

Scenario Generalization of Data-driven Imitation Models in Crowd Simulation

Gang Qiao
Rutgers University
gq19@cs.rutgers.edu

Honglu Zhou
Rutgers University
hz289@scarletmail.rutgers.edu

Mubbasir Kapadia
Rutgers University
mk1353@cs.rutgers.edu

Sejong Yoon
The College of New Jersey
yoons@tcnj.edu

Vladimir Pavlovic
Rutgers University
vladimir@cs.rutgers.edu

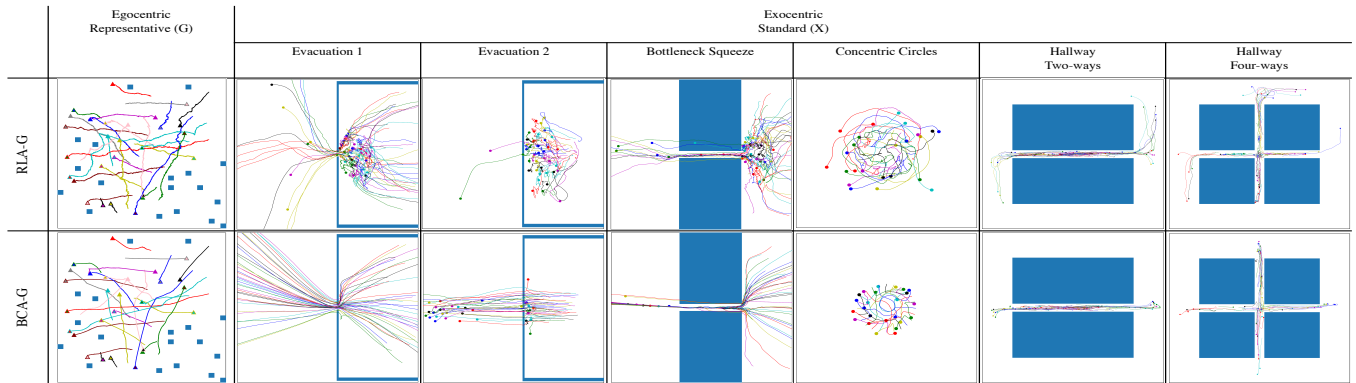


Figure 1: Visualization of trajectories of RLA-G and BCA-G generalized to egocentric representative and exocentric standard scenarios. The lower AO metric of RLA-G than that of BCA-G results from the fact that much fewer RLA-G agents can avoid obstacles and reach destinations.

ABSTRACT

Crowd simulation, the study of the movement of multiple agents in complex environments, presents a unique application domain for machine learning. One challenge in crowd simulation is to imitate the movement of expert agents in highly dense crowds. An imitation model could substitute an expert agent if the model behaves as good as the expert. This will bring many exciting applications. However, we believe no prior studies have considered the critical question of how training data and training methods affect imitators when these models are applied to novel scenarios. In this work, a general imitation model is represented by applying either the Behavior Cloning (BC) training method or a more sophisticated Generative Adversarial Imitation Learning (GAIL) method, on three typical types of data domains: standard benchmarks for evaluating crowd models, random sampling of state-action pairs, and egocentric scenarios that capture local interactions. Simulated results suggest that (i) simpler training methods are overall better

than more complex training methods, (ii) training samples with diverse agent-agent and agent-obstacle interactions are beneficial for reducing collisions when the trained models are applied to new scenarios. We additionally evaluated our models in their ability to imitate real world crowd trajectories observed from surveillance videos. Our findings indicate that models trained on representative scenarios generalize to new, unseen situations observed in real human crowds.

CCS CONCEPTS

• Computing methodologies → Multi-agent systems.

KEYWORDS

Crowd Simulation; Imitation Learning; Scenario Generalization

ACM Reference Format:

Gang Qiao, Honglu Zhou, Mubbasir Kapadia, Sejong Yoon, and Vladimir Pavlovic. 2019. Scenario Generalization of Data-driven Imitation Models in Crowd Simulation. In *Motion, Interaction and Games (MIG '19)*, October 28–30, 2019, Newcastle upon Tyne, United Kingdom. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3359566.3360087>

1 INTRODUCTION

Imitating the movement of a goal-directed expert agent in a complex scenario, involving obstacles and other agents, has recently received attention from machine learning community. Researchers aim to create data-driven models to predict the next movement

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MIG '19, October 28–30, 2019, Newcastle upon Tyne, United Kingdom

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6994-7/19/10...\$15.00

<https://doi.org/10.1145/3359566.3360087>

decision (velocity) of an agent given current state (local observation on environment and neighboring agents), by imitating the demonstrated crowd movement of an expert. A good imitator could substitute the expert, with potentialities in some applications. For instance, we may want to imitate the controlling signals (steering angle, acceleration, etc.) demonstrated by a real person steering a vehicle in parallel parking scenarios, whose decisions are based on the person's successive local observations, and then replace the human efforts with the imitator to provide controlling signals given the observations of a camera equipped on the vehicle in new parallel parking scenarios.

In this paper, the term “scenario” refers to the configuration of environment obstacles as well as the tasks (initial positions and destination positions) for all involved agents. Agents may have different destination positions. Existing works, e.g., [Qiao et al. 2018], train and test models over the same environment with only initial and goal positions and the number of agents varied, or over environments with small obstacle adjustment [Long et al. 2017]. To the best of our knowledge, no prior studies have considered the critical question of how training data and training paradigms affect imitation models when these models are generalized to substantially different scenarios.

The generalization ability of an imitator to new scenarios is subtly but essentially different from the regular generalization ability of a model, in three aspects. (1) For regular generalization, the model has full knowledge about a scenario such as the initial/destination positions of all agents and the positions of all obstacles, while in scenario generalization, each agent assigned with an imitator may only know its own destination and make a decision based on its own partial observation, without knowing the destinations and observations of other agents. (2) For regular generalization, test samples are usually isolated: a previous test sample does not influence the next test sample. In contrast, in scenario generalization all agents (each agent is equipped with an imitator) move step by step and synchronously, during which the previous observation and decision of an agent influence its next observation and decision successively. (3) Instead of measuring on isolated state-action pairs in regular generalization, measurement in scenario generalization is on the overall generated trajectories with multiple metrics, and some of them might be mutually balanced.

Unlike most previous works that focus on improving a specific expert model, or imitating an expert model for a specific behavior/in a specific scenario, our main goal is to investigate the effect of training paradigm and training data on the scenario generalization ability of an imitator, by comparing the combinations of representative training paradigms and representative data domains. Specifically, two training paradigms are studied: (1) **Behavior Cloning (BC)**: an approximation of maximum likelihood estimation by fitting a neural network regressor, capable of representing many classic regressors (support vector regressor, random forest, etc.) and (2) **Reinforcement Learning (RL)**: a Markov decision process solved by Generative Adversarial Imitation Learning (GAIL) [Ho and Ermon 2016], leading to a solution that is theoretically equivalent to any two-step reward estimation followed by policy search procedures. Although only two paradigms are studied, encompassing two distinct families of training approaches, the former focusing on imitating simple reactive behaviors, while the latter considering the

impact of local actions on accumulated outcomes, they are generic modeling approaches and represent most data-driven models in crowd simulation.

