SAGE: A Socially-Aware Generative Engine for Heterogeneous Multi-Agent Navigation

Lan Hu, Minghui Liwang, *Senior Member, IEEE*, Wenbo Zhu, Xinlei Yi, *Senior Member, IEEE*, Yiguang Hong, *Fellow, IEEE*, Xianbin Wang, *Fellow, IEEE*, Zhenzhen Jiao, Seyyedali Hosseinalipour, *Senior Member*, *IEEE*

Abstract*—Safe and socially compliant navigation in open human-robot environments requires robots to navigate among *heterogeneous agents*, such as pedestrians, cyclists, vehicles, and other robots, that exhibit fundamentally different motion dynamics, behavioral characteristics, and interaction patterns. Consequently, robots must simultaneously predict the future motions of surrounding agents and plan their own trajectories while satisfying both physical safety requirements and social interaction norms. Nevertheless, existing trajectory prediction and planning methods typically assume homogeneous interactions or enforce only geometric collision constraints, making them inadequate for modeling the distinct interaction patterns among different agent categories, jointly predicting surrounding-agent motions and robot trajectories, and incorporating social-compliance constraints (i.e., maintaining appropriate interpersonal distances and exhibiting socially acceptable navigation behaviors). Motivated by this shortcoming, we propose SAGE, a socially-aware generative engine for heterogeneous multi-agent navigation. SAGE represents robots and surrounding agents as a *directed heterogeneous graph* and employs a *heterogeneous graph transformer* (HGT) to learn interaction representations that distinguish different agent types and interaction directions. Conditioned on this representation, we develop a *diffusion-based generative model* that jointly generates surrounding-agent trajectory predictions and robot trajectory plans within a unified probabilistic framework, thereby explicitly modeling the inherent coupling between environment prediction and robot motion planning. During inference, we introduce a *training-free safety-social energy guidance mechanism* that injects differentiable collision-avoidance, kinematic-feasibility, task-progress, and role-dependent social-compliance objectives into the diffusion sampling process, thereby refining the generated robot trajectories without retraining. Through extensive experiments on real-world (ETH/UCY and SDD) and synthetic datasets, we demonstrate that SAGE consistently improves safety and social compliance while maintaining task performance; furthermore, its training-free guidance mechanism reduces collision and social-violation rates, scales to teams of up to 20 robots, and enables explicit control of the trade-off among safety, task performance, and trajectory accuracy without retraining.

***Index Terms*—Socially-aware navigation, heterogeneous multi-agent systems, diffusion models, trajectory generation, heterogeneous graph transformer.**

I. Introduction

L. Hu (2453781@tongji.edu.cn) is with the Guohao School, Tongji University, Shanghai, China. M. Liwang (minghuiliwang@tongji.edu.cn), W. Zhu (wbzhu@tongji.edu.cn), X. Yi (xinleiyi@tongji.edu.cn) and Y. Hong (yghong@tongji.edu.cn) are with the Shanghai Research Institute for Intelligent Autonomous Systems, the State Key Laboratory of Autonomous Intelligent Unmanned Systems, Department of Control Science and Engineering, Tongji University, Shanghai, China. X. Wang (xianbin.wang@uwo.ca) is with the Department of Electrical and Computer Engineering, Western University, Ontario, Canada. Z. Jiao (jiaozhenzhen@gmail.com) is with Aeromind Technology Co., Ltd., Shenzhen, China. S. Hosseinalipour (alipour@buffalo.edu) is with the Department of Electrical Engineering, University at Buffalo-SUNY, USA. Corresponding author: M. Liwang.

DRIVEN by recent advances in large-scale robotic systems [1], autonomous agents are rapidly transitioning from structured and controlled environments to unstructured, open environments characterized by dense human-robot coexistence and interaction. In such mixed-traffic scenarios, achieving safe, efficient, and socially compliant autonomous navigation remains a major challenge [2], [3]. This is because unlike structured environments, these scenarios feature heterogeneous agents with diverse behavioral patterns and interactions. Specifically, these agents can be classified into four categories based on their levels of autonomy and control authority: *(i) Purely human agents* (PHAs, e.g., pedestrians and runners), who are driven entirely by human will, exhibit highly random behavior and adhere to a wide range of social interaction norms [4]; *(ii) Human-controlled non-autonomous agents* (HNAAs, e.g., skateboarders and cyclists), who rely on manual power or simple mechanical propulsion without autonomous sensing, making their trajectories highly dependent on users' real-time reflexes; *(iii) Shared-control semi-autonomous agents* (SSAAs, e.g., human-driven vehicles with advanced driver-assistance systems), where human decision-making operates in tandem with intelligent assistance, retaining inherent behavioral unpredictability despite partial rule compliance [5]; and *(iv) Execution-autonomous agents* (EAAs, e.g., automated robots), which are data-driven, with their behavior being relatively predictable under limited compatibility with human social norms [6]. Among these four categories, PHAs, HNAAs, and SSAAs are independently behaving entities whose future motions cannot be directly controlled, requiring robots to continuously anticipate their future behaviors and interactions [7]. In contrast, EAAs correspond to the controllable robots whose trajectories must be planned according to these predicted behaviors while simultaneously satisfying physical safety and social interaction requirements.

This tight coupling between environment prediction and robot trajectory planning is a defining characteristic of socially-aware navigation in heterogeneous environments. However, jointly solving these two tasks (i.e., predicting the future behaviors of surrounding heterogeneous agents and planning safe and socially compliant trajectories for autonomous robots) remains fundamentally challenging because robots must reason about human-driven and AI/data-driven agents that exhibit diverse motion dynamics, autonomy levels, and social interaction behaviors. Addressing these diverse interactions requires navigation frameworks capable of reasoning about both heterogeneous agent behaviors and their influence on robot decision-making. Nevertheless, existing trajectory prediction and planning methods typically rely on homogeneous interaction assumptions [8], [9] or focus primarily on geometric collision avoidance [10], [11], making them unable to adequately capture these heterogeneous interactions and the associated social behaviors. Consequently, achieving safe and socially compliant navigation in heterogeneous multi-agent environments remains a challenging open problem [6], [12]–[14].

A. Core Motivation

To address the aforementioned challenges, we investigate the following key research questions (RQs):

• *RQ 1: How can we develop principled models that capture structural and behavioral asymmetries in heterogeneous multi-agent interactions, where agents differ in dynamics, perception, and influence over the shared environment?* Most mainstream trajectory generation models are developed under the homogeneous-agent assumption, employing shared parameters to uniformly model diverse agent types [8], [9], thereby overlooking asymmetric interaction mechanisms across heterogeneous agent categories. However, in real-world environments, interactions vary significantly across agent categories. For example, a robot may proactively yield to nearby pedestrians while instead anticipating and tracking the future motion of vehicles. Ignoring these *role-dependent* interactions limits the ability of existing models to capture realistic cross-category behaviors, thereby compromising both trajectory prediction and robot trajectory planning. Since these interaction asymmetries naturally arise from the semantic roles of different agents and their relationships, they can be naturally represented as a heterogeneous interaction graph, where nodes correspond to different agent categories and edges encode their interaction relationships [15]. Based on the heterogeneous graph representation of the scene, we develop a Heterogeneous Graph Transformer (HGT), inspired by [16], that learns distinct interaction patterns across different agent categories. By explicitly distinguishing semantic types of nodes and edges, HGT learns independent attention projection matrices for each interaction relationship (e.g., "robot→pedestrian", "pedestrian→robot", and "robot→robot"), thereby capturing asymmetric cross-category interactions and producing socially-aware scene representations.

• *RQ 2: How can physical feasibility constraints and culturally induced behavioral norms be jointly modeled within a unified framework that reconciles hard kinematic limitations with context-dependent social regularities?* Most existing generative models primarily enforce hard physical constraints, such as collision avoidance and kinematic feasibility, while largely overlooking soft social constraints. Nevertheless, in human-robot coexistence scenarios, navigation should ensure physical safety and, at the same time, comply with social norms and human comfort expectations [17]. For instance, even if a trajectory is physically collision-free, it may still be socially unacceptable when it frequently intrudes into pedestrians' personal space or disrupts ongoing social interactions [17]. To address this, we employ a *conditional diffusion model* [18], [19] as the generative backbone and propose a *safety-social energy guidance mechanism* based on a *differentiable energy function* [20] during the inference stage. Specifically, social norms, such as maintaining comfortable interpersonal distances and exhibiting polite avoidance behaviors, are formulated as heterogeneous anisotropic potential fields, whose gradients are incorporated into each diffusion denoising step to iteratively steer the generated trajectories toward lower-energy solutions. Unlike CVAE- and GAN-based methods [7], [21], where constraints are implicitly learned during training and are difficult to modify afterward, diffusion models enable differentiable guidance to be injected directly into the sampling process. Consequently, new physical or social constraints can be incorporated, adjusted, or removed at inference time without retraining the generative model. Through this safety-social energy guidance, generated trajectories are encouraged toward kinematic feasibility [22] and social compliance in a training-free, plug-and-play manner.

B. Novelty and Contribution

To address the aforementioned research questions, we develop a socially-aware generative engine, called *SAGE*, that integrates heterogeneous interaction modeling, diffusion-based trajectory generation, and safety-social guidance. The key contributions of SAGE are summarized as follows:

• *A novel problem formulation for safe and socially compliant navigation in heterogeneous multi-agent environments.* We formulate a new navigation problem in which autonomous robots operate alongside heterogeneous entities (e.g., pedestrians, cyclists, and semi-autonomous vehicles) exhibiting distinct dynamics, autonomy levels, and social behaviors. Unlike existing formulations, our problem explicitly captures the tight coupling among heterogeneous-agent interactions, surrounding-agent trajectory prediction, robot trajectory planning, and the joint consideration of hard physical safety constraints and soft social interaction norms. This unified formulation provides the foundation for developing socially-aware navigation algorithms in heterogeneous environments.

• *A unified generative framework for heterogeneous interaction modeling, trajectory prediction, and robot trajectory planning.* We develop SAGE, a socially-aware generative framework that unifies heterogeneous interaction modeling, surrounding-agent trajectory prediction, and robot trajectory planning within a single probabilistic framework. Specifically, SAGE represents heterogeneous agents and their interactions using a directed heterogeneous graph and develops an HGT to explicitly learn role-dependent and asymmetric interaction patterns across different agent categories. Building upon the resulting socially-aware scene representation, SAGE further develops a joint diffusion-based generative model that simultaneously predicts the future trajectories of surrounding agents and generates robot trajectory plans, thereby explicitly capturing the inherent coupling between environment prediction and robot motion planning.

• *A training-free safety-social guidance framework for controllable robot trajectory generation.* We develop a novel safety-social energy guidance mechanism that enables explicit constraint-aware correction during diffusion-based trajectory generation. Specifically, the proposed framework formulates collision avoidance, kinematic feasibility, task progress, and role-dependent social norms as differentiable energy functions, where heterogeneous anisotropic potential fields capture category-specific social behaviors. By injecting the resulting energy gradients into each diffusion denoising step, the proposed guidance iteratively reduces physical-safety and social-compliance violations at inference time, enabling flexible and controllable correction without modifying or retraining the underlying generative model.

• *Empirical validation.* We validate the effectiveness of *SAGE* through extensive experiments on real-world (ETH/UCY and SDD) and synthetic datasets, demonstrating its notable performance on reducing personal space intrusion and collision rates while maintaining trajectory diversity and kinematic feasibility.

II. Literature Review

In the following, we provide an overview of the existing literature while highlighting the key differences between prior studies and this work.

A. Trajectory Prediction for Socially-Aware Navigation in Heterogeneous Environments

Safe robot navigation in mixed human-robot environments fundamentally relies on accurate prediction of surrounding entities' future trajectories, which governs collision avoidance and motion planning. Early studies in this domain predominantly relied on recurrent neural networks (RNNs) to model temporal motion dependencies. For example, Social-LSTM [8] introduced a social pooling mechanism to aggregate hidden states across neighboring agents, enabling initial modeling of inter-agent interactions. Social-GAN [7] subsequently incorporated adversarial learning to produce multimodal trajectory distributions, addressing the stochastic nature of human motion. Nevertheless, these methods predominantly employ simple spatial pooling strategies, which lack the representational expressiveness required to model agents with fundamentally diverse behavioral dynamics and interaction semantics. Consequently, their applicability to robot navigation in heterogeneous environments remains limited.

Motivated by these shortcomings, a body of work adopted attention mechanisms and graph-based architectures to capture richer interaction structures. For instance, SoPhie [23] and Social-BiGAT [15] integrated physical scene contexts with social attention for expressive prediction. Social-STGCNN [9] reformulated interactions as spatio-temporal graphs, achieving significant gains in inference efficiency. Trajectron++ [21] introduced dynamic graph structures to handle variable agent counts. AgentFormer [14] used self-attention to jointly model agent-agent and agent-environment dependencies. Despite their improved prediction accuracy, existing methods generally assume similar interaction mechanisms across different agent categories or rely on lightweight semantic embeddings that cannot fully characterize their distinct behavioral and social interaction patterns. In addition, most of these approaches are designed solely for trajectory prediction, rather than jointly reasoning about surrounding-agent trajectory prediction and robot trajectory planning, limiting their applicability to socially-aware robot navigation in heterogeneous environments.

