [20] that enables physics-based methods and deep learning methods to learn from each other for crowd simulation. As shown in Figure 2, the framework contains three key components, including the the physical model, deep learning model, and interaction mechanism for the two models to learn from each other. These components works with two basic processes: a physics-informed machine learning process and a machine-learning-aided physics discovery process. Specifically, the former process lets the neural network model take advantage of the physics model's robustness and generalizability through physics-informed machine learning [26], a learning philosophy that introduces observational, inductive, or learning biases to constrain the model's learning process to physically consistent solutions. Existing works have proved it effective in microscopic fields such as modeling molecular dynamics [15], where we know lots of physics. We extend it to the field where little physics is known in this paper. The latter process enables physical models to learn from the well-trained deep learning models to improve their capability of characterizing complex and heterogeneous crowd dynamics. Research has shown that this process can be done automatically with symbolic regression on the trained neural network model's components [5]. We further suggest that it is better to carry out this process in a human-machine teaming manner, where researchers can find better physical models more easily with the assistance of the well-trained neural network model by symbolic regression on its components. In this way, as we iterate back and forth between the two processes, we can potentially obtain a robust and generalizable neural network model for simulation and discover a better physical model to deepen our understanding of crowd dynamics at the same time.

In this work, our principal aim is to realize the PIML framework for crowd simulation and demonstrate its effectiveness. To this end, we select the most widely used social force model as our physical model and demonstrate how we go through the two processes of the PIML framework. (1) For the physics-informed machine learning process, we present a novel neural network model, Physical-informed Crowd Simulator (PCS), with three novel designs. First, we introduce strong inductive biases by directly basing the neural network on the social force model and substituting its term that characterizes pedestrian interactions with a graph network. Second, we design a student-teacher co-forcing training algorithm to let our model learn from both the real-world data and the data generated by the social force model since it provides supervised signals with long-term generalizability. Third, to facilitate our model to learn natural micro pedestrian dynamics, we design a collision avoidance learner with a self-supervised collision prediction loss and a collision focal loss that can effectively identify and upsample the sparse supervised signal for collision avoidance from the data. (2) In terms of the machine-learning-aided physics discovery process, we perform symbolic regression on the learned edge function of the graph network in the well-trained PCS. To stabilize this process, we propose a rejection probability-based sampling algorithm and work in a human-machine teaming manner.

We highlight our contributions as follows:

- To the best of our knowledge, we propose to use a Physics-Infused Machine Learning (PIML) framework for crowd simulation for the first time, which integrates the advantage of both physical and deep learning models to achieve accurate and generalizable simulations.
- We realize the PIML framework with novel designs, including the physics-informed crowd simulator with tailored architecture and training algorithms for long-term simulation, and the interaction mechanisms that enable the physical and deep learning models to facilitate each other.
- Extensive experiments on two large-scale real-world datasets demonstrate the effectiveness of the proposed framework. Specifically, both the neural network and physical model we obtained from the framework outperform the state-of-the-art simulation methods by more than 10% on both accuracy and fedelity. They also show the best ability to generalize beyond the training distributions. Further ablation study shows the effectiveness of each design. We validate that enabling the neural networks and the physical models to learn from each other is the key to success.

[1] https://www.ptvgroup.com/en/solutions/products/ptv-viswalk/
[2] https://www.oasys-software.com/products/pedestrian-simulation/massmotion/

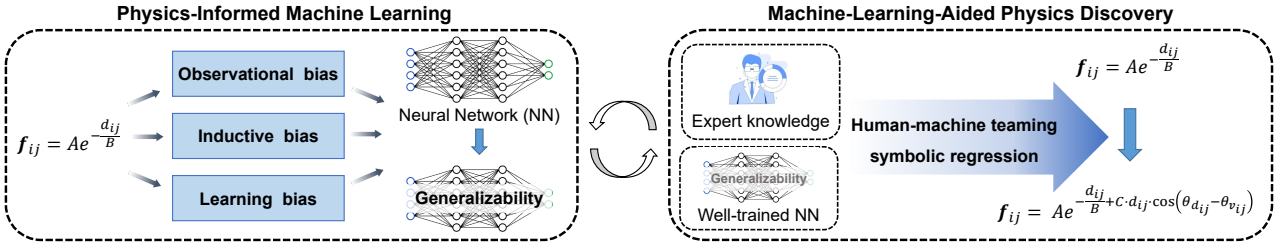

Figure 2: Illustrations of the Physics-Infused Machine Learning (PIML) framework.

- With the framework, we find new social force models with valid physical meanings that characterize micro crowd dynamics better than the original social force model, which demonstrates this framework's ability to assist scientific discovery for crowd dynamics.

2 PROBLEM FORMULATION

With all the pedestrians' initial states, including their positions, velocities, accelerations, and destinations, together with the environment states, including the obstacles' positions, velocities, and accelerations, crowd simulation requires the model to output how pedestrians' states evolve for an arbitrarily long time. We formulate it as an iterative prediction problem, that is, the simulator iteratively maps pedestrians' current states to their future states to get a whole simulation given the environment states. Formally, it can be formulated as follows,

$$S_p^{t+1} = f(S_p^t, S_e^t). \tag{1}$$

where S_p^t and S_e^t is the the pedestrians' states and the environment states at time t. To get physically consistent results, we further refine this problem by letting the simulator predict the accelerations, and then we use an Euler integration to update the velocities and positions based on it, i.e.,

$$\begin{aligned} v^{t+1} &= v^t + \Delta t \cdot a^{t+1}, \\ p^{t+1} &= p^t + \Delta t \cdot v^{t+1}, \end{aligned} \tag{2}$$

where p, v, a represents the position, velocity, and acceleration, respectively).

3 PHYSICS-INFUSED MACHINE LEARNING

To realize our proposed PIML framework, we need to go through three key design processes. (1) Start with a physical model that well-fit the problem. (2) Design a neural network model with strong inductive biases well matched to the physical model. (3) Design interaction mechanisms to let the neural network model and the physical model learn from each other. In this section, we elaborate on each of our designs.

Note that since our principal aim is to demonstrate the effectiveness of our proposed PIML framework in crowd simulation, we only develop minimal designs needed for the framework, which can be easily extended to complicated ones to further improve the performance.

