

Grounded Relational Inference: Domain Knowledge Driven Explainable Autonomous Driving

Chen Tang[✉], Member, IEEE, Nishan Srishankar, Sujitha Martin, and Masayoshi Tomizuka[✉], Life Fellow, IEEE

Abstract—Explainability is essential for autonomous vehicles and other robotics systems interacting with humans and other objects during operation. Humans need to understand and anticipate the actions taken by machines for trustful and safe cooperation. In this work, we aim to develop an explainable model that generates explanations consistent with both human domain knowledge and the model’s inherent causal relation. In particular, we focus on an essential building block of autonomous driving—multi-agent interaction modeling. We propose Grounded Relational Inference (GRI). It models an interactive system’s underlying dynamics by inferring an interaction graph representing the agents’ relations. We ensure a semantically meaningful interaction graph by grounding the relational latent space into semantic interactive behaviors defined with expert domain knowledge. We demonstrate that it can model interactive traffic scenarios under both simulation and real-world settings and generate semantic graphs explaining the vehicle’s behavior through their interactions.

Index Terms—Autonomous vehicles, explainable AI, deep learning, driving behavior modeling.

I. INTRODUCTION

DEEP learning has been utilized to address various autonomous driving problems [1], [2], [3]. However, deep neural networks lack the transparency that helps people understand their underlying mechanisms. It is a crucial drawback for safety-critical applications with humans involved (e.g., autonomous vehicles). Humans need to understand and anticipate the actions taken by machines for trustful and safe cooperation. In response to this problem, the concept of explainable AI (XAI) was introduced. It refers to machine learning techniques that provide details and reasons that make a model’s mechanism easy to understand [4]. Most of the existing works for deep learning models focus on post-hoc explanations [4]. They enhance model explainability by

Manuscript received 18 June 2022; revised 11 June 2023; accepted 6 June 2024. Date of publication 15 July 2024; date of current version 29 August 2024. The work of Chen Tang and Nishan Srishankar was supported in part by the Honda Research Institute as Research Interns, through the support of Honda Research Institute’s resources and funding. The Associate Editor for this article was S. Sun. (*Corresponding author: Chen Tang*.)

Chen Tang was with the Honda Research Institute, San Jose, CA 95134 USA, and with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 USA. He is now with the Department of Computer Science, UT Austin, Austin, TX 78712 USA (e-mail: chen_tang@berkeley.edu).

Nishan Srishankar was with the Honda Research Institute, San Jose, CA 95134 USA. He is now with JPMorgan Chase, New York City, NY 10081 USA.

Sujitha Martin was with the Honda Research Institute, San Jose, CA 95134 USA. She is now with Amazon Web Services, Seattle, CA 98109 USA.

Masayoshi Tomizuka is with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 USA.

Digital Object Identifier 10.1109/TITS.2024.3424667

unraveling the underlying mechanisms of a trained model: Vision-based approaches, such as visual attention [5] and deconvolution [6], illustrate which segments of the input image affect the outputs; Interaction-aware models, such as social LSTM with social attention [7], [8] and graph neural networks (GNN) with graph attention [9], [10], [11], [12], identify the agents that are critical to the decision-making procedure.

Although promising, post-hoc explanations could be ambiguous and falsely interpreted by humans. For instance, a visual attention map only illustrates which regions of the input image the output of the model depends on. The semantic meaning behind the causal relation is left for human users to interpret. Kim et al. [13] attempted to resolve the ambiguity by aligning textual explanations with visual attention. However, the underlying mechanism of the model is not necessarily consistent with the textual explanations. To truly build trust with humans, we argue that a deep learning model for an autonomous system should be equipped with explanations consistent with both *human domain knowledge* and the model’s *inherent causal relation*.

In this work, we explore how to approach such an explainable model for an essential building block of autonomous driving—multi-agent interaction modeling. In particular, we focus on the relational inference problem studied in [12]. Kipf et al. propose the Neural Relational Inference (NRI) model, which models an interactive system by explicitly inferring its inherent interactions. Formally, the NRI model aims to solve a reconstruction task. Given the observed trajectories of all the objects, an encoder first infers the interactions between objects represented by a latent interaction graph, whose edges are aligned with discrete latent variables corresponding to a cluster of pairwise interaction behaviors between the objects. Afterward, a decoder learns the dynamical model conditioned on the inferred interaction graph and then reconstructs the trajectories given the initial states. If the decoder can accurately reconstruct the trajectories, it indicates that the latent space effectively models the interactions.