The above limitations suggest that heterogeneous graph learning can provide a promising direction for modeling multi-agent interactions, as it can explicitly distinguish different agent categories and their interaction relationships. For example, Frameworks such as Heterogeneous-GNN [24] and the Heterogeneous Graph Transformer (HGT) [16] provide powerful representation learning capabilities by assigning distinct representations to different node and edge types. However, their application to socially-aware robot navigation remains largely unexplored. Specifically, while recent studies have highlighted the importance of jointly considering task-motion planning and heterogeneous multi-robot coordination in dynamic environments [25], [26], heterogeneous graph learning has rarely been leveraged to jointly reason about surrounding-agent trajectory prediction and robot trajectory planning. To bridge this gap, we develop an HGT-based interaction modeling framework that learns rich semantic, social, and kinematic representations of heterogeneous agents, providing a socially-aware scene representation that supports both surrounding-agent trajectory prediction and downstream robot trajectory planning.

B. Diffusion-Based Trajectory Generation for Robot Navigation

Diffusion probabilistic models [18], [27] have recently emerged as a powerful generative paradigm by formulating data generation as an iterative denoising process, enabling flexible and controllable trajectory synthesis. In robotics, Diffuser [28] and Decision Diffuser [29] reformulated trajectory planning as a conditional diffusion process for long-horizon decision making. For trajectory generation, Motion Indeterminacy Diffusion (MID) [19] first exploited diffusion models to capture the uncertainty of human motion, while MotionDiffuser [30] extended this framework to controllable multi-agent trajectory prediction. SafeDiffuser [31] further incorporated scene constraints to generate collision-free trajectories, and guided diffusion has also been explored for controllable traffic simulation [32]. Beyond trajectory forecasting, diffusion models have also been adopted for interactive robot navigation [33]. Collectively, these studies demonstrate that diffusion models provide a powerful and flexible framework for controllable trajectory generation.

Despite these advances, existing diffusion-based navigation methods primarily focus on enforcing hard physical constraints, such as collision avoidance through scene constraints or classifier-guided sampling [31], [34], while providing only limited support for modeling the soft social interactions required in heterogeneous human-robot environments. In particular, they lack explicit mechanisms for incorporating interpersonal comfort distances, category-dependent yielding behaviors, and other socially-aware interaction principles during trajectory generation, although robust trajectory generation remains an active research topic in cybernetic systems [35]. Moreover, in most existing approaches, these behavioral constraints must be implicitly learned during training, making them difficult to modify or extend once the model has been trained. To address these limitations, we leverage the controllable sampling property of diffusion models and develop a differentiable safety-social energy guidance framework that explicitly injects physical and social objectives into the diffusion sampling process. Specifically, collision avoidance, kinematic feasibility, task progress, and role-dependent social norms are formulated as differentiable energy functions, allowing their gradients to iteratively guide the denoising process. Consequently, our proposed framework favors physically feasible and socially compliant robot trajectories while enabling flexible, training-free, and plug-and-play incorporation of new objectives without retraining the underlying diffusion model.

C. Social Norm Modeling and Human-Aware Robot Navigation

Socially-aware navigation aims to enable robots to navigate while respecting implicit human social norms, such as maintaining appropriate interpersonal distances and avoiding behaviors that cause discomfort or disrupt ongoing interactions [6], [36]. Early work in this domain relied on handcrafted behavioral models: the Social Force Model (SFM) [4] represented pedestrian interactions through attractive and repulsive forces, while ORCA [10] achieved efficient collision avoidance through reciprocal velocity optimization, albeit often producing overly conservative and mechanically rigid robot behaviors. More recently, data-driven approaches have learned navigation policies directly from demonstrations or interactions. For example, deep reinforcement learning (DRL) methods such as SARL [37] optimize socially-aware navigation policies through reward-driven learning, while interactive model predictive control (MPC) has been investigated for navigation in dense crowds [3]. Recent studies have further emphasized richer social reasoning by incorporating group behavior and proxemics (i.e., interpersonal comfort zones) [17], and Transformer-based approaches such as Social-Transmotion [38] have improved promptable human trajectory prediction. Moreover, recent evaluation studies have highlighted that socially-aware navigation should be assessed not only by trajectory accuracy, but also by safety, comfort, and social-compliance metrics [39]–[41].

Despite these advances, existing socially-aware navigation methods primarily focus on learning socially plausible behaviors through policy optimization or trajectory prediction, rather than explicitly incorporating social principles into the trajectory generation process. Consequently, social norms are often implicitly encoded in learned model parameters, making them difficult to interpret, modify, or adapt to different environments and interaction contexts. Moreover, most existing approaches emphasize surrounding-agent trajectory prediction, with limited integration of downstream robot trajectory planning within a unified generative framework. To address these limitations, we formulate physical safety objectives and social interaction principles, including interpersonal comfort distances and category-dependent social margins, as differentiable energy functions that directly guide diffusion-based trajectory generation. By injecting their gradients into the diffusion sampling process, SAGE favors physical feasibility and social compliance in a training-free and controllable manner, avoiding both opaque black-box supervision and rigid rule-based engineering.

Beyond human-robot interactions, multi-robot coordination has also been extensively studied, including switching-topology coordination [42], distributed collision and deadlock avoidance [11], swarm formation control [43], time-constrained task allocation [44], and learning-based flocking with repulsive interactions [45]. While these studies provide important foundations for coordinated robot behavior, they typically treat robot-robot coordination separately from human-robot interaction modeling. In contrast, SAGE unifies these two perspectives by representing robot-robot and robot-entity interactions within the same heterogeneous graph and regulating them through a common safety-social energy guidance mechanism, enabling coordinated, socially-aware navigation for heterogeneous multi-agent systems.

III. Core Models and Problem Formulation

In heterogeneous urban environments characterized by dynamically evolving human-machine interactions, robot navigation necessitates the tight coupling of heterogeneous-entity trajectory prediction and socially-aware robot trajectory planning. We formulate this problem under dynamic multi-agent interactions, where each robot, i.e., EAA, is assigned a navigation task such as moving from its current/start state toward an assigned waypoint while satisfying both physical feasibility and social compliance. Meanwhile, surrounding heterogeneous entities, including PHAs, HNAAs, and SSAAs, exhibit independently evolving and uncontrollable behaviors, requiring the robot to continuously anticipate their future motions and interaction patterns for reliable navigation[1]. For clarity, we use *agents* as a general term for all participants, *robots* to denote controllable agents, *entities* to describe heterogeneous non-robot participants, and *neighbors* to represent entities within the perception neighborhood of a specific robot. As illustrated in Fig. 1, SAGE, which is a multi-agent social perception and navigation framework, consists of the following three synergistic phases:

- *Phase 1: Heterogeneous scene encoding and interaction modeling.* Unlike traditional trajectory planning methods that treat agents homogeneously, this phase integrates neighbors' historical trajectories with semantic and kinematic attributes (such as pedestrian vulnerability and vehicle motion characteristics, etc.). Through a directed heterogeneous interaction graph, the framework captures asymmetric interaction dynamics and implicit social hierarchies, enabling socially aware navigation behaviors, including pedestrian yielding and predictive vehicle interaction. The resulting scene representation, which utilizes HGT, provides structured social semantics for downstream robot trajectory generation.
- *Phase 2: Joint trajectory generation via conditional diffusion.* Given the inherent uncertainty in future entity behaviors, this phase does not produce a single deterministic trajectory. Instead, we leverage a conditional diffusion model to generate a multimodal joint trajectory distribution that simultaneously encompasses both the predicted future behaviors of heterogeneous entities and the preliminary task-directed trajectory plans of robots. This establishes a tightly coupled prediction-planning paradigm, capturing robot-environment interaction dynamics at a distributional level.
- *Phase 3: Social norm-guided trajectory refinement.* Phase 2 samples are statistically plausible but may violate safety, social, or task-progress requirements. Phase 3 encodes these requirements as differentiable energies and applies a bounded, robot-only correction at each denoising step without retraining.

The remainder of this section defines agents and trajectories (Secs. III-A–III-B), introduces perception neighborhoods and kinematic quantities (Sec. III-C), and formulates the joint prediction–planning objective and safety-social constraints (Sec. III-D).

[1] In the problem description throughout the text, we use *navigation* to denote the overall task, *prediction* to describe the future trajectories of entities, *trajectory planning* to describe the trajectory-planning problem for the main robots, and *trajectory generation* to describe the diffusion-based solution process.

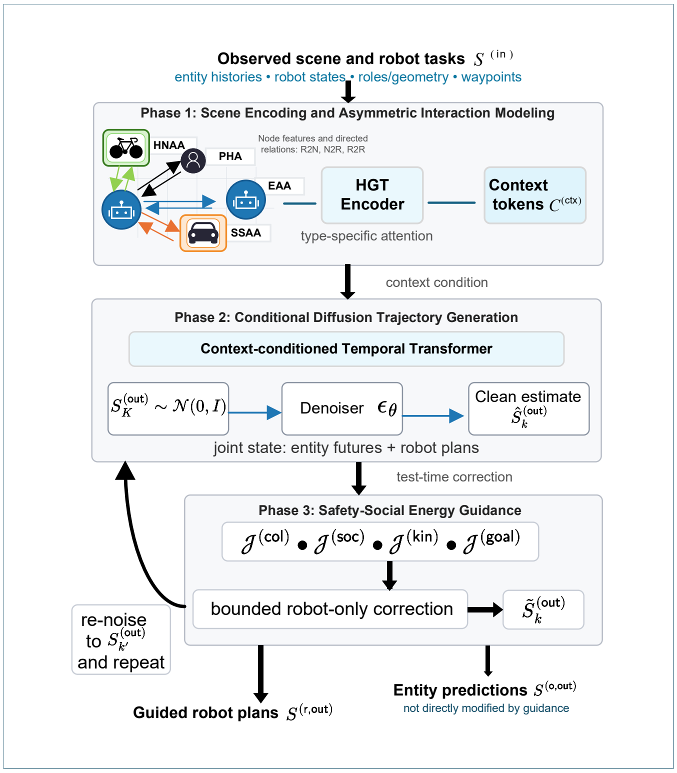


Fig. 1. Overview of SAGE inference. Phase 1 encodes scene histories, agent attributes, and robot waypoints into context tokens through a directed heterogeneous graph and HGT. At each reverse step, Phase 2 estimates clean joint robot–entity trajectories, and Phase 3 applies bounded robot-only energy guidance before re-noising. Iteration yields robot trajectory plans and entity trajectory predictions.

A. Agent Modeling in Heterogeneous Environments

We consider two key agent types: *(i) environmental entities* such as heterogeneous PHAs, HNAAs, and SSAAs, which are dynamic and uncontrollable, gathered by set $\mathcal{O} = \{o_1, \ldots, o_i, \ldots, o_{|\mathcal{O}|}\}$; and *(ii) robots* that are falling in the category of EAAs, collected by set $\mathcal{R} = \{r_1, \ldots, r_j, \ldots, r_{|\mathcal{R}|}\}$. For each entity o_i, we collect its semantic and geometric attributes in the tuple

$$\boldsymbol{a}_i^{(\mathrm{o})} = \left(\mathcal{C}_i^{(\mathrm{o,role})}, \mathcal{C}_i^{(\mathrm{o,geom})}\right). \tag{1}$$

where $\mathcal{C}_i^{(\mathrm{o,role})} \in \mathcal{C}^{(\mathrm{role})} \setminus \{\mathrm{robot}\}$ represents the semantic category (e.g., $\mathcal{C}^{(\mathrm{role})} = \{\mathrm{pedestrian}, \mathrm{cyclist}, \mathrm{vehicle}, \ldots, \mathrm{robot}\}$), describing its behavioral patterns and social roles; and $\mathcal{C}_i^{(\mathrm{o,geom})} \in \mathbb{R}^2$ represents the geometric properties (i.e., length and width) that define physical dimensions and collision boundaries. All robots share fixed geometric dimensions $\mathcal{C}^{(\mathrm{r,geom})}$ and kinematic constraints, while each robot r_j is assigned a waypoint $\mathbf{g}_j = (x_j^{(\mathrm{g})}, y_j^{(\mathrm{g})})$ as its navigation objective within the current prediction horizon $T^{(\mathrm{prd})}$. To standardize with the interaction representation, we assign all robots the same semantic category $\mathcal{C}^{(\mathrm{r,role})} = \{\mathrm{robot}\}$.

