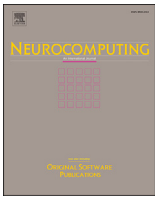
Neurocomputing 698 (2026) 134274

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Mamba-MAGAIL: Mamba-based multi-agent generative adversarial imitation learning for crowd dynamics simulation

Wen Zhou*, Yehui Qiu

School of Computer and Information, Anhui Normal University, Wuhu, 241002, China

ARTICLE INFO

Dataset link: UCY/ETH GC (Original data)

Communicated by Q. Li

Keywords:
Crowd dynamics
Long-range temporal dependencies
Causal buffering mechanisms
Temporal agent encoder
Mamba-SSM modules

ABSTRACT

Modeling realistic crowd dynamics from observational trajectories remains challenging due to the difficulty of capturing long-range temporal dependencies and heterogeneous motion patterns in pedestrian interactions. Existing multi-agent generative adversarial imitation learning methods typically rely on MLP-based backbones, which struggle to model complex scenes involving dense interactions, abrupt motion changes, and disorderly behaviors. We propose Mamba-MAGAIL, a framework that introduces a temporal agent encoder combining Mamba state-space models with causal buffering mechanisms. The encoder decomposes agent states into three components, self-dynamics, interaction context, and external obstacles, each with independent normalization for stable feature learning. A temporal state buffer with causal masking handles variable-length trajectories while preserving temporal causality. Mamba-SSM modules efficiently capture long-range dependencies across sequential positions, velocities, and accelerations, enabling the framework to learn intricate motion patterns characteristic of heterogeneous pedestrians in crowded environments. Experimental validation on multiple benchmark datasets demonstrates competitive performance in trajectory prediction accuracy and realism metrics. The principled approach to decomposed state representation and temporal encoding provides a more effective alternative to traditional MLP architectures for multi-agent generative modeling.

1. Introduction

Crowd trajectory modeling has become an important research topic in artificial intelligence, computer vision, and intelligent systems because of its wide applications in autonomous driving, robot navigation, intelligent surveillance, and virtual human interaction. A reliable crowd simulation framework should not only reproduce the spatial evolution of individual pedestrians, but also capture the dynamic interactions among multiple agents under changing environments. With the rapid development of pedestrian detection and multi-object tracking techniques, large amounts of trajectory data can now be extracted from videos, providing an important foundation for learning-based crowd behavior modeling. However, generating realistic and physically plausible trajectories from limited observations remains a challenging task, especially in crowded, uncertain, and abnormal scenarios.

A key difficulty lies in the fact that pedestrian motion is governed by complex latent rules rather than by position observations alone. Real-world trajectories are influenced by individual intent, neighboring interactions, scene constraints, and temporal evolution. Therefore, similar spatial traces may correspond to different motion tendencies when higher-order dynamic factors such as velocity and acceleration are considered. In practical applications, especially in emergency or disorderly scenarios, such subtle motion variations can strongly affect the realism and safety of trajectory generation. Consequently, effective representation and exploitation of these weak but informative dynamic features are essential for high-quality multi-agent simulation.

Existing methods [1–3] for trajectory prediction and generation can be broadly divided into rule-based methods, optimization-based methods, and data-driven learning methods. Rule-based methods depend heavily on manually defined behavioral assumptions, which often limit their generalization to complex scenes. Optimization-based approaches can model interactions explicitly, but they usually require carefully designed objectives and may suffer from high computational complexity. In contrast, imitation learning provides a more flexible alternative by allowing agents to learn policies directly from expert demonstrations. In particular, generative adversarial imitation learning has shown strong potential for modeling complex behaviors without explicit reward design, making it appealing for multi-agent crowd trajectory generation.

Despite this advantage, many existing MAGAIL-based methods still use simple MLP structures as their feature extraction and policy

* Corresponding author.
Email address: w.zhou@ahnu.edu.cn (W. Zhou).

https://doi.org/10.1016/j.neucom.2026.134274
Received 21 April 2026; Received in revised form 3 June 2026; Accepted 11 June 2026
Available online 16 June 2026

representation backbone. Such architectures are effective for low-complexity mappings, but their capacity to encode long-range temporal dependencies and subtle motion variations remains limited. As a result, they may fail to fully characterize the intrinsic dynamic laws embedded in pedestrian trajectories, especially in scenes with complex interactions or long-term dependencies. This issue motivates the search for more expressive sequence modeling mechanisms that can better support multi-agent crowd dynamics simulation.

Recently, state space sequence models have attracted increasing attention because of their strong ability to model long-context dependencies with favorable computational efficiency. Among them, Mamba has demonstrated promising performance in a variety of sequence modeling tasks. Its structured state space mechanism provides an effective way to encode temporal evolution while maintaining efficient computation. From the perspective of crowd behavior modeling, this property is particularly valuable because pedestrian trajectories are sequential, interaction-dependent, and dynamically evolving. Therefore, incorporating Mamba into multi-agent imitation learning is expected to improve the representation of latent motion knowledge and enhance trajectory generation quality.

Motivated by the above observations, this paper proposes Mamba-MAGAIL, a Mamba-enhanced multi-agent generative adversarial imitation learning framework for crowd dynamics simulation. Different from conventional MAGAIL frameworks, the proposed method replaces the MLP backbone with Mamba-SSM to strengthen trajectory feature extraction and dynamic representation. In addition, to improve the quality of expert demonstrations, interpolation-based numerical processing is used to construct refined trajectory data, and perspective transformation is adopted to map image coordinates to world coordinates. Furthermore, a multi-metric constrained loss is introduced to jointly guide the optimization of benchmark and generated trajectories, enabling the framework to better adapt to different scene conditions and produce more reasonable motion paths.

Compared with existing studies, the contribution of this work does not only lie in pursuing marginal improvements on saturated benchmarks. More importantly, it explores a new knowledge-guided modeling paradigm for multi-agent trajectory generation by integrating state space sequence modeling into adversarial imitation learning. This integration improves the ability of the framework to encode latent motion knowledge from multiple dynamic factors and provides a more expressive basis for simulating realistic crowd behaviors.

The main contributions of this paper are summarized as follows:

1) **Mamba-Based Agent Encoder for State Heterogeneity**. A temporal state encoder combining Mamba-SSM with causal buffering to uniformly process heterogeneous agent states (self-dynamics, interaction, external) and capture long-range temporal dependencies, replacing traditional MLPs in MAGAIL-based imitation learning.
2) **MAGAIL Framework with Normalized Agent Representations**. Integration of the agent encoder into multi-agent GAIL with per-component state normalization and clamping, enabling balanced gradient flow across the discriminator and actor-critic modules during simultaneous training.
3) **Empirical Validation on Crowd Dynamics**. Demonstrations that Mamba-encoded agents achieve faster convergence and stable trajectory generation on standard benchmarks, validating the effectiveness of SSM-based agent modeling over attention and RNN baselines.

The remainder of this paper is organized as follows. Section 2 reviews the related work on crowd trajectory modeling, imitation learning, and state space sequence models. Section 3 presents the proposed Mamba-MAGAIL framework in detail. Section 4 describes the datasets, experimental settings, and evaluation metrics. Section 5 reports and discusses the experimental results. Finally, Section 6 concludes the paper and outlines future research directions.

2. Related works

In recent years, pedestrian and multi-agent trajectory planning has become a prominent research focus, driven by its critical role in autonomous driving, crowd simulation, and intelligent transportation systems. To position our proposed method within the current research landscape, we review the literature from three distinct perspectives: traditional planning methods, reinforcement learning approaches, and imitation learning paired with advanced sequence modeling.

2.1. Traditional trajectory planning and multi-agent navigation

For several decades, researchers have proposed various solutions for guiding agents and planning paths in complex environments. Classical path planning algorithms, such as A* [4], ant colony optimization [5], the bee colony algorithm [6], and particle swarm optimization [7], have been widely adopted due to their straightforward optimization objectives. In multi-agent collaborative tasks, physics-based theories and heuristic rules have also played a significant role. For instance, Van et al. [8] proposed the Optimal Reciprocal Collision Avoidance (ORCA) approach, which effectively mitigates collision issues in crowd simulations by optimizing the reciprocal distances between agents. Furthermore, Hou et al. [9] introduced a group leadership paradigm for multi-agent organization, which aligns with the observed mobility behaviors of structured crowds.

However, these traditional methods inevitably face limitations in their expressive capabilities. They often rely heavily on handcrafted rules, predefined physical constraints, or rigid avoidance strategies, which can lead to high computational demands or unnatural congestion in dense scenarios. More importantly, they struggle to model the highly non-deterministic, complex, and habit-driven behaviors inherent in real-world pedestrian movement, making them difficult to apply to large-scale, dynamic multi-agent environments.

2.2. Reinforcement learning for navigation

With the rapid advancement of artificial intelligence, Reinforcement Learning (RL) has become a vital paradigm for addressing dynamic decision-making tasks. Early RL algorithms, such as Monte Carlo and standard Q-learning [10], were successfully applied to crowd evacuation [12], robotic motion control [11,13], and UAV trajectory optimization [14]. However, as environmental complexity increases, the state and action spaces expand dramatically, causing these tabular methods to suffer from the dimensionality explosion problem.

To overcome this, deep reinforcement learning (DRL) algorithms such as the Deep Q-Network (DQN) [15] and its advanced variants [16–19] were developed to leverage the perceptual power of deep neural networks. Recent applications have explored multi-agent coordination [21–23] and cooperative formation maintenance [20,25] using DRL. Furthermore, Proximal Policy Optimization (PPO) [26] has emerged as a leading algorithm for continuous control tasks due to its training stability and sample efficiency. Furthermore, Gao et al. [35]develops a decentralized iterative learning control framework for collaborative tracking with reduced computation. Peng et al. [36] address resilient consensus control for multi-helicopter slung-load systems, incorporating vibration suppression, but their control lacks adaptive handling of uncertain load dynamics and wind disturbances. Collectively, these works advance trajectory prediction, and cooperative control. Despite these successes, RL-based navigation faces significant hurdles. First, the reward functions in real-world pedestrian scenarios are highly complex, dynamic, and difficult to engineer manually. Second, methods relying on traditional Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) for state processing [27] either provide limited long-term temporal modeling or incur high computational costs,

restricting their ability to capture the long-range dependencies crucial for multi-agent trajectory prediction.

2.3. Imitation learning and sequence modeling in multi-agent environments

Given the difficulties of reward design in RL, Imitation Learning (IL) [28,29] provides a powerful alternative by utilizing expert demonstrations to guide agent behavior. Generative Adversarial Imitation Learning (GAIL) [30] formulates IL as a minimax game between a policy and a discriminator, enabling agents to learn realistic behaviors directly from expert trajectories. To improve training stability, Wasserstein GAN variants with gradient penalty (WGAN-GP) [31] have been successfully integrated into IL frameworks. For multi-agent scenarios, Song et al. [32] innovatively proposed Multi-Agent GAIL (MAGAIL) to learn coordinated behaviors without depending on handcrafted rewards. However, existing MAGAIL frameworks are predominantly constrained by simple Multi-Layer Perceptron (MLP) backbone architectures, leaving considerable room for improvement in handling complex spatio-temporal data.

Besides, Huang et al. [34] propose trajectory Mamba, an efficient selective state-space model for trajectory forecasting, demonstrating improved computational efficiency, but trajectory Mamba lacks real-world validation under complex occlusions and long-term drift analysis. SocialGAIL [37] is put forward for realistic crowd simulation in robot navigation, however, its generalization across cultural contexts and novel crowd behaviors remains unaddressed. Thereby, the focus on computational efficiency and decentralized approaches reflects practical deployment considerations. In addition, incorporating uncertainty quantification, communication-robust designs, physics-informed learning for complex dynamics, and multi-cultural datasets would strengthen practical applicability.