We find this discrete latent space interesting because the inferred interaction graph could potentially serve as an explanation directly: it explains the reconstructed trajectories as a sequence of interaction behaviors among agents. Moreover, the reconstructed trajectories are governed by the same interaction graph. Therefore, the NRI model seems promising to fulfill our goal to make the explanation consistent with the model’s underlying mechanism. However, since the NRI model learns the latent space in an unsupervised manner, it is

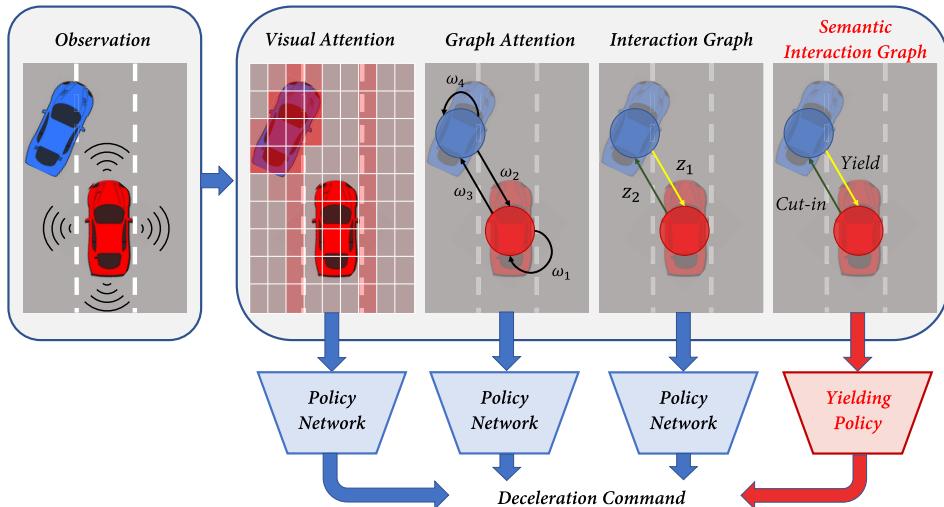


Fig. 1. A motivating lane-changing scenario where we ask different models to control the red vehicle. All the models generate deceleration commands but have different intermediate outputs. With the aid of visual attention, we generate a heat map indicating the critical pixels of the input image. Graph attention network assigns edge weights ω_i to specify the importance of surrounding vehicles to the controlled vehicle. However, the attention mechanisms cannot recognize different effects—the two cars are mutually important but affect each other in distinct ways. The NRI model can distinguish between different interactive behaviors by assigning different values to the latent variables z_i in the interaction graph. Still, the latent space does not have explicit semantic meaning. In contrast, our model ensures a semantic interaction graph, which illustrates the model’s understanding of the scenario and explains the action it takes. It determines the interaction graph with a latent space grounded in yielding and cutting-in behaviors. It learns the control policies that generate behaviors consistent with their definitions in domain knowledge (e.g., traffic rules) and executes the corresponding policies according to the inferred edge types.

difficult for humans to interpret the semantic meaning behind those interaction behaviors, which makes the interaction graph ambiguous as an explanation. To address this issue, we propose to ground the latent space in a set of interactive behaviors defined with human domain knowledge.

As a running example, consider the scenario depicted in Fig. 1, where we ask different models to control the red vehicle. Attention mechanisms can indicate the critical pixels or agents, but they cannot recognize different effects—the two cars are mutually important but affect each other in distinct ways. The NRI model can distinguish between different interactive behaviors. Still, the latent space does not have explicit semantic meaning. In contrast, our model should determine the interaction graph with a latent space grounded in yielding and cutting-in behaviors. It learns the control policies that generate behaviors consistent with their definitions in domain knowledge (e.g., traffic rules) and executes the corresponding policies according to the inferred edge types. This semantic interaction graph illustrates the model’s understanding of the scenario and explains the action it takes.