3.1 Physical Model

In this work, we use the original form of Social Force Model (SFM) [13], the most widely adopted model in real-world applications, as our physical model. It conducts crowd simulation through modeling the interactions among individuals, obstacles, and destinations as forces, which can be formulated as follows,

$$\begin{aligned} m_i \boldsymbol{a}_i &= \boldsymbol{f}_{iD} + \sum_{j \neq i, j \in \mathcal{P}} \boldsymbol{f}_{ji} + \sum_{o \in O} \boldsymbol{f}_{oi}, \\ \boldsymbol{f}_{iD} &= m_i \frac{v_{id} \boldsymbol{n}_{iD} - \boldsymbol{v}_i}{\tau}, \\ \boldsymbol{f}_{ji} &= \lambda_1 e^{-d_{ji}/\lambda_2} \cdot \boldsymbol{n}_{ji}, \\ \boldsymbol{f}_{oi} &= \lambda_3 e^{-d_{oi}/\lambda_4} \cdot \boldsymbol{n}_{oi}, \end{aligned} \tag{3}$$

where $\mathcal{P}$ and O refers to the set of pedestrians and obstacles, respectively. $\boldsymbol{f}_{iD}$, $\boldsymbol{f}_{ji}$ and $\boldsymbol{f}_{oi}$ represent the traction force of destination D, the repulsive force of pedestrian j and obstacle o on pedestrian i, respectively. The repulsive force between pedestrian j and i is correlated with the relative distance between them and in the direction of their relative position. The repulsive force between pedestrian j and obstacle o takes a similar form. The magnitude and direction of the traction force $\boldsymbol{f}_{iD}$ depend on the desired walking speed v_{id}, the unit vector to the target direction $\boldsymbol{n}_{iD}$, and the current velocity $\boldsymbol{v}_i$ of pedestrian i. m_i is the mass of pedestrian i, and τ is the simulation time step. λ_1 and λ_2 are tunable parameters with a physical meaning of force intensity and force radius, respectively.

3.2 Neural Network Model

Our core idea that the physics and neural network models can complement each other in crowd simulation is based on the premise that the designed neural network model can outperform the physical model on the real-world dataset by learning from both the physical model and the real-world data. However, this is non-trivial. Besides the problem of how to design the interaction mechanism that helps the neural network model learn from the physical model, learning microscopic crowd dynamics from real-world data is challenging itself. First, a good simulation requires long-term accuracy. In other words, the neural network model should have the ability to generalize to scenarios with a time longer than that in the training set, which is generally difficult. Second, micro human behaviors are heterogeneous and complex [14]. For example, to avoid collisions, some people may slow down and wait, while others may speed up to bypass. Further, the timing people take action to avoid collision varies. As a result, the supervised signal in the real-world data can be contradictory when the environment state is the same. Therefore, how to learn physically consistent results from real-world data is also challenging.

Facing these challenges, we design a Physics-informed Crowd Simulator (PCS) with strong inductive biases. As shown in Figure 3, it contains three novel designs, including the graph-network-based

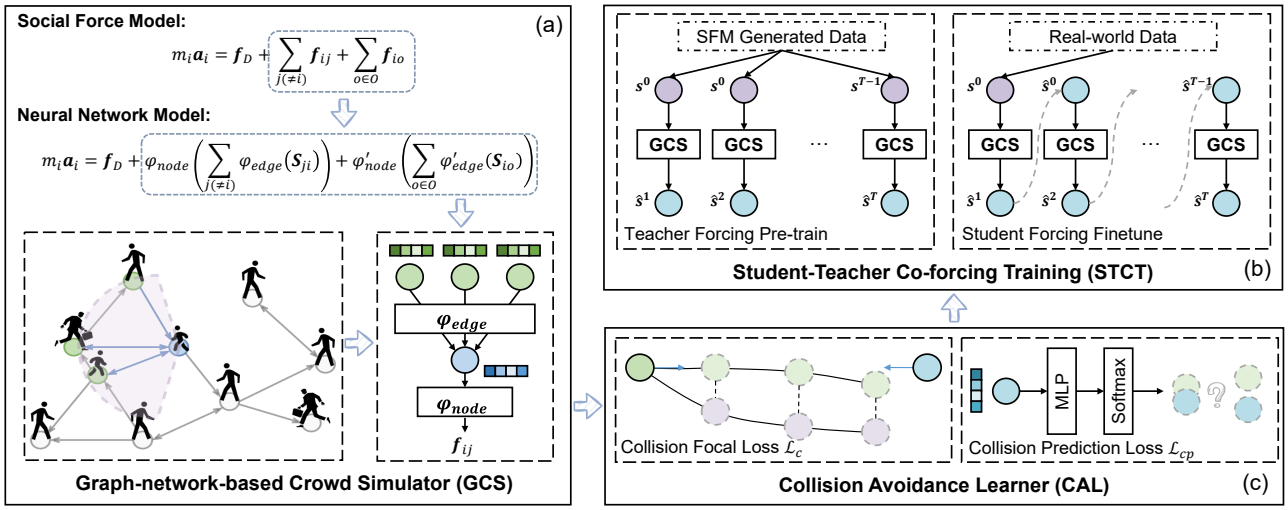

Figure 3: The architecture of our proposed Physics-informed Crowd Simulator (PCS). We base the model on the original social force model and use a graph network as our backbone to introduce strong inductive biases. We further introduce a collision avoidance learner and a student-teacher co-forcing training algorithm to equip our model with long-term generalizability and more realistic micro crowd behaviors.

crowd simulator, the student-teacher co-forcing training algorithm, and the collision avoidance learner.

3.2.1 ***Graph-network-based Crowd Simulator (GCS).*** To ensure we get physically consistent results and reduce the difficulty of the model interaction mechanism's design, we directly base our neural network model on the original social force model and replace its core term, the interaction term that describes how pedestrians and obstacles interact with each other, with our designed neural network model. In this way, our model natually fits the underlying assumptions of social force model and can easily interact with it.

As illustrated in Figure 3(a), the interactions between pedestrians are close to the interactions of particles in a particle system, where one dynamically identifies nearby pedestrians in sight and obstacles and takes proper actions to avoid collisions. As such, we can model pedestrians and obstacles as nodes and the potential interactions as directed edges to build a pedestrian graph. Recent work has demonstrated the effectiveness of using graph networks [12] to model the interactions in particle systems in many domains, such as simulating fluid dynamics [3, 30], estimating molecular properties [2], and simulating deforming surfaces and volumes [25]. We extend this method to crowd simulation and use Graph Networks (GNs) as our neural network model's backbone, which naturally incorporates strong and physically consistent inductive biases. Specifically, a graph network can be seen as a message-passing framework that learns to propagate latent information through graph-structured data [9]. In general, this process learns two distinct functions: an edge function that maps the embeddings of a pair of connected nodes and the embeddings of the corresponding edge to a message, and a node function that aggregates the messages to a node and use it to update the node embeddings, i.e.,

$$GN(\cdot) = \varphi_{node}(\varphi_{edge}(\cdot)). \tag{4}$$

This framework is coherent with the physical nature of how pedestrians interact. The edge function learns how people pay attention to each other, and the node function learns how people collectively consider the environments to make the next move.