In addition to training paradigms, three data domains are developed: (1) a set of six standard scenarios which serve as benchmarks for evaluating crowd simulation, (2) a random sampling of inter-agent interactions at a single time step, and (3) a set of representative scenarios to capture inter-agent and agent-obstacle interactions during the overall navigation procedure. These data domains span the spectrum of a few but complex and crowded scenarios, to many random discrete snapshots for the immediate response of a model to inter-agent interactions, and a large number of sampling of small-scale, but general interaction situations that individuals encounter.

Combinations of training paradigms and data domains are systematically evaluated in the ability to emulate expert trajectories while avoiding collisions with other agents and environment obstacles in substantially new scenarios. Our empirical results suggest that (i) a simpler training method is better than a more complex training method, (ii) training samples with diverse agent-agent and agent-obstacle interactions are beneficial for reducing collisions when the trained models are applied to new scenarios.

We additionally evaluated all five models in their ability to imitate real world crowd trajectories observed from surveillance videos. Results indicate that models trained on representative scenarios generalize to new, unseen situations observed in real human crowds.

2 PRIOR WORK

Crowd simulation and analysis are paramount examples of distributed AI modeling, with application across a variety of domains including computer graphics, crowd tracking, crowd trajectory estimation and optimization [Ali et al. 2013; Cheng et al. 2018; Junior et al. 2010; Kapadia et al. 2015; Lee et al. 2018; Liu et al. 2017; Qiao et al. 2018]. We provide a brief summary of the most related literature below.

2.1 Crowd Simulation Approach

Methods in this approach rely on pre-determined physical, social or geometric rules or computational procedures to decide a velocity for an agent to execute in the next time duration [Karamouzas et al. 2014; Kim et al. 2013, 2012; Knob et al. 2019; Ren et al. 2017; Vicsek et al. 1995], and hence they are not data-driven models. In social force method [Helbing and Molnar 1995], an agent is simultaneously attracted by its goal and repelled by other agents and obstacles. Each force obeys the gravitation-like inverse-square law, and the composition of all forces of an agent determines the acceleration of that agent. Geometric methods such as velocity obstacles [Fiorini and Shiller 1998] define a geometrical cone in the relative velocity space, inside which a collision will occur. Extensions to this work [Van Den Berg et al. 2011] define the set of collision-avoiding velocities and induce Optimal Reciprocal Collision Avoidance (ORCA) that provides a sufficient condition for collision avoidance if agents are not densely packed.

2.2 Behavior Cloning (BC) Approach

This approach view state-action (s, a) pairs as independent samples and use these samples to fit a regression model based on maximum likelihood estimation (MLE). Thus, models [Long et al. 2017; Qiao et al. 2018; Torabi et al. 2018] within this approach are data-driven models. If the regression model is represented by a neural network (NN), it stands for a general function and covers many traditional learning models. [Long et al. 2017] randomly places neighboring agents around a reference agent and randomly samples the current velocities for all agents. Given a preferred velocity for the reference agent, they use ORCA [Van Den Berg et al. 2011] to produce the corresponding action (velocity) for the reference agent in that state. Such a uniform sampling over state space yields a sufficient amount of state-action pairs to fit an NN model. Similarly, [Qiao et al. 2018] simulates the social force model to collect expert trajectories, and treat state-action pairs from the expert trajectories as independent samples to fit an NN model. However, the trained model is used to provide a velocity prior used for trajectory interpolation, where the actions of individual agents become seemingly decoupled from each other, leading to a computationally efficient solution.

2.3 Reinforcement Learning (RL) Approach

RL methods [Arora and Doshi 2018; Finn et al. 2016; Ho and Ermon 2016; Kuefler et al. 2017; Pautrat et al. 2018; Schulman et al. 2015; Ziebart et al. 2008] alternate between sampling trajectories with a policy model in an environment and updating the policy model based on reward signal. The goal is to maximize the expected accumulated reward by balancing environment exploration and reward exploitation. [Torrey 2010] introduces RL to crowd simulation and proposes several new challenges when it scales from single-agent to multi-agent setting. A recent work presents an agent-based, RL navigation method that learns a single unified policy to be applicable to several scenarios and settings, without considering environmental obstacles [Lee et al. 2018]. Some other works [Casadiago 2014] also use RL to approach the problem of data-driven trajectories learning [Cheng et al. 2018] in crowd simulation.

The reward function in RL is either human-defined [Long et al. 2018], or learned with inverse-RL (IRL) methods [Arora and Doshi 2018; Finn et al. 2016; Ziebart et al. 2008]. For fair comparison, we consider only data-driven models and thus the reward function is estimated via IRL from demonstrated expert trajectories.

[Finn et al. 2016] proposes guided cost learning for IRL, which alternates between (1) estimating the partition function (so as to search for the current optimal parameter point) by sampling the proposal distribution, and (2) optimizing the proposal distribution to reduce the variance of the partition function. Given estimated reward function, [Schulman et al. 2015] proposes to optimize the policy by searching at each iteration within a region centered at previously estimated parameter point, which could be considered as KL-constrained natural gradient ascend. Recently, [Ho and Ermon 2016] proposes generative adversarial imitation learning (GAIL), an imitator of demonstration. It is model-free, without the need to estimate the dynamics explicitly. More importantly, they proved that any two-step reward estimation and policy optimization procedures (IRL-RL) are equivalent to the one-step adversarial learning. Thus GAIL covers most traditional data-driven RL methods, avoiding

us the need to develop a specific RL model. We will describe this training paradigm in detail in the following section, and apply it within the context of multi-agent goal-directed collision avoidance.

2.4 Comparison of Three Approaches

The three categories of approaches have their own characteristics, which make them complementary to others. (1) Some methods describe certain movement knowledge of physical particles, geometric objects, animals or humans, and represent the knowledge explicitly for making velocity decisions in crowd simulation, rather than focusing on imitating/learning implicit knowledge from demonstrated data. (2) Provided with expert trajectories, BC suffers from the well-known compounding error problem [Ross and Bagnell 2010]. That is, when BC's decision deviates a little from the expert's decision, the next state would be less represented in expert trajectories, leading to further deviation from the expert decisions. When such error accumulates, it might end up with invalid situation (e.g., off-road driving). (3) RL methods are much more sophisticated in training compared with BC. (4) One can anticipate that the physics-based approach has the best scenario generalization ability, followed by BC, while RL have the least scenario generalization ability. This might be explained by Occam's razor law: physics-based methods follow a few rules or computational procedures, BC learn independently from (s, a) pairs, while RL explore and learn from the same environment repeatedly.

Despite these insights, it is still not clear to which extent the data-driven models differ from each other, in the sense of generalization capacity to new scenarios. Therefore, we specifically seek to determine what training paradigm / training data is the most suitable for developing generalizable models for multi-agent goal-directed collision avoidance. Considering the above-mentioned characteristics of the three approaches, we use physics-based methods to generate different types of expert trajectories and utilize these trajectories for training with BC/RL approaches, followed by comparing the scenario generalization capacities of those trained models.