If we merely want to make the interaction graph consistent with humans’ labeling of the scenes, a straightforward approach is training the encoder directly via supervised learning. Interaction labels can be either obtained from human experts [14] or rule-based labeling functions [15]. However, labels for the interaction graph are insufficient to induce the decoder to synthesize the interactive behaviors suggested by the labels, because the model cannot capture the semantic meaning behind those interaction labels. Instead, we recast relational inference into an inverse reinforcement learning (IRL) problem and introduce structured reward functions to ground the latent space. Concretely, we model the system as a multi-agent Markov decision process (MDP), where the

agents share a reward function that depends on the relational latent space. We design structured reward functions based on expert domain knowledge to explicitly define the interactive behaviors corresponding to the latent space. To solve the formulated IRL problem, we propose Grounded Relational Inference (GRI). It has a variational-autoencoder-like (VAE) GNN in NRI [12] as the backbone model. Additionally, we incorporate the structured reward functions into the model as an additional reward decoder. A variational extension of the adversarial inverse reinforcement learning (AIRL) algorithm is derived to train all the modules simultaneously.

Compared to direct supervision via interaction labels, we provide implicit supervision to GRI in terms of the structures of the reward functions. Since each reward function defines a type of interactive behavior, we confine the latent space to a cluster of interactive behaviors. It mainly has two advantages over supervision through labeling: 1) First, since the policy decoder learns to maximize the cumulative reward given the inferred interaction graph, the structured reward functions guide the policy to synthesize the corresponding semantic behaviors rather than simply mimicking the demonstrated trajectories; 2) Second, the end-to-end training scheme leaves the model to identify the underlying interaction graph of the observed trajectories and learn the characteristics of different behaviors (i.e., parameters of reward functions) from data. It avoids the undesired bias introduced during the labeling procedure. Labels generated by human experts are subjective. Different people may interpret an interacting scenario in different ways. In contrast, there exist systematic and principled ways to investigate what reward functions human behavior is subject to from data [16].

The remaining content is organized as follows. In Section II, we give a concise review of existing works that are closely

related to ours in terms of methodology or motivation. In Section III, we briefly summarize NRI and AIRL to prepare the readers for the core technical content. In Section IV, we introduce how we reformulate relational inference into a multi-agent IRL problem with relational latent space. In Sec V, we present the GRI model in a general context. In Section VI, we demonstrate how we apply the proposed framework to model some simple traffic scenarios in both simulation and real-world settings. The experimental results show that GRI can model interactive traffic scenarios, and generate semantic interaction graphs that are consistent with both human domain knowledge and the modeled interactive behaviors.

II. RELATED WORK

Our model combines graph neural networks and adversarial inverse reinforcement learning for interactive system modeling. This section gives a concise review of these two topics and summarizes the existing works closely related to ours. We also discuss some additional works on explainable driving models as a complement to the discussion in Sec. I.

1) Interaction Modeling Using GNN: GNN refers to the class of neural networks that operate on graph-structured data, which consist of local message-passing functions shared across all nodes to aggregate the information of neighboring nodes. The inductive biases introduced by the shared message-passing operations make GNN particularly effective for modeling multi-agent interactive systems [11], [17], [18]. Multi-agent systems can be naturally represented as graphs. Furthermore, sharing the message-passing functions across nodes improves learning efficiency and enables modeling systems with varying numbers of agents with a single GNN. One category of models we find interesting is those with graph attention mechanisms. One seminal work is Graph Attention Network (GAT) [10], which performed well on large-scale inductive classification problems. VAIN [9] applied attention in multi-agent modeling. The attention map unravels the interior interaction structure to some extent which improves the explainability of VAIN. An approach closely related to ours is NRI [12], which modeled the interaction structure explicitly with discrete relational latent space compared to the continuous graph attention. We explain the difference between NRI and our proposed method in Sec. I and V. One related work in the autonomous driving domain is [15], which also modeled interactive driving behavior with semantically meaningful interactions but in a supervised manner.