In our design, We consider the visual field of each pedestrian as a sector with a radius of r_s meters and a central angle of θ_s degrees. The sector locates at the pedestrian's current location and is symmetrical about the pedestrian's heading direction. We construct a pedestrian graph for each time step with only the links within the pedestrians' visual fields. The edge function φ_{edge} takes each pedestrian's current state as inputs and first processes it into relative values by making a difference since only relative values affect pedestrians' decisions when interacting with others. Then, the relative values are fed into a multi-layer perceptron with residual bypass. Formally, it can be formulated as follows,

$$\varphi_{edge}\left(s_i, s_j\right) = \text{ResMLP}(s_j - s_i), \tag{5}$$

where s_i and s_j denote the states of pedestrians, and $\varphi_{edge}(s_i, s_j)$ learns the message that is passed from the pedestrian v_j to v_i. In terms of the node function, we simply adopt a summation operation, which assumes that each nearby pedestrian affects the focal pedestrian independently. We model obstacle boundaries as a different types of nodes spaced 0.1 meters apart with other settings the same as pedestrians.

3.2.2 ***Student-Teacher Co-forcing Training Algorithm (STCT).*** To acquire long-term generalizability, training with single-step predictions is not enough because the supervised signals are generally noisy in real-world pedestrian data. To be more specific, people are not machines, and they do not always walk in a regular pattern. Some may just randomly hang around. As such, training on single-step predictions can easily result in a degenerated model. A feasible solution is to train on multiple-step rollouts, which is also referred to as student forcing in the NLP literature [27]. In other words, we first simulate for T steps and use them together to compute losses and update model parameters. However, this solution is highly inefficient and very hard to converge. To deal with this problem,

we propose a student-teacher co-training training algorithm that integrates the idea of student-forcing [27] and teacher-forcing [10] mechanisms to train with multiple-step rollouts, as shown in Figure 3(b). Specifically, to improve sample efficiency and facilitate convergence, we first pre-train the model with teacher-forcing that uses the real labels to compute the inputs of each rollout step, and the corresponding loss $\mathcal{L}_T$ is computed as follows,

$$\mathcal{L}_T = \frac{1}{N}\sum_{t=1}^{T}\mathcal{L}_p(\varphi(\boldsymbol{s^{t-1}}), \boldsymbol{p^t}). \tag{6}$$

where N denotes the total number of training samples in T time steps, $\varphi(\cdot)$ is the neural network model, and $\mathcal{L}_p(\cdot)$ is the prediction loss function. In this paper, we use the mean square loss calculated from pedestrians predicted position $\hat{\boldsymbol{p}}$ and real position $\boldsymbol{p}$. Then, we finetune our model with student-forcing and add design a reverse long-term discounted factor λ_s to it to make our model focus on long-term accuracy. In this way, our model is able to deal with the error accumulation in multiple rollouts and get robust predictions. Formally, the student training loss $\mathcal{L}_S$ can be formulated as follows,

$$\mathcal{L}_S = \frac{1}{N}\sum_{t=1}^{T}\lambda_s^{T-t}\mathcal{L}_p(\varphi(\hat{\boldsymbol{s}}^{\boldsymbol{t-1}}), \boldsymbol{p^t}), \tag{7}$$

where $\hat{s}_{t-1}$ denotes the state that calculated from the predictions of the $t-1$ time step, and λ_s is the discounted coefficient.

3.2.3 ***Collision Avoidance Learner.*** With the above designs, we found that our model still fails to learn complex collision avoidance behaviors from the real-world data. After an in-depth analysis of the data, we found the reason is that people only take least efforts to avoid collisions [38]. For example, two people walking towards each other only need to take a small lateral movement to avoid the collision even without slowing down. As a result, the supervised signal of collision avoidance, i.e., the lateral movement, is very small compared to the supervised signal of moving forwards, which leads to degenerated models.

To facilitate our model to capture the dynamic collision avoidance behaviors, we introduce two designs, as shown in Figure 3(c). First, we designed a collision focal loss that upsamples the collision signals from the data. Specifically, for simplicity, we use the direction pointing from the first step of the training rollout to the last step to approximate pedestrians' heading direction. Then, we calculate the mean square distance error between the predictions and labels in the orthogonal direction as the collision focal loss, which can be formulated as follows,

$$\mathcal{L}_c = \frac{1}{N}\sum_{t=1}^{T}\left(\lambda_c\left(\hat{\boldsymbol{p}}^{\boldsymbol{t}} - \boldsymbol{p}^{\boldsymbol{t}}\right)\cdot\boldsymbol{n}\right)^2, \tag{8}$$

where $\boldsymbol{n}$ represents the unit vector on the orthogonal direction, and λ_c represents the focal coefficient. Second, we design an auxiliary collision prediction task that feeds the learned messages into a two-layer MLP with a softmax layer to predict whether two pedestrians will collide in the next second if their states keep the same. The intuition of this task is to introduce a self-supervised signal from the data to help the model distinguish the currently interacting pedestrians so that it can focus on the collision avoidance behaviors better. We use binary cross-entropy loss for training, which can be formulated as follows,

$$\mathcal{L}_{cp} = \frac{1}{N}\sum_{t=1}^{T}\lambda_{cp}\left(\boldsymbol{y_c^t}\log(\hat{\boldsymbol{y}}_{\boldsymbol{c}}^{\boldsymbol{t}}) + (1-\boldsymbol{y_c^t})\log(1-\hat{\boldsymbol{y}}_{\boldsymbol{c}}^{\boldsymbol{t}})\right)^2, \tag{9}$$

where $\boldsymbol{y_c}$ represents the true collision label calculated from data, and $\hat{\boldsymbol{y}}_{\boldsymbol{c}}$ is the predicted collisions.

3.3 Interaction Mechanisms

Physics-informed Machine Learning. To enable the neural network model to learn from the physical model, we extend existing physical-informed machine learning methods [17, 26] to crowd simulation, where we only know little physics and the existing physical models, such as the social force model, are only a rough approximation of the real-world scenarios. Specifically, we replace the supervised labels in our designed teacher-forcing training stage from real-world data to the data generated by the physical model. Compared with the real-world data with heterogeneous and noisy signals, the physical model provides homogeneous signals with long-term generalizability. As such, although the physical model itself may not be able to perfectly describe the real situation, learning from it gives our neural network model a perfect initial point to learn from real-world data. In this way, after finetuned in the student-forcing stage on the real-world data, the neural network model can learn to better approximate the real-world data with physically consistent representations, which can inform the physical model in turn.

Machine-Learning-Aided Physics Discovery. After training, we obtain a neural network model that encodes physical laws closer to reality than the original social force. Thus, if we can decode the law from the well-trained network, we can potentially find a better physical model and deepen our understanding of crowd dynamics. Inspired by previous work [5], this process can be done by performing symbolic regression on the learned edge functions of our model. Specifically, we use a bottleneck design for the edge function, i.e., we set the output to be a 2-dimensional vector, so that the learned messages have a straightforward physical meaning: forces. We consider operators, including $+, -, \times, /$, exp, and cos, and use a high-performance symbolic regression package PySR [4]. We fit the magnitude and the degree of the learned forces separately.