3 PROBLEM FORMULATION

Let $\mathcal{S}$ and $\mathcal{A}$ be the state and the action space, respectively, of an agent given an environment $\mathcal{E}$. Let $s_t \in \mathcal{S}$ denote the state of an agent at time t , where t is the discrete step index with $t = 0, 1..T$ and T is the maximal number of steps. An agent's state typically includes what the agent locally observes about the world around itself, and may also incorporate some guidance signals received from external sources. Let $a_t \in \mathcal{A}$ denote the action of the agent at time t , determined by the agent's policy function (decision-making function) based on s_t . That is, $a_t = \pi(s_t)$, with $\pi(\cdot)$ representing the policy function adopted by the agent. The action could be high-dimensional controlling signal (steering angle, acceleration, etc.), but in its simplest form, it may represent the velocity that will take the agent to a new position, leading to a new local observation of the world. At each step t , assume the next state s_{t+1} of an agent depends only on its current state s_t and current action a_t . For comparison, we further assume all agents are homogeneous, i.e., they utilize the same policy π for their execution, however no agent knows what policies other agents adopt. Therefore, the dynamics, $s_{t+1} \sim P(\cdot | s_t, a_t)$, is probabilistic due to partial observation of the agent

and unknowing about other agents' decisions at step t . Furthermore, a state-action pair $(s_t, \pi(s_t))$ can be evaluated by a cost function associated with the world system: $c(s_t, \pi(s_t)) = r_t$, where $r_t \in \mathbb{R}$ is a reward value for the action the policy decides based on s_t . For instance, the cost function may evaluate a lower reward if executing a_t incurs agent-agent/agent-obstacle collisions and a higher reward otherwise.

Given the above definitions, the problem can be formulated as a Markov Decision Process (MDP). For a given cost function $c(\cdot, \cdot)$, the goal is to find π^* that maximizes the accumulated rewards along the expected trajectory:

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=1}^T \gamma_t c(s_t, a_t) \right], \quad (1)$$

where $\gamma_t \in [0, 1)$ denotes the discount factor.

One issue is that the cost function is usually unknown or implicit, and the demonstrated expert trajectories also conceal the reward signals. In other words, the demonstrated expert trajectories are $\{(s_0, a_0, s_1, a_1, \dots, s_T, a_T)\}$, not $\{(s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_T, a_T, r_T)\}$. Another important issue is that neither the stochastic dynamics nor the expert policy π_E is known, stemming from the complex nature of the crowd simulation task. Typically, there are four ways to handle these challenges: (1) use IRL to estimate a cost function that favors the expert trajectories with high accumulated rewards (in the following, we denote the cost function estimated from expert trajectories as c^*), (2) estimate the dynamics from data, (3) use RL to estimate π^* to mimic the expert policy π_E using the IRL-found cost function c^* , and (4) use BC to directly estimate π^* from the expert trajectories. We focus on (3) and (4) in this work.

4 BEHAVIOR CLONING AGENTS

Behavior cloning methods could be viewed as a special case of the formulation in Eq. 1: a reduction when the cost function $c(s_t, a_t)$ of BC is a differentiable training loss function, with discount factor $\gamma_t \equiv 1$, and the dynamics of BC depends only on the data distribution, independent from the current (s_t, a_t) pair.

Training a model in BC paradigm is identical to fitting a supervised regressor. For instance, one can fit a Neural Network (NN) regressor with the cost function $c(\cdot, \cdot)$ set to L2 loss:

$$a_t = f_{NN}(s_t | \theta_{NN}), \quad (2)$$

where s_t is the state of an expert agent, including its local visibility (e.g., a range map, a velocity map) from the center point of this agent, as well as a local guidance velocity and a global guidance velocity – see details on state representation in the evaluation part. θ_{NN} is the model parameter. Such NN based model can also represent many traditional regressors including support vector regressor, random forest, etc.

As mentioned earlier, in crowd simulation agents are goal directed. To arrive the final destination, it is critical for the state s_t of an agent to contain not only the local observation about where neighboring agents/obstacles exist and what the relative velocities of neighboring agents/obstacles are wrt the agent, but also a local guidance direction (or local guidance velocity) that leads the agent to its own nearest sub-goal location. Such local guidance velocity is agent-specific and dependent on the current location of the agent.

In addition, due to the existence of environmental obstacles, the local guidance velocity may not coincide with the global guidance velocity that directly points to the final destination of the agent.

The local guidance velocity can be either learned from experience (e.g., from expert trajectories) or planned by an external planner provided with the environment configuration and the initial/destination positions of an agent. When the movement of expert agents forms a flow pattern, indicating that two nearby agents have similar trajectories, the flow can be learned with a Gaussian Process (GP). With the learned GP, when an imitator is generalized to that environment, the prediction of the GP could provide the local guidance velocity for the imitator in s_t :

$$a_t^{\text{local}} \sim GP(\cdot | (x, y, t), \mathbf{X}_{\text{train}}, \theta_{GP}), \quad (3)$$

where $(x, y, t)^T \in \mathbb{R}^3$ is the spatial-temporal location of the imitator, $\mathbf{X}_{\text{train}}$ is the training data, θ_{GP} is the hyper-parameter, and a_t^{local} is the local guidance velocity at the current spatial-temporal location.

On the other hand, when the movement of expert agents does not form a flow pattern, one may use a path-planning algorithm to provide the local guidance velocity in s_t .

5 REINFORCEMENT LEARNING AGENTS

Reinforcement learning first estimates c^* from expert trajectories, then estimates the optimal policy π^* to approximate the underlying but unknown expert policy π_E . One approach to recover c^* is the maximum causal entropy IRL [Ziebart et al. 2008]:

$$\arg \max_{c \in \mathcal{C}} \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi} [c(s, a)] - \mathbb{E}_{\pi_E} [c(s, a)], \quad (4)$$

where $H(\pi) \triangleq \mathbb{E}_{\pi} [-\log \pi(a|s)]$, $\mathcal{C}$ is the family of cost functions, Π is the family of policy functions, and π_E denotes the expert policy that generates the expert trajectories. Here c^* minimizes the expected cost of expert trajectories while maximizes the cost of the policy trajectories. If such c^* is obtained, the optimal policy π^* satisfies

$$\arg \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi} [c(s, a)], \quad (5)$$

and can be estimated in a regularized RL procedure.

The two-step IRL-RL are complex. Recently, [Ho and Ermon 2016] have proposed GAIL, in which they have shown that the two-step IRL-RL is identical to a one-step occupancy matching procedure.

To induce GAIL paradigm, they first add a closed, proper convex cost function regularizer $\psi : \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \rightarrow \mathbb{R}$ to alleviate the overfitting issue stemming from the finite dataset size. With this regularizer, the IRL objective is given by

$$\arg \max_{c \in \mathcal{C}} -\psi(c) + \left(\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi} [c(s, a)] \right) - \mathbb{E}_{\pi_E} [c(s, a)]. \quad (6)$$

On the other hand, they define an occupancy measure $\rho_{\pi} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ of a stochastic policy π as $\rho_{\pi}(s, a) = \pi(a|s) \sum_t \gamma_t P(s_t = s | \pi)$. ρ_{π} describes the distribution of (s, a) pairs that an agent encounters when navigation with policy π . (policy is stochastic in training but deterministic in testing to trade off exploitation for exploration).

With this definition, they show that RL and IRL solve the primal and the dual problems of occupancy measure matching, with optimal solutions forming a saddle point. That means any two-step IRL-RL is equivalent to the following one-step formulation:

$$\pi^* = \arg \min_{\pi \in \Pi} \psi^*(\rho_\pi - \rho_{\pi_E}) - \lambda H(\pi) \quad (7)$$

where ψ^* (the convex conjugate of function ψ) is the convex function measuring the deviation of ρ_π from ρ_{π_E} . This suggests that finding π^* to approach π_E can be transformed to matching the occupancy measure between ρ_π and ρ_{π_E} . Here λ is an introduced regularization parameter to control the entropy term.