B. Trajectory Modeling over Observation and Prediction Horizons

At each planning instant, SAGE processes a single observation window of length $T^{(\mathrm{obs})}$ and generates joint trajectories over a prediction horizon of length $T^{(\mathrm{prd})}$. Using a relative time index, $t = 0$ denotes the current planning time, the observed time steps are $t \in \{-T^{(\mathrm{obs})} + 1, \ldots, 0\}$, and the future time steps are $t \in \{1, \ldots, T^{(\mathrm{prd})}\}$. The same procedure can be repeatedly invoked as new observations become available, while each planning instance is formulated independently. For entities in $\mathcal{O}$, we use the observed planar position $\mathbf{p}_i^{(\mathrm{o},t)} = (x_i^{(\mathrm{o},t)}, y_i^{(\mathrm{o},t)})$ to construct the position history $\boldsymbol{s}_i^{(\mathrm{past})} = \{\mathbf{p}_i^{(\mathrm{o},t)} \mid t \in \{-T^{(\mathrm{obs})} + 1, \ldots, 0\}\}$ supplied to the entity LSTM. Velocity components can be obtained by finite differences when required. For each robot, the current Cartesian state is $\boldsymbol{s}_j^{(\mathrm{r},0)} = (x_j^{(\mathrm{r},0)}, y_j^{(\mathrm{r},0)}, v_j^{(\mathrm{rx},0)}, v_j^{(\mathrm{ry},0)})$, where $x_j^{(\mathrm{r},0)}$ and $y_j^{(\mathrm{r},0)}$ denote its current planar position coordinates, and $v_j^{(\mathrm{rx},0)}$ and $v_j^{(\mathrm{ry},0)}$ denote the corresponding velocity components along the x- and y-axes, estimated from the last two observed positions. Each robot r_j aims to generate a trajectory from this current state toward its assigned waypoint $\mathbf{g}_j$ over $T^{(\mathrm{prd})}$, subject to safety and social-interaction requirements.

C. Perception Neighborhoods and Kinematic Derivatives

To facilitate downstream constraint modeling, we formalize the robot's local perception neighborhood and relevant kinematic quantities. Let $\mathbf{p}_j^{(\mathrm{r},t)} = (x_j^{(\mathrm{r},t)}, y_j^{(\mathrm{r},t)})$ and $\mathbf{p}_i^{(\mathrm{o},t)} = (x_i^{(\mathrm{o},t)}, y_i^{(\mathrm{o},t)})$ denote the planar positions of robot r_j and neighbor o_i, respectively.

(i) Heterogeneous-aware neighborhood. In dense and highly populated real-world environments, exhaustively modeling all entities in $\mathcal{O}$ incurs substantial computational overhead and may introduce redundant or weakly relevant interaction dependencies that degrade representation quality. Therefore, for each robot $r_j \in \mathcal{R}$ at time t, we define its effective perception neighborhood as the set of nearby entities within a pair-dependent interaction radius $R_{j,i}^{(\mathrm{val})}$:

$$\mathcal{N}_j^t = \left\{ o_i \;\middle|\; o_i \in \mathcal{O}, \; \left\| \mathbf{p}_j^{(\mathrm{r},t)} - \mathbf{p}_i^{(\mathrm{o},t)} \right\|_2 \le R_{j,i}^{(\mathrm{val})} \right\}. \tag{2}$$

where $R_{j,i}^{(\mathrm{val})}$ is a pair-dependent interaction radius jointly constrained by two factors: the sensing range of robot r_j, which is bounded by its maximum observable range, and the category-dependent social influence range of neighbor o_i, which depends on its semantic role, motion state, and geometric properties. For instance, for pedestrians, we use Hall's proxemics theory [17] as a reference (e.g., the boundary between the social and public zones is approximately $3.6\,\mathrm{m}$); beyond this range, interpersonal social influence is typically weakened. For other entity categories, such as cyclists or vehicles, the interaction radius can be adjusted according to their speed, size, and safety-relevant motion characteristics[2].

(ii) Kinematic derivatives. To characterize the nonholonomic constraints of a robot, we define three derived quantities from its state vector in the following. Let $\mathbf{v}_j^t = (v_j^{(\mathrm{rx},t)}, v_j^{(\mathrm{ry},t)})$ denote the planar translational velocity of robot r_j at time

[2]Note that, $\mathcal{N}_j^t$ contains only heterogeneous non-robot neighbors observed by robot r_j at time t, while nearby robots are later modeled separately through R2R interactions.

t. First, the robot's heading angle at time t is computed from its velocity:

$$\theta_j^t = \operatorname{atan2}\Big(v_j^{(\mathsf{ry},t)}, v_j^{(\mathsf{rx},t)} + \varepsilon\Big), \tag{3}$$

Then, the lateral velocity is the component of the current velocity perpendicular to the preceding heading:

$$v_j^{(\mathsf{lat},t)} = -v_j^{(\mathsf{rx},t)} \sin(\theta_j^{t-1}) + v_j^{(\mathsf{ry},t)} \cos(\theta_j^{t-1}), \tag{4}$$

Finally, the angular velocity uses the wrapped heading difference:

$$\omega_j^t = \begin{cases} 0, & t = 1, \\ \dfrac{\operatorname{atan2}(\sin\Delta\theta_j^t, \cos\Delta\theta_j^t)}{\Delta t}, & t \geq 2, \end{cases} \tag{5}$$

where $\Delta\theta_j^t = \theta_j^t - \theta_j^{t-1}$, $\varepsilon > 0$ is a numerical stabilizer, and Δt is the sampling interval in seconds (set to $0.4\,\mathrm{s}$ in our experiments). For $t = 1$, θ_j^0 estimated from the last two observations is used as the reference for $v_j^{(\mathsf{lat},1)}$.
For an ideal differential-drive robot, lateral motion should remain small and the translational and angular velocities should remain within their prescribed limits.

D. Optimization Objective and Safety-Social Compliance Constraints

Next, we formulate the navigation problem of our interest, with two coupled components. *First*, we define a joint generative objective that maps current robot states, assigned waypoints, and historical observations of neighbor entities to a distribution over predicted entity trajectories and task-directed robot trajectory plans. *Second*, we impose task and safety-social feasibility constraints on samples from this distribution, including task progress, collision avoidance, kinematic feasibility, and social compliance. We begin by specifying the joint generative objective.

• *Joint generative objective.* Given the robot states at the current planning time $t = 0$ and their assigned waypoints, together with the historical observations of heterogeneous entities, let $\mathcal{O}_{\mathcal{N}}^0 = \bigcup_{r_j \in \mathcal{R}} \mathcal{N}_j^0$ denote the set of neighbor entities currently perceived by the robot team. We define the scene-level conditioning input as

$$S^{(\mathsf{in})} := \left\{ \bigcup_{r_j \in \mathcal{R}} s_j^{(\mathsf{r},0)}, \bigcup_{r_j \in \mathcal{R}} \mathbf{g}_j, \bigcup_{o_i \in \mathcal{O}_{\mathcal{N}}^0} \boldsymbol{s}_i^{(\mathsf{past})} \right\}. \tag{6}$$

Here, robot-side conditioning is restricted to the current robot state and assigned waypoint, while trajectory-history inputs are used only for heterogeneous neighbor entities. The diffusion target is the joint future velocity sequence $S^{(\mathsf{out})} = \{S^{(\mathsf{r},\mathsf{out})}, S^{(\mathsf{o},\mathsf{out})}\}$, where $S^{(\mathsf{r},\mathsf{out})}$ contains robot plan velocities and $S^{(\mathsf{o},\mathsf{out})}$ contains predicted entity velocities over $T^{(\mathsf{prd})}$. Positions used for navigation and energy evaluation are recovered by $\mathbf{p}_a^t = \mathbf{p}_a^0 + \Delta t \sum_{\tau=1}^{t} \mathbf{v}_a^\tau$ for each robot or entity a. Note that trajectory prediction and planning are fundamentally coupled. Without anticipating neighboring agents, navigation reduces to open-loop planning in a static environment and cannot account for the reciprocal, evolving dynamics of multi-agent interactions. We therefore formulate them as a unified problem and use a conditional generative model to parameterize the joint trajectory distribution $p_\theta(S^{(\mathsf{out})} \mid S^{(\mathsf{in})})$, where θ denotes the learnable parameters of the scene encoder and the diffusion denoising network. The model parameters θ are learned by minimizing the diffusion reconstruction objective $\mathcal{L}_{\mathsf{diff}}$ (detailed in Sec. IV-B).

• *Safety-social constrained trajectory planning.* Although samples from the learned conditional joint trajectory distribution $p_\theta(S^{(\mathsf{out})} \mid S^{(\mathsf{in})})$ are statistically plausible, they may still fail to make task progress or violate safety, kinematic, or social requirements, especially in dense or long-tail interaction scenarios. Hence, we seek robot trajectories that are both likely under this distribution and feasible with respect to task, physical, and social constraints, leading to our overarching problem formulation:

$$\mathcal{P}: \quad S^{(\mathsf{out})} \sim p_\theta\Big(\cdot \mid S^{(\mathsf{in})}\Big), \quad \text{s.t.} \quad \text{(C1)–(C5)}. \tag{7}$$

Specifically, constraints (C1)-(C5) are detailed below.
(C1) Robot–neighbor collision avoidance. Each robot should maintain its pairwise safety margin from neighbors:

$$\begin{gathered} \left\| \mathbf{p}_j^{(\mathsf{r},t)} - \mathbf{p}_i^{(\mathsf{o},t)} \right\|_2 \geq m_{j,i}, \\ \forall t \in \{1, \ldots, T^{(\mathsf{prd})}\},\ \forall r_j \in \mathcal{R},\ \forall o_i \in \mathcal{N}_j^t. \end{gathered} \tag{8}$$

(C2) Robot-to-robot (R2R) collision avoidance. Any pair of robots should preserve their pairwise safety margin:

$$\begin{gathered} \left\| \mathbf{p}_j^{(\mathsf{r},t)} - \mathbf{p}_{j'}^{(\mathsf{r},t)} \right\|_2 \geq m_{j,j'}, \\ \forall t \in \{1, \ldots, T^{(\mathsf{prd})}\},\ \forall r_j, r_{j'} \in \mathcal{R},\ j' \neq j. \end{gathered} \tag{9}$$

Here, $d^{(\mathsf{safe})}$ is the base clearance threshold. When geometry-aware margins are enabled, $\mathcal{C}_a^{(\mathsf{geom})}$ denotes the length–width tuple of agent a, and $\frac{1}{2}\|\mathcal{C}_a^{(\mathsf{geom})}\|_2$ is the radius of its circumscribed circle; thus, for any interacting pair a, b, $m_{a,b} = d^{(\mathsf{safe})} + \frac{1}{2}\|\mathcal{C}_a^{(\mathsf{geom})}\|_2 + \frac{1}{2}\|\mathcal{C}_b^{(\mathsf{geom})}\|_2$. Otherwise, agents are treated as points for margin calculation and $m_{a,b} = d^{(\mathsf{safe})}$.
(C3) Kinematic feasibility. The generated robot trajectories should satisfy the prescribed kinematic limits:

$$\begin{gathered} \|\mathbf{v}_j^t\|_2 \leq v_{\max}, \quad |v_j^{(\mathsf{lat},t)}| \leq v_{\max}^{\mathsf{lat}}, \quad |\omega_j^t| \leq \omega_{\max}, \\ \forall t \in \{1, \ldots, T^{(\mathsf{prd})}\},\ \forall r_j \in \mathcal{R}. \end{gathered} \tag{10}$$

(C4) Social compliance. Beyond physical collision avoidance, robot trajectories should remain outside the category-conditioned social compliance region induced by neighbors in the robot's forward half-plane:

$$\begin{gathered} \left(\frac{d_{i,j}^{(\mathsf{lon},t)}}{\sigma^{\mathsf{lon}}\Big(\mathcal{C}_i^{(\mathsf{o},\mathsf{role})}\Big)} \right)^2 + \left(\frac{d_{i,j}^{(\mathsf{lat},t)}}{\sigma^{\mathsf{lat}}\Big(\mathcal{C}_i^{(\mathsf{o},\mathsf{role})}\Big)} \right)^2 \geq 1, \\ \forall t \in \{1, \ldots, T^{(\mathsf{prd})}\},\ \forall r_j \in \mathcal{R},\ \forall o_i \in \mathcal{N}_j^t. \end{gathered} \tag{11}$$

Here, $d_{i,j}^{(\mathsf{lon},t)}$ and $d_{i,j}^{(\mathsf{lat},t)}$ can be obtained by projecting the relative position from robot r_j to neighbor o_i into the robot-centric frame:

$$\begin{bmatrix} d_{i,j}^{(\mathsf{lon},t)} \\ d_{i,j}^{(\mathsf{lat},t)} \end{bmatrix} = \begin{bmatrix} \cos(\theta_j^t) & \sin(\theta_j^t) \\ -\sin(\theta_j^t) & \cos(\theta_j^t) \end{bmatrix} \begin{bmatrix} x_i^{(\mathsf{o},t)} - x_j^{(\mathsf{r},t)} \\ y_i^{(\mathsf{o},t)} - y_j^{(\mathsf{r},t)} \end{bmatrix}. \tag{12}$$

The functions $\sigma^{\mathsf{lon}}(\cdot)$ and $\sigma^{\mathsf{lat}}(\cdot)$ define the role-conditioned semi-axes of the elliptical comfort region. In words, for each forward neighbor, (11) requires the robot to remain on or outside this ellipse. The right-hand side is one because the separations are normalized by these semi-axes: values below one indicate intrusion, whereas values at or above one satisfy the constraint; any other positive threshold would merely rescale the margins. Shared category-level margins provide an interpretable social prior, with larger pedestrian margins reflecting interpersonal comfort and tighter vehicle margins primarily reflecting safety.