Well-known Transformer models have recently been applied to sequential decision-making due to their strong dependency modeling capabilities. However, their $O(n^2)$ attention complexity becomes prohibitive as the number of agents and the length of trajectories increase. Conversely, RNNs offer linear complexity but inherently struggle with long-term memory retention.

Recently, Mamba state space models (SSMs) [33] have emerged as a highly efficient alternative. Through selective state updates and parallelizable sequence processing, Mamba overcomes the limitations of both RNNs and Transformers, delivering superior performance with linear-time complexity. Its ability to handle long, variable-length sequences makes it exceptionally well-suited for dynamic multi-agent environments.

Recognizing the sparsity and variable temporal lengths typical of pedestrian trajectories, this paper proposes an enhanced framework, **Mamba-MAGAIL**. By integrating the Mamba architecture with MAGAIL and PPO, and incorporating a novel sparsity gate module, our approach effectively captures latent dynamic motion knowledge and long-range dependencies, overcoming the architectural bottlenecks of previous multi-agent simulation models.

3. Details of the proposed framework

This section details the proposed Mamba-MAGAIL framework for multi-agent crowd trajectory simulation. As illustrated in Fig. 1, the overall pipeline consists of three primary stages: (1) expert trajectory data preparation, (2) adversarial imitation learning via the proposed Mamba-MAGAIL network, and (3) downstream multi-agent trajectory generation in virtual environments.

During the data preparation stage, high-quality expert trajectories are extracted from real-world pedestrian datasets, such as ETH/UCY and Grand Central. To seamlessly bridge real-world data with virtual 3D simulation environments, coordinate transformation techniques are applied to map image-plane coordinates into the world coordinate system. Furthermore, multi-object tracking algorithms (e.g., ByteTrack) are utilized to augment the dataset, capturing continuous long-horizon pedestrian movements. Critical dynamic features—including velocity, movement direction, and acceleration—are systematically extracted to provide the network with a comprehensive representation of complex multi-agent behavioral patterns.

In the training phase, the extracted expert data is fed into the Mamba-MAGAIL architecture. Through an adversarial learning mechanism, the model optimizes multiple objective functions to capture the underlying motion knowledge and interaction rules of the crowd. Upon convergence, the trained policy is deployed into virtual simulation scenarios to independently govern multi-agent navigation, generating highly realistic and socially compliant interactive trajectories.

3.1. Problem description

Multi-agent pedestrian navigation is formulated as a Markov Decision Process (MDP), defined by the following core elements:

1. **State Space (S)**: Comprises the agent's internal state (e.g., current velocity), interactive state (e.g., relative position to the destination), and external environmental state (e.g., positions and velocities of neighboring pedestrians and static obstacles);
2. **Action Space (A)**: Formulated as continuous actions representing linear velocity (v) and angular velocity (ω);
3. **Reward Function (R)**: Within our adversarial imitation learning framework, the primary reward signal is dynamically generated by the Discriminator to quantify the behavioral similarity between the generated trajectories and expert demonstrations. This is further augmented with task-specific constraints, integrating factors for goal progress, collision avoidance, energy efficiency, and motion smoothness.

Consequently, the learning objective is to optimize a behavioral policy $\pi(a|s)$ that maximizes the expected cumulative reward $\mathbf{R}_\pi = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\gamma^t r(s_t,a_t)\right]$, where $\tau = (s_0, a_0, s_1, a_1, \ldots)$ denotes a trajectory, $\gamma \in [0,1)$ is the temporal discount factor, and $r(\cdot)$ represents the immediate reward.

3.2. Data preprocessing

In imitation learning, the quality of expert demonstrations directly determines the upper bound of the learned policy. Undeniably, extracting high-fidelity expert trajectories from real-world surveillance videos is a critical prerequisite for our framework.

In practical applications, multiobject tracking methodologies frequently employ mapping projection operations to mitigate limitations in visual correspondence estimation. A notable example is UCMCTrack [38], which substitutes the conventional Intersection over Union (IoU) distance metric with a mapped Mahalanobis distance measure. This approach enhances tracking accuracy by providing a more robust distance calculation that accounts for spatial uncertainty and correlation between tracked objects, thereby improving the overall performance of multiobject tracking systems in complex scenarios.

However, raw trajectory data extracted from video streams often suffers from sensor noise, missing frames, and perspective distortion (being based on image pixel coordinates). To enable rigorous quantitative training and ensure the physical plausibility of the generated behaviors, data ingestion, homographic transformation, and trajectory interpolation with temporal filtering are performed. The specific procedures are outlined as follows:

Data ingestion. Given a raw pedestrian video spanning a spatial domain S and temporal duration T, multi-object tracking algorithms are deployed to extract initial pedestrian trajectories, denoted as $\boldsymbol{\Phi}_N^T$ (where N is the number of pedestrians). A rigorous filtering step is then performed to eliminate anomalous data. Specifically, trajectories associated with tracking errors, non-pedestrian entities, or physically implausible motion patterns (e.g., sudden teleportation or prolonged

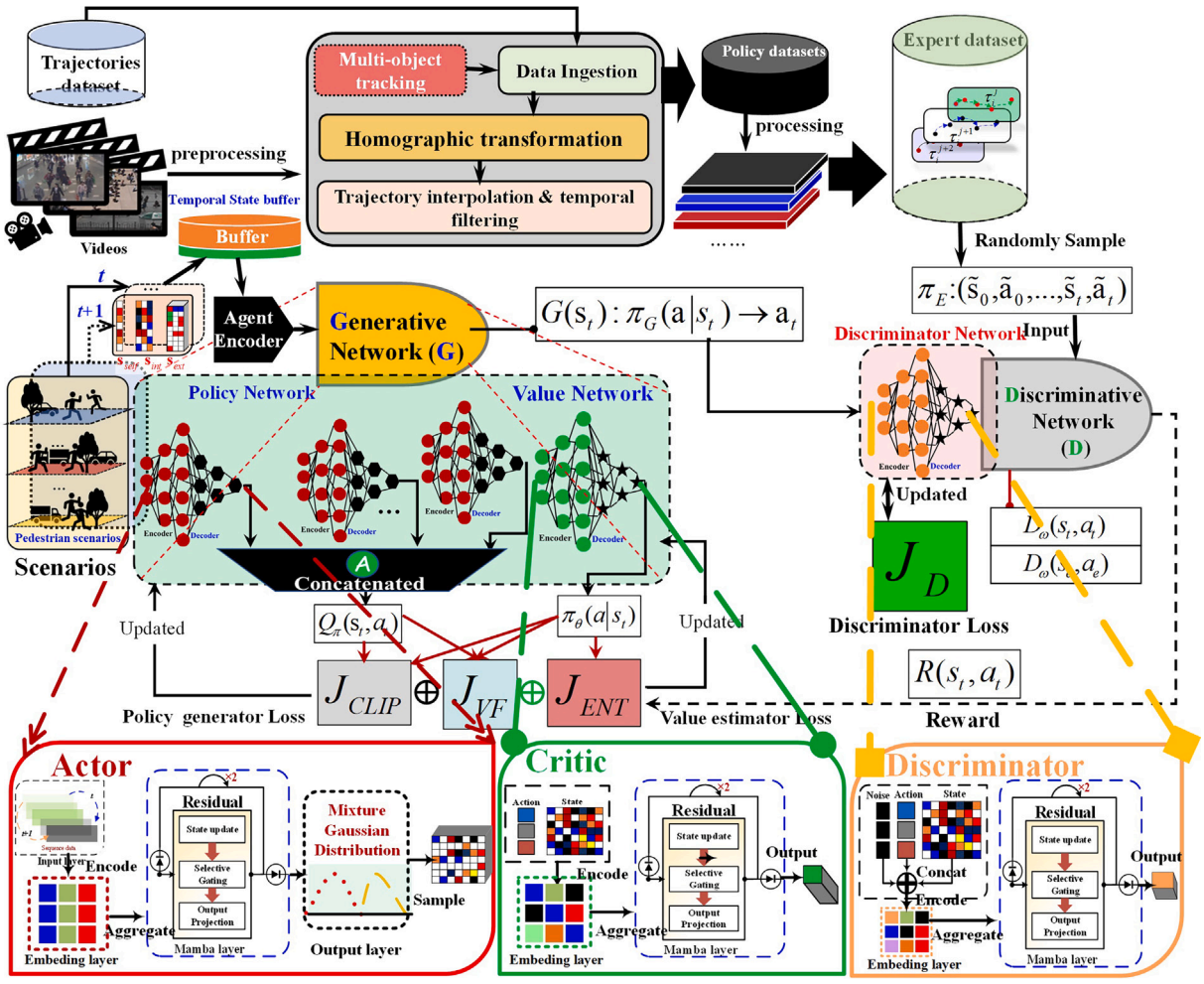


Fig. 1. Overview of the proposed Mamba-MAGAIL framework. The pipeline comprises three main stages: expert dataset preparation, model training via adversarial imitation learning, and deployment for trajectory generation in virtual scenarios. The core Mamba-MAGAIL architecture features three distinct Mamba-based backbone modules (Actor, Critic, and Discriminator) and employs an attention mechanism to fuse multi-agent state features for effective Proximal policy optimization (PPO) training.

unnatural immobility) are systematically excluded to ensure the behavioral authenticity of the expert dataset.

Homographic transformation. The raw trajectories in Φ_N^T are initially expressed in 2D image pixel coordinates, which are subject to camera perspective distortion and lack physical scale. To convert these into metric world coordinates, a perspective transformation matrix $\mathbf{M}$ is applied.

Assuming pedestrians move on a planar surface (i.e., $\mathbf{Z} = 0$), the transformation maps 2D image coordinates(u, v) to homogeneous 3D world coordinates $(\tilde{X}, \tilde{Y}, \tilde{Z})$ as follows Eq. (1):

$$\begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{W} \end{bmatrix} = \mathbf{M} * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the homography matrix derived from known correspondences between image pixels and real-world reference points. The standard metric world coordinates, (X, Y) are subsequently obtained via normalization: $X = \frac{\tilde{X}}{\tilde{W}}$, $Y = \frac{\tilde{Y}}{\tilde{W}}$. This transformation ensures that the imitation learning model processes physically meaningful spatial representations.

Trajectory interpolation and temporal filtering. Due to occlusions or sensor limitations, Φ_N^T may occasionally contain missing frames. To ensure the temporal continuity required by sequence models like Mamba, an adaptive interpolation mechanism is employed. Let $\phi_i \in \Phi_N^T$ denote the observed discrete trajectory points for pedestrian i between their first observed frame t_{min} and their last observed frame t_{max}.

The continuous fitted trajectory $\hat{\phi}_i$ is defined as follows Eq. (2).

$$\hat{\phi}_i = \begin{cases} \Delta(\phi_i^{t_{min}, t_{max}}(x, y)) & |\phi_i| > 3 \\ \Gamma(\phi_i^{t_{min}, t_{max}}(x, y)) & \text{otherwise} \end{cases} \tag{2}$$

where Δ and Γ represent spline interpolation and linear interpolation, respectively, and $|\phi_i|$ denotes the number of available observation points. Finally, to align the interpolated trajectories with the discrete temporal intervals used by the predictive model, a temporal filtering mask ω is applied to $\hat{\phi}_i$ as Eq. (3).

$$\Theta(\hat{\phi}_i^t) = \begin{cases} 1 & t \in [t_i^{min}, t_i^{max}] \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

This alignment ensures that all expert demonstrations are chronologically consistent and compatible with the recurrent step updates of the proposed multi-agent framework.