They further show that there exists a specific ψ :

$$\psi_{GA}(c) \triangleq \begin{cases} \mathbb{E}_{\pi_E} [g(c(s, a))] & \text{if } c < 0 \\ +\infty & \text{otherwise} \end{cases} \quad (8)$$

where $g(x) = -x - \log(1 - e^x)$ if $x < 0$, otherwise $g(x) = +\infty$, such that $\psi_{GA}(\rho_\pi - \rho_{\pi_E})$ can be represented as:

$$\psi_{GA}^*(\rho_\pi - \rho_{\pi_E}) = \max_D \mathbb{E}_\pi [\log(D(s, a))] + \mathbb{E}_{\pi_E} [\log(1 - D(s, a))]$$

where $D : \mathcal{S} \times \mathcal{A} \rightarrow (0, 1)$, which is employed to predict the probability that a given state-action pair comes from π rather than π_E , with the relation $c(s, a) = \log D(s, a)$.

In that case, the one-step formulation in Eq. 7 is reduced to an adversarial form:

$$\min_{\pi} \max_D \mathbb{E}_\pi [\log(D(s, a))] + \mathbb{E}_{\pi_E} [\log(1 - D(s, a))] - \lambda H(\pi). \quad (9)$$

Therefore, the final objective given by Eq. 9 can be optimized adversarially with gradient descend and policy optimization (e.g., trust region policy optimization [Schulman et al. 2015]). Eventually, both cost function and policy function can be obtained simultaneously, capable of representing a general two-step IRL-RL models.

6 DATA DOMAINS

We identify three scenario domains in this work: exocentric standard scenarios (X), egocentric representative scenarios (G), and egocentric random scenarios (R). In all domains, an agent is represented as a circle with the radius of 0.5 meters.

6.1 Exocentric Standard Scenarios (X)

This domain provides a few but complex and crowded scenarios, including six environment benchmarks used to evaluate computational models of crowd movement [Qiao et al. 2018; Singh et al. 2009; Yoon et al. 2016]. The six scenarios (with variation in agent density and initial/destination positions) include:

- (1) **Evacuation 1.** Many agents must evacuate a room, with only one small doorway of width 2.4 m. Agents are heading toward distinct target locations outside the room.
- (2) **Evacuation 2.** Similar to Evacuation 1 but the doorway width is narrowed to 1.4 m. Also agents are heading toward the same target location outside of the room.
- (3) **Bottleneck squeeze.** All agents begin on one side of the area, and enter and traverse a hallway to reach the target.
- (4) **Concentric circles.** Agents are symmetrically placed along a circle and aim to reach antipodal positions.
- (5) **Hallway two-way.** Many agents traveling in either direction through a hallway. Agents are expected to form lanes.

Table 1: characteristics of three data domains

X	A few challenging obstacle configurations; with inter-agent interactions
G	Diverse inter-agent and agent-obstacle interactions; many test scenarios
R	Numerous diverse snapshots on inter-agent interactions; no complete trajectories; no obstacles

- (6) **Hallway four-way.** Many agents arriving from and traveling to any of the four cardinal directions.

Illustrations of the six scenarios are shown in Fig. 2.

6.2 Egocentric Representative Scenarios (G)

Exocentric standard scenarios provide challenging crowd tasks, but may not be able to sufficiently provide a *representative space* of challenging local interactions individuals encounter in crowd. Egocentric random scenarios provide random samples of state-action pairs, but these samples can not form complete trajectories, and there are no agent-obstacle interactions.

In an effort to produce a data domain with a large number of sampling of small-scale, but general inter-agent and agent-obstacle interactions that individuals encounter, we refer to [Kapadia et al. 2011], which characterizes the representative space of scenarios observed in crowds, and a sampling strategy to generate a finite set of scenarios with sufficient coverage. Specifically, a considerable amount of simulation scenarios are uniformly sampled from this scenario space for both training and testing (4000 for training, 100 for testing). Each scenario contains randomly distributed obstacles and randomly assigned initial/destination positions of agents, with expert driven by the social force model. Fig. 3 illustrates two samples in this domain.

6.3 Egocentric Random Scenarios (R)

The randomly generated scenarios proposed in [Long et al. 2017] constitute the second domain, where a sufficient number of samples are collected by uniformly and independently sampling over the state space. The positions of neighboring agents, previous velocities of neighboring agents and the preferred velocity of a reference agent are randomly set to construct a particular state for the reference agent at a step, while the expert decision of the reference agent at this step is queried from ORCA [Van Den Berg et al. 2011] given the same state. This produces many discrete and independent snapshots for immediate responses of an expert to inter-agent interactions. Note that in any sample of this domain, there are no obstacles, thus no agent-obstacle interactions involved.

6.4 Summary of Three Data Domains

In the following sections, abbreviations X, G, and R are used to denote the domain of exocentric standard, egocentric random and egocentric representative scenarios respectively. Tab. 1 summarizes the characteristics of each domain.

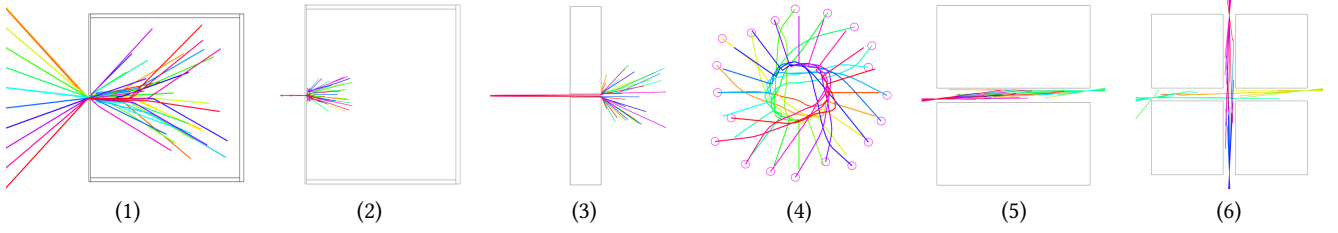


Figure 2: Exocentric standard scenarios (X). (1) Evacuation 1, (2) Evacuation 2, (3) Bottleneck squeeze, (4) Concentric circles, (5) Hallway two-way, (6) Hallway four-way.

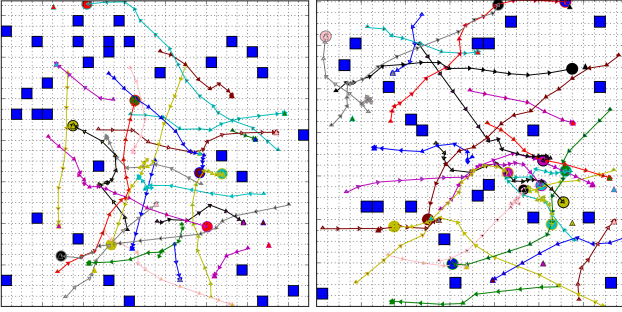


Figure 3: Example scenarios from egocentric representative (G) domain, shown with expert trajectories. Each agent (denoted as a circle) aims to reach its destination (a triangle of the same color) while avoiding other agents and obstacles.

7 EVALUATING SCENARIO GENERALIZATION CAPABILITY

Bidirectional experiments are conducted: models trained on egocentric representative (G) and egocentric random (R) are tested on exocentric standard scenarios (X); models trained on exocentric standard (X) and egocentric random (R) are tested on egocentric representative scenarios (G).