(C5) Task progress. The generated robot trajectory should reach the assigned waypoint $\mathbf{g}_j$ or terminate sufficiently close to it within the planning horizon:

$$\left\| \mathbf{p}_j^{(\mathsf{r}, T^{(\mathsf{prd})})} - \mathbf{g}_j \right\|_2 \leq d^{(\mathsf{goal})}, \quad \forall r_j \in \mathcal{R}. \tag{13}$$

Here, $d^{(\mathsf{goal})}$ is the tolerance for reaching the assigned waypoint within the prediction horizon.

The constrained formulation $\mathcal{P}$ specifies both the generative target and the task-feasibility requirements of socially compliant robot navigation. Directly solving $\mathcal{P}$ is fundamentally challenging due to the intricate coupling among multimodal entity trajectory prediction and robot trajectory planning, heterogeneous interaction reasoning, collision avoidance, kinematic feasibility, and socially compliant behavior modeling. These are inherently interdependent and often exhibit competing constraints across spatial, temporal, and social dimensions. To address this complexity, we develop *SAGE*, a structured guided generative framework.

Algorithm 1: *SAGE*

Input: Scene-level conditioning input $S^{(\mathsf{in})}$, robot and entity attributes, perception radius $R_{j,i}^{(\mathsf{val})}$, diffusion steps K, sampling stride s

Output: Robot trajectory plans $S^{(\mathsf{r,out})}$ and entity trajectory predictions $S^{(\mathsf{o,out})}$

Construct the heterogeneous graph $\mathcal{G}$ with R2N, N2R, and R2R edges following (14);

Encode node features and obtain context tokens $C^{(\mathsf{ctx})}$ following (15)–(20);

Initialize $S_K^{(\mathsf{out})} \sim \mathcal{N}(0, I)$ according to (21);

for $k = K, K-s, \ldots$ *while* $k > 0$ **do**

  Predict the noise $\epsilon_\theta(S_k^{(\mathsf{out})}, \beta_k, C^{(\mathsf{ctx})})$ as in (23);

  Estimate the clean velocity sequence $\hat{S}_k^{(\mathsf{out})}$ following (23);

  Compute the four energy terms following (24)–(28);

  Apply the bounded robot-only correction following (29);

  Set $k' = \max(k-s, 0)$ and sample $S_{k'}^{(\mathsf{out})}$ following (29);

return $S^{(\mathsf{r,out})}$ *as robot trajectory plans and* $S^{(\mathsf{o,out})}$ *as entity trajectory predictions*

IV. SAGE: A Three Phase Framework

Building on Sec. III-D, this section presents how *SAGE* parameterizes and subsequently samples from the task-aware generative prior $p_\theta(S^{(\mathsf{out})} \mid S^{(\mathsf{in})})$. During training, *SAGE* learns this prior to capture the multimodal joint future distribution of controllable robots and heterogeneous entities within their perception neighborhoods. During inference, samples from this prior are further refined by safety-social energy guidance to promote task progress, collision avoidance, kinematic feasibility, and social compliance while remaining likely under the learned distribution. Specifically, SAGE consists of three synergistic phases. First, an HGT encodes asymmetric robot-entity interactions into a contextual representation $C^{(\mathsf{ctx})}$ that captures social semantics and interaction hierarchies (*Phase 1*). Second, conditioned on $C^{(\mathsf{ctx})}$, a diffusion model jointly generates robot and entity trajectories, enabling coherent reasoning over coupled multi-agent dynamics (*Phase 2*). Third, differentiable safety-social energy guidance is integrated into the denoising process to reduce physical-safety and social-compliance energy penalties (*Phase 3*). The overall inference procedure is summarized in Algorithm 1.

A. Phase 1: Scene Encoding and Asymmetric Interaction Modeling

This phase encodes the heterogeneous scene described in Sec. III-A as a context-token sequence $C^{(\mathsf{ctx})}$, providing the diffusion model with structured awareness of entity identities, categories, and social interaction patterns. To this end, as we will explain in the following, the scene is formulated as a directed heterogeneous graph, in which information passing is conducted via a tailored HGT to yield socially enriched node embeddings. Robot-centered interaction relationships are thus captured within this directed heterogeneous graph representation:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \tag{14}$$

where $\mathcal{V} = \mathcal{R} \cup \mathcal{O}_{\mathcal{N}}^0$ uses the team-level perceived entity set defined in Sec. III-D; under full-observability simulation, all active entities are treated as perceived. The encoder connects every robot–entity pair in this set through directed robot-to-neighbor (R2N) and neighbor-to-robot (N2R) edges, and all ordered robot pairs, including robot self-attention, through robot-to-robot (R2R) edges. We next construct the context-token sequence in three steps.

• *Step 1. Node feature embedding.* Before enabling information sharing across $\mathcal{G}$, we construct a layer-0 embedding for each node. Because entity motion must be inferred from observation histories whereas robot plans are conditioned on current states and assigned waypoints, we use an LSTM for entities and a state MLP for robots, and fuse shared role, control-status, geometry, and waypoint features into a common latent space for subsequent HGT interaction modeling. Let Φ_{fuse} denote the implemented fusion projection, which applies layer normalization and an MLP after concatenation ($\oplus$). For each $o_i \in \mathcal{O}_{\mathcal{N}}^0$, the initial embedding is

$$\begin{aligned} \mathbf{h}_i^{(\mathsf{o},0)} &= \Phi_{\mathsf{fuse}}\big(\mathrm{LSTM}(\boldsymbol{s}_i^{(\mathsf{past})}) \oplus \mathrm{Emb}_{\mathsf{role}}(\mathcal{C}_i^{(\mathsf{o,role})}) \\ &\oplus \mathrm{Emb}_{\mathsf{ctrl}}(0) \oplus \mathrm{MLP}_{\mathsf{geom}}(\mathcal{C}_i^{(\mathsf{o,geom})}) \oplus \mathrm{MLP}_{\mathsf{goal}}(\mathbf{0})\big), \end{aligned} \tag{15}$$

where a zero-initialized LSTM recursively maps each observed position and its preceding hidden/cell states to the next states, with its final hidden state $\mathbf{h}_{i,0}$ used as $\mathrm{LSTM}(\boldsymbol{s}_i^{(\mathsf{past})})$. The learned role embedding represents semantic identity, $\mathrm{Emb}_{\mathsf{ctrl}}(0)$ marks an uncontrollable entity, the geometry MLP encodes physical dimensions, and the zero waypoint displacement masks task conditioning for entity nodes. The LSTM, role/control embeddings, feature MLPs, fusion projection, HGT, and diffusion denoiser are jointly optimized through $\mathcal{L}_{\mathsf{diff}}$. For a robot $r_j \in \mathcal{R}$, we define its initial node embedding as $\mathbf{h}_j^{(\mathsf{r},0)}$:

$$\begin{aligned}\mathbf{h}_j^{(\mathsf{r},0)} = \Phi_{\mathsf{fuse}}\big(&\mathrm{MLP}_{\mathsf{state}}(\boldsymbol{s}_j^{(\mathsf{r},0)}) \oplus \mathrm{Emb}_{\mathsf{ctrl}}(1)\\ &\oplus \mathrm{Emb}_{\mathsf{role}}(\mathcal{C}^{(\mathsf{r},\mathsf{role})}) \oplus \mathrm{MLP}_{\mathsf{geom}}(\mathcal{C}^{(\mathsf{r},\mathsf{geom})})\\ &\oplus \mathrm{MLP}_{\mathsf{goal}}\Big(\mathbf{g}_j - \mathbf{p}_j^{(\mathsf{r},0)}\Big)\big),\end{aligned} \tag{16}$$

where the state MLP encodes the robot's current position and velocity, the control embedding marks a controllable node, and the goal MLP encodes its waypoint displacement. These layer-0 embeddings integrate motion information, semantic identity, physical attributes, and robot task intent, and serve as the inputs to the relation-specific HGT in Step 2.

• *Step 2. Meta-relation-based attention mechanism.* To model asymmetric role-specific interactions that a standard shared-parameter GAT [46] does not explicitly parameterize, we assign HGT parameters according to ordered source–target role pairs:

$$\psi(u_b \to u_a) = \iota(u_b)N_{\mathsf{role}} + \iota(u_a), \tag{17}$$

where $\iota(u) \in \{0,\dots,N_{\mathsf{role}}-1\}$ is the semantic-role ID and N_{role} is the number of roles; thus, opposite directions receive different relation IDs. One HGT layer performs relation-specific attention, neighborhood aggregation, and node-state update. At layer $l \in \{1,\dots,L\}$, target u_a receives sources from $\mathcal{M}_a = \{u_b \in \mathcal{V} \mid (u_b \to u_a) \in \mathcal{E}\}$, with $\psi = \psi(u_b \to u_a)$. Each relation has head-specific projections $(W^{\mathsf{Q}}_{\psi,h}, W^{\mathsf{K}}_{\psi,h}, W^{\mathsf{V}}_{\psi,h})$ and logit biases $b_{\psi,h}$, together with a relation gate $\boldsymbol{\gamma}_\psi$; their layer indices are omitted for brevity. The compatibility logit and its normalized attention weight are

$$\begin{aligned}e^{(l)}_{a,b,h} &= \frac{(\mathbf{h}_a^{(l-1)}W^{\mathsf{Q}}_{\psi,h})(\mathbf{h}_b^{(l-1)}W^{\mathsf{K}}_{\psi,h})^\top}{\sqrt{d_h}} + b_{\psi,h},\\ \alpha^{(l)}_{a,b,h} &= \mathrm{Softmax}_{u_b\in\mathcal{M}_a}\big(e^{(l)}_{a,b,h}\big),\end{aligned} \tag{18}$$

where d_h is the per-head projection dimension. Node u_a then aggregates the projected source representations and applies the relation gate before its state is updated:

$$\begin{aligned}\mathbf{m}^{(l)}_{a,h} &= \sum_{u_b\in\mathcal{M}_a} \alpha^{(l)}_{a,b,h}\mathbf{h}_b^{(l-1)}W^{\mathsf{V}}_{\psi,h},\\ \boldsymbol{\eta}_a^{(l)} &= \mathrm{sigmoid}\left(\frac{1}{\max(1,|\Psi_a|)}\sum_{\psi\in\Psi_a}\boldsymbol{\gamma}_\psi\right),\\ \mathbf{h}_a^{(l)} &= \rho_{\mathsf{HGT}}\left(\mathbf{h}_a^{(l-1)}, W^{\mathsf{O}}\left[\boldsymbol{\eta}_a^{(l)} \odot \big\Vert_{h=1}^{H} \mathbf{m}^{(l)}_{a,h}\right]\right),\end{aligned} \tag{19}$$

where $\Psi_a = \{\psi(u_b \to u_a) \mid u_b \in \mathcal{M}_a\}$, $\Vert$ concatenates heads, and $\odot$ is element-wise multiplication. The mapping ρ_{HGT} applies the implemented residual, layer-normalization, and GELU feed-forward updates. In words, attention selects relevant incoming nodes, relation-specific values determine what they contribute, and $\boldsymbol{\eta}_a^{(l)}$ controls their aggregate influence; repeating the update propagates information over up to L hops.

• *Step 3. Scene context construction.* After L HGT layers, we retain the node tokens and append a normalized mean-pooled graph token:

$$\begin{aligned}\mathbf{c}^{(\mathsf{G})} &= \mathrm{Norm}\left(\Phi_{\mathsf{ctx}}\left(\frac{1}{|\mathcal{V}|}\sum_{u\in\mathcal{V}}\mathbf{h}_u^{(L)}\right)\right),\\ C^{(\mathsf{ctx})} &= \left[\mathbf{h}_{u_1}^{(L)},\dots,\mathbf{h}_{u_{|\mathcal{V}|}}^{(L)},\mathbf{c}^{(\mathsf{G})}\right].\end{aligned} \tag{20}$$

Here, Φ_{ctx} projects the pooled feature into the node-token dimension and Norm denotes ℓ_2 normalization. Thus, $C^{(\mathsf{ctx})}$ provides Phase 2 with agent-specific interaction tokens and a global scene summary. HGT supplies an implicit interaction prior for joint generation, whereas Phase 3 provides explicit constraint-aware correction.

B. Phase 2: Conditional Diffusion Trajectory Generation

Phase 1 maps $S^{(\mathsf{in})}$ to context tokens $C^{(\mathsf{ctx})}$ encoding agent-specific interactions, a global scene summary, and robot waypoints; Phase 2 therefore models $p_\theta\big(S^{(\mathsf{out})} \mid C^{(\mathsf{ctx})}\big)$, so $S^{(\mathsf{in})}$ conditions the future through this learned representation. We adopt denoising diffusion probabilistic models (DDPM) [18] because stochastic denoising represents multiple coupled robot–entity futures and exposes intermediate clean estimates for Phase 3 correction. Joint generation preserves robot–entity dependencies by producing entity velocity predictions and robot plan velocities together. We first describe forward corruption and then present the context-conditioned denoiser, training objective, and reverse sampling process.