However, directly applying the existing method is highly unstable and typically comes up with meaningless results. The reasons lie in two aspects. First, different from the previous work's scenario, we do not know the exact form of the underlying physics expressions. Thus, we are not sure what variables should be included in the symbolic regression. Directly including all possible variables results in a latent space too large to find a good solution [33]. Second, as we illustrated before, the interactions are sparse in the data, which results in highly unbalanced samples for symbolic regression.

To handle the first problem, we propose to use a human-machine teaming approach in the physics discovery procedure. Specifically, we let experts specify several sets of possible combinations of operators and input variables. Then, for each combination, the symbolic regression solver returns a list of possible equations with their fitting MSE. Finally, the experts jointly evaluate the equations and select one that achieves both interpretability and accuracy to be the improved physical model. We finetune the parameters on the

Groups	Models	GC				UCY			
		MAE	OT	MMD	Collision	MAE	OT	MMD	Collision
Physics-based Models	SFM [13]	1.259	2.114	0.015	622	2.539	6.571	0.129	434
	CA [32]	2.708	5.499	0.062	1492	8.336	79.42	2.022	4504
Data-driven Models	Social-STGCNN [22]	7.669	20.31	0.613	> 9999	8.304	23.31	0.698	> 9999
	Social-GAN [11]	7.513	25.21	0.387	> 9999	8.698	54.54	0.557	> 9999
	Social-LSTM [1]	6.922	11.34	0.345	> 9999	7.291	16.41	0.476	> 9999
Ours	**PCS**	**1.097** (+13%)	1.774 (+16%)	0.015 (+3.3%)	558 (+10%)	**2.330** (+8.2%)	**6.250** (+4.9%)	**0.109** (+16%)	264 (+39%)
	MLAPM	1.136 (+9.8%)	**1.740** (+17.7%)	**0.012** (+22%)	**398** (+36%)	2.406 (+5.2%)	6.383 (+2.9%)	0.125 (+3.1%)	**204** (+53%)
	PCS/IM	2.666	6.498	0.103	616	5.657	21.78	0.426	730
	MLAPM/IM (SFM)	1.259	2.114	0.015	622	2.539	6.571	0.129	434

Table 1: The performance evaluation results on the GC and UCY datasets.

real-world dataset to get the best fit. To solve the second problem, we develop a sampling algorithm based on the assumption that the learned forces are an injective function of the inputs. It first divides the samples into m groups according to the volume of the forces. After that, each group is sampled according to a rejection probability formulated as follows,

$$p_k = 1 - \log(N_k)^2 / N_k, \tag{10}$$

where p_k and N_k represent the rejection probability and the number of samples of the kth group, respectively. Note that the key to the success of physics discovery is the aid of the neural network model because it provides the supervised signal of how two different people interact with each other for symbolic regression solver, which does not exist in the raw data.

4 EXPERIMENTS

4.1 Experiment Setup

4.1.1 ***Datasets.*** We evaluate our proposed model on two public large-scale real-world datasets, including the GC[3] dataset and the UCY[4] dataset. These two datasets differ in the scene, scale, pedestrian density, pedestrian demographics, and pedestrian behavior patterns. The GC dataset is a large pedestrian trajectories dataset from a public transport interchange with 12684 annotated trajectories from a one-hour video. We took 5 minutes out of it with rich pedestrian interactions for training and testing. The UCY dataset is from an outdoor scene at a university containing 528 annotated pedestrian trajectories from a 216-second video. Please refer to Appendix Section A.1 for more details.

4.1.2 ***Baseline Methods.*** We compare our framework with five state-of-the-art models from two categories, including physics-based models and data-driven models. Physics-based models include the Social Force Model (SFM) [13] and the Cellular Automaton (CA) [32], which are widely used in the state-of-the-art commercial crowd simulators. Data-driven methods include Social-LSTM [1], Social-GAN [11], and Social-STGCNN [22], which are the state-of-the-art methods that integrate the recent advance in sequence prediction models, generative adversarial networks, and graph neural networks to model crowd trajectories. Please refer to Appendix Section A.2 for details.

4.1.3 ***Experiment Settings and Reproducibility.*** For the GC dataset, we use three minutes for training, one for validation, and one for testing. For the UCY dataset, we use 108 seconds for training, 54 seconds for validation, and 54 seconds for testing. We use four widely adopted metrics, including the Mean Absolute Error (MAE), characterizing microscopic simulation accuracy, Optimal Transport divergence (OT) and Maximum Mean Discrepancy (MMD), measuring the differences between the generated simulation and the ground truth, and #Collision, characterizing the simulation's fidelity in terms of the collision avoidance behaviors. We perform a grid search on all the hyperparameters for all models. For reproducibility, we make our codes available[5], and further implementation details are given in Appendix A.3.

4.2 Overall Performance Comparison

To examine the effectiveness of our proposed framework, we compare the performance of the Physics-informed Crowd Simulator (PCS) and the Machine-Learning-Aided Physical Model (MLAPM) with different types of state-of-the-art baselines on two large large-scale datasets in Table 1. We also compare them with its ablation version that removes the interaction mechanism, where the physical model and the neural network model do not learn from each other. Here, we summarize key observations and insights as follows:

The Superior Performance of PCS and MLAPM: Both PCS and MLAPM outperform all state-of-the-art baselines across all four evaluation metrics on both datasets. Specifically, taking PCS as an example, it provides a relative performance gain of 13%, 16%, 3.3%, and, 10% on the GC dataset and 8.2%, 4.9%, 16%, and, 39% on the UCY dataset, in terms of MAE, OT, MMD, and #Collisions, respectively, which demonstrates the effectiveness of our proposed model. Further, MAE, OT, and MMD measure whether our model gives accurate simulations from both micro and macro perspectives, while #Collisions reflects the simulations' fidelity. The consistent superior performance across all metrics suggests that our method improves both accuracy and fidelity.

The Effectiveness of the PIML Framework: Removing the interaction mechanisms directly leads to the failure of both the neural network and the physical model. Specifically, the MAE of the neural network model and the physical model increase 59% and

[3]https://www.dropbox.com/s/7y90xsxq0l0yv8d/cvpr2015_pedestrianWalkingPathDataset.rar
[4]https://paperswithcode.com/dataset/ucy
[5]https://github.com/tsinghua-fib-lab/PIML

Groups	Models	GC2 MAE	GC2 OT	GC2 MMD	GC2 Collision	UCY MAE	UCY OT	UCY MMD	UCY Collision
Physics-based Models	SFM [13]	1.545	2.998	0.033	338	2.435	6.535	0.133	358
	CA [32]	2.900	5.952	0.112	1404	8.511	80.10	2.067	4618
Data-driven Models	Social-STGCNN [22]	7.691	24.61	0.509	> 9999	7.763	19.06	0.5719	> 9999
	Social-GAN [11]	8.948	44.18	0.581	> 9999	6.884	32.25	0.512	> 9999
	Social-LSTM [1]	6.889	14.63	0.387	> 9999	7.744	19.04	0.569	> 9999
Ours	**PCS**	1.500 (+2.9%)	2.910 (+2.9%)	0.036 (−9.1%)	330 (+2.4%)	**2.327** (+4.4%)	**6.165** (+5.7%)	**0.107** (+20%)	296 (+17%)
	MLAPM	**1.394** (+9.8%)	**2.765** (+7.8%)	**0.029** (+12%)	**218** (+36%)	2.366 (+2.8%)	6.268 (+4.1%)	0.123 (+7.5%)	**208** (+42%)

Table 2: The generalization performance of the models trained on GC dataset.