7.1 Trained Models

Given the two training paradigms and three data domains, five training paradigm – training domain combinations are studied:

- (1) BCA-X: BC agents trained on X
- (2) BCA-G: BC agents trained on G
- (3) BCA-R: BC agents trained on R
- (4) RLA-X: RL agents trained on X
- (5) RLA-G: RL agents trained on G

RL agents are not trained on egocentric random scenarios as RL require complete trajectories, not independent state-action pairs.

7.2 State Representation

Similar to [Qiao et al. 2018], we simulate that each agent observes the world around it using a collection of local measurements. The first local measurement is a range map, a measure of radial distances from the center of the agent to the surface of the environment (including surfaces of neighboring agents and surfaces of obstacles), typically at a resolution of one degree over 360 degrees. We also

simulate that an agent can detect the relative movements of neighboring agents and obstacles, perceiving a radial velocity map. In addition, an agent receives local and global guidance velocities. The local guidance velocity is provided by an external source (either GP or A-star), which is capable of sensing obstacles in the environment but lacks knowledge of the existence of other moving agents, thus guiding the agent’s movement independent of other agents, like a GPS. The global signal provides an overall heading direction towards the final destination position, much like a compass.

Following [Long et al. 2017; Qiao et al. 2018], GP provides the local guidance velocity in exocentric standard scenarios, while the sampled preferred velocity acts as the local guidance in egocentric random scenarios. However, in egocentric representative scenarios, the movement of agents does not form a flow pattern. Therefore, we use A-star to plan a route for each agent from its initial to its destination position. Influenced by neighboring moving agents, an agent does not follow strictly with its A-star way points. Instead, at each step it aims at its furthest A-star way point it sees without visual occlusion as the current local goal.

7.3 Main Training Configuration

For the size of the training data, the amount of state-action pairs for training in three domains are nearly the same, about 1.6M.

All BCA-X, BCA-G, BCA-R adopt a six-layer fully connected network, with each layer containing 100 neurons. They are trained by RMSprop [Tieleman and Hinton 2012] with L2 loss and learning rate 0.0001.

For training reinforcement learning (RL) agent, both policy and reward functions adopt the same architecture as BCA-X, BCA-G, BCA-R to ensure that all policies share the same model complexity. The policy learning rate for RL agent is set to 0.01. During sampling model trajectories in the training phase, a zero-mean Gaussian random noise with standard deviation 0.5 is added to the output trading off for exploration. The policy entropy regularizer λ is set to be 0. The network is trained at 10K iterations for exocentric standard scenarios and 6K iterations for egocentric representative scenarios.

7.4 Metrics

The five models are evaluated on three metrics, following [Qiao et al. 2018]. All metrics are the lower, the better.

- (1) **DTW metric:** Dynamic Time Warping distance [Salvador and Chan 2007] measures the spatial deviation of a model trajectory from an expert trajectory averaged over agents. To

eliminate the influence of different number of steps in model trajectories, a min-match version of DTW is adopted, by registering each of the nodes (positions) of a model trajectory to its closest node of the corresponding expert trajectory using dynamic programming, and accumulating the minimal distance of registered pairs of each node along the expert trajectory.

- (2) **AA metric:** AA stands for agent-agent collisions, the total number of collisions for all pairs of agents accumulated over all steps. During one-step movement, a collision between one pair of agents occurs if their distance is less than the sum of their radii at any real-valued time point within that time duration, which could be verified by solving a distance-related quadratic equation.
- (3) **AO metric:** AO denotes agent-obstacle collisions, the total number of collisions between all pairs of an agent and an edge of an obstacle during a simulation, also accumulated over timesteps. An agent-obstacle collision can be detected based on (1) the intersection of two line segments (one for an edge of an obstacle, the other for the trace of an agent's center during a one-step movement) and (2) the distance between a point (the center of an agent) and a line segment (an edge of an obstacle).

Note that within one step if an agent collides with more than one edge of an obstacle, only one AO collision is counted. Two agents keep overlapping or an agent moving within an obstacle is only counted once for the first contacting of their edges until they depart from each other. Also for simplicity, if an agent-agent or agent-obstacle collision occurs, it does not change the velocity of involved agents within that temporal duration.

7.5 Generalization to Test Scenarios

Based on the above experimental setup, bidirectional experiments are conducted to test scenario generalization ability of the training paradigm-training domain combinations on test domains.

7.5.1 Test on Exocentric Standard Scenarios. In this test domain, models are evaluated on the six types of standard scenarios, varying in agent density from 10 to 50 and initial/destination positions. Fig. 4 left shows the averaged rankings for the three metrics.

For DTW, BCA-G, BCA-R, RLA-G ranks first, second and third respectively. This indicates that BCA paradigm is better at inferring a route than RLA when the testing scenarios are widely divergent from the training scenarios. For AA, BCA-G, BCA-R, RLA-G ranks first, second and third respectively. For AO, surprisingly, RLA-G is the best while BCA-G, BCA-R ranks second and third respectively. Therefore, one can see that, under the same training paradigm (BCA), training on egocentric representative scenarios (G) incurs less AA and AO collisions than training on egocentric random scenarios (R), when applied to exocentric standard scenarios (X). This evidence that egocentric representative scenarios (G) provide a suite of challenging local agent-agent interactions and sufficient samples on avoiding collisions in a myriad of obstacle configurations. It also implies that when applying a model to a few challenging unseen environments (e.g., X), it might be better to train a model with a sufficient number of environment configurations (training on egocentric representative scenarios (G) enables the model to learn from

4000 different environments), than applying a model without environment knowledge (BCA-R learns from snapshots of surrounding agents, not from any specific environment).

To understand why RLA-G incurs less AO collisions than BCA-G, we further list Tab. 2 to show detailed comparisons along agent densities. We notice that the DTW metric of RLA-G is much higher than those of BCA-G. From simulation videos and trajectories illustrated in Fig. 1, we observe that only a few RLA-G agents can go through the doorway with slow speed, while other RLA-G agents have to wander near the doorway until the maximum number of simulation steps. The cautious behaviors of RLA-G agents bring benefits in terms of lower AO. This explains the outlier RLA-G in AO metric.

7.5.2 Test on Egocentric Representative Scenarios. In this evaluation, models are tested over 100 scenarios from the egocentric representative scenarios (G) domain. Fig. 4 right shows averaged ranking results over three metrics. For DTW, BCA-X, RLA-X, BCA-R ranks first, second and third respectively. For AA, BCA-R, BCA-X, RLA-X ranks first, second and third respectively. For AO, again, BCA-R, BCA-X, RLA-X ranks first, second and third.

On the one hand, given the same training domain (exocentric standard scenarios (X)), training with BCA paradigm is better than training with RLA paradigm for all metrics of DTW, AA, and AO. On the other hand, the training domain egocentric random scenarios (R) is better than exocentric standard scenarios (X), in term of reducing AA and AO collisions when generalized to many new scenarios. This implies an even more interesting insight: when a model needs to be applied to many new environments (testset of egocentric representative scenarios (G) comprises of 100 new environments), having no knowledge about any environment (BCA-R) is more advantageous than having a little knowledge about a few environments (BCA-X and RLA-X are trained only on six different environments).

7.5.3 Overall Summary on Bidirectional Experiments. According to the above bidirectional results and analysis, it is clear that BCA training paradigm is overall better than RLA training paradigm, and the data domain egocentric representative scenarios (G) and egocentric random scenarios (R) are better than exocentric standard scenarios (X) in reducing AA and AO collisions. Considering the coverage of these paradigms and domains, we conclude that (i) a simpler training paradigm is better than a more sophisticated training paradigm, (ii) training samples with diverse agent-agent and agent-obstacle interactions are beneficial for reducing collisions when the trained models are applied to new scenarios.