Another type of model we want to mention is the spatio-temporal graphs (st-graph). St-graph decomposes a complex problem into components and their spatio-temporal interactions, which are represented by nodes and edges of a factor graph. It makes st-graph a ubiquitous representation for interacting systems, e.g., human motion [19], human-robot interaction [20], and traffic flow [21]. Jain et al. [19] proposed a general method to transform any st-graph to a mixture of RNNs called structural-RNN (S-RNN). When using GRUs, our GNN policy decoder is similar to S-RNN, as they capture the same spatio-temporal dependency. In particular, Liu et al. [20] combined S-RNN with model-free RL to obtain a structured policy for robot crowd navigation. In terms of

the underlying MDP, our GRI model is developed based on a multi-agent MDP, whereas theirs has a single robot as the agent and regards the surrounding humans as parts of the environment. In addition, we adopt a structured reward function for each agent based on the graph and introduce a relational latent space into the MDP.

2) Adversarial IRL and Imitation Learning: Now we give a brief review of related works on adversarial IRL. We also include prior works related to generative adversarial imitation learning (GAIL) [22], because GAIL is closely connected to AIRL [23]. Both methods have GANs as the backbone models, and learn the discriminator through MaxEntIRL. The difference is that GAIL uses an unstructured discriminator and does not use the generator's density.

Our work is mainly related to two categories of methods: multi-agent and latent AIRL/GAIL algorithms. Yu et al. [24] proposed a multi-agent AIRL framework for Markov games under correlated equilibrium. It is capable of modeling general heterogeneous multi-agent interactions. The PS-GAIL algorithm [25] considered a multi-agent environment in the driving domain that is similar to ours—cooperative and homogeneous agents with shared policies—and extended GAIL [22] to model the interactive behaviors. In [26], they augmented the reward in PS-GAIL as a principle manner to specify prior knowledge, which shares the same spirit with the structured reward functions in GRI.

Latent AIRL models integrate a VAE into either the discriminator or the generator for different purposes. Wang et al. [27] conditioned the discriminator on the embeddings generated by a VAE trained separately using behavior cloning. The VAE encodes trajectories into low-dimensional space, enabling the generator to produce diverse behaviors from the limited demonstration. VDB [28] constrained information contained in the discriminator's internal representation to balance the training procedure for adversarial learning algorithms. The PEMIRL framework [29] achieved meta-IRL by encoding demonstration into a contextual latent space. Though studied in a different context, PEMIRL is conceptually similar to our framework as both its generator and discriminator depend on the inferred latent variables.

3) Explainable Autonomous Driving : At the end of this section, we discuss some additional works related to explainable autonomous driving as a complement to those we have mentioned in Sec. I. They addressed some shortcomings of the discussed approaches, especially those methods based on attention mechanisms. Kim et al. [13] trained a textual explanation generator concurrently with a visual-attention-based controller in a supervised manner. It generates sentences explaining the control action as a consequence of certain objects highlighted in the attention map, which can be easily interpreted compared to visual attention. Another issue of attention that has been raised in the literature is causal confusion [30]. The model does not necessarily assign high attention weights to objects/regions that influence the control actions. In [5], a fine-grained decoder was proposed to refine visual attention maps and detect critical regions through causality tests. In [31], Li et al. adopted a similar idea for object-level reasoning. Causal inference was applied to identify risk objects

in driving scenes. One interesting observation was that the detection accuracy was improved with intervention during the training stage, i.e., augmenting the training data by masking out non-causal objects. However, intervention requires explicit prior knowledge of the causal relations to label the causal and non-causal objects in a scene. Similar to intention labels, such kinds of labels are generally prohibitive to obtain due to the intricate nature of human cognition.

III. BACKGROUND

In this section, we would like to briefly summarize two algorithms that are closely related to our approach, in order to prepare the readers for the core technical content.

A. Neural Relational Inference (NRI)

Kipf et al. [12] represent an interacting system with N objects as a complete bi-directed graph $\mathcal{G}_{\text{scene}} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} = \{v_i\}_{i=1}^N$ and edges $\mathcal{E} = \{e_{i,j} = (v_i, v_j) \mid i \neq j\}$. The edge $e_{i,j}$ refers to the one pointing from the vertex v_i to v_j . Each vertex corresponds to an object in the system. The NRI model is formalized as a VAE with a GNN encoder inferring the underlying interactions and a GNN decoder synthesizing the system dynamics given the interactions.