3.3. Details of proposed framework

The proposed Mamba-MAGAIL architecture forms the core of the MAGAIL framework. To effectively model the highly dynamic

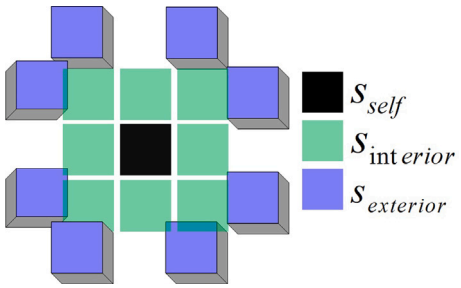


Fig. 2. Illustration of agent state information. The spatial awareness is divided into three distinct zones: the agent's self-state, the interactive state reflecting environmental goals, and the external state focusing on collision avoidance with dynamic neighbors.

interactions in pedestrian navigation, each agent learns a continuous action policy modeled as a Gaussian Mixture Model (GMM), with a shared encoder that captures both spatial neighbor interactions and temporal motion dynamics.

3.3.1. State representation

In a multi-agent environment, an agent must process not only its own kinematics but also the contextual information of its surroundings (illustrated in Fig. 2). For a given agent i, its state observation at timestamp t, denoted as $\mathbf{o}_i^t$, is formulated as the concatenation of three components as $\mathbf{o}_i^t = \{\mathbf{s}_{\text{self}}^t, \mathbf{s}_{\text{int}}^t, \mathbf{s}_{\text{ext}}^t\}$.

Self state $\mathbf{s}_{self}^t \in \mathbb{R}^1$ encodes the agent's scalar self-state (e.g., current velocity), interaction state $\mathbf{s}_{int}^t \in \mathbb{R}^8$ encodes goal-directed and kinematic interaction features (e.g., relative goal distance and heading, relative velocity, etc.), and external state $\mathbf{s}_{ext}^t \in \mathbb{R}^{K\times 8}$ encodes the relative states of up to $\mathcal{K} = 20$ neighboring pedestrians and obstacles in the current frame. Consequently, to efficiently process sequential data, a temporal state buffer of fixed length $\mathcal{T} = 20$ is implemented to accumulate per-step tuples throughout the episode. This fixed-size buffer standardizes sequence length across all training samples. For episodes shorter than $\mathcal{T}$ steps, zero-padding is applied to maintain consistent dimensionality, enabling seamless batch processing and compatibility with the model architecture.

3.3.2. Agent encoder

In conventional MAGAIL frameworks, state features are typically processed using Multi-Layer Perceptrons (MLPs), which struggle to capture complex, long-term sequential dependencies. To overcome this, we introduce an agent encoder to extract dynamic motion cues from sequential pedestrian data with linear computational complexity. Specifically, the encoder processes the temporal state buffer and produces a unified feature vector via two parallel branches (i.e., Branch A and Branch B).

Ego temporal encoding. To efficiently process these temporal inputs, we employ a dual-stream Mamba block structure (shown in Fig. 3) as the temporal encoder $\mathbb{H}^A$. In particular, the concatenation of $\mathbf{s}_{self}$ and $\mathbf{s}_{int}$ across $\mathcal{T}$ timesteps forms a sequence $\mathbf{X}^A \in \mathbb{R}^{N\times\mathcal{T}\times 9}$, which is passed through a Mamba-based temporal encoder $\mathbb{H}^A$. The output feature is $\mathbf{h}^A = \mathbb{H}^A(\mathbf{X}^A) \in \mathbb{R}^{N\times 32}$.

Spatial-temporal encoding. The proposed module implements a sophisticated two-stage feature processing pipeline that systematically integrates spatial and temporal information to generate comprehensive

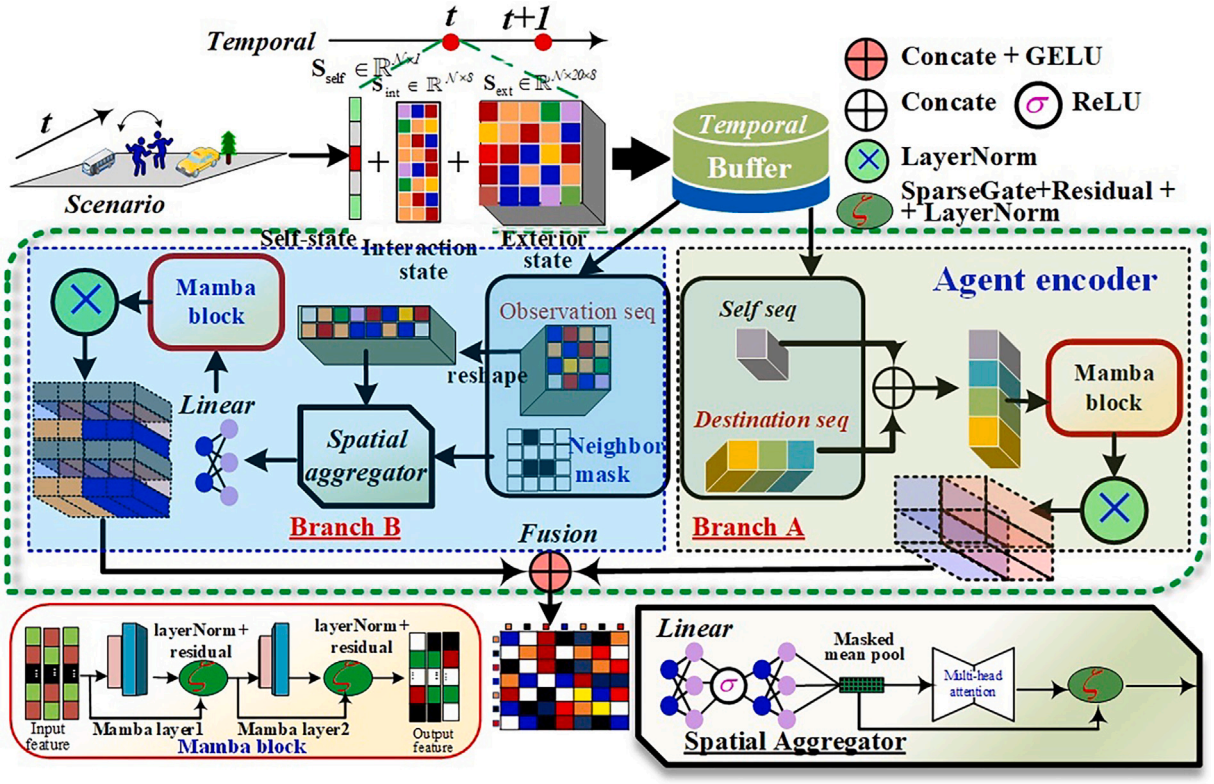


Fig. 3. Overview of the proposed AgentEncoder paradigm architecture. The temporal state buffer at the top is sequentially populated with state information extracted from expert trajectory datasets, encompassing three fundamental components: self-state (representing individual agent properties), interactive state (capturing inter-agent relationships), and exterior state (reflecting environmental context). Subsequently, appropriately sampled rollout sequences are processed through the AgentEncoder module, which comprises two principal branches, i.e., Branch A and Branch B. Additionally, a spatial aggregator is used for feature extraction and a Mamba block for sequential state encoding. This modular design enables efficient feature representation learning while maintaining explicit separation between spatial feature aggregation and temporal sequence modeling mechanisms.

spatial-temporal representations. Initially, the module undergoes a reshaping operation on the exterior state feature, which is subsequently transmitted to the spatial aggregator in conjunction with a neighbor mask that delineates relevant spatial relationships within the feature space.

The spatial aggregator itself is architected as a series of linear transformation layers interspersed with Sigma activation functions, which facilitate non-linear feature mapping while maintaining computational efficiency. Following this foundational processing, a multi-head attention mechanism is integrated to enable the selective fusion of correlated spatial features by allowing the model to dynamically weight and combine information from multiple spatial dimensions simultaneously. The computational flow continues with the sequential application of a sparse gate module and residual connection pathways, both of which contribute to improved gradient propagation and effective feature preservation throughout the network depth. A LayerNorm normalization layer then standardizes the spatial features $\mathbf{X}^B \in \mathbb{R}^{N\times\mathcal{T}\times 32}$, ensuring stability and consistency in the feature distributions before downstream processing.

Subsequently, the module transitions to temporal feature extraction by employing a Mamba temporal module, which is specifically designed to capture complex temporal dependencies and sequential patterns inherent in the exterior state data. This temporal processing effectively distills meaningful temporal relationships that complement the previously extracted spatial information. The architecture concludes with a final LayerNorm operation that normalizes and consolidates the integrated spatial-temporal features, producing a unified representation $\mathbf{h}^B = \mathbb{H}^B(\mathbf{X}^B)$ that encapsulates both spatial structure and temporal dynamics for downstream tasks or applications requiring comprehensive multi-dimensional feature understanding.

Feature fusion mechanism. The abovementioned two branches are concatenated and projected into the feature $\mathbf{z} \in \mathbb{R}^{N\times 160}$ as described in Eq. (4).

$$\mathbf{z} = \breve{\sigma}\left(\mathbf{MLP}\left(\mathbf{h}^A \oplus \mathbf{h}^B\right)\right) \tag{4}$$

where $\breve{\sigma}$ is the layer normalization function, and **MLP** represents the multi-layer perceptron.

3.3.3. Spatial aggregator with multi-head attention

For tasks requiring explicit neighbor weighting (e.g. social force computation), a multi-head attention (**MHA**) module aggregates the spatial neighbor features $\mathbf{s} \in \mathbb{R}^{N\times\mathcal{K}\times d}$ with a validity mask $\mathbf{m} \in \{0,1\}^{N\times\mathcal{K}}$. The output feature $\mathbf{O}$ can be denoted as Eq. (5).

$$\mathbf{O} = \breve{\sigma}\left(\mathbf{MHA}\left[\mathring{\sigma}(\mathbf{s}\odot\mathbf{m}), \mathbf{s}, \mathbf{s}, \sim\mathbf{m}\right] + \mathring{\sigma}\left(\mathbf{s}\odot\mathbf{m}\right)\right) \tag{5}$$

Where, the term $\mathring{\sigma}$ is the Mean Pooling operator, and $\odot$ is the masked multiply operator. Besides, for the **MHA** module, the number l of heads is empirically set, i.e., $l = 4$. The complete procedure encompassing the state encoder and observation representation is illustrated in Fig. 3.

3.4. Actor-critic networks

The complete adversarial architecture relies on three distinct networks: the Actor, the Critic, and the Discriminator, all of which leverage the extracted Mamba features (illustrated in Fig. 4).

Actor network. Actor network concatenates the outputs of the dual-stream Mamba blocks, resulting in a 160-dimensional fused feature vector (32 + 128). A final Mamba block maps these fused features to the parameters of a Gaussian Mixture Model (GMM) with $\mathbf{K} = 3$ components. Given the encoded state $\phi(\mathbf{s}) \in \mathbb{R}^{160}$, the actor network produces the feature $\mathbf{h}_\pi = \mathbb{H}(\phi(\mathbf{s}))$. In this manner, the output is partitioned as $\mathbf{h}_\pi = \Big[\underbrace{p}_{K}, \underbrace{\mu}_{2K}, \underbrace{\sigma_x}_{K}, \underbrace{\sigma_y}_{K}, \underbrace{\rho}_{K}\Big]$. Thereby the mixture distribution is shown as Eq. (6).