7.6 Discussion

Results (Fig. 4) suggest that while RLA-based training methods have a potentially powerful paradigm of aggregate behavior imitation through a combination of IRL and RL, it may not possess the desired cross-domain generalization observed in a simpler BCA paradigm, provided that all models have the same architecture and the same number of parameters. One reason for this may stem from the underlying modeling assumptions.

As evident from the expression of occupancy measure, RLA relies on matching the occupancy measures between the estimated policy

Table 1: Rankings of models over test domains. Each color ring represents a model, and each axis indicates averaged rankings over test scenarios for a metric. For all three metrics, the smaller the better. Numbers in legend denote averaged ranking over three metrics.

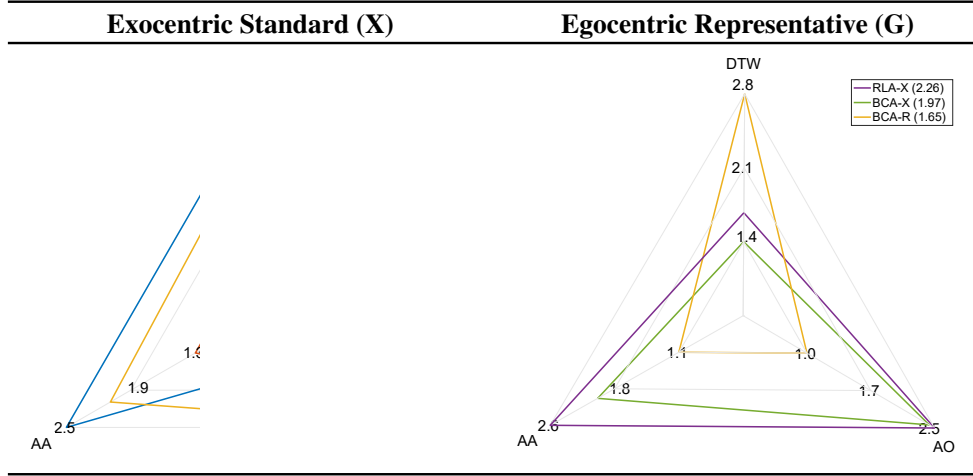
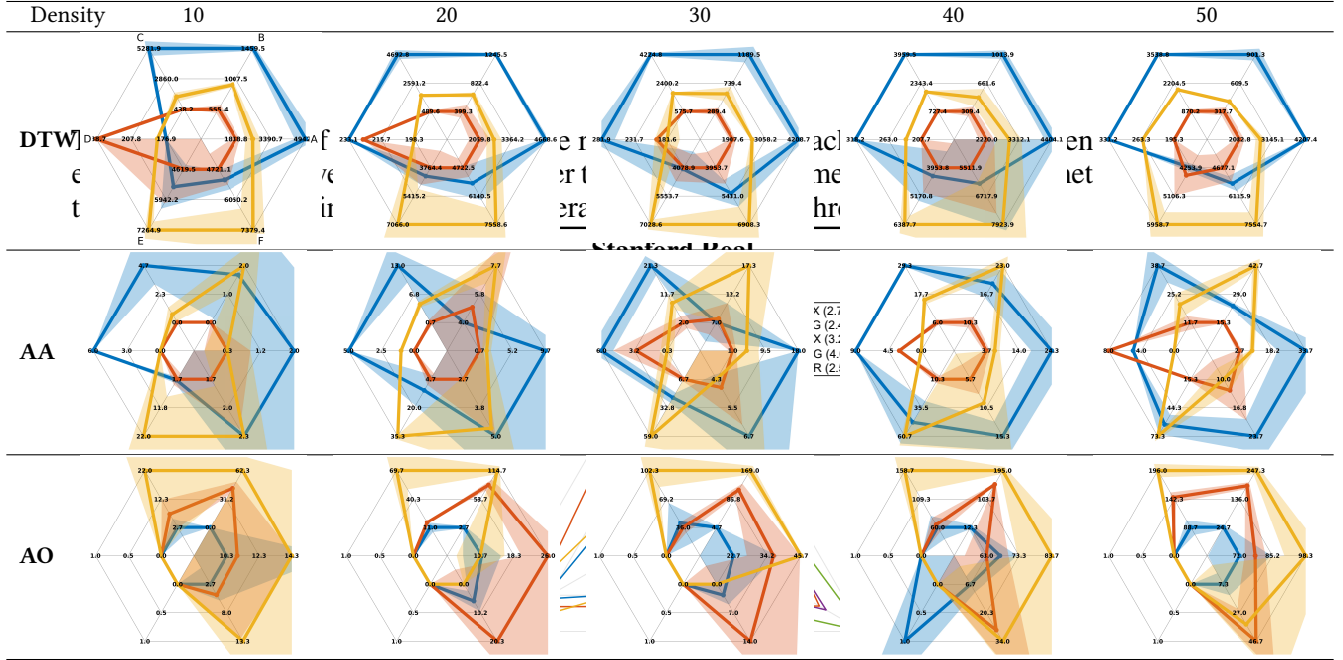


Figure 4: Rankings of models over test domains. Each color represents a model, and each axis indicates averaged rankings over test scenarios for a metric. For all three metrics, the smaller the better. Number in legend denote averaged ranking over three metrics.

Table 2: Comparison of RLA-G (blue), BCA-G (red) and BCA-R (yellow), using the three metrics with different agent densities increasing from left to right. Each axis in a plot denotes one type of the six scenarios: A, B, C, D, E and F denotes evacuation 1, evacuation 2, bottleneck squeeze, concentric circles, hallway two-way, and hallway four-way, respectively. Line thickness in plots indicates each metric’s standard deviation. Polygons closer to the origin imply a better (lower) metric value.



π^* and the expert policy π_E . [Puterman 2014] shows that a valid set of occupancy measures $\mathcal{D} \triangleq \{\rho_\pi | \pi \in \Pi\}$ satisfies a set of affine constraints: $\sum_a \rho(s, a) = p_0(s) + \gamma \sum_{(s', a)} P(s|s', a) \rho(s', a), \forall s \in \mathcal{S}$, where $p_0(s)$ denotes the distribution of initial states. Moreover, there is a one-to-one correspondence between $\mathcal{D}$ and Π : $\pi_\rho(a|s) \triangleq \frac{\rho(s, a)}{\sum_{a'} \rho(s, a')}$, with π_ρ the unique policy whose occupancy measure is

ρ , Thm.2 of [Syed et al. 2008]. Taking this into account, we obtain:

$$\pi_\rho(a|s) = \frac{\rho(s, a)}{p_0(s) + \gamma \sum_{(s', a)} P(s|s', a) \rho(s', a)}, \forall s \in \mathcal{S}.$$

Thus, when modeling the movement of agents in an environment, the dynamics $P(s|s', a)$ encodes complex scenario information, including positions of other moving agents and the obstacles

in the environment, occlusions, etc. These dynamics are, as noted, implicitly encoded in the policy. Therefore, an RLA model trained on a particular training domain implicitly learns its environments. Transferring this model directly to a new, test scenario with significantly different dynamics is bound to result in a weaker match, thus reduced generalization capacity. On the other hand, less biased BCA models will have the ability to surmount those differences more easily, and generalize better.