Formally, the model aims to reconstruct a given state trajectory, denoted by $\mathbf{x} = (\mathbf{x}^0, \dots, \mathbf{x}^{T-1})$, where T is the number of timesteps and $\mathbf{x}^t = \{\mathbf{x}_1^t, \dots, \mathbf{x}_N^t\}$. The vector $\mathbf{x}_i^t \in \mathbb{R}^n$ denotes the state vector of object v_i at time t . Alternatively, the trajectory can be decomposed into $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$, where $\mathbf{x}_i = [\mathbf{x}_i^0, \dots, \mathbf{x}_i^{T-1}]$. The encoder operates over $\mathcal{G}_{\text{scene}}$, with \mathbf{x}_i as the node feature of v_i . It infers the posterior distribution of the edge type $z_{i,j}$ for all the edges, collected into a single vector \mathbf{z} . The decoder operates over an interaction graph $\mathcal{G}_{\text{interact}}$ and reconstructs \mathbf{x} . The graph $\mathcal{G}_{\text{interact}}$ is constructed by assigning sampled \mathbf{z} to the edges of $\mathcal{G}_{\text{scene}}$ and assigning the initial state to the nodes of $\mathcal{G}_{\text{scene}}$. If $\mathcal{G}_{\text{interact}}$ represents the interactions sufficiently, the decoder should be able to reconstruct the trajectory accurately.

The model is trained by maximizing the evidence lower bound (ELBO):

$$\mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\gamma(\mathbf{x}|\mathbf{z})] - D_{KL} [q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z})],$$

where $q_\phi(\mathbf{z}|\mathbf{x})$ is the encoder output which can be factorized as:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \prod_{i=1}^N \prod_{j=1, j \neq i}^N q_\phi(z_{i,j}|\mathbf{x}), \quad (1)$$

where ϕ refers to the parameters of the encoder. The decoder output $p_\gamma(\mathbf{x}|\mathbf{z})$ can be written as:

$$p_\gamma(\mathbf{x}|\mathbf{z}) = \prod_{t=0}^{T-1} p_\gamma(\mathbf{x}^{t+1}|\mathbf{x}^t, \dots, \mathbf{x}^0, \mathbf{z}),$$

where γ refers to the parameters of the decoder.

B. Adversarial Inverse Reinforcement Learning (AIRL)

The AIRL algorithm follows the principle of maximum entropy IRL [32]. Consider an MDP defined by $(\mathcal{X}, \mathcal{A}, \mathcal{T}, r)$, where \mathcal{X}, \mathcal{A} are the state space and action space respectively. In the rest of the paper, we use \mathbf{x} and \mathbf{a} with any superscript or subscript to represent a state and action in \mathcal{X} and \mathcal{A} . \mathcal{T} is the transition operator given by $\mathbf{x}_{t+1} = f(\mathbf{a}_t, \mathbf{x}_t)$,¹ and $r : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function. The maximum entropy IRL framework assumes a suboptimal expert policy $\pi^E(\mathbf{a}|\mathbf{x})$. The demonstration trajectories generated with the expert policy, $\mathcal{D}^E = \{\tau_1^E, \dots, \tau_M^E\}$ where $\tau_i^E = (\mathbf{x}_i^{E,0}, \mathbf{a}_i^{E,0}, \dots, \mathbf{x}_i^{E,T-1}, \mathbf{a}_i^{E,T-1})$, have probabilities increasing exponentially with the cumulative reward. Concretely, they follow a Boltzmann distribution:

$$\tau_i^E \sim \pi^E(\tau) = \frac{1}{Z} \exp \left(\sum_{t=0}^{T-1} r_\lambda(\mathbf{x}_t, \mathbf{a}_t) \right),$$

where r_λ is the reward function with parameters denoted by λ . Maximum entropy IRL aims to infer the underlying reward function parameters of the expert policy. It is formalized as a maximum likelihood problem:

$$\lambda^* = \arg \max_{\lambda} \mathbb{E}_{\tau^E \sim \pi^E(\tau)} \left[\sum_{t=0}^{T-1} r_\lambda(\mathbf{x}_t^E, \mathbf{a}_t^E) \right] - \log Z.$$