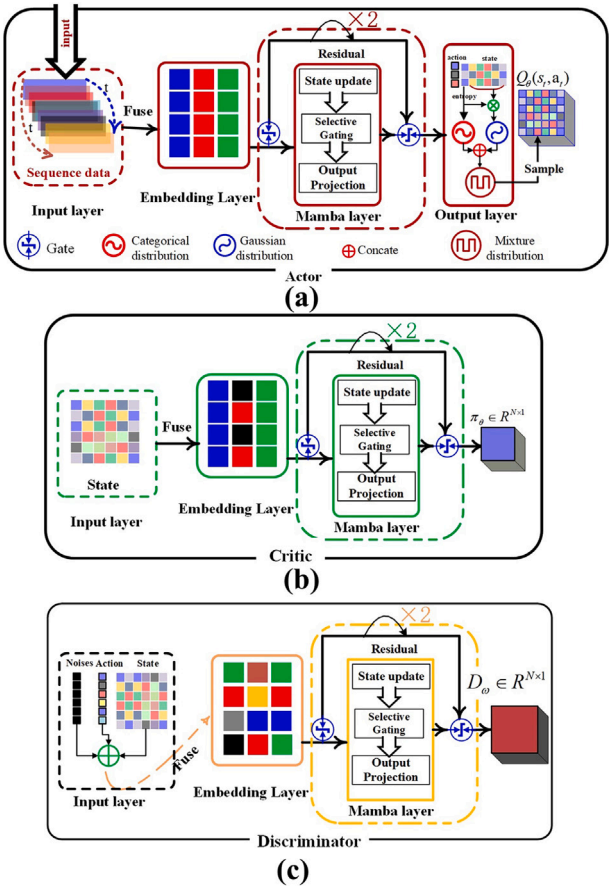


Fig. 4. Overview of the proposed Mamba-MAGAIL network architecture, highlighting its three principal components: (a) the Actor, (b) the Critic, and (c) the Discriminator. The feature extraction backbone incorporates dual-stream Mamba blocks, a selective gating logic module, and a linear output layer equipped with a sparsity gate.

$$\pi(a|s) = \sum_{k=1}^{\mathbf{K}} p_k \mathbb{N}(a; \mu_k(s), \Sigma_k(s)) \tag{6}$$

Here, p_k denotes the mixing probabilities, while μ_k and Σ_k represent the mean and covariance matrices, respectively, and $\mathbb{N}$ refers to the normal distribution.

Critic network. In parallel, the Critic network employs an identical Mamba-based feature extraction pipeline to map the state observations to a scalar state-value estimate $V(s)$, which is used to compute advantages during PPO updates. Finally, the Discriminator evaluates the realism of the generated state-action pairs against the expert data, providing the adversarial reward signal that guides the Actor.

Discriminator network. The discriminator D_ω is formulated in accordance with the Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) framework. It operates on the concatenation of the raw policy feature vector $f_{policy} \in \mathbb{R}^{169}$ and the action $a \in \mathbb{R}^2$, yielding the expression $D_\omega(f_{policy}, a) = \mathbb{H}(f_{policy} \oplus a) + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, 0.1)$ denotes an instance noise term introduced to enhance training stability, as further illustrated in Fig. 3(c).

3.5. PPO–MAGAIL joint optimization

In scenarios guided by expert demonstrations, the proposed framework integrates Multi-Agent Generative Adversarial Imitation Learning

(MAGAIL) with a Mamba-enhanced Proximal Policy Optimization (PPO) mechanism. This combination allows the agents to learn robust sequence-dependent behaviors without relying on manually engineered reward functions. The details of the Actor-Critic module and the Discriminator network are presented below.

3.5.1. Objective functions of the proposed module

To incorporate Mamba's superior sequence modeling capabilities into the PPO algorithm, we adapt the Actor-Critic (AC) update mechanism. To reduce the variance in advantage estimation, Generalized Advantage Estimation (GAE) is employed. The advantage $\hat{\mathbb{A}}_t$ is defined as $\hat{\mathbb{A}}_t = \sum_{j=0}^{\infty} (\gamma \times \lambda)^j \delta_{t+j}$, where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ represents the temporal difference (TD) error, and $\lambda \in [0, 1)$ is the GAE smoothing parameter.

The policy is optimized using a clipped surrogate objective, $\mathcal{J}_{CLIP}$ which prevents destructively large policy updates as shown in Eq. (7):

$$\mathcal{J}_{CLIP} = \min \left[\rho_t \hat{\mathbb{A}}_t, clip\Big(\rho_t, 1-\epsilon, 1+\epsilon\Big)\hat{\mathbb{A}}_t \right] \tag{7}$$

Where, $\rho_t = \frac{\pi(a_t|s_t)}{\pi_{old}(a_t|s_t)}$ is the probability ratio between the current and previous policies, and ϵ controls the clipping bounds.

Simultaneously, the Critic network is trained by minimizing the Mean Squared Error (MSE) between the predicted value and the discounted return $\hat{R}_t$, formulated as $\mathcal{J}_{VF} = \mathbb{E}_t\left[(V_\phi(s_t) - \hat{R}_t)^2\right]$. To encourage sufficient exploration and avoid premature convergence, an entropy bonus $\mathcal{J}_{ENT} = \mathbb{H}(\pi_\theta(\cdot|s_t))$ is introduced. The overall objective of the Mamba-MAGAIL module to be maximized is given by Eq. (8).

$$\mathcal{J} = \mathcal{J}_{CLIP} - \lambda_{vf}\mathcal{J}_{VF} + \lambda_{ent}\mathcal{J}_{ENT} \tag{8}$$

Here, λ_{vf} and λ_{ent} are weighting coefficients for the value and entropy terms, respectively. In this work, we empirically set $\lambda_{vf} = 0.5$. The encoder is updated end-to-end jointly with the actor and critic in each PPO epoch. The entropy coefficient λ_{ENT} is adaptively adjusted based on the approximate KL divergence. In this regard, the term λ_{ENT} is increased when $\mathbf{KL} < \delta/2$ and decreased when $\mathbf{KL} > 2\delta$, with $\delta = 0.01$.

3.5.2. Discriminator loss

As illustrated in Fig. 4(c), the Discriminator network utilizes a Mamba-based architecture, denoted as $\mathbb{D}_\psi$,to differentiate between expert trajectories (τ_E) and those generated by the Actor policy (τ_π). The discriminator evaluates state-action pairs using the formulation $\mathbb{D}_\psi = \sigma(\mathbb{O}(\oplus(s, a)))$, where, $\oplus$ denotes concatenation, $\mathbb{O}$ represents the output of the Mamba discriminator blocks, and σ is a sparsity gate designed to filter out invalid predictions and enhance computational efficiency.

To guarantee stable adversarial training and prevent mode collapse, the Wasserstein GAN with Gradient Penalty (WGAN-GP) framework is adopted. The optimization objective for the Discriminator is formulated as Eq. (9):

$$\mathcal{J}_D = \mathbb{E}_{\tau^\pi}\left[\mathbb{D}(s,a)\right] - \mathbb{E}_{\tau^E}\left[\mathbb{D}(s,a)\right] + \lambda_{GP}\mathbb{E}_{\hat{\tau}}\left[\left(\left\|\nabla_{\hat{\tau}}\mathbb{D}(\hat{\tau})\right\|_2 - 1\right)^2\right] \tag{9}$$

where $\hat{\tau}$ denotes interpolated state-action pairs linearly sampled between the expert data and the generated data, and λ_{GP} regulates the strength of the gradient penalty. Concurrently, the Actor policy acts as the generator, optimized to maximize the discriminator's output. The discriminator's output is therefore integrated into the PPO environment as an augmented reward signal: $\hat{r}_t = r_t + \eta \log(\mathbb{D}_\psi(s_t, a_t))$. This forces the agent's generated trajectories to closely align with expert behaviors. The overarching Mamba-MAGAIL training procedure is detailed in Algorithm 1.

3.5.3. State representation heterogeneity and normalization

In our state-space model, the state representation is formed by concatenating three heterogeneous components that can exhibit substantially different numerical ranges, and the absence of explicit normalization may therefore impede gradient-based optimization in architectures such as Mamba. To address this concern, we apply per-component Z-score normalization during a 50-episode warm-up phase, using the transformation $\hat{\mathbf{s}}_i = \left\{ clamp\left(\frac{s_i - \mu_i}{\sigma_i + \epsilon}, -5, 5\right)\right\}$ with temporal window $\mathcal{T} = 20$. Here, μ_i and σ_i denote the empirically estimated mean and standard deviation for each state component, $\epsilon = 10^{-8}$ ensures numerical stability, and the clamp bounds $[-5, 5]$ limit the influence of statistical outliers (e.g., rare collision events). After normalization is introduced, we conduct a related sensitivity analysis to confirm that concatenating the normalized components remains informative and improves or stabilizes downstream learning.

Algorithm 1: Mamba-MAGAIL training algorithm.

Input: Expert dataset $\mathbb{E}_{data} = \{\tau_E\}$,initial policy π_θ,discriminator $\mathbb{D}_\psi$,hyperparameters (λ_{GP}, η)
Output: optimized policy π_θ and discriminator $\mathbb{D}_\psi$

```
Initialize Mamba-based discriminator D_ψ;
for each episode do
    Collecdt agent tranjectories τ_π using current policy π_θ;
    /* Discriminator update phase                    */
    for each epoch e do
        Sample expert batch S_E, a_E from D and agent
          batch(s_π, a_π) from τ_π;
        Compute D_ψ(s_{π,a_π}) and D_ψ(s_E, a_E);
        Compute gradient penalty using interpolated ŝ, â;
        Update ψ to minimize L_D;
        r̃_t = r_t + η log(D_ψ(s_t, a_t)) ;
        Update discriminator learning rate scheduler;
        if convergence criteria met then
            break;
```

4. Experiments

The primary objective of this section is to rigorously evaluate the performance of the proposed Mamba-MAGAIL framework for multi-agent crowd simulation and trajectory generation. We aim to empirically substantiate that integrating the Mamba State Space Model (SSM) provides a distinct advantage in capturing complex, long-range temporal dependencies within dense crowd interactions.

As discussed in Section 2, conventional crowd simulation methods predominantly rely on either rigid rule-based systems (e.g., social forces) or standard Recurrent Neural Networks (RNNs). While these baselines generate plausible local behaviors, they frequently fail to model the intricate, non-deterministic interactions that evolve over extended time horizons. Rule-based models struggle with generalizability in novel, dynamic environments, whereas RNN-based approaches suffer from vanishing gradients and high computational costs when applied to long sequences with numerous interacting agents.

By contrast, the proposed Mamba-MAGAIL framework addresses these bottlenecks. The selective state-space mechanism of the Mamba blocks efficiently manages long trajectory sequences with linear computational complexity, allowing the model to capture subtle spatio-temporal interactions over extended periods. Coupled with the adversarial imitation learning (MAGAIL) objective, the framework learns directly from real-world expert demonstrations, ensuring the generated multi-agent trajectories are both physically plausible and socially compliant.

Expert trajectory data is sourced from three widely recognized pedestrian datasets—ETH/UCY, the Stanford Drone Dataset (SDD), and Grand Central Station (GC)—to ensure diverse and realistic training scenarios. To enhance environmental realism, non-RL agents serving as dynamic obstacles are governed by the Social Force Model (SFM), which evaluates

pairwise interactions, while periodic random resets at constant time intervals are applied to improve robustness. Denoting the number of pedestrians as $\mathcal{N}$ and the time horizon as $\mathcal{T}$, environment stepping incurs a worst-case complexity of $\mathcal{O}(\mathcal{N}^2)$ per time step and $\mathcal{O}(\mathcal{T}\cdot\mathcal{N}^2)$ per episode; however, spatial partitioning reduces this to $\mathcal{O}(\mathcal{T}\cdot\mathcal{N}\log\mathcal{N})$ in practice. Each agent i observes a self-state $\mathbf{s}^i_{\text{self}}\in\mathbb{R}^1$, intra-group interactions $\mathbf{s}^i_{\text{int}}\in\mathbb{R}^8$, and external context $\mathbf{s}^i_{\text{ext}}\in\mathbb{R}^{20\times8}$, yielding a fixed observation size of 169. The Mamba architecture processes this input with $\mathcal{O}(L\cdot d)$ complexity per agent per step, remaining linear in sequence length and thereby avoiding the prohibitive $\mathcal{O}(\mathcal{T}^2)$ cost associated with standard attention mechanisms.

4.1. Settings and metrics

The hardware and software configurations used to execute the experimental framework are detailed in Table 1. Furthermore, the primary hyperparameters for training the Mamba-MAGAIL framework are summarized in Table 2.