9.8% in the GC dataset and 59% and 5.2% in the UCY dataset, respectively, when the interaction mechanism is removed. The results support our core idea that physical models and neural networks can complement each other in crowd simulation, and we can improve them both by the proposed PIML framework.

The Different Advantages of Neural Network and Physical Models: Although PCS and MLAPM achieve comparable performance, they show advantages in different aspects. On both datasets, PCS reaches higher MAE, while MLAPM yields a better collision performance, which is reasonable since neural network models excel at learning large-scale data, and the physical models naturally generate physically consistent results.

Analyses on the Baselines' Performance. All physics-based baselines outperform data-driven baselines. The key reason is that, neural networks models typically have a poor generalization performance compared with knowledge-based models, especially when the simulation time is much longer than that of the training settings. On the contrary, our neural network model generalizes well, which further demonstrates our designs' effectiveness.

4.3 Physics Discovery

With the well-trained neural network model, we perform symbolic regression on the learned edge function characterizing the interaction among pedestrians and successfully find formulas with valid physical meanings. Specifically, on the GC dataset, the best formula fitted on the trained PCS takes the following form,

$$\begin{aligned} \|\boldsymbol{f_{ij}}\| &= \lambda_1 e^{(-d_{ji}/\lambda_2 + \lambda_3 \cos(\theta_{d_{ji}} - \theta_{v_{ji}}) + \lambda_4 \cos(\theta_{d_{ji}} - \theta_{v_{ji}}) d_{ji})}, \\ \theta_{f_{ij}} &= \theta_{d_{ji}} + \delta, \end{aligned} \tag{11}$$

where d_{ji}, $\theta_{d_{ji}}$, $\theta_{v_{ji}}$ are the relative distance between pedestrian i and j, the direction of the relative position, and the direction of the relative velocity, respectively, with λ_1, λ_2, λ_3, λ_4, and δ as the formula's coefficients.

Compared with the original social force model, there are two new terms. First, the new term $\cos(\theta_{d_{ji}} - \theta_{v_{ji}})d_{ji}$ characterizes whether there is a risk of collision between two pedestrians, and the interaction forces are positively correlated with the risk, which makes physical sense. Second, the formula also suggests a small deflection angle between the direction of the force and the direction of the relative position, which echos previous work [35] that demonstrates that the deflection angle describes a more natural collision avoidance behavior. More details on the physics discovery are given in Appendix A.5.

4.4 Generalizability

To examine whether our proposed method can generalize well beyond its training distributions, we test the performance of all methods trained on the GC Dataset on the UCY dataset and a new time period of the GC dataset, which we referred to as GC2 Dataset. Both datasets have a data distribution fundamentally different from the original GC dataset, and the degree of the differences is different. As shown in Table 3, compared with the GC dataset, the GC2 Dataset has a different pedestrian density, average pedestrian speed, and pedestrian distribution. On top of these, the UCY dataset also differs in scenarios and pedestrian demographics.

We show the results in Table 2. Both PCS and MLAPM outperform all the baselines on both datasets, which demonstrates their strong generalizability. In particular, the neural network model, PCS, achieves a comparable generalization performance to the physical model, MLAPM, and it even achieves a better MAE on the UCY dataset, which suggests the effectiveness of the PIML framework.

4.5 Ablation Study

In addition to the interaction mechanism, we further examined how different parts of our designs, including the Student-Teacher Coforcing Training Algorithm, and the Collision Avoidance Learner, contribute to the performance. To this end, we consider four variants of our proposed methods, including *PINNSF w/o STCT-Teacher, PINNSF w/o STCT-Stududent, PINNSF w/o CAL-Lcp*, and *PINNSF w/o CAL-Lc*. For each of the variants, we remove or replace one of the key designs. Specifically, in terms of the STCT, *w/o STCT-Teacher* and *w/o STCT-Student* stands for replacing the training algorithm with only student forcing and only teacher forcing, respectively. In terms of the CAL, *w/o CAL-Lcp* and *w/o CAL-Lc* means removing the collision prediction loss and removing the collision focal loss, respectively.

We report the evaluation results in Figure 4, and we have the following three key observations: First, removing any of the components results in a certain level of performance decrease compared with the full model, which suggests that all the designed components are effective. Second, the STCT successfully combines the advantage of the student-forcing and the teacher-forcing algorithm. Specifically, training the model with the student-forcing algorithm is fast yet results in bad simulation performance. On the contrary, training with teacher-forcing yields a comparable performance with the full model, yet the training takes 7 to 10 times longer than the full model with STCT. Third, both the Collision Prediction Loss

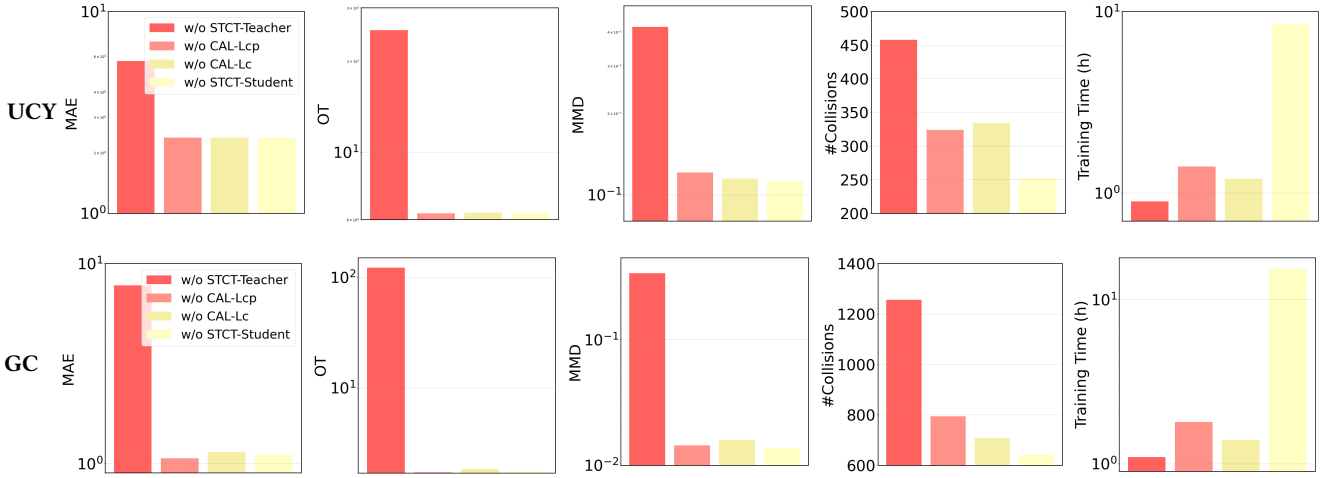

Figure 4: The ablation study on different modules of PCS.