• *Forward diffusion process:* To create controlled denoising examples, the forward process progressively corrupts the ground-truth joint future velocity sequence $S^{(\mathsf{out})}$. Its one-step transition and corresponding one-shot sampling form are:

$$\begin{aligned}q\Big(S_k^{(\mathsf{out})} \mid S_{k-1}^{(\mathsf{out})}\Big) &= \mathcal{N}\Big(\sqrt{\alpha_k}S_{k-1}^{(\mathsf{out})}, (1-\alpha_k)I\Big),\\ S_k^{(\mathsf{out})} &= \sqrt{\bar\alpha_k}S^{(\mathsf{out})} + \sqrt{1-\bar\alpha_k}\,\epsilon.\end{aligned} \tag{21}$$

Here, $k = 1,\dots,K$ indexes diffusion steps, β_k is the fixed noise variance at step k, $\alpha_k = 1-\beta_k$, $\bar\alpha_k = \prod_{\kappa=1}^{k}\alpha_\kappa$, and $\epsilon \sim \mathcal{N}(0, I)$. The one-shot form samples $S_k^{(\mathsf{out})}$ directly from $S^{(\mathsf{out})}$ without simulating the intermediate Markov transitions. For sufficiently large k, $S_k^{(\mathsf{out})}$ approaches isotropic Gaussian noise.

• *Reverse denoising process:* During training, $S_k^{(\mathsf{out})}$ denotes the joint robot–entity velocity sequence corrupted to diffusion level k; thus, k is a diffusion index rather than the physical trajectory time t. The Transformer denoiser [14] combines temporal self-attention within each trajectory, agent-wise attention across robot and entity trajectories at each future time step, and cross-attention to $C^{(\mathsf{ctx})}$; these operations model motion evolution, inter-agent coupling, and scene/waypoint conditioning, respectively. Given $(S_k^{(\mathsf{out})}, \beta_k, C^{(\mathsf{ctx})})$, ϵ_θ predicts the added noise. For each training example, we sample k and ϵ, construct $S_k^{(\mathsf{out})}$ using (21), and minimize

$$\mathcal{L}_{\mathsf{diff}} = \mathbb{E}\left[\left\Vert\epsilon - \epsilon_\theta\left(S_k^{(\mathsf{out})}, \beta_k, C^{(\mathsf{ctx})}\right)\right\Vert^2\right] \tag{22}$$

The expectation in (22) averages over training trajectories and sampled k and ϵ; minimizing it teaches the network to denoise robot plans and entity trajectories jointly.

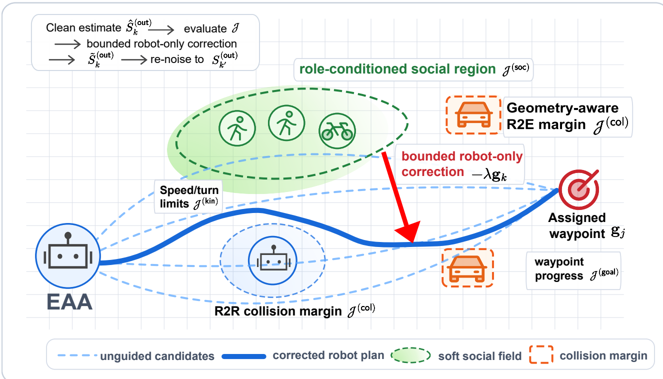


Fig. 2. Illustration of one safety-social guidance step during reverse diffusion. Geometry-aware R2E/R2R collision margins, role-conditioned social regions, kinematic limits, and waypoint progress define the total energy. Its clipped gradient produces a bounded robot-only correction before re-noising to the next diffusion level; entity components are not directly modified by guidance.

At inference, SAGE converts Gaussian noise into a joint future-velocity sample. Starting from $S_K^{(\text{out})} \sim \mathcal{N}(0, I)$, the denoiser first reconstructs a clean estimate at level k. With sampling stride s and next level $k' = \max(k - s, 0)$, the unguided implementation re-noises this estimate at level k':

$$
\begin{aligned}
\hat{S}_k^{(\text{out})} &= \frac{S_k^{(\text{out})} - \sqrt{1-\bar{\alpha}_k}\, \epsilon_\theta(S_k^{(\text{out})}, \beta_k, C^{(\text{ctx})})}{\sqrt{\bar{\alpha}_k}}, \\
S_{k'}^{(\text{out})} &= \begin{cases} \sqrt{\bar{\alpha}_{k'}}\, \hat{S}_k^{(\text{out})} + \sqrt{1-\bar{\alpha}_{k'}}\, z, & k' > 0, \\ \hat{S}_k^{(\text{out})}, & k' = 0, \end{cases}
\end{aligned} \tag{23}
$$

Here, $z \sim \mathcal{N}(0, I)$, $\hat{S}_k^{(\text{out})}$ is the clean joint velocity estimate, and $S_{k'}^{(\text{out})}$ is the less-noisy input to the next step. Repetition to $k' = 0$ yields robot plan and entity-prediction velocities, with different noise draws producing alternative joint futures; Phase 3 modifies only the robot components before re-noising.

C. Phase 3: Safety-Social Energy Guidance

Individual statistically plausible Phase 2 samples need not satisfy the constraints in Section III-D. Following gradient-guided diffusion [30], [34], Phase 3 evaluates positions integrated from each clean velocity estimate with a differentiable energy $\mathcal{J}$, where lower values indicate greater compliance, and corrects only robot components before re-noising in (23); entity predictions remain unchanged. Fig. 2 illustrates the collision, role-conditioned social, and waypoint effects. We first define the four energy components and then detail the implemented correction.

• *Energy function design:* We define $\mathcal{J}$ as the weighted differentiable counterpart of (C1)–(C5), evaluated on integrated robot plans and predicted entity trajectories:

$$
\begin{aligned}
\mathcal{J}(S^{(\text{out})}) = w_{\text{col}} \mathcal{J}^{(\text{col})} + w_{\text{soc}} \mathcal{J}^{(\text{soc})} \\
+ w_{\text{kin}} \mathcal{J}^{(\text{kin})} + w_{\text{goal}} \mathcal{J}^{(\text{goal})}.
\end{aligned} \tag{24}
$$

The nonnegative weights $w_{\text{col}}, w_{\text{soc}}, w_{\text{kin}}, w_{\text{goal}}$ set the relative contributions of collision avoidance, social compliance, kinematic regularity, and task progress.

(i) Physical anti-collision potential energy ($\mathcal{J}^{(\text{col})}$): For pairwise distance d and margin m, the kernel $\phi_{\text{col}}(d; m) = \exp[-d^2/(2\sigma_{\text{col}}^2)] + \text{ReLU}(m-d)^2$ combines smooth repulsion with a stronger barrier inside the margin. Summing it once over each robot–entity and unordered robot pair gives

$$
\begin{aligned}
\mathcal{J}^{(\text{col})} = \sum_{t=1}^{T^{(\text{prd})}} \Bigg[\sum_{r_j \in \mathcal{R}} \sum_{o_i \in \mathcal{O}_{\mathcal{N}}^0} \phi_{\text{col}}(d_{j,i}^t; m_{j,i}) \\
+ w_{\text{R2R}} \sum_{j<j'} \phi_{\text{col}}(d_{j,j'}^t; m_{j,j'}) \Bigg].
\end{aligned} \tag{25}
$$

Here, $d_{a,b}^t = \|\mathbf{p}_a^t - \mathbf{p}_b^t\|_2$, w_{R2R} controls R2R repulsion, and $m_{a,b}$ is the pairwise safety margin defined in (C1)–(C2).

(ii) Heterogeneous anisotropic social potential ($\mathcal{J}^{(\text{soc})}$): Using the margins from (11), let $\boldsymbol{\delta}_{i,j}^t = (d_{i,j}^{(\text{lon},t)}/\sigma_i^{\text{lon}}, d_{i,j}^{(\text{lat},t)}/\sigma_i^{\text{lat}})$ denote normalized relative position, where $\sigma_i^q = \sigma^q(\mathcal{C}_i^{(\text{o,role})})$. The front-oriented potential is

$$
\begin{aligned}
\mathcal{J}^{(\text{soc})} = \sum_{t=1}^{T^{(\text{prd})}} \sum_{r_j \in \mathcal{R}} \sum_{o_i \in \mathcal{O}_{\mathcal{N}}^0} \mathbb{I}\Big[d_{i,j}^{(\text{lon},t)} \geq 0\Big] \\
\times \exp\big(-\|\boldsymbol{\delta}_{i,j}^t\|_2^2\big).
\end{aligned} \tag{26}
$$

(iii) Kinematic potential energy ($\mathcal{J}^{(\text{kin})}$): Using $[x]_+ = \text{ReLU}(x)$, the differentiable counterpart of (C3) is

$$
\begin{aligned}
\mathcal{J}^{(\text{kin})} = \sum_{t=1}^{T^{(\text{prd})}} \sum_{r_j \in \mathcal{R}} \Big[[\|\mathbf{v}_j^t\|_2 - v_{\max}]_+^2 \\
+ [|v_j^{(\text{lat},t)}| - v_{\max}^{\text{lat}}]_+^2 + [|\omega_j^t| - \omega_{\max}]_+^2 \Big].
\end{aligned} \tag{27}
$$

(iv) Task-progress potential energy ($\mathcal{J}^{(\text{goal})}$): The differentiable counterpart of (C5) penalizes terminal waypoint error:

$$
\mathcal{J}^{(\text{goal})} = \sum_{r_j \in \mathcal{R}} \Big\| \mathbf{p}_j^{(\text{r},T^{(\text{prd})})} - \mathbf{g}_j \Big\|_2^2. \tag{28}
$$

Thus, $\mathcal{J}^{(\text{col})}$ separates robots from entities and other robots, $\mathcal{J}^{(\text{soc})}$ enforces role-conditioned comfort regions, $\mathcal{J}^{(\text{kin})}$ suppresses velocity-limit violations, and $\mathcal{J}^{(\text{goal})}$ draws terminal positions toward assigned waypoints.

• *Guided correction algorithm design:* At level k, SAGE evaluates $\mathcal{J}$ on the clean estimate $\hat{S}_k^{(\text{out})}$ from (23). Let $\text{Clip}_{\text{grad}}$ and $\text{Clip}_{\text{step}}$ denote trajectory-wise norm clipping of the raw gradient and scaled update, and let Π_{vel} cap robot speeds at $\gamma_{\text{vel}} v_{\max}$ ($\gamma_{\text{vel}} = 1.25$) without changing entity components. The guided update is

$$
\begin{aligned}
\mathbf{g}_k &= \text{Clip}_{\text{grad}}\Big(\mathcal{M}_{\mathcal{R}} \nabla_{\hat{S}_k^{(\text{out})}} \mathcal{J}(\hat{S}_k^{(\text{out})})\Big), \\
\tilde{S}_k^{(\text{out})} &= \Pi_{\text{vel}}\Big(\hat{S}_k^{(\text{out})} - \text{Clip}_{\text{step}}(\lambda \mathbf{g}_k)\Big), \\
S_{k'}^{(\text{out})} &= \begin{cases} \sqrt{\bar{\alpha}_{k'}}\, \tilde{S}_k^{(\text{out})} + \sqrt{1-\bar{\alpha}_{k'}}\, z, & k' > 0, \\ \tilde{S}_k^{(\text{out})}, & k' = 0, \end{cases}
\end{aligned} \tag{29}
$$

Here, $z \sim \mathcal{N}(0, I)$, $\mathcal{M}_{\mathcal{R}}$ retains robot gradient components and zeros entity components, and λ controls correction strength.

V. Experiments

In the following, we evaluate SAGE in terms of task effectiveness, safety, social compliance, guidance controllability, and R2R coordination.

A. Experimental Setup

We consider two complementary evaluation regimes: recorded real-world trajectories for practical safety and social-compliance assessment, and controlled heterogeneous simulation for commanded-waypoint navigation and scalability analysis.

• *Datasets and rationale.* We employ three data sources, each serving a distinct purpose. *(i) ETH/UCY* [8] contains five pedestrian scenes (ETH, HOTEL, UNIV, ZARA1, and ZARA2). Because it lacks semantic labels and commanded waypoints, we use an offline proxy-navigation protocol: a target pedestrian serves as the robot proxy and its future endpoint as the proxy waypoint. This evaluates generation and guidance, but not genuine navigation, because the waypoint is derived from the recorded future. *(ii) SDD* provides aerial trajectories with official object annotations [47]. We map Pedestrians to PHAs, Bikers/Skaters/Carts to HNAAs, and Cars/Buses to SSAAs, making SDD the primary source for role-aware social-compliance evaluation. *(iii) Controlled heterogeneous simulation* complements recorded data by varying robot population, entity density, role composition, and start-waypoint conflicts. Robots receive commanded waypoints independent of their future trajectories across open, crossing, corridor, bottleneck, and intersection layouts, enabling genuine task and scalability evaluation in Section V-E.