To derive a feasible algorithm to solve the problem, we need to estimate the partition function Z . One practical solution is co-training a policy model with the currently estimated reward function through reinforcement learning [33]. Finn et al. [23] found the equivalency between it and a special form of the generative adversarial network (GAN). The policy model is the generator, whereas a structured discriminator is defined with the reward function to distinguish a generated trajectory τ^G from a demonstrated one τ^E . Fu et al. [34] proposed the AIRL algorithm based on it, using a discriminator that identifies generated samples based on the pairs of state and action instead of the entire trajectory to reduce variance:

$$\mathcal{D}_{\lambda, \eta}(\mathbf{x}, \mathbf{a}) = \frac{\exp \{r_\lambda(\mathbf{x}, \mathbf{a})\}}{\exp \{r_\lambda(\mathbf{x}, \mathbf{a})\} + \pi_\eta(\mathbf{a}|\mathbf{x})}, \quad (2)$$

where $\pi_\eta(\mathbf{a}|\mathbf{x})$ is the policy model with parameters denoted by η . The models $\mathcal{D}_{\lambda, \eta}$ and π_η are trained adversarially by solving the following min-max optimization problem:

$$\begin{aligned} \min_{\eta} \max_{\lambda} & \mathbb{E}_{\mathbf{x}^E, \mathbf{a}^E \sim \pi^E(\mathbf{x}, \mathbf{a})} [\log (\mathcal{D}_{\lambda, \eta}(\mathbf{x}^E, \mathbf{a}^E))] \\ & + \mathbb{E}_{\mathbf{x}^G, \mathbf{a}^G \sim \pi_\eta(\mathbf{x}, \mathbf{a})} [\log (1 - \mathcal{D}_{\lambda, \eta}(\mathbf{x}^G, \mathbf{a}^G))], \end{aligned} \quad (3)$$

where $\pi^E(\mathbf{x}, \mathbf{a})$ denotes the distribution of state and action induced by the expert policy, and $\pi_\eta(\mathbf{x}, \mathbf{a})$ is the distribution induced by the learned policy.

¹The transition is assumed deterministic to simplify the notation. A more general form of the algorithm can be derived for stochastic systems, which is essentially the same with the deterministic case.

IV. PROBLEM FORMULATION

Our GRI model grounds the relational latent space in a clustering of semantically meaningful interactions by reformulating the relational inference problem into a multi-agent IRL problem. Since the framework has the potential to be generalized to interactive systems in other domains apart from autonomous driving, we will introduce our approach in a general tone. However, it should be noted that we limit our discussion in this paper to autonomous driving problems, without claiming that it can be directly applied to other domains. GRI relies on expert domain knowledge to identify all possible semantic behaviors and design the corresponding reward functions. There exists a broad range of literature on interactive driving behavior modeling [14], [35], which we can refer to when designing the rewards. We can extend the proposed framework to other fields if proper domain knowledge is available, which is left for future investigation.

We start with modeling the interactive system as a multi-agent MDP with graph representation. As in NRI, the system has an underlying interaction graph $\mathcal{G}_{\text{interact}}$. The discrete latent variable $z_{i,j}$ takes a value from $0, 1, \dots, K - 1$, where K is the number of interactions. It indicates the type of relation between v_i and v_j with respect to its effect on v_j . Additionally, we assume the objects of the system are homogeneous intelligent agents who make decisions based on their interactions with others.