To evaluate the performance of our proposed approach against SocialGAIL [37], Final Displacement Error (FDE), Fréchet Distance, and Hausdorff Distance were employed. Specifically, FDE measures the Euclidean distance between predicted and ground-truth endpoints, providing a direct assessment of prediction accuracy at the trajectory terminus. Fréchet Distance captures the similarity between trajectory curves by computing the minimum distance between corresponding points along the paths. Hausdorff Distance evaluates the maximum deviation between trajectories, thereby identifying the worst-case prediction errors. These metrics collectively offer a comprehensive evaluation framework for assessing trajectory prediction quality across different spatial scales and temporal characteristics.

Furthermore, to ensure thorough assessment of the proposed framework's effectiveness, we employ a comprehensive suite of metrics that capture performance across multiple scales. These metrics encompass microscopic measures that evaluate individual-level behaviors, such as decision-making patterns and agent interactions, alongside macroscopic measures that assess crowd-level dynamics, including aggregate flow patterns and collective phenomena. This dual-scale approach enables systematic validation of both the framework's granular behavioral accuracy and its capacity to replicate realistic large-scale social dynamics, thereby providing robust evidence of overall system performance.

Table 1
Details of experimental settings.

Name	Value
Operating System	Ubuntu 24.04
CPU	Intel Xeon Gold 5218R
GPU	NVIDIA GeForce RTX 4096
RAM	DDR4 128 GB
Physics Engine	Unity3D
Programming Languages	Python & C#

Table 2
Primary experimental parameters during training stage.

HyperParameters	Value
Expert's data time horizon $\mathcal{T}$	60
Number of pedestrains $\mathcal{N}$	20
Maximum Episode Per-Step	20
Number of Obstacles	10
Batch size $\mathcal{B}$	20
Maximum Episodes $\mathbf{T}_{ep}$	10,000
Generator's epochs	128
Discriminator's epochs	10
Discounted rate γ	0.95
Learning rate of Actor	1e-4
Learning rate of Critic	1e-4
Learning rate of Discriminator	5e-5
Environmental randomly updated interval	1000
Number of Mamba layers $\mathcal{L}$	2

(1) **Microscopic Metrics:**

Count of collisions (Q_{col}). Count of Collisions Q_{col} quantifies the number of instances in which the inter-agent distance falls below a predefined minimum safety threshold d as Eq. (10). In this study, the threshold $\mathbf{d}$ (i.e., the radius of social force model ω_{sfm}) is set to 0.1 m based on empirical observations conducted, meaning that any two agents approaching within this proximity are registered as a collision event (it is detailedly explained in sensitivity analysis section).

$$Q_{col}(\mathbf{d})=\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{I}\Big(\exists j:\Big\|p_i(t)-p_j(t)\Big\|\le\mathbf{d}\Big) \tag{10}$$

The indicator function $\mathbb{I}$ is employed to detect collision events, where $p_i(t)$ represents the spatial coordinates of agent i at time t.

Displacement ($\bar{D}$). It refers to the average stepwise movement distance per agent, quantifying the mean spatial shift an agent undergoes between consecutive steps throughout the simulation, as defined in Eq. (11).

$$\bar{D}=\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T_i-1}\sum_{t=1}^{T_i-1}\Big\|p_i(t+1)-p_i(t)\Big\|\right) \tag{11}$$

Where, T_i represents the total visible time stems for pedestrian i.

Speed deviation (Speed $\bar{v}$). It is a metric that quantifies the average perpendicular distance between a pedestrian's actual trajectory and the ideal straight-line path connecting their origin to their destination, as formalized in Eq. (12).

$$\bar{v}=\frac{1}{T_i}\sum_{t=1}^{T}\frac{\Big|Ax_i(t)+By_i(t)+C\Big|}{\sqrt{A^2+B^2}} \tag{12}$$

Where $Ax+By+C=0$ is the standard line equation derived from the start and goal coordinates.

Speed change rate. (Velocity $\Delta\dot{v}$) It is defined as the absolutive speed squared norm change of agent i between consecutive timesteps, formally expressed as $\Delta\hat{v}_i=\|v_i(t+1)\|-\|v_i(t)\|$, as presented in Eq. (13). A lower magnitude of $\Delta\dot{v}$ indicates smoother locomotion with minimal abrupt accelerations or decelerations, which is generally desirable for energy efficiency and trajectory stability.

$$\Delta\dot{v}=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i-1}\left(\sum_{t=1}^{T_i-1}\Big|\Delta\hat{v}_i\Big|\right) \tag{13}$$

Direction change rate (Angle $\Delta\theta$). It quantifies the average angular deviation of an agent's trajectory, normalized within the range $[-\pi,\pi]$, serving as a key indicator of collision avoidance capability. A higher value of this metric reflects a greater capacity for dynamic obstacle avoidance, as it signifies more pronounced directional adjustments during navigation.The overall Direction Change Rate $\Delta\theta$ is then computed as the normalized average across all agents and timesteps, as formulated in Eq. (14).

$$\Delta\theta=\frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{T_i-1}\sum_{t=1}^{T_i-1}\Big|\Delta\theta_i \bmod 2\pi\Big|\right] \tag{14}$$

Here, the operator mod denotes the modulo operator, a mathematical function that returns the remainder of a division operation, thereby constraining the resulting value to a defined range.

Energy consumption (Energy $\bar{E}_c$). It is a physically grounded metric derived from fundamental mechanical principles. By assuming a standardized agent mass of $m = 60\,\text{kg}$, the energy consumption can be formally defined and quantified through a consistent and reproducible framework. The precise mathematical formulation governing $\bar{E}_c$ is presented in Eq. (15), where the underlying physical relationships are explicitly established.

$$\bar{E}_c = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T_i}\int_{t=1}^{T_i} m\left(e_s + e_{\boldsymbol{w}}\left\|v_i(t)\right\|^2\right)dt\right) \tag{15}$$

In this formulation, the parameters e_s and $e_{\boldsymbol{w}}$ represent constant coefficients that govern the behavior of the model. For the purposes of this study, these coefficients are assigned fixed values of $e_s = 2.23$ and $e_{\boldsymbol{w}} = 1.26$, respectively. These values were carefully selected based on empirical calibration and are held constant throughout all analyses presented herein, ensuring consistency and reproducibility of the results.

Steering energy (Steer $\bar{E}_s$). It captures the cumulative effect of acceleration and deceleration events, providing a comprehensive measure of how dynamic speed changes influence overall energy consumption. By incorporating this parameter into the evaluation framework, it becomes possible to assess driving efficiency more accurately and identify opportunities for optimizing energy use through smoother speed regulation, as formally expressed in Eq. (16).

$$\bar{E}_s = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{T_i-1}\int_{t=1}^{T_i-1} m\left(e_s + e_{\boldsymbol{w}}\left\|v_i(t)\right\|^2\right)dt\right] \tag{16}$$

(2) **Macroscopic Metrics:**

Average speed difference (Δv). It is a macroscopic metric used to evaluate the fidelity of pedestrian simulations by comparing the mean pedestrian speed observed in real-world environments against that produced by simulated models as shown Eq. (17). This metric provides a high-level assessment of how accurately a simulation captures overall pedestrian movement dynamics, without focusing on individual trajectories. A smaller Δv value indicates a higher degree of alignment between the simulated and observed pedestrian speeds, reflecting a more realistic and reliable simulation performance.

$$\Delta v = \frac{1}{T}\sum_{t=1}^{T}\left\|\overline{v}_{real}(t) - \overline{v}_{imit}(t)\right\| \tag{17}$$

The variables $\overline{v}_{real}(t)$ and $\overline{v}_{imit}(t)$ denote the average speed of visible pedestrians at time t in the real dataset and imitation scenario, respectively. This metric is derived by aggregating the instantaneous velocities of all pedestrians present within the observable field at each given moment, thus offering a collective measure of pedestrian flow dynamics.

*Structural similarity (**SSIM**).* SSIM is applied to compare pedestrian distribution heatmaps, providing a more perceptually meaningful measure of similarity than pixel-wise error metrics as formulated in Eq. (18).

$$\mathbf{SSIM}(\mathbb{I}_1, \mathbb{I}_2) = \frac{(2\mu_1\mu_2 + C_1)(2\sigma_{12} + C_2)}{(\mu_1^2 + \mu_2^2 + C_1)(\sigma_1^2 + \sigma_2^2 + C_2)} \tag{18}$$

In this formulation, $\mathbb{I}_1$ and $\mathbb{I}_2$ represent the heatmaps corresponding to the real and simulated scenarios, respectively. The statistical parameters μ_1 and μ_2 denote their respective mean values, while σ_1 and σ_2 capture their variances, and σ_{12} quantifies the covariance between the two heatmaps. To ensure numerical stability, the terms $C_1 = 1e^{-5}$ and $C_2 = 1e^{-5}$ are introduced, preventing unstable outputs in the similarity metric.

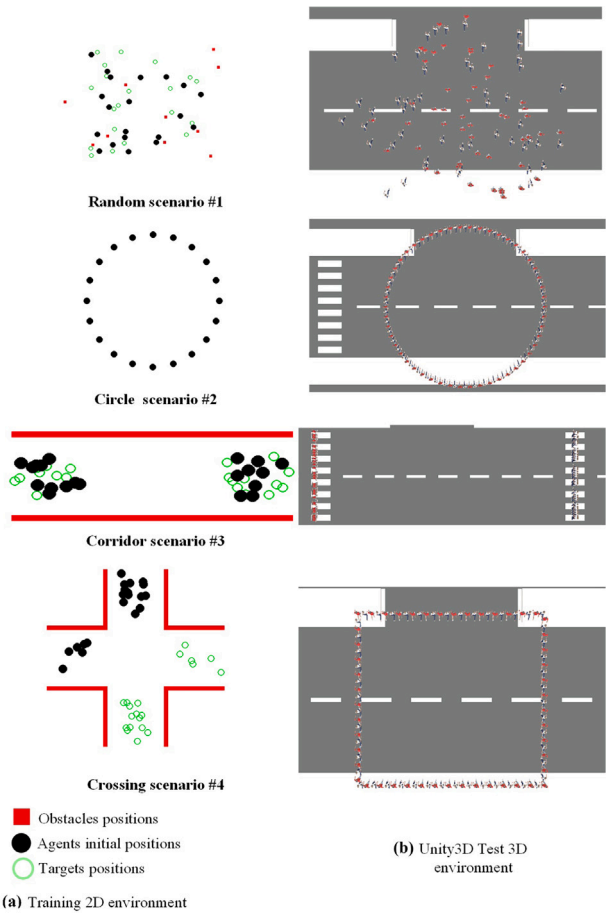


Fig. 5. The performance of our framework was evaluated using four distinct training and testing environments. In 2D, we employed random, circle, corridor, and crossing layouts to assess fundamental navigation capabilities. To further test the framework's robustness, complex 3D environments developed in Unity3D were also used, providing a simulated real-world setting for rigorous validation of its efficacy.

For benchmarking, we evaluate our Mamba-MAGAIL framework against five representative methods: Multi-Agent Optimal Reciprocal Collision Avoidance (ORCA) [8], Social Force Model (SFM) [9], MAGAIL [32], standard PPO [26], LSTM-Attention [24] and SocialGAIL [37].

4.2. Comparison and analysis

To rigorously verify the effectiveness of the proposed method, extensive quantitative and qualitative comparative experiments were carried out. Performance was evaluated based on six key metrics across four distinct scenarios: Random, Circle, Corridor, and Crossing. These scenarios were chosen because they represent fundamental navigation challenges commonly encountered in real-world environments, thereby enabling a thorough and convincing assessment. The 2D training and corresponding 3D testing environments for these scenarios are illustrated in Fig. 5.

4.2.1. Quantitative experiments

Extensive experiments were conducted to evaluate the proposed method across multiple metrics on the ETH/UCY, GC and SDD datasets, respectively. The results are summarized in Tables 3–5.