• *Settings and metrics.* We set $T^{(\mathsf{obs})} = 8$, $T^{(\mathsf{prd})} = 12$, and $\Delta t = 0.4\,\mathrm{s}$. Unless noted, evaluation uses $N_{\mathrm{samp}} = 20$ stochastic samples with stride 10. SAGE and SAGE w/o guidance share the same checkpoint and differ only in whether the safety-social gradient is applied; w/ and w/o denote these settings. For *physical safety*, collision rate (CR) is the fraction of robot–entity and unordered robot–robot pair-time separations below the collision margin, and minimum separation (MD) is the minimum such distance over the prediction horizon. For *social compliance*, personal-space intrusion rate (PIR) is the fraction of robot–entity pair-time separations within the isotropic personal-space margin, social-violation rate (SVR) is the fraction inside the front-facing role-conditioned ellipse in (11), speed-violation rate (Sp-VR) is the fraction of predicted robot states exceeding the maximum speed, and Energy is the mean guidance objective $\mathcal{J}$. For *task progress*, Goal-FDE is the terminal distance to the assigned waypoint, and Goal-SR is the fraction satisfying Goal-FDE$\leq d^{(\mathsf{goal})}$. ADE/FDE measure mean/final Euclidean errors against recorded trajectories; R- and E- denote robots and entities. ADE/FDE and goal metrics use the best of 20 samples, whereas safety-social metrics average all 20. Thus, the aggregation measures candidate-set coverage and typical candidate safety rather than an oracle-selected executed plan. Lower values are better except for MD and Goal-SR. R2E/R2R denote robot-to-entity/robot-to-robot metric pairs and are distinct from the R2N/N2R graph relations in Phase 1. Detailed preprocessing, comfort margins, thresholds, and model settings are provided in Supplementary Section A-A.

• *Comparative benchmarks.* On ETH/UCY, we compare against three reference methods under the same proxy-navigation protocol: *Constant Velocity* (deterministic short-horizon extrapolation), *Trajectron++* [21] (CVAE-based stochastic prediction), and *MID* [19] (diffusion-based prediction without guidance). On SDD, we report Constant Velocity and Trajectron++. We also report the complete framework (*SAGE*) and an ablation that disables Phase 3 safety-social guidance (*SAGE w/o guidance*).

TABLE I

MAIN RESULTS ON RECORDED REAL-WORLD SCENES. ETH/UCY VALUES ARE AVERAGED OVER FIVE SCENES; SDD USES OFFICIAL SEMANTIC ANNOTATIONS. ARROWS INDICATE THE DESIRED DIRECTIONS.

Dataset	Method	R-ADE↓	Goal-FDE↓	CR↓	MD↑	PIR↓	SVR↓	Energy↓
ETH/UCY	Constant Velocity	0.4760	1.0092	0.0471	0.0398	0.3407	0.1456	20.4553
	Trajectron++	0.1461	0.0968	0.0416	0.0469	0.3440	0.1355	20.0778
	MID	0.1163	0.0929	0.0409	0.0510	0.3547	0.1482	20.1621
	SAGE w/o guidance	0.1152	0.0937	0.0441	0.0471	0.3560	0.1479	20.3761
	SAGE	0.1174	0.0947	0.0353	0.0623	0.3457	0.1248	19.1257
SDD	Constant Velocity	0.7290	1.6818	0.1701	0.0572	0.1560	0.0892	28.7860
	Trajectron++	0.1391	0.1188	0.1798	0.0875	0.1645	0.0965	27.6866
	SAGE w/o guidance	0.1151	0.0700	0.1763	0.0582	0.1612	0.0925	26.1494
	SAGE	0.1241	0.0813	0.1729	0.0694	0.1576	0.0679	24.4608

• *Implementation.* Training and inference use an NVIDIA 4090 GPU; per-scene runtime is reported in Table III. Preprocessing, model/training hyperparameters, and guidance settings are provided in the Supplementary Material (Section A-A).

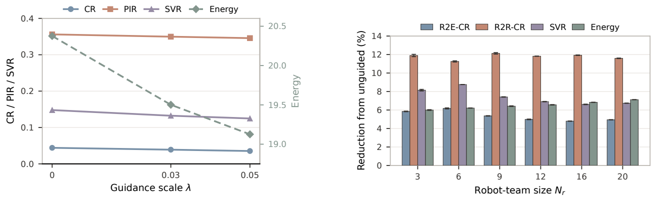
B. Real-World Safety/Social Compliance on ETH/UCY and SDD

We first evaluate the complete SAGE framework on real trajectory data and isolate the contribution of its inference-time safety-social guidance by comparing it with SAGE w/o guidance. Table I reports summarized results across ETH/UCY (five-scene average) and SDD, where two findings emerge. *First*, guidance produces consistent safety improvements. On ETH/UCY, SAGE reduces CR by 20.0% (0.0441→0.0353), SVR by 15.6%, and Energy by 6.1%, while increasing MD by 1.32× relative to SAGE w/o guidance. Notably, SAGE w/o guidance is slightly *less* safe than Trajectron++ and MID on several metrics (e.g., CR 0.0441 vs. 0.0416 and 0.0409), confirming that the improvement comes from guidance rather than the diffusion prior alone. On SDD, guidance reduces SVR by 26.6% and Energy by 6.5%, while the CR improvement is smaller (1.9%, 0.1763→0.1729) because the recorded-data proxy protocol leaves limited collision-rate headroom. *Second*, the accuracy trade-off is modest: SAGE w/o guidance is competitive with the best learned baseline on ETH/UCY (R-ADE 0.1152 vs. MID 0.1163 and Trajectron++ 0.1461), and guidance increases R-ADE by only 1.9% (0.1152→0.1174). ETH/UCY Goal-FDE is an auxiliary consistency check rather than a navigation-success metric because the proxy waypoint coincides with the recorded endpoint. Commanded-waypoint evaluation is presented in Section V-E; complete per-scene and role-wise results are provided in the Supplementary Material.

C. Guidance Mechanism: Controllability and Safety–Accuracy Trade-off

The above results treat guidance as a binary switch. We now sweep the guidance scale λ, which controls the strength of the safety-social energy gradient during reverse diffusion, to answer two questions: *(i)* is the correction strength controllable, and *(ii)* what accuracy cost accompanies each safety level?

As λ increases from 0 to 0.05, CR decreases by 20.0%, SVR by 15.6%, and Energy by 6.1%, while R-ADE increases by only 1.9%. Fig. 3(a) visualizes this consistent trend, with exact



(a) (b)

Fig. 3. Guidance effects across evaluation regimes. (a) Safety and social-compliance metrics versus guidance scale λ on ETH/UCY (five-scene average); CR, PIR, and SVR use the left axis, while Energy uses the right axis. (b) Relative safety-social improvement in the multi-robot scalability sweep; bars report reductions over unguided sampling averaged over three seeds, and error bars denote standard deviation.

TABLE II
HGT VERSUS HOMOGENEOUS-GNN ENCODING ON SDD UNDER GUIDED SAMPLING.

Variant	R-ADE↓	Goal-FDE↓	CR↓	SVR↓	Energy↓
Homogeneous GNN	0.1170	0.0621	0.1806	0.0743	24.8181
HGT	0.1241	0.0813	0.1729	0.0679	24.4608

values in Supplementary Table V. The modest degradation indicates that the learned prior already proposes plausible trajectories and guidance acts as a lightweight correction layer. Unless otherwise specified, subsequent experiments use $\lambda = 0.05$. Together with w_{goal} (Supplementary Table X), λ forms a two-dimensional inference-time control interface for the safety–accuracy–task trade-off without retraining.

D. Architecture and Planner Diagnostics

SAGE comprises heterogeneous graph encoding (Phase 1), joint diffusion generation (Phase 2), and safety-social guidance (Phase 3). Rather than attributing every metric change to an entire phase, we use focused diagnostics to clarify the roles of representative design choices.

• *Heterogeneous encoder.* Replacing HGT with a shared-parameter GNN removes relation-specific attention projections. As shown in Table II, the homogeneous encoder achieves slightly better displacement accuracy, whereas HGT improves CR, SVR, and Energy. Thus, semantic relation modeling contributes primarily to interaction quality and asymmetric social reasoning rather than raw trajectory precision; the dominant safety improvement still comes from Phase 3.

• *Learned generation versus deterministic rules.* The Social Force planner [4] aggressively repels robots from surrounding participants using hand-crafted potentials. It achieves low collision and social-violation rates but at a severe task cost: across the controlled-simulation sweep, its Goal-FDE is 1.68× worse and Robot ADE is 3.33× worse than SAGE. This diagnostic exposes the task–safety trade-off directly; complete values are provided in Supplementary Table XII.

Joint-versus-two-stage generation and pseudo-role robustness are reported in Supplementary Tables IX and XI. The two-stage variant can be locally more conservative, while joint generation preserves better task consistency in large-team settings; the pseudo-role ablation confirms that ETH/UCY safety gains are not artifacts of one labeling rule.

TABLE III
MULTI-ROBOT SCALABILITY ($N_e = 50$, $N_{\text{samp}} = 20$). RUNTIME IS THE PER-SCENE MEAN OVER 1,024 SCENES AND THREE SEEDS.

N_r	Guidance	Goal-FDE↓	R2E-CR↓	R2R-CR↓	R2E-MD↑	R2R-MD↑	SVR↓	Energy↓	Runtime
3	w/o	5.872	0.0418	0.0851	0.0206	0.0578	0.0216	292.2	85.0 ms
3	w/	5.928	0.0394	0.0750	0.0241	0.0838	0.0198	274.7	92.1 ms
6	w/o	5.813	0.0372	0.0781	0.0136	0.0308	0.0185	594.9	90.5 ms
6	w/	5.868	0.0349	0.0693	0.0171	0.0465	0.0169	557.9	99.6 ms
9	w/o	5.430	0.0397	0.0924	0.0122	0.0194	0.0202	966.3	96.6 ms
9	w/	5.465	0.0375	0.0812	0.0146	0.0286	0.0187	904.3	106.1 ms
12	w/o	5.420	0.0399	0.0951	0.0106	0.0138	0.0205	1369.8	103.1 ms
12	w/	5.448	0.0379	0.0839	0.0117	0.0207	0.0191	1279.9	113.8 ms
16	w/o	5.319	0.0383	0.0910	0.0090	0.0116	0.0193	1881.1	123.0 ms
16	w/	5.330	0.0365	0.0801	0.0111	0.0165	0.0180	1752.6	137.0 ms
20	w/o	5.432	0.0387	0.0869	0.0078	0.0094	0.0196	2473.6	132.0 ms
20	w/	5.444	0.0368	0.0769	0.0093	0.0141	0.0183	2297.5	147.2 ms

E. Controlled Multi-Robot Evaluation: Coordination and Scalability

We use controlled heterogeneous simulation to evaluate commanded-waypoint navigation as the robot team grows. Unlike the recorded proxy protocol, robots receive independently assigned, potentially conflicting waypoints. We investigate two questions: *(i)* whether guidance remains effective with increasing team size, and *(ii)* which R2R mechanism—graph edges or collision energy—drives coordination.

• Table III sweeps $N_r = 3$ to 20 with $N_e = 50$. Because each scale induces a different normalized interaction topology, raw collision rates are scale-wise diagnostics rather than a monotonic density curve. Guidance improves every team size (Fig. 3(b)). Averaged over all N_r, it reduces R2E-CR by 5.35%, R2R-CR by 11.75%, SVR by 7.43%, and Energy by 6.52%, while increasing R2E-MD by 1.19× and R2R-MD by 1.48×. The smaller R2E-CR gain reflects the substantial R2E separation already maintained by synthetic expert trajectories; guidance primarily corrects the more challenging R2R interactions. At the dense $N_r = 20$ stress case, it still reduces R2R-CR from 0.0869 to 0.0769 and increases R2R-MD from 0.0094 to 0.0141. Goal-FDE changes by less than 2% across all scales, showing that safety correction preserves waypoint progress. Guided runtime grows from 92.1 ms at $N_r = 3$ to 147.2 ms at $N_r = 20$.

• Table IV isolates R2R graph edges (Phase 1) and R2R collision energy (Phase 3) through four variants: neither mechanism, edges only, energy only, and complete SAGE. Relative to no coordination, the energy-only variant reduces R2R-CR by 10.4% at $N_r = 12$ (0.0892→0.0799) and 11.0% at $N_r = 20$ (0.0858→0.0764), while increasing R2R-MD by 1.47× at both scales. Hence, collision energy is the dominant local safety mechanism. SAGE obtains the best Goal-FDE in both settings, indicating that R2R graph edges help integrate the safety signal with task-conditioned generation. Energy is not comparable across variants with different active terms because it includes the R2R collision term when enabled. Supplementary Table XIII provides complementary recorded-scene proxy evidence with official SDD semantic annotations.