Figure 5: A Case Study on the learned collision avoidance behaviors of (a) SFM, (b) PCS, and (c) MLAPM.

and the Collision focal loss facilitate the model to capture pedestrians' dynamic collision avoidance behaviors and yield a more realistic simulation. To be specific, removing CAL-Lcp and CAL-Lc results in a performance decrease of 42.3% and 26.8% in terms of #Collisions on the GC dataset, and 22.7% and 26.5% on the UCY dataset, respectively. This observation indicates that the supervised signal for collision avoidance is indeed sparse, and finding ways to identify and amplify it is indeed an effective approach.

4.6 Case Study

To further validate whether our model learns a more natural collision avoidance behavior, we conduct a case study on the model's rollout simulations on the test set of the GC dataset. We visualize the simulation results of the SFM, PCS, and MLAPM and present the results in Figure 5. As we can see, in the initial frame, pedestrian No.66 and pedestrian No.63 are about to collide, and the three models generate different simulations. In the SFM's simulation, the two pedestrians collide first and quickly separate as if they were bounced off. This is a typical problem of the social force model, and the reason is that pedestrians are modeled as elastic balls. On the contrary, in the simulation generated by PCS or MLAPM, the two persons naturally passed by, which demonstrates our model's capability to yield more realistic simulations.

5 RELATED WORK

5.1 Microscopic Crowd Simulation

Microscopic crowd simulation models can be mainly divided into two categories, including physics-based methods and deep learning methods [28]. Physics-based methods typically use calibrated physical models, such as forces and fluid dynamics, to model the interactions between individuals [7, 13, 24] or groups of pedestrians [16]. These methods generally yield a robust performance with strong generalizability, yet they fall short in fidelity. With the development of deep learning, many researchers seek to use it to improve simulation performance. For example, some research uses the latest advancement such as GANs and GCNs to predict pedestrians' future behaviors [11, 22]. Nevertheless, they still fail to yield generalizable simulations. In this work, we jointly leverage the strength of the physical and neural network models for crowd simulation through a Physics-Infused Machine Learning framework, which achieves accurate and generalizable simulations.

5.2 Physics-informed Machine Learning

Physics-informed machine learning is a rising research topic that tries to integrate the governing physical laws into machine learning models to help them generate physically consistent predictions [17]. Existing methods typically make a model physics-informed by introducing observational, inductive, or learning biases to guide the models' designing and learning process [17]. Researchers have demonstrated its effectiveness in microscopic fields such as modeling molecular dynamics [15] and fluid dynamics [19], where we know lots of physics. In this work, we extend it to crowd simulation, where little physics is known.

5.3 Machine-learning-aided Physical Discovery

Using neural networks to aid scientific discovery is an emerging field. It originates from the current dilemma of scientific domains that the collected observational data grows much faster than our ability to analyze and further understand them [29]. Neural networks excel at learning in high-dimensional data, and thus it is

possible to leverage it for scientific discovery [21]. Previous work has successfully recovered known formulas, such as Newton force laws, by conducting symbolic regression on learned neural network models with strong inductive biases [5, 18]. We adapt this method to find new formulas in unknown fields by proposing a rejection probability-based sampling algorithm and working in a human-machine teaming manner.

6 CONCLUSION

In this work, we propose a PIML framework for crowd simulation. It jointly leverages the strength of the physical and neural network models by enabling them to learn from each other. Extensive experiments show that with the framework, we are able to get a neural network model with strong generalizability and a new physical model with valid physical meanings. Since the principal aim of this work is to prove the effectiveness of the framework, we only developed minimal designs and went through the PIML iteration for a single round. Future work could consider going through more iterations to further improve the simulation performance and designing better interaction mechanisms. Finally, we also suggest the potential for the PIML framework to generalize to fields beyond crowd simulation.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under 2020AAA0106000, the National Natural Science Foundation of China under 61971267, U21B2036, U20B2060.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 961–971.

[2] Brandon Anderson, Truong Son Hy, and Risi Kondor. 2019. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems* 32 (2019).

[3] Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. 2020. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In *International Conference on Machine Learning*. PMLR, 2402–2411.

[4] M Cranmer. 2020. PySR: Fast & parallelized symbolic regression in Python/Julia.

[5] Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. 2020. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems* 33 (2020), 17429–17442.

[6] Mark Cutler and Jonathan P How. 2016. Autonomous drifting using simulation-aided reinforcement learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5442–5448.

[7] Felix Dietrich and Gerta Köster. 2014. Gradient navigation model for pedestrian dynamics. *Physical Review E* 89, 6 (2014), 062801.

[8] Tian Feng, Lap-Fai Yu, Sai-Kit Yeung, KangKang Yin, and Kun Zhou. 2016. Crowd-driven mid-scale layout design. *ACM Trans. Graph.* 35, 4 (2016), 132–1.

[9] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[11] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2255–2264.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[13] D. Helbing and P Molnar. 1995. Social force model for pedestrian dynamics. *Physical Review E* (1995).

[14] Kiran Ijaz, Shaleeza Sohail, and Sonia Hashish. 2015. A survey of latest approaches for crowd simulation and modeling using hybrid techniques. In *17th UKSIMAMSS international conference on modelling and simulation*. 111–116.

[15] Weile Jia, Han Wang, Mohan Chen, Denghui Lu, Lin Lin, Roberto Car, E Weinan, and Linfeng Zhang. 2020. Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. In *SC20: International conference for high performance computing, networking, storage and analysis*. IEEE, 1–14.

[16] Ioannis Karamouzas and Mark Overmars. 2011. Simulating and evaluating the local behavior of small pedestrian groups. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (2011), 394–406.

[17] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3, 6 (2021), 422–440.

[18] Samuel Kim, Peter Y Lu, Srijon Mukherjee, Michael Gilbert, Li Jing, Vladimir Čeperić, and Marin Soljačić. 2020. Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE Transactions on Neural Networks and Learning Systems* 32, 9 (2020), 4166–4177.

[19] Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. 2021. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences* 118, 21 (2021).