As shown in the tables above, our framework achieves the best performance across most evaluation indicators. In particular, the SSIM metric exhibits outstanding results, demonstrating the benefits of learning from real-world datasets. This allows our approach to generate

Table 3
Comparison of our framework with other methods on the ETH/UCY dataset: bold text indicates the best result, and underlined text indicates the second-best. Blue column headings represent macroscopic indicators.

Criteria								
Methods	Speed↑	Displacement↓	Deviation↑	Velocity↑	Angle↑	Energy↑	Steer↑	SSIM↑
PPO [26]	0.698	0.251	8.389	0.256	0.0278	8.525	12.95	0.85
LSTM-Attention [24]	0.75	0.268	8.45	0.245	0.037	8.66	15.102	0.86
MAGAIL [32]	0.7218	0.0282	8.3749	**0.0317**	**0.0299**	8.4682	13.785	0.87
ORCA [8]	0.3179	0.058	8.5449	0.0189	0.0284	4.8894	8.1612	0.8075
SFM [9]	0.3298	0.0361	8.4985	0.0179	0.0205	4.6596	8.1751	0.78
Ours	**1.0367**	**0.0149**	**8.3356**	0.0192	0.0222	**8.7684**	**15.863**	**0.9054**

Table 4
Comparison of our framework with other methods on the GrandCentre (GC) dataset: bold text denotes the best result, and underlined text indicates the second-best. Blue column headings represent macroscopic indicators.

Criteria								
Methods	Speed↑	Displacement↓	Deviation↑	Velocity↑	Angle↑	Energy↑	Steer↑	SSIM↑
PPO [26]	0.641	0.0291	**11.389**	0.048	0.0408	8.825	13.075	0.86
LSTM-Attention [24]	**0.706**	0.03	12.45	0.045	0.037	8.79	13.52	0.86
MAGAIL [32]	0.1226	0.0274	12.725	0.051	**0.0477**	4.2736	8.3756	0.877
ORCA [8]	0.6064	0.0562	12.8875	0.0253	0.0375	6.7761	10.1153	0.7699
SFM [9]	0.1925	0.0367	12.9382	0.0261	0.033	5.203	9.2633	0.819
Ours	0.648	**0.0101**	12.71	**0.0528**	0.0429	**9.2639**	**13.729**	**0.9065**

Table 5
Comparison of our framework with other methods on the SDD dataset: bold text denotes the best result, and underlined text indicates the second-best. Blue column headings represent macroscopic indicators.

Criteria								
Methods	Speed↑	Displacement↓	Deviation↑	Velocity↑	Angle↑	Energy↑	Steer↑	SSIM↑
PPO [26]	2.18	0.11	26.02	1.25	0.79	38.78	43.08	0.89
LSTM-Attention [24]	2.16	0.13	28.95	1.39	0.71	42.94	47.58	0.91
MAGAIL [32]	2.33	**0.04**	28.75	1.56	0.77	42.08	**49.25**	0.90
ORCA [8]	**2.42**	0.15	**30.25**	1.49	0.84	35.25	28.69	0.86
SFM [9]	1.28	0.27	27.97	1.47	**0.86**	37.89	48.02	0.83
Ours	2.34	0.05	29.05	**1.61**	0.83	**45.77**	**49.25**	**0.96**

realistic imitation outcomes, a crucial factor in accurately simulating real-world scenarios. Moreover, for macroscopic indicators such as speed, our framework exhibits superior capability in simulating collective crowd behaviors. Overall, in most microscopic indicators, our method consistently achieves top results, clearly demonstrating its overall superiority.

Furthermore, as discussed earlier, four representative virtual scenarios were designed to evaluate performance in terms of time consumption and collision occurrences. The comparative results are shown in Fig. 6.

To evaluate our approach against attention-based mechanisms, we conduct a comparative analysis with SocialGAIL on the GC dataset, utilizing results reported in the original paper. To ensure a fair and convenient comparison, we employ standardized evaluation metrics to assess performance consistency. The comparison employs three primary indicators: Frechet Distance, Final Distance Error (FDE), and Hausdorff Distance. These metrics collectively provide a comprehensive assessment of trajectory prediction accuracy and spatial deviation. The experimental results are presented below Table 6.

Additionally, four imitation environments—crossing, corridor, circular, and random—were developed to represent typical real-world scenarios with 10 random static obstacles. Total time consumption and the number of collisions were used as performance indicators. As shown in Fig. 6, although our framework does not achieve the shortest pathfinding time, compared with ORCA and SFM—which excessively prioritize minimizing time consumption while partially neglecting collision avoidance—our framework ensures that almost no collisions occur. This advantage stems from learning patterns observed in real-world datasets, where most pedestrians prefer to wait rather than forcibly moving forward and risking collisions. Consequently, our framework emphasizes collision avoidance in every form, making it particularly well-suited for simulating realistic crowd movement behaviors.

4.2.2. Qualitative experiments

In this section, qualitative comparison experiments are conducted to demonstrate explicit results. A heatmap is used to visually present the similarity between our approach and other methods.

Fig. 7 displays the heatmaps for the ETH/UCY and GC datasets. The degree of crowd activity represents pedestrian density. It is clear that our proposed method is best suited for benchmark evaluations. Compared with the benchmark, our framework provides a more accurate representation of key areas with high crowd activity.

4.2.3. Sensitivity analysis

To effectively assess the robustness of the proposed framework and identify influential factors, we performed a one-factor-at-a-time sensitivity analysis. Specifically, for each parameter, we varied its value while keeping all other settings fixed, and evaluated performance using the final evaluation episodic reward. We report the mean reward μ_R and its standard deviation σ_R across evaluation rollouts. In addition, we report the arrival rate (i.e., fraction of agents reaching the given goal) to offer a task-level success indicator beyond reward.

Sensitivity results. As Table 7 shows, the observed sensitivity across all tested parameters is presented. Overall, the method is highly robust to the interaction geometry parameter (the radius of the Social Force model), while it exhibits greater sensitivity to the exploration regularizer (Entropy) and to task difficulty changes induced by $\mathcal{N}_{ped}$ (crowd density). Besides, seed to seed variability is also non-negligible, reinforcing the need for multi-seed reporting.

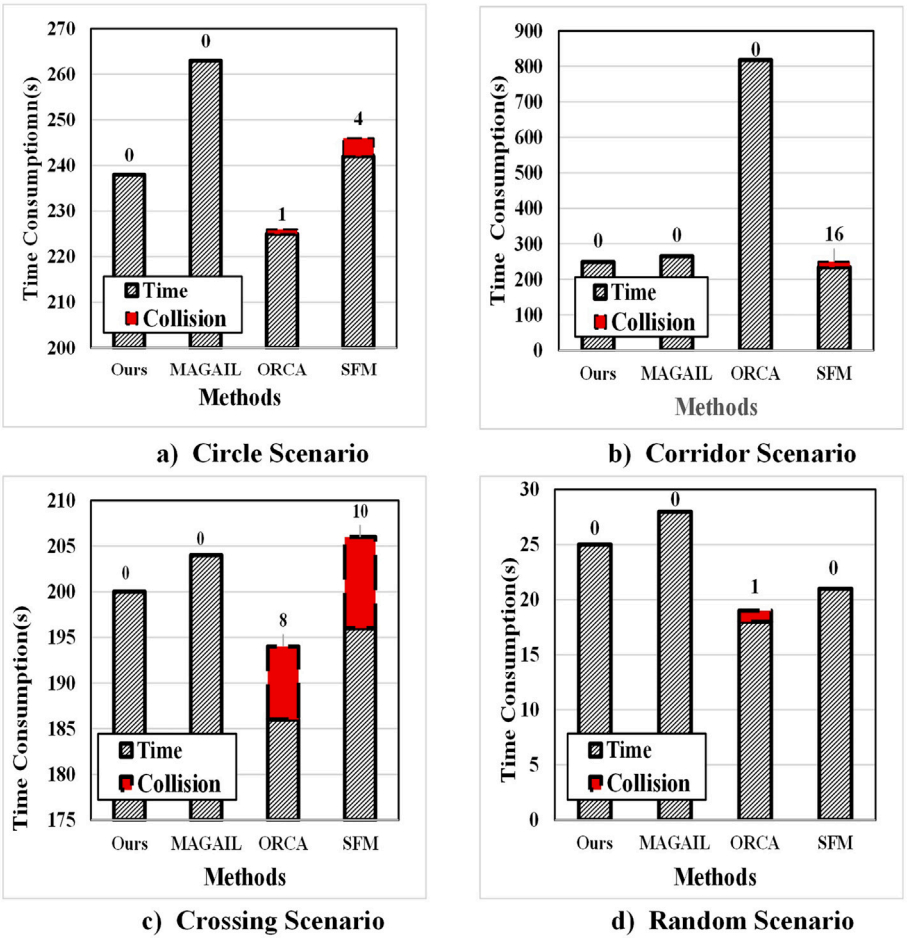


Fig. 6. Comparison results between our method and others across four different virtual scenarios—Circle, Corridor, Crossing, and Random—in terms of time consumption and collision count based on a static obstacle scenario. The collision count is indicated on the bar chart.

Table 6
Comparison of our framework with other methods on Grand Center dataset (Bold indicates the best).

Methods	Frechet↓	FDE↓	Hausdorff↓
SocialGAIL [37]	3.8	0.44	285.5
RadialGAIL [37]	4.31	1.41	341.0
SocialBC [37]	4.50	2.01	343.1
RadarRNN	6	4.4	380.5
Ours	**2.75**	**0.36**	**219.5**

PPO hyperparameters. Entropy coefficient (Entropy λ_{ent}) has the strongest impact among the PPO-related factors. In fact, moderate entropy (10^{-3}) yields the best mean reward ($\mu_R = -2.876$), whereas overly large entropy (10^{-1}) substantially degrades performance ($\mu_R = -5.263$), corresponding to $\delta_{\mu_R} = 2.408$. The arrival rate remains low across the sweep ($\kappa_{arr} \in [0.017, 0.083]$), suggesting that entropy primarily affects interaction quality and reward shaping rather than dramatically changing success frequency.

Environmental factors. Crowd density ($\mathcal{N}_{ped}$) affects both reward and variablity. Increasing to 80 pedestrians produces the worst mean reward ($\mu_R = -3.696$) and the highest dispersion (almost $\sigma_R = 1.218$), while 40 pedestrians yields the best mean reward ($\mu_R = -2.596$). Notably, the arrival rate κ_{arr} increases with density in our runs (κ_{arr} up to 0.25), implying that the reward is influenced by more than pure success or failure (e.g., efficiency, collision penalties or comfort terms), and that density alters the interaction regime in a way that changes the reward trade-off. Scenario reset frequence (λ_{update}) has a comparatively small effect ($\delta\mu_R = 0.422$). The setting 100 achieves the best mean reward ($\mu_R = -2.592$) and a higher arrival rate ($\kappa_{arr} = 0.150$), while higher reset values cluster near $\mu_R \sim 0.089$. This suggestis performances is not dominated by reset artifacts, although resets may modestly improve learning conditions.

Stochasticity and reproducibility (SEED). Random seed changes induce moderate fluctuations ($\delta\mu_R = 0.605, \kappa_{arr} \in [0.067, 0.183]$), consistent with stochasticity from initialization and sampling. Consequently, results are averaged over multiple seeds, particularly when comparing methods with small expected margins.

Interaction geometry ω_{sfm}. It has a negligible effect on performance (i.e., $\delta\mu_R = 0.0049$) and does not change arrival rate κ_{arr}. This indicates that the learned policy and environment interactions are effectively invariant to this geometric scale.

To assess the effectiveness of state representation heterogeneity and normalization, a sensitivity analysis was conducted on the normalized state encoder using the SDD dataset. The SDD dataset contains densely populated scenarios that may insufficiently capture local social context. The findings are presented in Table 8, which demonstrate the impact of normalization techniques on model performance and state representation quality.