F. Qualitative Analysis

Fig. 5 visualizes representative multi-robot cases in controlled heterogeneous simulation. In each case, the robot team, waypoints, heterogeneous entities, and scene layout are fixed; only the guidance switch changes. We observe three patterns. *First*, guidance corrections are localized: unguided

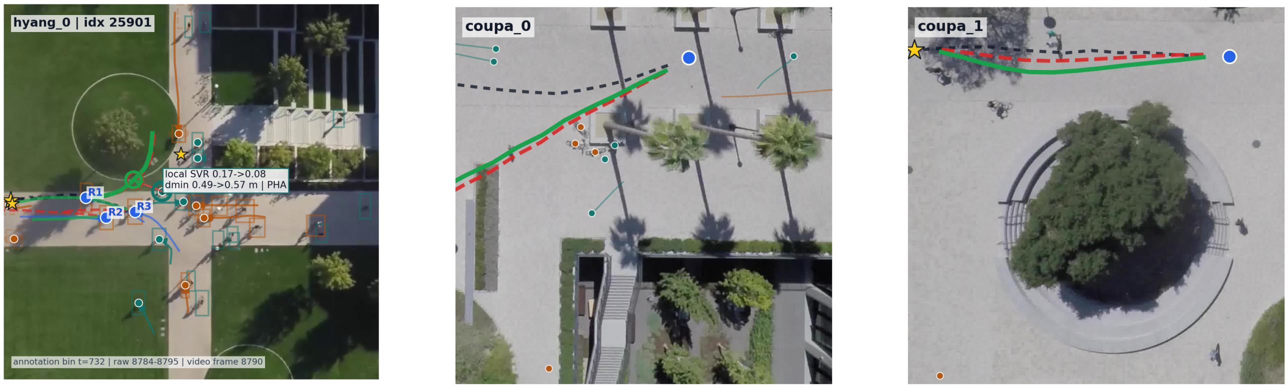


Fig. 4. Real-scene qualitative examples on SDD. Left: three-robot proxy case (Hyang). Center/right: single-robot proxy cases (Coupa). Dashed red: unguided. Solid green: SAGE-guided. Gray traces: observed history.

TABLE IV
R2R COORDINATION ABLATION; ENERGY IS INCOMPARABLE ACROSS DIFFERENT ACTIVE TERMS.

N_r	Variant	Goal-FDE↓	R2R-CR↓	R2R-MD↑	Energy
12	No R2R coord.	5.574	0.0892	0.0144	1057.7
12	Edges only	5.488	0.0934	0.0146	1068.1
12	Energy only	5.534	**0.0799**	**0.0212**	1259.6
12	SAGE	**5.448**	0.0839	0.0207	1279.9
20	No R2R coord.	5.557	0.0858	0.0095	1755.1
20	Edges only	5.510	0.0859	0.0094	1750.8
20	Energy only	5.486	**0.0764**	0.0140	2299.0
20	SAGE	**5.444**	0.0769	**0.0141**	2297.5

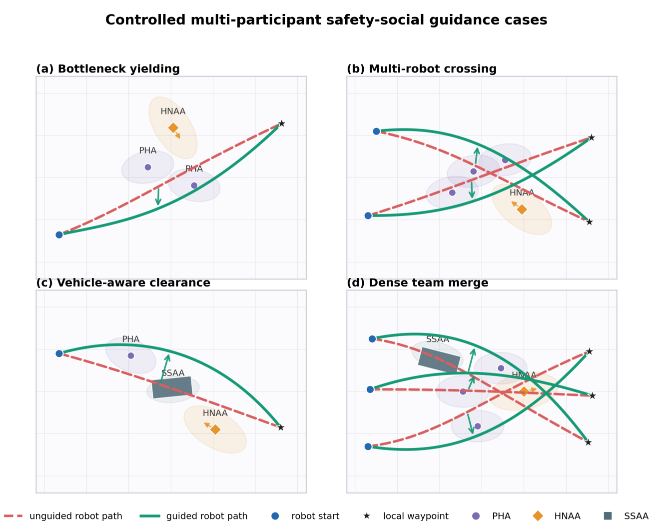


Fig. 5. Controlled cases: bottleneck yielding, multi-robot crossing, vehicle-aware clearance, and dense merging. Dashed red: unguided; solid green: guided; blue circles: robot starts.

trajectories that already avoid entities remain largely unchanged, while those encroaching on social fields or collision boundaries receive targeted repulsion. *Second*, the correction magnitude adapts to entity type—robots maintain larger clearance from PHAs than from SSAAs, consistent with the role-conditioned social field design. *Third*, guidance preserves waypoint progress: in all four panels, guided trajectories terminate near the same waypoints as unguided ones, illustrating how w_{goal} balances safety against task completion. Fig. 4 shows that analogous local corrections also occur in recorded SDD semantic scenes.

G. Summary of Evaluations

• *Conclusions.* The evidence supports four conclusions.
(i) Safety-social guidance is the primary driver of improvement. On ETH/UCY, guidance reduces CR by 20.0% and SVR by 15.6% with minimal accuracy cost. On SDD, it reduces SVR by 26.6%, although the CR gain is limited to 1.9% under the proxy protocol. The λ sweep confirms controllable correction over the tested values.
(ii) The components play distinct roles. HGT modestly improves safety-social diagnostics over a homogeneous GNN at the cost of displacement accuracy; joint generation mainly preserves task consistency in multi-robot settings; role-conditioned fields provide category-specific comfort constraints and interpretable diagnostics. The main quantitative safety gains come from inference-time guidance, while the other components support heterogeneous representation and prediction–planning coupling.
(iii) Guidance remains effective over the tested multi-robot range. It reduces collision rates and social violations as team size grows from 3 to 20. R2R collision energy is the dominant local safety mechanism, while R2R graph edges help preserve task consistency. SDD multi-robot proxy results provide complementary feasibility evidence but should not be equated with commanded robot-team deployment.
(iv) The safety–accuracy trade-off is tunable. Stronger guidance consistently improves safety at the cost of displacement accuracy. The controls λ and w_{goal} allow context-dependent prioritization of safety, trajectory fidelity, and waypoint progress without retraining.
• *Limitations.* First, ETH/UCY provides pseudo-heterogeneous roles inferred from speed heuristics; true semantic heterogeneity is evaluated on SDD and depends on its official annotations. Second, recorded-data multi-robot experiments retrospectively treat pedestrian tracks as controllable robots rather than evaluating commanded real-robot teams. Third, controlled simulation enables systematic stress tests but uses simplified entity dynamics. Fourth, guidance parameters (λ, w_{goal}, and comfort margins) are fixed per experiment and may benefit from scene-adaptive or online

adjustment. Fifth, the 100-level diffusion schedule with stride-10 reverse updates still requires offline computation. Physical deployment, richer behavior models, adaptive guidance, and faster sampling therefore remain important future directions.

VI. Conclusion

SAGE combines HGT interaction encoding, joint diffusion prediction/planning, and training-free safety-social guidance. Recorded and controlled experiments show improved safety/social compliance with preserved task progress and a controllable trade-off. Future work will target adaptive sampling and physical-robot validation.

References

[1] A. I. Karoly, P. Galambos, J. Kuti, and I. J. Rudas, "Deep learning in robotics: Survey on model structures and training strategies," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 266–279, Jan. 2021.

[2] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, Sep. 2016.

[3] Y. Chen, F. Zhao, and Y. Lou, "Interactive model predictive control for robot navigation in dense crowds," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 4, pp. 2289–2301, Apr. 2022.

[4] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, May 1995.

[5] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proc. Natl. Acad. Sci. USA*, vol. 116, no. 50, pp. 24 972–24 978, Dec. 2019.

[6] C. Mavrogiannis *et al.*, "Core challenges of social robot navigation: A survey," *ACM Trans. Hum.-Robot Interact.*, vol. 12, no. 3, pp. 1–39, Sep. 2023.

[7] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 2255–2264.

[8] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 961–971.

[9] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 14 424–14 432.

[10] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Proc. 14th Int. Symp. Robot. Res. (ISRR)*, 2011, pp. 3–19.

[11] Y. Zhou, H. Hu, Y. Liu, and Z. Ding, "Collision and deadlock avoidance in multirobot systems: A distributed approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1712–1726, Jul. 2017.

[12] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, Jul. 2020.

[13] M. Li, J. Qin, J. Li, Q. Liu, Y. Shi, and Y. Kang, "Game-based approximate optimal motion planning for safe human-swarm interaction," *IEEE Trans. Cybern.*, vol. 54, no. 10, pp. 5649–5660, Oct. 2024.

[14] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 9813–9823.

[15] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-BiGAT: Multimodal trajectory forecasting using Bicycle-GAN and graph attention networks," in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 137–146.

[16] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf. (WWW)*, 2020, pp. 2704–2710.

[17] E. T. Hall, *The Hidden Dimension*. Garden City, NY, USA: Doubleday, 1966.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[19] T. Gu *et al.*, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 17 113–17 122.

[20] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[21] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 683–700.

[22] G. Campion, G. Bastin, and B. Dandrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 47–62, Feb. 1996.

[23] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 1349–1358.

[24] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019, pp. 793–803.

[25] B. Miloradovic, B. Curuklu, M. Ekstrom, and A. V. Papadopoulos, "GMP: A genetic mission planner for heterogeneous multirobot system applications," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10 627–10 638, Oct. 2022.

[26] M. Faroni, A. Umbrico, M. Beschi, A. Orlandini, A. Cesta, and N. Pedrocchi, "Optimal task and motion planning and execution for multiagent systems in dynamic environments," *IEEE Trans. Cybern.*, vol. 54, no. 6, pp. 3366–3377, Jun. 2024.

[27] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 2256–2265.

[28] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 9902–9915.

[29] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?" in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[30] C. M. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, and D. Anguelov, "MotionDiffuser: Controllable multi-agent motion prediction using diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 9644–9653.

[31] W. Xiao, T.-H. Wang, C. Gan, R. Hasani, M. Lechner, and D. Rus, "SafeDiffuser: Safe planning with diffusion probabilistic models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

[32] Z. Zhong *et al.*, "Guided conditional diffusion for controllable traffic simulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 3560–3566.

[33] M. Niedoba *et al.*, "A diffusion-model of joint interactive navigation," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 27 440–27 459.

[34] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8780–8794.

[35] Z. Lin, Z. C. Chen, G. Zhu, J. Chen, and J. Li, "GALC: Guided amplified learning with Lipschitz constraint for robust trajectory generation," *IEEE Trans. Cybern.*, pp. 1–11, Mar. 2026, early access, doi: 10.1109/TCYB.2026.3668987.

[36] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013.

[37] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 6015–6022.

[38] S. Saadatnejad, Y. Gao, K. Messaoud, and A. Alahi, "Social-Transmotion: Promptable human trajectory prediction," arXiv:2312.16168, 2023. [Online]. Available: https://arxiv.org/abs/2312.16168

[39] Y. Gao and C.-M. Huang, "Evaluation of socially-aware robot navigation," *Front. Robot. AI*, vol. 8, p. 721317, Jan. 2022.

[40] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "SocNavBench: A grounded simulation testing framework for evaluating social navigation," *ACM Trans. Hum.-Robot Interact.*, vol. 11, no. 3, pp. 1–24, Sep. 2022.

[41] S. Saadatnejad, M. Bahari, P. Khorsandi, M. Saneian, S.-M. Moosavi-Dezfooli, and A. Alahi, "Are socially-aware trajectory prediction models really socially-aware?" *Transp. Res. C, Emerg. Technol.*, vol. 141, p. 103705, Aug. 2022.

[42] G. Wen, X. Yu, W. Yu, and J. Lu, "Coordination and control of complex network systems with switching topologies: A survey," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 10, pp. 6342–6357, Oct. 2021.

[43] J. Wu, C. Luo, Y. Luo, and K. Li, "Distributed UAV swarm formation and collision avoidance strategies over fixed and switching topologies," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10 969–10 979, Oct. 2022.

[44] J. Turner, Q. Meng, G. Schaefer, A. Whitbrook, and A. Soltoggio, "Distributed task rescheduling with time constraints for the optimization of total task allocations in a multirobot system," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2583–2597, Sep. 2018.

[45] C. Bai, Y. Peng, H. Piao, W. Pan, and J. Guo, "Learning-based multi-UAV flocking control with limited visual field and instinctive repulsion," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 462–475, Jan. 2024.

[46] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[47] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 549–565.

Appendix A
Supplementary Experimental Details

This supplementary material collects implementation settings, complete tables, additional ablations, and diagnostics referenced from the main text. It supports—rather than replaces—the evidence in Section V.

A. Implementation and Guidance Settings

ETH/UCY retains up to eight neighboring entities and SDD up to 12 in single-robot settings. ETH/UCY and controlled simulation use metric coordinates; SDD coordinates are divided by 50 before training and evaluation. For these three regimes, respectively, $(d^{(\mathsf{safe})}, v_{\max}, d^{(\mathsf{goal})})$ is $(0.45, 2.5, 0.5)$, $(0.30, 3.0, 0.5)$, and $(0.35, 3.0, 0.6)$ in the working coordinates; $v_{\max}^{\mathsf{lat}} = 0.15$ and $\omega_{\max} = 1.2$ are shared. Following proxemics theory [17] and social-navigation literature [4], $(\sigma^{\mathsf{lon}}, \sigma^{\mathsf{lat}})$ is $(2.0, 1.0)$ for PHAs, $(2.6, 1.2)$ for HNAAs, and $(1.2, 0.7)$ for SSAAs.