[20] Alexander Lavin, Hector Zenil, Brooks Paige, David Krakauer, Justin Gottschlich, Tim Mattson, Anima Anandkumar, Sanjay Choudry, Kamil Rocki, Atılım Güneş Baydin, et al. 2021. Simulation Intelligence: Towards a New Generation of Scientific Methods. *arXiv preprint arXiv:2112.03235* (2021).

[21] Gary Marcus. 2020. The next decade in AI: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177* (2020).

[22] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. 2020. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14424–14432.

[23] Takuma Otsuka, Hitoshi Shimizu, Tomoharu Iwata, Futoshi Naya, Hiroshi Sawada, and Naonori Ueda. 2019. Bayesian optimization for crowd traffic control using multi-agent simulation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 1981–1988.

[24] Nuria Pelechano, Jan M Allbeck, and Norman I Badler. 2007. Controlling individual agents in high-density crowd simulation. (2007).

[25] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. 2020. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409* (2020).

[26] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378 (2019), 686–707.

[27] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* (2015).

[28] Amir Rasouli. 2021. Pedestrian simulation: A review. *arXiv preprint arXiv:2102.03289* (2021).

[29] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, et al. 2019. Deep learning and process understanding for data-driven Earth system science. *Nature* 566, 7743 (2019), 195–204.

[30] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. 2020. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*. PMLR, 8459–8468.

[31] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. 2018. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*. PMLR, 4470–4479.

[32] Siamak Sarmady, Fazilah Haron, and Abdullah Zawawi Talib. 2010. Simulating crowd movements using fine grid cellular automata. In *2010 12th International Conference on Computer Modelling and Simulation*. IEEE, 428–433.

[33] Michael Schmidt and Hod Lipson. 2009. Distilling free-form natural laws from experimental data. *science* 324, 5923 (2009), 81–85.

[34] Ameya Shendarkar, Karthik Vasudevan, Seungho Lee, and Young-Jun Son. 2006. Crowd simulation for emergency response using BDI agent based on virtual reality. In *Proceedings of the 2006 winter simulation conference*. IEEE, 545–553.

[35] Tomer Weiss, Alan Litteneker, Chenfanfu Jiang, and Demetri Terzopoulos. 2017. Position-based multi-agent dynamics for real-time crowd simulation. In *the ACM SIGGRAPH / Eurographics Symposium*.

[36] Shanwen Yang, Tianrui Li, Xun Gong, Bo Peng, and Jie Hu. 2020. A review on crowd simulation and modeling. *Graphical Models* 111 (2020), 101081.

[37] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* 64, 3 (2021), 107–115.

[38] George Kingsley Zipf. 2016. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books.

(a) GC (b) UCY

Figure 6: The snapshots of two datasets.

Statistics	GC	GC2	UCY
Average duration of a trajectory (s)	11.02	11.72	13.25
Average #Pedestrian per minute (min^{-1})	203	111	132
Pedestrian density (m^{-2})	0.094	0.055	0.058
Average speed ($m \cdot s^{-1}$)	1.155	1.132	1.072
Std of the average speed ($m \cdot s^{-1}$)	0.565	0.534	0.646

Table 3: The basic statistics of the datasets.

A APPENDIX FOR REPRODUCIBILITY

To support the reproducibility of the results in this study, we elaborate on the dataset details, baselines, the experiment settings, and the implementation details of our model. We further give more details on the symbolic regression results.

A.1 Dataset Statistics

To evaluate our models, we introduce two large-scale real-world datasets that differ in the scene, scale, pedestrian density, pedestrian demographics, and pedestrian behavior patterns. The basic statistics of all four datasets are shown in Table 3, and the details of the datasets are as follows:

GC: The GC dataset is built on a one-hour crowd surveillance video. It annotates the walking routes of 12684 pedestrians on a 30m × 35m square in image coordinates. We take a range of 20m×20m, duration of 5-minute space-time slice out of it, which has rich pedestrian interactions for training and testing. As our models are based on the physical laws, we need the pedestrian positions in real-world coordinates but not image coordinates. Therefore, we perform a projective transformation to transform the coordinates. Specifically, we choose the four vertices of the square shown in Figure 6(a), estimate its length and width with the average height of pedestrians, and get the positions of these four vertices in real-world coordinates. With four points' positions in two coordinates, we can calculate the transform matrix and map all points in image coordinates to real-world coordinates. We further fine-tune the previously estimated lengths and widths until the round obstacle in the middle of the snapshot looks round in real-world coordinates. As GC dataset is captured at 1.25Hz ($\Delta t = 0.8$s), which is too long for Euler integration and will introduce huge error, we perform cubic interpolation with SciPy to reduce the time step to 0.08s.

UCY: The UCY dataset is composed of three sequences, but as we focus on long-term simulation in this work, we only choose the sequence of university students which has a long duration and is rich in pedestrian interactions. We also perform a projective transformation taking the four vertices shown in Figure 6(b) of the rectangular street in Figure 6(b) as reference points, and perform cubic interpolation to reduce time step from 0.4s to 0.08s.

A.2 Baselines

As all existing data-driven models need an observed sequence as input, for a fair comparison, we permit all models to observe each pedestrian for 25 steps.

Physics-based Models:

- Social Force Model (SFM): For SFM, we let it observe each pedestrian for 25 steps and calculate the mean speed of each pedestrian as their desired walking speeds respectively, which makes it perform better than the classical SFM, in which all pedestrians have the same desired speed determined manually.
- Cellular Automaton (CA): For CA, We calculate movement probabilities based on how pedestrian cells move in the grid for the initial 25 steps to imitate their different desired walking speeds. We also divide the grid according to the size of the scene, to make the side length of the cells in the grid is 0.5m, which is roughly the occupied area of a standing pedestrian.

Data-driven Models: For all three data-driven models, We use their official implementations, convert our data to their accepted data format, and train the model with the observation length of 25 time steps. We then perform grid searches on the learning rate, batch size, and predicted length. Specifically, we search the learning rates $\in [10^{-4}, 10^{-2}]$, batch sizes $in\{64, 128, 256\}$ and predicted length $\in \{8, 25\}$. We also deal with the dynamic entry and exit of pedestrians with zero padding when testing their performance on long-term simulation. Specifically, we set the position of pedestrians not in the scene currently to zero.

- Social-LSTM[6]: This network predicts each pedestrian's trajectory with a correlative LSTM unit, and introduces an aggregation layer that enables LSTMs to share hidden-states with other spatially nearby LSTMs.
- Social-GAN[7]: This model is a Generative Adversarial Network (GAN) composed of an RNN Encoder-Decoder generator and an RNN based encoder discriminator, and also introduces an aggregation layer in the generator. As this model can only generate a trajectory sequence of the given length in a single prediction, we take its predicted sequence as the new observe sequence and rollout to the end.
- Social-STGCNN[8]: Instead of using the aggregation layers, this network model the pedestrian trajectories as a spatio-temporal graph, and model the social interactions between the pedestrians with graph edges. Same as Social-GAN, we take its predicted sequence as the new observe sequence and rollout to the end.