Table 8 presents comprehensive evidence of the proposed method's robustness across reasonable hyperparameter configurations. The normalization bounds demonstrate minimal sensitivity, with final displacement error (FDE) variations not exceeding 0.08 meters across tested ranges, thereby validating the selection of $[-5, 5]$ as appropriate

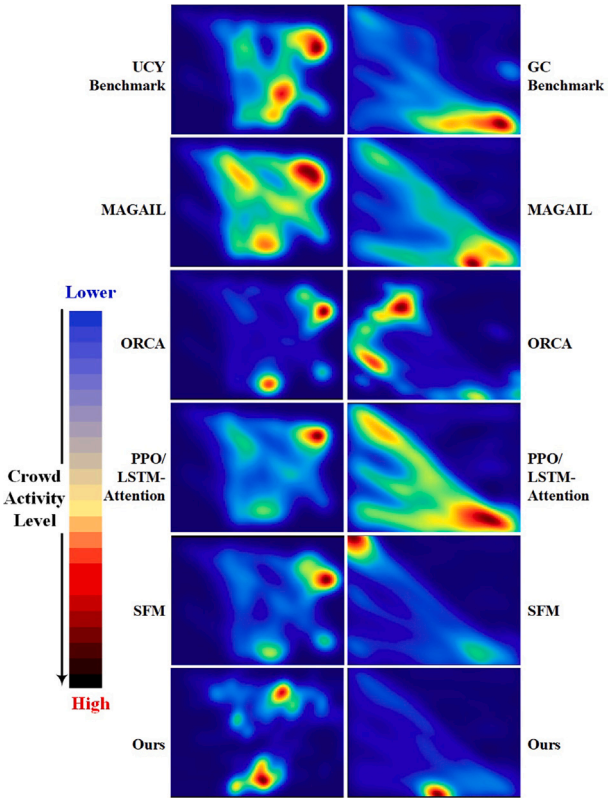


Fig. 7. Heatmap comparison of various methods applied to two real-world datasets—ETH/UCY and GC. The first row shows the benchmark results for the real datasets. It should be noted that, due to the high similarity between the PPO and LSTM-Attention methods, their combined heatmap results are presented.

bounds. The designated warm-up period of 50 episodes effectively balances statistical accuracy with computational efficiency. Furthermore, the relatively modest convergence change number indicates stable and reliable training dynamics, confirming that the method maintains consistent performance without requiring excessive fine-tuning of hyperparameters.

4.2.4. Ablation studies

To systematically evaluate the contribution of each component within the proposed Agent encoder paradigm, comprehensive ablation experiments were conducted. A key methodological consideration involved the treatment of static obstacles, which inherently present challenges in capturing temporal dynamics. To address this limitation, obstacles were modeled as dynamic entities driven by a social force model, thereby enhancing the framework's capacity to process temporal long-sequence information. The Agent encoder architecture comprises two primary branches: Branch A and Branch B. Branch B integrates a spatial aggregator and a temporal Mamba encoder to facilitate sophisticated feature extraction. Performance evaluation was conducted using

Table 8
Sensitivity analysis on state representation heterogeneity and normalization.

Parameter	Baseline	Variation	Δ FDE (m)	Δ Conv. (%)
Clamp bounds	[−5,5]	[−3,3] [−10,10]	+0.08 + 0.06	+3.2 + 2.1
Warm-up episodes	50	20 100	+0.15 −0.04	+7.5 −1.2
Temporal window $\mathcal{T}$	20	10 30	+0.22 −0.06	+9.8 −2.4

Table 9
Ablation study results showing the effects of different architectural components on the SDD dataset.

Paradigm	Collisions ↓	Reward ↑
Full	**80**	**−1.12**
w/o Branch A (Spatial only)	368	−2.84
w/o Branch A	362	−2.71
w/o Branch B (self-state only)	184	−2.99
w/o Temporal mamba encoder	368	−2.99
only Self mlp	320	−3.57
only Spatial aggregator	160	−3.17
Branch A + Spatial aggregator	192	−2.48

two primary metrics, count of collisions across all episodes and average rewards (i.e., $-\frac{\sum_i \mathcal{L}_i}{\mathcal{N}\times \mathbf{T}_{ep}}$) achieved by the agent. These metrics provide comprehensive insights into both safety and efficiency aspects of the proposed approach. Detailed results of the ablation study are presented in Table 9, which systematically demonstrate the individual contribution of each architectural component to overall system performance.

The ablation study presented in Table 9 reveals a clear distinction in the modeling capabilities of different architectural components. The Branch A architecture combined with spatial aggregation primarily captures only individual dynamic features, such as ego-vehicle speed and acceleration, while fundamentally limiting its capacity to handle dynamic crowd interactions. This limitation stems from the temporal information dropout inherent in the aggregation strategy, which restricts neighbor information integration to the final frame only, effectively treating the crowd as static. In contrast, the proposed Temporal Mamba encoder substantially enhances predictive performance by sequentially processing spatially-aggregated neighbor embeddings across the full temporal horizon of T = 20 timesteps. This design enables comprehensive capture of critical multi-agent phenomena, including neighbor velocity trends, collision avoidance patterns, and flow dynamics that are essential for accurate crowd behavior prediction. The empirical results thus demonstrate that maintaining temporal continuity in neighbor information processing is crucial for developing robust trajectory prediction models in complex, interactive environments.

4.3. Results

In this section, our framework is applied to simulate four virtual scenarios to further demonstrate its feasibility and robustness. The Unity3D engine is utilized to accomplish these tasks. The results are shown in Figs. 8 and 9. Furthermore, by leveraging the video data from the ETH/UCY dataset and fully exploiting the proposed framework, a highly realistic virtual environment can be constructed through transfer

Table 7
Sensitivity analysis on UCY/eth dataset,higher mean reward μ_R is better. Error statistics are summarized by the observed range of reward standard deviation σ_R across tested values.

Parameter	Values tested	Best μ_R (value)	Worst μ_R (value)	$\Delta\mu_R$	σ_R range	Arrival range
λ_{ent}	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$	−16.891 (10^{-4})	−18.552 (10^{-1})	1.660	[1.445, 1.603]	[0.040, 0.048]
$\mathcal{N}_{ped}$	{10, 20, 40, 80}	−16.177 (40)	−17.080 (10)	0.903	[1.290, 2.730]	[0.020, 0.133]
λ_{update}	{100,500, 1000, 9999999}	−16.967 (100)	−17.009 (1000)	0.042	[1.366, 1.444]	[0.045, 0.045]
SEED	{0, 1, 2, 3, 4}	−16.980 (3)	−17.441 (0)	0.462	[1.395, 2.235]	[0.030, 0.047]
ω_{sfm}	$\{0.1, 0.2, 0.3, 0.5\}m$	−17.007 (0.1)	−17.008 (0.5)	0.0003	[1.443, 1.443]	[0.045, 0.045]

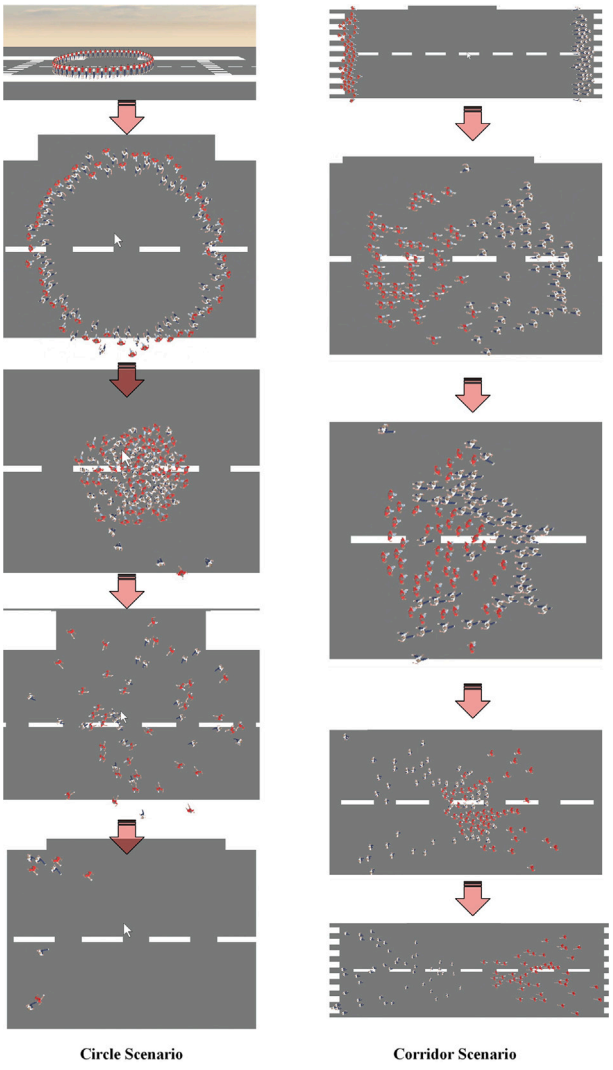


Fig. 8. Simulation results for the Circle and Corridor virtual scenarios from the initial to the final timestamp.

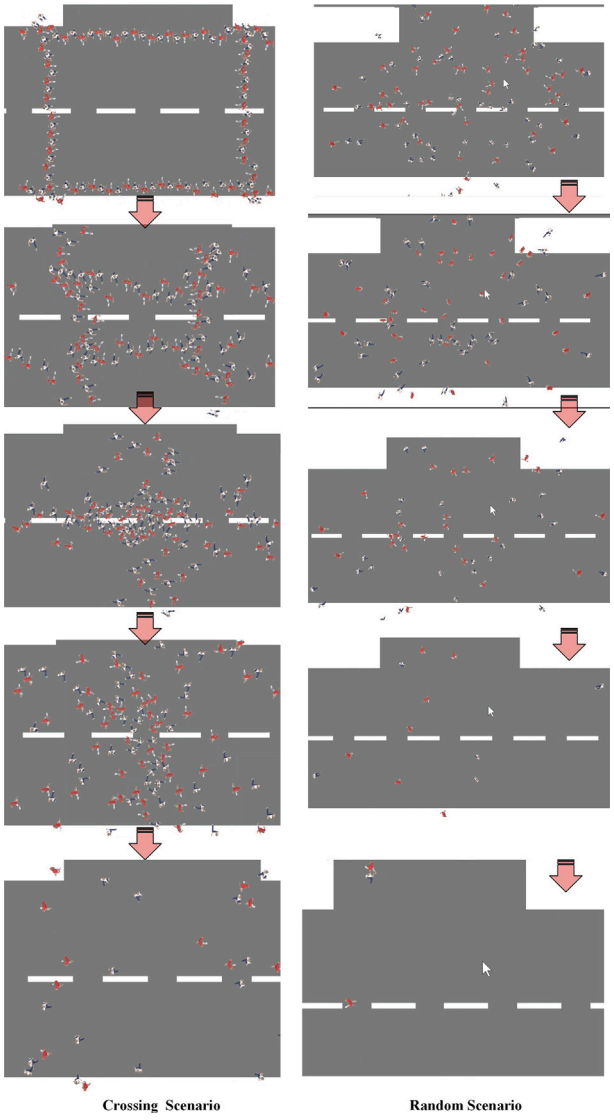


Fig. 9. Simulation results for Crossing and Random virtual scenarios.

learning across two mutually imitative environments of different scales, thereby enabling real-world scenario simulation.

Fig. 8 depicts pedestrian dynamics throughout the process, from start to finish, from a bird's-eye perspective. The overall process closely mirrors real-world conditions. Notably, our framework effectively prevents potential collisions, as the responsive agents, guided by the proposed model, employ a stop-and-wait strategy to ensure a zero-collision outcome. In this regard, the simulation more accurately represents real-world situations.

Meanwhile, Fig. 9 illustrates the crowd dynamics in the other two scenarios. Clearly, these cases involve more frequent multi-agent interactions. Nevertheless, our framework successfully guides the agents to maintain zero collisions while ensuring that each agent reaches its respective destination.