8 GENERALIZATION TO REAL DOMAIN

In this section, we apply the above five combinations of training paradigms and training domains to a real test domain to visualize their scenario generalization abilities and verify the conclusion in a real world domain.

8.1 Real Domain Description

The real domain we considered is Stanford crowd trajectory dataset, introduced in [Alahi et al. 2016]. It consists of a large set of real pedestrian trajectories collected at a train station of size $25\text{m} \times 100\text{m}$ for 12×2 hours by a set of distributed cameras. Identity numbers, position histories with timestamps of the pedestrians are extracted from the image sequences with detection and tracking algorithms. The dataset is challenging since (1) The agent density is quite high. In a time duration of 4 minutes, there are about 500 pedestrians moving in the train station. (2) Pedestrians are highly asynchronous. They enter into and exit from the train station at different timestamps, without a unified time controller. (3) the data is noisy, due to the detection, tracking and localization error, and the difficulty to measure the accurate positions of the obstacles (infeasible areas).

8.2 Dataset Preprocessing

First, the positions of the obstacles in the environment layout are determined by drawing the provided pedestrians' trajectories on the layout, and manually finding out the obstacle positions based on the occupancy areas of the drawn trajectories. Second, the long-lasting trajectories are aligned with timestamps and further split with a temporal sliding window of 4-minute length and 2-minute stride. Within each time window, all pedestrians are retrieved, including those emerge after the starting time and/or exit before the ending time of the window, and those whose destinations have to be retrieved in the next time window. Third, to reduce the noise in the data, pedestrians whose initial position or destination position are within the obstacles are removed. Last but not the least, a Gaussian convolution operation is applied to the binary representation of the environment layout (obstacle pixels are represented as 1, other feasible pixels are 0), to yield an obstacle-probability map. Based on the map, the cost from a node to its child node in A-star is modified according to the obstacle probability, so as to prevent the planned A-star nodes from being too close to the obstacles, to reduce the risk of agent-obstacle collisions.

8.3 Visualization of Model Trajectories

Fig. 5 illustrates trajectories of the above five combinations of training methods and training domains on the Stanford real dataset. The obstacles (infeasible areas) are in blue color, and the trajectories

are also colored. According to our experiment setting, we know that for agents even slightly entering into an obstacle, they will not perceive the obstacle wherein. However, in this specific test domain, some agents slightly entering into the obstacle may see their far-away planned nodes (e.g., the final destination node), and thus would be guided to directly approach to their final destinations, leading to visually obvious agent-obstacle collisions. We can see that even under such challenging scenarios, with high agent density and easy-to-cause obstacle crossing, BCA-R and BCA-G are still visually generalized better than other combinations. Thus the visualization strengthens our conclusion.

8.4 Quantitative Results

Fig. 6 presents the averaged rankings of all models when generalized to the real domain on the three metrics. We can see that for DTW, BCA-R and RLA-G ranks first and second respectively. For AA, RLA-X and BCA-X ranks first and second respectively. For AO, BCA-R and RLA-G ranks first and second respectively. Overall, RLA-G and BCA-R models are better than others.

From the rankings we have three observations. (1) Training domain egocentric random (R) and training domain egocentric representative (G) are beneficial for reducing AO collisions, which accords with the simulated bidirectional experiments. (2) Training domain exocentric standard (X) is better at reducing AA collisions. This suggests that even though the exocentric standard (X) domain is not suggested by the simulated bidirectional experiments, it contains a few challenging obstacle configurations and can still benefit a model when applied to real challenging scenarios with high agent density. (3) For the training paradigm, both RLA and BCA are involved in the first and second ranked models in each of the three metrics and in the overall ranking. The lack of a dominant training paradigm implies the need to trade off when choosing a training paradigm for generalizing to real challenging domains.

9 CONCLUSION

In this study, our main goal is to analyze the effect of different training paradigms and training domain characteristics on scenario generalization capacities of data-driven imitation models in crowd modeling settings. Our empirical results and analysis indicate that for training method, the simpler behavior cloning method is overall better than the more complex reinforcement learning method. According to our experiment results, it is also noticeable that the training domains have substantial impact on the generalization ability of models to new scenarios. In particular, training samples with diverse agent-agent and agent-obstacle interactions are beneficial for reducing collisions when models are applied to new scenarios.

Future work includes: (1) a comparison to scenario generalization capacities of RL agents whose reward functions are pre-defined, for example, as a combination of the three metrics (DTW, AA, AO); (2) the improvement of scenario generalization capacity. For instance, train a model in a training domain and then adopt it in a testing domain using limited testing samples [Le et al. 2018; Zhao et al. 2018], where a domain is the dynamics belonging to a specific type of scenarios.

Table 2: Rankings of all models over the real test domain. Each color ring represents each axis indicates averaged rankings over test scenarios for a metric. For all three metrics the better. Numbers in legend denote averaged ranking over three metrics.

MIG '19, October 28–30, 2019, Newcastle upon Tyne, United Kingdom

Qiao, et al.

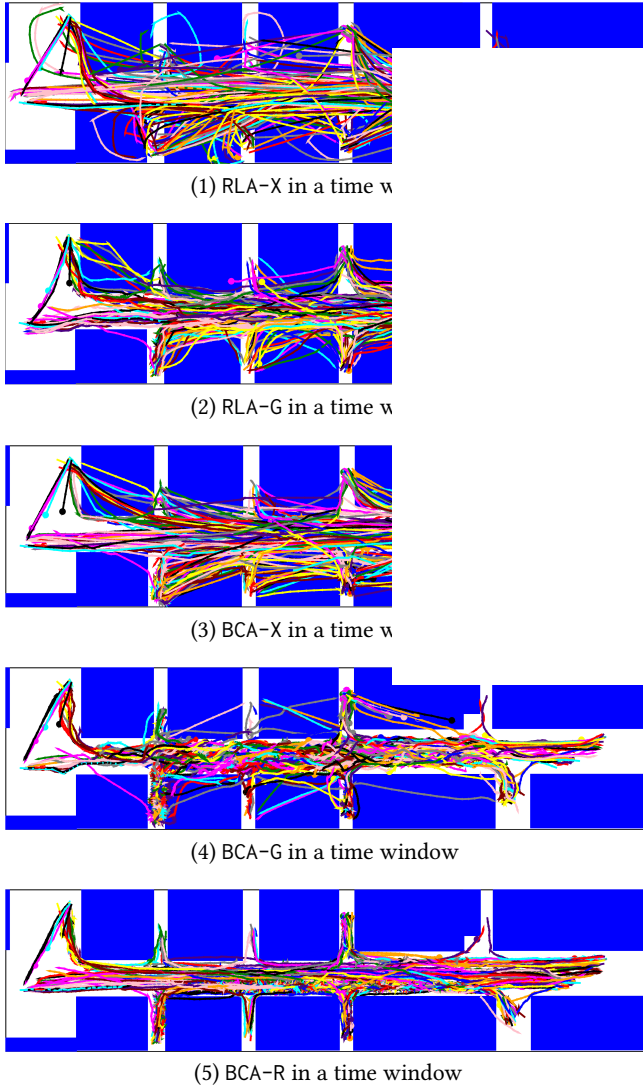


Figure 5: Visualization of different combinations of training methods and training domains generalized to real dataset. The obstacles (infeasible areas) are in blue color.

ACKNOWLEDGMENTS

Kapadia has been funded in part by NSF IIS-1703883, and NSF S&AS-1723869. Yoon has been funded in part by TCNJ SOSA 2017-2019 grant.