A.3 PCS Implementation Details

PCS is based on graph network models, and we construct the graph based on each pedestrian's visual field. In the implementation, we set the sector-based visual fields' radius r_s as 4 meters and the central angle θ_s as 180 degrees. For the fully-connected layers with

[6]https://github.com/quancore/social-lstm
[7]https://github.com/agrimgupta92/sgan
[8]https://github.com/abduallahmohamed/Social-STGCNN

residual bypass in the edge function, we use a node embedding size of 128. During training, we set the rollout steps as a hyper-parameter with its values ranging from 5 - 15 time steps, i.e., 0.4s - 1.2s. Intuitively, this parameter determines the temporal receptive field of our model. Thus, if the value is too small, the learning process will be strongly affected by the random noise in each time step and cannot take advantage of the designed student-training stage. We jointly optimize all the designed losses, including the position prediction loss, the collision prediction loss, and the collision focal loss through the Adam optimizer. We also add an l_2 regularization of $1e-6$ and a dropout rate of 0.5 to prevent overfitting. To find the best hyper-parameters, we perform a grid search on the learning rate, batch size, l_2 regularization coefficient, and dropout rate in both the student-forcing training and teacher-forcing training stages. Specifically, we search learning rate $\in [1e-3, 1e-6]$, batch size $\in \{32, 64, 128\}$, l_2 regularization coefficient $\in [1e-3, 1e-6]$. All the evaluated models are implemented with Pytorch and trained on a server with two CPUs (AMD Ryzen 2990WX * 2) and four GPUs (NVIDIA GeForce RTX 2080 * 4). Note that all the training time we reported in the paper is trained on two GPUs with the server.

A.4 Evaluation Metrics

Mean Absolute Error (MAE): It can be expressed as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathbf{p}}_i - \mathbf{p}_i\|_2, \tag{12}$$

where N denotes the total number of predicted instances, $\hat{\mathbf{p}}_i$ denotes the prediction of the pedestrian position and $\mathbf{p}_i$ denotes the real position, and the $\|\cdot\|_2$ is the l_2 norm of a given vector.

Optimal Transport (OT): It measures the distance between two distributions as the minimum cost to transport from distribution P to distribution Q. Specifically, it has the form as

$$\begin{aligned} OT(P\|Q) &= \inf_{\pi} \int_{X\times Y} \pi(x,y)c(x,y)\mathrm{d}x\mathrm{d}y, \\ s.t. \int_Y \pi(x,y)\mathrm{d}y = P(x), &\int_X \pi(x,y)\mathrm{d}x = Q(y), \end{aligned} \tag{13}$$

where $\pi(x,y)$ can be approximated from $P(x)$ and $Q(y)$ by Sinkhorn Algorithm[9]. In this work, We use 2D Wasserstein distance, i.e. $c(x,y) = \|x-y\|_2^2$, and $X \equiv Y$ is the simulation duration, $P, Q : \mathbb{R} \to \mathbb{R}^2$ is the prediction trajectory and the true trajectory.

Maximum Mean Discrepancy (MMD): It takes the maximum difference between two distributions' moments in any order as their distance. This metric can be calculated by kernel embedding of distributions and implemented easily on PyTorch[10].

#Collision: We count two pedestrians with a distance less than 0.5m as a collision and take the summation of collisions in all frames as #Collision. But considering that pedestrians could walk with their friends, to whom they won't keep a large social distance, We take the pair of pedestrians that have a collision in more than 2 seconds as friends and do not count the collisions between them into #Collision.

[9] https://dfdazac.github.io/sinkhorn.html

[10] https://github.com/easezyc/deep-transfer-learning/blob/master/MUDA/MFSAN/MFSAN_3src/mmd.py

A.5 Detailed Results of Physics Discovery

After training, we obtain a neural network model that encodes physical laws closer to reality than the original social force. We then perform symbolic regression on the learned edge functions of our model to find a better physical model and deepen our understanding of crowd dynamics. We try to perform symbolic regression in cartesian coordinates, yet obtain no valid results. Thus, we propose to perform symbolic regression in polar coordinates and fit the magnitude and the degree of the learned forces separately. We assume that interaction is determined only by relative quantities, and give the variable set including:

- **Relative position** $\mathbf{d}_{ji}$: Including the magnitude d_{ji} and the degree $\theta_{d_{ji}}$ of the relative position between two pedestrians, i.e. the coordinates of a pedestrian in the polar coordinates which takes the direction of a given pedestrian's velocity as the polar axis.
- **Relative velocity** $\mathbf{v}_{ji}$: Including the magnitude v_{ji} and the degree $\theta_{v_{ji}}$ of the relative velocity between two pedestrians.
- **Relative acceleration** $\mathbf{a}_{ji}$: Including the magnitude a_{ji} and the degree $\theta_{a_{ji}}$ of the relative acceleration between two pedestrians.
- **Cosine of the angle between relative position and relative velocity** $\cos(\theta_{d_{ji}} - \theta_{v_{ji}})$: This variable provides a rough estimate of the possibility of collision between two pedestrians.
- **Collision flag in 1 second** coll_{ji}: This item is set to 1 when two pedestrians with relative position $\mathbf{d}_{ji}$, relative velocity $\mathbf{v}_{ji}$, and zero acceleration will collide in 1 second, and is set to 0 when they will not collide.

As directly including all possible variables results in a latent space too large to find a good solution, we let experts specify several sets of possible combinations of operators and input variables. For example, we obtain the best fit on the UCY dataset with a symbol set of d_{ij}, v_{ij}, $\cos(\theta_{d_{ij}} - \theta_{v_{ij}})$, and a operator set of $+$, $-$, $\times$, $/$ and exp. Here, we report the best formula fitted on the UCY dataset:

$$\begin{aligned} \left\|\mathbf{f}_{ij}\right\| &= \lambda_1 e^{\mathrm{coll}_{ji} \times d_{ji}/\lambda_2 + \lambda_3 \mathrm{coll}}, \\ \theta_{f_{ij}} &= \theta_{d_{ji}} + \delta, \end{aligned} \tag{14}$$

where λ_1, λ_2, λ_3 and δ are the formula's coefficients. Compared with the original social force model, the coll_{ji} item acts as a switch. Specifically, when two pedestrians will collide in 1 second, coll_{ji} will be 1 and the magnitude of the force will have the same form as original social force, suggesting that pedestrians will react strongly to avoid predictable collisions. When two pedestrians will not collide in 1 second, coll_{ji} will be 0 and the magnitude of the force will be a constant, meaning that pedestrian will not pay extra attention on others who will not generate collision. The formula also suggests a small deflection angle between the direction of the force and the direction of the relative position, which can give a more natural collision avoidance behavior.