Concretely, each of them is modeled with identical state space \mathcal{X} , action space \mathcal{A} , transition operator \mathcal{T} and reward function $r : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$. At time step t , the reward of agent v_j depends on the states and actions of itself and the pairwise interactions between itself and all its neighbors:

$$\begin{aligned} r_{\xi,\psi}(v_j^t, \mathbf{z}_j) &= r_{\xi}^n(\mathbf{x}_j^t, \mathbf{a}_j^t) \\ &+ \sum_{i \in \mathcal{N}_j} \sum_{k=1}^K \mathbf{1}(z_{i,j} = k) r_{\psi_k}^{e,k}(\mathbf{x}_i^t, \mathbf{a}_i^t, \mathbf{x}_j^t, \mathbf{a}_j^t), \end{aligned} \quad (4)$$

where \mathbf{z}_j is the collection of $\{z_{i,j}\}_{i \in \mathcal{N}_j}$, r_{ξ}^n is the node reward function parameterized by ξ , \mathcal{N}_j is the set of v_j 's neighbouring nodes, $\mathbf{1}$ is the indicator function, and $r_{\psi_k}^{e,k}$ is the edge reward function parameterized by ψ_k for the k^{th} type of interaction. We utilize expert domain knowledge to design $r_{\psi_k}^{e,k}$, so that the corresponding interactive behavior emerges by maximizing the rewards. Particularly, the edge reward equals zero for $k = 0$, indicating that the action taken by v_j does not depend on its interaction with v_i .

We assume the agents act cooperatively to maximize the cumulative reward of the system:

$$\mathcal{R}_{\xi,\psi}(\tau, \mathbf{z}) = \sum_{t=0}^{T-1} \mathbf{r}_{\xi,\psi}(\mathbf{x}^t, \mathbf{a}^t, \mathbf{z}) = \sum_{t=0}^{T-1} \sum_{j=1}^N r_{\xi,\psi}(v_j^t, \mathbf{z}_j),$$

with a joint policy denoted by $\pi_{\eta}(\mathbf{a}^t | \mathbf{x}^t, \mathbf{z})$. The cooperative assumption does not generally hold for real-world traffic scenarios. Each individual driver may value the utilities differently. In practice, though, it has been shown that realistic highway driving behavior can be learned from real-world data under the cooperative assumption [26], [36]. It implies that

the assumption can approximately apply in practical circumstances, especially in the highway scenarios tested in this work. Thus, we adopt the cooperative assumption to simplify the problem formulation and training procedure. In future work, we will investigate a non-cooperative extension of the proposed method to tackle interactive traffic scenarios where the cooperative assumption has to be removed [24].

Given a demonstration dataset, we aim to infer the underlying reward function and policy. Different from a typical IRL problem, both $r_{\xi,\psi}$ and π_{η} depend on \mathbf{z} . Therefore, we need to infer the distribution $p(\mathbf{z}|\tau)$ to solve the IRL problem.

V. GROUNDED RELATIONAL INFERENCE

We now present the Grounded Relational Inference model to solve the IRL problem specified in Sec. IV. The model consists of three modules modeled by message-passing GNNs [37]: an encoder inferring the posterior distribution of edge types, a policy decoder generating control actions conditioned on the edge variables sampled from the posterior distribution, and a reward decoder modeling the rewards conditioned on the inferred edge types.

A. Architecture

The overall model structure is illustrated in Fig. 2. Given a demonstration trajectory $\tau^E \in \mathcal{D}^E$, the encoder operates over $\mathcal{G}_{\text{scene}}$ and approximates the posterior distribution $p(\mathbf{z}|\tau^E)$ with $q_{\phi}(\mathbf{z}|\tau^E)$. Following NRI, we parameterize the distribution of each edge latent variable with a softmax function, e.g., $q_{\phi}(z_{i,j}|\tau^E) = \text{softmax}(f_{\text{enc},\phi}(\tau^E)_{i,j})$, where $f_{\text{enc},\phi}(\cdot)$ denotes the encoder GNN. The policy decoder operates over a $\mathcal{G}_{\text{interact}}$ sampled from the inferred $q_{\phi}(\mathbf{z}|\tau^E)$ and models the policy $\pi_{\eta}(\mathbf{a}^t | \mathbf{x}^t, \mathbf{z})$. The policy distribution is parameterized as a diagonal Gaussian distribution whose mean and variance are generated by the decoder GNN. Given an initial state, we can generate a trajectory by sequentially sampling \mathbf{a}^t from $\pi_{\eta}(\mathbf{a}^t | \mathbf{x}^t, \mathbf{z})$ and propagating the state. The state is propagated with either the transition operator \mathcal{T} if given, or a simulating environment if \mathcal{T} is not accessible. We denote a generated trajectory given the initial state of τ^E as τ^G . Since these two modules are essentially the same in NRI, we omit the detailed model structures here and include them in Appx. A.