Fig. 10 demonstrates the similarity between pedestrian dynamics captured in real video footage and those produced by our framework. However, slight discrepancies exist due to perspective issues arising from our preprocessing method. Future work will incorporate more robust view-projection techniques to mitigate this limitation.

4.4. Discussion

The experimental results demonstrate the superior capabilities of the Mamba-MAGAIL architecture. The selective state updates within the Mamba blocks effectively capture long-range temporal dependencies, a critical requirement for anticipating and avoiding collisions in dynamic environments.

Despite these advantages, several notable limitations warrant consideration. Mamba models exhibit considerable sensitivity to hyperparameter initialization, particularly with respect to state expansion factors, thereby necessitating meticulous tuning procedures. Furthermore, the inherently recurrent nature of Mamba architectures demands strict gradient clipping constraints during Proximal Policy Optimization (PPO) to mitigate the risk of exploding gradients. Finally, inherited limitations from the MAGAIL framework, including implicit representation, model collapse, and unstable adversarial rewards, contribute to performance degradation when the model is deployed in environments where pedestrian densities substantially exceed those encountered during training.

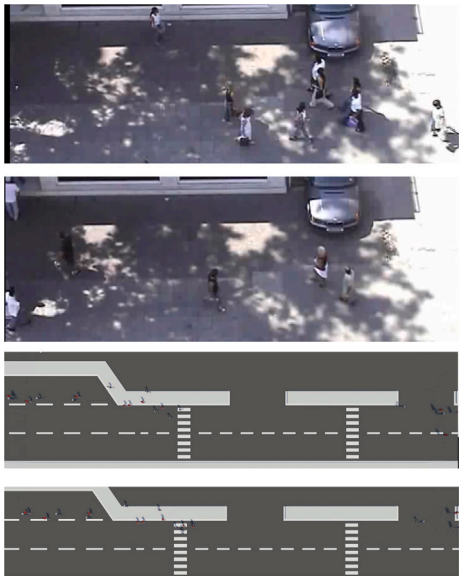


Fig. 10. Simulation results for virtual scenario learning based on the ETH/UCY dataset. Specifically, the first two rows display pedestrian movements from the video dataset, followed by the corresponding replicated crowd dynamics generated by our framework.

5. Conclusions

This paper presents Mamba-MAGAIL, a novel multi-agent reinforcement learning framework that addresses the persistent challenges of temporal modeling and reward design in crowd navigation. By integrating Mamba state-space models into the Actor-Critic and Discriminator networks, the framework enables linear-time extraction of complex, long-range pedestrian behaviors from expert demonstrations.

Extensive quantitative and qualitative experiments on the ETH/UCY, Grand Central (GC), Stanford Drone Dataset (SDD) confirm that the proposed architecture outperforms traditional heuristics and standard RL baselines in terms of safety, realism, and spatial similarity. The successful deployment of the model into 3D simulation environments further underscores its practical potential.

Future research will focus on three key areas: (1) Implementing adaptive neighbor selection based on proximity thresholding rather than fixed $K = 20$; (2) exploring multi-scale temporal architectures to handle both micro-level collision avoidance and macro-level route planning simultaneously; and (3) integrating explicit trajectory prediction modules to enhance proactive obstacle avoidance in highly unstructured environments.

CRediT authorship contribution statement

Wen Zhou: Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Yehui Qiu:** Software, Resources, Investigation.

Funding

This work was supported in part by the University Natural Science Research Project of Anhui Province under Grant No. 2023AH050142. The authors gratefully acknowledge this financial support, which contributed to the successful completion of the research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors express their appreciation to all anonymous reviewers for their valuable feedback and suggestions, which have significantly improved this paper.

Data availability

Data sharing is not applicable because all data utilized in the study were obtained from publicly available open-source datasets. Additionally, the study did not require any specialized physical materials or exceptional resources for its execution. As a result, there are no additional datasets or materials to provide beyond those already publicly accessible.

References

[1] W. Zhou, W. Jiang, B. Jie, W. Bian, Multiagent evacuation framework for a virtual fire emergency scenario based on generative adversary imitation learning computer animation and virtual worlds, Computer Animation and Virtual Worlds 33 (1) (2022) e2035.
[2] W. Zhou, C. Zhang, S. Chen, Dual deep Q-learning network guiding a multiagent path planning approach for virtual fire emergency scenarios applied intelligence, Appl. Intell. 53 (2023) 21858–21874.
[3] W. Zhou, W. Shen, X. Meng, An improved social force model-driven multiagent generative adversarial imitation learning framework for pedestrian trajectory prediction, Computer Animation and Virtual Worlds 36 (1) (2025) e2035.
[4] W. Xiaohong, Ye Tao.Research on robot path planning based on improved A* algorithm, Computer Measurement and Control 26 (7) (2018) 282–286.
[5] A. Lissovoi, C. Witt, Runtime analysis of ant colony optimization on dynamic shortest path problems, Theor. Comput. Sci. 561 (2015) 73–85.
[6] S. Wang, H. Liu, K. Gao, et al., A multi-species artificial bee colony algorithm and its application for crowd simulation, IEEE Access 7 (1) (2018) 2549–2558.
[7] T. Mac, C. Copot, D.T. Tran, et al., A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization, Appl. Soft Comput. 59 (2017) 68–76.
[8] J. Van Den Berg, et al., "Optimal reciprocal collision avoidance for multi-agent navigation, in: Proc. of the IEEE International Conference on Robotics and Automation, Anchorage (AK), USA, 2010.
[9] L. Hou, et al., A social force evacuation model with the leadership effect, Phys. a Stat. Mech. Appl. 400 (2014) 93–99.
[10] J. Kober, J.A. Bagnell, J. Peters, Reinforcement learning in robotics: a survey, Int. J. Robot. Res. 32 (11) (2013) 1238–1274.
[11] Q. Zhou, Y. Lian, J. Wu, et al., An optimized Q-learning algorithm for mobile robot local path planning, Knowl.-based Syst. 286 (2024) 111400.
[12] A. Konar, I.G. Chakraborty, S.J. Singh, et al., A deterministic improved Q-learning for path planning of a mobile robot, IEEE Trans. Syst. Man Cybern. Syst. 43 (5) (2013) 1141–1153.
[13] L. Yehezkel, S. Berman, D. Zarrouk, Overcoming obstacles with a reconfigurable robot using reinforcement learning, IEEE Access 8 (1) (2020) 217541–217553.
[14] N. Imanberdiyev, C. Fu, et al., Autonomous navigation of UAV by using real-time model-based reinforcement learning, in: 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2016.
[15] V. Mnih, K. Kavukcuoglu, D. Silver, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[16] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, Proceedings of the AAAI conference on artificial intelligence. 30 (1) (2016).
[17] Z. Wang, T. Schaul, M. Hessel, et al., Dueling network architectures for deep reinforcement learning international conference on machine learning, in: PMLR, 2016, pp. 1995–2003.
[18] J. Foerster, N. Nardelli, G. Farquhar, et al., Stabilising experience replay for deep multi-agent reinforcement learning international conference on machine learning, in: PMLR, 2017, pp. 1146–1155.
[19] S. Yin, Z. Xiang, Adaptive operator selection with dueling deep Q-Network for evolutionary multi-objective optimization, Neurocomputing 581 (2024) 127491.
[20] S.K. Pattanayak, M. Bhoyar, T. Adimulam, Deep reinforcement learning for complex decision-making tasks, Int. J. Innov. Res. Sci. Eng. Technol. 13 (11) (2024) 18205–18220.
[21] S. Swapno, S. Nobel, P. Meena, et al., A reinforcement learning approach for reducing traffic congestion using deep Q learning, Sci. Rep. 14 (1) (2024) 30452.

[22] Q. He, L. Zhang, H. Fang, et al., Multistage competitive opinion maximization with Q-learning-based method in social networks, IEEE Trans. Neural Netw. Learn. Syst. 36 (4) (2024) 7158–7168.
[23] F. Martinez-Gil, M. Lozano, F. Fernández, Multi-Agent Reinforcement Learning for Simulating Pedestrian Navigation International Workshop on Adaptive and Learning Agents, Springer, Berlin Heidelberg, 2011, pp. 54–69.
[24] D.O. Oyewola, S.A. Akinwunmi, T.O. Omotehinwa, Deep LSTM and LSTM-attention Q-learning based reinforcement learning in oil and gas sector prediction, Knowl.-based Syst. 284 (2024) 111290.
[25] L. Sun, J. Yan, Y. Qiu, et al., The crowd cooperation approach for formation maintenance and collision avoidance using multi-agent deep reinforcement learning, The Visual Compututer 41 (2025) 4081–4095.
[26] Y. Gu, Y. Cheng, C.L.P. Chen, et al., Proximal policy optimization with policy feedback, IEEE Trans. Syst. Man Cybern. Syst. 52 (7) (2021) 4600–4610.
[27] T. Van Der Heiden, F. Mirus, H. Van Hoof, Social Navigation with Human Empowerment Driven Deep Reinforcement Learning International Conference on Artificial Neural Networks, Springer International Publishing, Cham, 2020, pp. 395–407.
[28] S.A. Alizadeh Kolagar, A. Taheri, A.F. Meghdari, NAO robot learns to interact with humans through imitation learning from video observation, J. Intell. Robot. Syst. 109 (1) (2023) 4–16.
[29] J. Xie, Y. Zhang, H. Yang, et al., Crowd perception communication-based multi-agent path finding with imitation learning, in: IEEE Robotics and Automation Letters, vol. 9(10, 2024, pp. 8929–8936.
[30] J. Ho, S. Ermon, Generative adversarial imitation learning, Adv. Neural Inf. Process. Syst. (2016) 29–35.
[31] I. Gulrajani, F. Ahmed, M. Arjovsky, et al., Improved training of Wasserstein GANs, Adv. Neural Inf. Process. Syst. (2017) 30–41.
[32] J. Song, H. Ren, D. Sadigh, et al., Multi-agent generative adversarial imitation learning, Advances in neural information processing systems, pp. 31–38, 2018.
[33] G. Albert, T. Dao, Mamba: linear-time sequence modeling with selective state spaces, arXiv preprint arXiv:2312.00752, 2023. 1–6.
[34] Y. Huang, Y. Cheng, K. Wang, Trajectory Mamba: efficient attention-Mamba forecasting model based on selective ssm, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 12058–12067.
[35] L. Gao, Z. Zhuang, H. Tao, et al., A decentralized optimal iterative learning control approach with efficient computation for collaborative tracking, Int. J. Robust Nonlinear Control 35 (16) (2025) 6652–6669.
[36] Z. Peng, X. Song, S. Song, et al., Resilient consensus tracking control and vibration suppression of moving multiple helicopter flexible slung-load systems, J. Vib. Control 32 (7–8) (2026) 1893–1909.
[37] B. Ling, Y. Lyu, D. Li, G. Gao, et al., Socialgail: faithful crowd simulation for social robot navigation 2024, in: IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 16873–16880.
[38] Y. Kefu, et al., UCMCTrack: multi-object tracking with uniform camera motion compensation, Proc. AAAI Conf. Artif. Intell. 38 (7) (2024) 6702–6710.

Author biography

Wen Zhou, received the Ph.D. degree from the School of Software Engineering, Tongji University in 2018. Since 2018, he has been in the School of Computer and Information, Anhui Normal University, Wuhu, China, where he has been an associate professor since 2021. Besides, he is a senior member of the China Graphics Society, an IEEE member, and a member of the Chinese Computer Federation (CCF). His research interests include reinforcement learning, data visualization, virtual reality, sketch-based retrieval, deep learning, and so on.

Yehui Qiu, is a Master's degree candidate at Anhui Normal University in China. He received his B.Sc. degree in New Media Technology from Guangxi University of Science and Technology in 2024 and is currently studying for a Master's degree in Computer Technology at Anhui Normal University. His research interests mainly focus on multiagent reinforcement learning.