REFERENCES

- A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 961–971. <https://doi.org/10.1109/CVPR.2016.110>
- Saad Ali, Ko Nishino, Dinesh Manocha, and Mubarak Shah. 2013. Modeling, simulation and visual analysis of crowds: a multidisciplinary perspective. In *Modeling, simulation and visual analysis of crowds*. Springer, 1–19.
- Saurabh Arora and Prashant Doshi. 2018. A survey of inverse reinforcement learning: Challenges, methods and progress. *arXiv:1806.06877* (2018).
- Luiselena Casadiego. 2014. *Social crowd controllers using reinforcement learning methods*. Master's thesis. Universitat Politècnica de Catalunya. Departament de Llenguatges

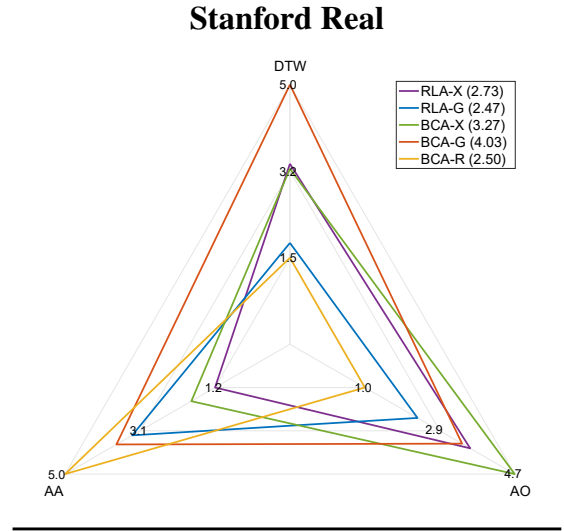


Figure 6: Rankings of all models over real test domain. Each color represents a model. Each axis indicates averaged rankings over test scenarios for a metric. All metrics are the smaller the better. Numbers in legend denote averaged ranking over three metrics.

i Sistemes Informàtics.

- Qingrong Cheng, Zhiping Duan, and Xiaodong Gu. 2018. Data-Driven and Collision-Free Hybrid Crowd Simulation Model for Real Scenario. In *Neural Information Processing*. Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa (Eds.). Springer International Publishing, Cham, 62–73.
- Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. 2016. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852* (2016).
- Paolo Fiorini and Zvi Shiller. 1998. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research* 17, 7 (1998), 760–772.
- Dirk Helbing and Peter Molnar. 1995. Social force model for pedestrian dynamics. *Physical review E* 51, 5 (1995), 4282.
- Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*. 4565–4573.
- Julio Cesar Silva Jacques Junior, Soraia Raupp Musse, and Claudio Rosito Jung. 2010. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine* 27, 5 (2010), 66–77.
- Mubbasir Kapadia, Nuria Pelechano, Jan Allbeck, and Norm Badler. 2015. Virtual crowds: Steps toward behavioral realism. *Synthesis lectures on visual computing: computer graphics, animation, computational photography, and imaging* 7, 4 (2015), 1–270.
- Mubbasir Kapadia, Matt Wang, Shawn Singh, Glenn Reinman, and Petros Faloutsos. 2011. Scenario space: characterizing coverage, quality, and failure of steering algorithms. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 53–62.
- Ioannis Karamouzas, Brian Skinner, and Stephen J Guy. 2014. Universal power law governing pedestrian interactions. *Physical review letters* 113, 23 (2014), 23821.
- Sujeong Kim, Stephen J Guy, and Dinesh Manocha. 2013. Velocity-based modeling of physical interactions in multi-agent simulations. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics symposium on computer animation*. ACM, 125–133.
- Sujeong Kim, Stephen J Guy, Dinesh Manocha, and Ming C Lin. 2012. Interactive simulation of dynamic crowd behaviors using general adaptation syndrome theory. In *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*. ACM, 55–62.
- Paulo Knob, Gabriel Wetzel Rockenbach, Claudio Rosito Jung, and Soraia Raupp Musse. 2019. Optimal Group Distribution based on Thermal and Psycho-Social Aspects. In *Proceedings of the 32nd International Conference on Computer Animation and Social Agents*. ACM, 59–64.
- Alex Kuefler, Jeremy Morton, Tim Wheeler, and Mykel Kochenderfer. 2017. Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 204–211.
- Trung Le, Khanh Nguyen, and Dinh Phung. 2018. Theoretical Perspective of Deep Domain Adaptation. *arXiv preprint arXiv:1811.06199* (2018).

- Jaedong Lee, Jungdam Won, and Jehee Lee. 2018. Crowd Simulation by Deep Reinforcement Learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games (MIG '18)*. ACM, New York, NY, USA, Article 2, 7 pages. <https://doi.org/10.1145/3274247.3274510>
- Weining Liu, Kaidong Hu, Sejong Yoon, Vladimir Pavlovic, Petros Faloutsos, and Mubbasir Kapadia. 2017. Characterizing the Relationship Between Environment Layout and Crowd Movement Using Machine Learning. In *Proceedings of the Tenth International Conference on Motion in Games (MIG '17)*. ACM, New York, NY, USA, Article 2, 6 pages. <https://doi.org/10.1145/3136457.3136474>
- Pinxin Long, Tingxiang Fanl, Xinyi Liao, Wenxi Liu, Hao Zhang, and Jia Pan. 2018. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6252–6259.
- Pinxin Long, Wenxi Liu, and Jia Pan. 2017. Deep-learned collision avoidance policy for distributed multiagent navigation. *IEEE Robotics and Automation Letters* 2, 2 (2017), 656–663.
- Rémi Pautrat, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. 2018. Bayesian optimization with automatic prior selection for data-efficient direct policy search. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 7571–7578.
- Martin L. Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
- Gang Qiao, Sejong Yoon, Mubbasir Kapadia, and Vladimir Pavlovic. 2018. The Role of Data-Driven Priors in Multi-Agent Crowd Trajectory Estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Zhiguo Ren, Panayiotis Charalambous, Julien Bruneau, Qunsheng Peng, and Julien Pettré. 2017. Group Modeling: A Unified Velocity-Based Approach. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 45–56.
- Stéphane Ross and Drew Bagnell. 2010. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 661–668.
- Stan Salvador and Philip Chan. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.
- John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International Conference on Machine Learning*. 1889–1897.
- Shawn Singh, Mubbasir Kapadia, Petros Faloutsos, and Glenn Reinman. 2009. Steer-Bench: a benchmark suite for evaluating steering behaviors. *Computer Animation and Virtual Worlds* 20, 5-6 (2009), 533–548.
- Umar Syed, Michael Bowling, and Robert E Schapire. 2008. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning*. ACM, 1032–1039.
- T Tieleman and G Hinton. 2012. Divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw. *Mach. Learn* (2012).
- Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954* (2018).
- Lisa Torrey. 2010. Crowd simulation via multi-agent reinforcement learning. In *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*.
- Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. 2011. Reciprocal n-body collision avoidance. In *Robotics research*. Springer, 3–19.
- Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. 1995. Novel type of phase transition in a system of self-driven particles. *Physical review letters* 75, 6 (1995), 1226.
- Sejong Yoon, Mubbasir Kapadia, Pritish Sahu, and Vladimir Pavlovic. 2016. Filling in the blanks: reconstructing microscopic crowd motion from multiple disparate noisy sensors. In *2016 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 1–9.
- Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. 2018. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*. 8559–8570.
- Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.