Unless varied, $(w_{\mathsf{col}}, w_{\mathsf{soc}}, w_{\mathsf{kin}}, w_{\mathsf{goal}}) = (2.0, 1.0, 0.1, 0.3)$, $\sigma_{\mathsf{col}} = 0.8$, and gradient/step clipping norms are $1.0/0.05$; $w_{\mathsf{R2R}} = 1.5$ in controlled simulation and 1.0 in the SDD multi-robot proxy. AdamW uses learning rate 3×10^{-4}, weight decay 10^{-4}, and batch size 128. ETH/UCY and SDD models train for 100 and 80 epochs, respectively. Each controlled-simulation model trains for 50 epochs on 8,192 episodes and is evaluated on 1,024 held-out episodes per seed. The HGT has $L = 2$, $d = 128$, and four heads; the denoiser has four temporal Transformer layers and 100 diffusion steps. Checkpoints and evaluation scripts are available in the project repository.

TABLE V
Guidance-strength sweep on ETH/UCY (five-scene average).

λ	R-ADE↓	Goal-FDE↓	CR↓	PIR↓	SVR↓	Energy↓
0.00	0.1152	0.0937	0.0441	0.3560	0.1479	20.3761
0.03	0.1160	0.0946	0.0390	0.3497	0.1324	19.5015
0.05	0.1174	0.0947	0.0353	0.3457	0.1248	19.1257

B. Complete Recorded-Scene Diagnostics

Table VI reports the full ETH/UCY per-scene proxy-navigation results underlying the averages in Table I. Table VII provides the complete SDD aggregate result including entity displacement, while Table VIII gives the role-wise decomposition.

TABLE VIII
SDD role-wise semantic diagnostics.

Role	Method	ADE↓	FDE↓	CR↓	MD↑	PIR↓	SVR↓	Count/scene
PHA	w/o	0.2781	0.5223	0.0141	0.1329	0.1786	0.0933	2.6130
PHA	w/	0.2787	0.5230	0.0119	0.1476	0.1748	0.0710	2.6130
HNAA	w/o	0.7382	1.4622	0.0134	0.1915	0.1040	0.0752	1.7265
HNAA	w/	0.7385	1.4631	0.0123	0.2098	0.1008	0.0508	1.7265
SSAA	w/o	0.1428	0.2569	0.0058	0.6455	0.0327	0.0103	0.0697
SSAA	w/	0.1406	0.2526	0.0057	0.6475	0.0320	0.0042	0.0697

Guidance reduces role-wise CR by 15.6% for PHAs (0.0141→0.0119), 8.2% for HNAAs (0.0134→0.0123), and marginally for SSAAs. SVR decreases by 23.9% (PHA, 0.0933→0.0710), 32.4% (HNAA, 0.0752→0.0508), and 59.2% (SSAA, 0.0103→0.0042). The improvement is distributed across all semantic categories rather than concentrated in one dominant class. SSAA samples are rare in the evaluated split (0.07 per scene), so their role-wise numbers should be interpreted as diagnostic rather than conclusive.

C. Additional Ablations and Diagnostics

Joint vs. two-stage generation. Table IX reports the two-stage decoupling diagnostic. In this variant, entity futures are generated first and then fixed while robot trajectories are generated with the same guidance interface.

TABLE IX
Multi-robot two-stage decoupling in controlled simulation.

Setting	N_r	Variant	Guidance	Goal-FDE↓	R2E-CR↓	R2R-CR↓	R2R-MD↑	SVR↓
Controlled sim.	12	Joint	w/	**5.448**	0.0379	0.0839	0.0207	0.0191
Controlled sim.	12	Two-stage	w/	5.725	**0.0369**	**0.0788**	**0.0626**	**0.0178**
Controlled sim.	20	Joint	w/	**5.444**	0.0368	0.0769	0.0141	0.0183
Controlled sim.	20	Two-stage	w/	5.675	**0.0360**	**0.0757**	**0.0391**	**0.0173**

The two-stage variant is competitive on local safety metrics, particularly R2R-MD where it substantially exceeds joint generation. However, the joint model preserves better Goal-FDE in large-team settings (e.g., 5.444 vs. 5.675 at $N_r = 20$), indicating that coupled generation helps maintain task consistency when multiple robots plan simultaneously.

Task-progress recovery via w_{goal}. Table X sweeps the task-progress weight w_{goal} on SDD under guided sampling. Note that this table reports results in raw coordinate units; the relative trends are directly interpretable.

TABLE VI
Full ETH/UCY per-scene proxy-navigation results.

Dataset	Method	R-ADE↓	Goal-FDE↓	Goal-SR↑	E-ADE↓	E-FDE↓	CR↓	MD↑	PIR↓	SVR↓	Energy↓
ETH	SAGE w/o guidance	0.1338	0.0550	1.0000	0.7319	1.2722	0.0723	0.0362	0.3580	0.1494	10.5465
ETH	SAGE	0.1339	0.0531	1.0000	0.7183	1.2350	0.0591	0.0549	0.3538	0.1221	9.9119
HOTEL	SAGE w/o guidance	0.0977	0.1132	0.9691	0.5364	1.0041	0.0491	0.0551	0.3884	0.1000	10.0647
HOTEL	SAGE	0.0997	0.1116	0.9783	0.5452	1.0207	0.0392	0.0684	0.3733	0.0802	9.2323
UNIV	SAGE w/o guidance	0.1567	0.1441	0.9681	0.7837	1.4014	0.0393	0.0213	0.2574	0.1298	35.4965
UNIV	SAGE	0.1588	0.1467	0.9686	0.7835	1.4011	0.0341	0.0258	0.2482	0.1170	33.5777
ZARA1	SAGE w/o guidance	0.0954	0.0661	0.9996	0.2899	0.4944	0.0240	0.0707	0.4006	0.1869	16.9963
ZARA1	SAGE	0.0993	0.0711	1.0000	0.2860	0.4893	0.0179	0.0942	0.3849	0.1722	15.9966
ZARA2	SAGE w/o guidance	0.0926	0.0900	0.9995	0.3076	0.5763	0.0358	0.0520	0.3753	0.1733	28.7766
ZARA2	SAGE	0.0955	0.0908	0.9993	0.3057	0.5707	0.0265	0.0680	0.3684	0.1326	26.9102
Avg.	SAGE w/o guidance	0.1152	0.0937	0.9873	0.5299	0.9497	0.0441	0.0471	0.3560	0.1479	20.3761
Avg.	SAGE	0.1174	0.0947	0.9892	0.5278	0.9434	0.0353	0.0623	0.3457	0.1248	19.1257

TABLE VII
FULL SDD SEMANTIC DIAGNOSTICS WITH OFFICIAL ANNOTATIONS ($N_{\text{samp}} = 20$ SAMPLES).

Method	R-ADE↓	Goal-FDE↓	Goal-SR↑	E-ADE↓	E-FDE↓	CR↓	MD↑	PIR↓	SVR↓	Sp-VR↓	Energy↓
SAGE w/o guidance	0.1151	0.0700	0.9945	0.4769	0.9274	0.1763	0.0582	0.1612	0.0925	0.0298	26.1494
SAGE	0.1241	0.0813	0.9913	0.4768	0.9274	0.1729	0.0694	0.1576	0.0679	0.0300	24.4608

TABLE X
SDD w_{goal} SWEEP UNDER GUIDED SAMPLING.

w_{goal}	R-ADE↓	Goal-FDE↓	Goal-SR↑	CR↓	PIR↓	SVR↓	Energy↓
0.0	1.4740	2.4349	0.2120	0.0118	0.0087	0.0025	5.5359
0.3	1.3487	2.1951	0.2247	0.0122	0.0090	0.0027	11.4233
0.6	1.2335	1.9741	0.2387	0.0126	0.0092	0.0029	15.9795
1.0	1.0945	1.7031	0.2595	0.0131	0.0096	0.0033	19.8706
2.0	0.8267	1.1677	0.3169	0.0146	0.0106	0.0044	23.5397

Increasing w_{goal} from 0 to 2.0 reduces Goal-FDE from 2.4349 to 1.1677 and improves Goal-SR from 0.2120 to 0.3169, at the cost of gradually higher CR, PIR, SVR, and Energy. This confirms w_{goal} as a tunable task–safety control knob.

Pseudo-role construction robustness. ETH/UCY lacks semantic labels. Table XI verifies that the guidance effect does not depend on a particular pseudo-labeling rule.

TABLE XI
PROXY ROLE CONSTRUCTION ABLATION ON ETH/UCY (FIVE-SCENE AVERAGE, $N_{\text{samp}} = 20$ SAMPLES).

Role Strategy	Method	R-ADE↓	Goal-FDE↓	CR↓	PIR↓	SVR↓	Energy↓
`speed_threshold`	w/o	0.5149	0.8900	0.0456	0.3885	0.1673	26.4692
`speed_threshold`	w/	0.6980	1.0756	0.0030	0.1018	0.0218	9.4370
`speed_quantile`	w/o	0.6054	1.0353	0.0459	0.3625	0.1714	25.7617
`speed_quantile`	w/	0.6748	1.0259	0.0034	0.0961	0.0261	9.6635
`all_pha`	w/o	0.5519	0.9574	0.0471	0.3810	0.2083	27.5432
`all_pha`	w/	0.7024	1.0839	0.0031	0.0989	0.0252	9.6558

Across all strategies, guided sampling sharply reduces CR, PIR, SVR, and Energy. The safety improvement is robust to the choice of pseudo-labeling rule. This diagnostic uses a separate evaluation protocol; relative improvements are directly comparable within the table, whereas its absolute R-ADE values should not be compared with Table I.

D. Social Force Baseline and Multi-Robot Diagnostics

Social Force comparison. Table XII details the deterministic Social Force planner comparison summarized in the main text.

TABLE XII
COMPARISON WITH DETERMINISTIC SOCIAL FORCE PLANNER.

Setting	Method	Goal-FDE↓	R-ADE↓	R2E-CR↓	R2R-CR↓	SVR↓	Sp-VR↓
Controlled sim.	Social Force	9.371	2.688	**0.0136**	**0.0172**	**0.0049**	0.0474
Controlled sim.	SAGE	**5.583**	**0.808**	0.0371	0.0772	0.0185	**0.0000**
SocialGym-style	Social Force	7.766	2.322	**0.0083**	**0.0198**	**0.0043**	0.0366
SocialGym-style	SAGE	**6.605**	**0.696**	0.0292	0.0823	0.0168	**0.0000**

SDD multi-robot proxy. Table XIII reports guided SDD multi-robot proxy results with official semantic annotations. Controlled simulation (Section V-E) provides the primary scalability evidence; these results show that the same pipeline can operate on real semantic scenes.

TABLE XIII
GUIDED SDD MULTI-ROBOT PROXY SWEEP WITH OFFICIAL SEMANTIC ANNOTATIONS.

N_r	Goal-FDE↓	Goal-SR↑	R2E-CR↓	R2R-CR↓	PIR↓	SVR↓	Energy↓
3	0.1260	0.9716	0.0639	0.2547	0.0567	0.0296	71.46
6	0.2229	0.9338	0.0249	0.1228	0.0215	0.0128	131.63
12	0.3254	0.8919	0.0130	0.0590	0.0105	0.0062	224.74

SocialGym-style scale and density. Here, SocialGym-style denotes our local generator of doorway, hallway, intersection, roundabout, and open-crowd layouts, rather than exported rollouts from the SocialGym simulator. Its model is trained for 60 epochs on 8,192 generated episodes and evaluated on 1,024 held-out episodes for each of three seeds. Table XIV reports the resulting scale and density diagnostics.

TABLE XIV
SOCIALGYM-STYLE SCALE AND DENSITY DIAGNOSTICS (THREE EVALUATION SEEDS).

N_r	N_e	Guidance	Goal-FDE↓	R2E-CR↓	R2R-CR↓	SVR↓	Energy↓
3	40	w/o	6.675	0.0315	0.1050	0.0188	248.4
3	40	w/	6.707	0.0292	0.0922	0.0169	233.7
6	40	w/o	6.543	0.0313	0.0907	0.0185	529.6
6	40	w/	6.573	0.0291	0.0807	0.0168	497.3
6	80	w/o	6.462	0.0319	0.0970	0.0194	789.5
6	80	w/	6.533	0.0296	0.0872	0.0176	740.0
9	40	w/o	6.518	0.0308	0.0853	0.0184	864.6
9	40	w/	6.548	0.0288	0.0765	0.0168	815.2

Across all SocialGym-style configurations, guidance consistently lowers R2E-CR, R2R-CR, SVR, and Energy, matching the pattern observed in the controlled-simulation scalability sweep.