The reward decoder computes the reward of a state-action pair given the sampled edge variables. We use it to compute the cumulative rewards of τ^G and τ^E conditioned on the sampled $\mathcal{G}_{\text{interact}}$. The reward decoder is in the form of Eqn. (4). Additionally, we augment the functions r_{ξ}^n and $r_{\psi_k}^{e,k}$ with MLP shaping terms to mitigate the reward shaping effect [34], resulting in:

$$f_{\xi,\omega}^n(\mathbf{x}_j^t, \mathbf{a}_j^t, \mathbf{x}_j^{t+1}) = r_{\xi}^n(\mathbf{x}_j^t, \mathbf{a}_j^t) + h_{\omega}^n(\mathbf{x}_j^{t+1}) - h_{\omega}^n(\mathbf{x}_j^t), \quad (5)$$

and

$$\begin{aligned} f_{\psi_k,\chi_k}^{e,k}(\mathbf{x}_i^t, \mathbf{a}_i^t, \mathbf{x}_i^{t+1}, \mathbf{x}_j^t, \mathbf{a}_j^t, \mathbf{x}_j^{t+1}) &= r_{\psi_k}^{e,k}(\mathbf{x}_i^t, \mathbf{a}_i^t, \mathbf{x}_j^t, \mathbf{a}_j^t) \\ &+ h_{\chi_k}^{e,k}(\mathbf{x}_i^{t+1}, \mathbf{x}_j^{t+1}) \\ &- h_{\chi_k}^{e,k}(\mathbf{x}_i^t, \mathbf{x}_j^t), \end{aligned} \quad (6)$$

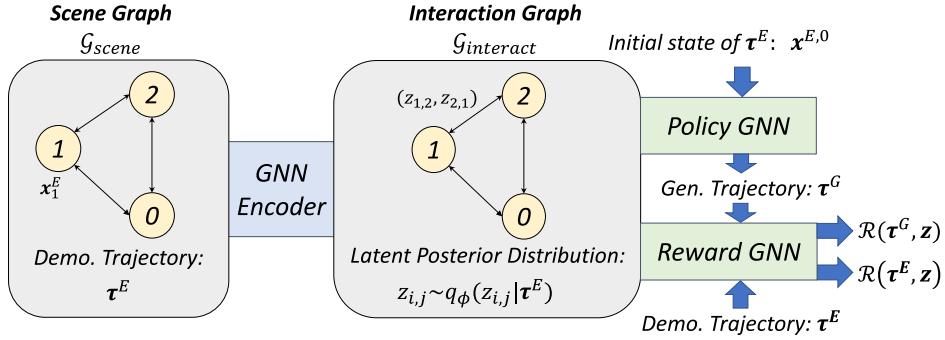


Fig. 2. Architecture of grounded relational inference model. Given a demonstration trajectory $\tau^E \in \mathcal{D}^E$, the encoder operates over $\mathcal{G}_{\text{scene}}$ and approximates the distribution $p(\mathbf{z}|\tau^E)$ with $q_\phi(\mathbf{z}|\tau^E)$. The policy decoder operates over a $\mathcal{G}_{\text{interact}}$ sampled from the inferred $q_\phi(\mathbf{z}|\tau^E)$ and models the policy $\pi_\eta(\mathbf{a}^t|\mathbf{x}^t, \mathbf{z})$. Given the initial state of τ^E , we sample a trajectory τ^G by sequentially sampling \mathbf{a}^t from $\pi_\eta(\mathbf{a}^t|\mathbf{x}^t, \mathbf{z})$ and propagating the state. Finally, We use the reward GNN to compute the cumulative rewards of τ^G and τ^E conditioned on the sampled $\mathcal{G}_{\text{interact}}$.

where h_ω^n and $h_{\chi k}^{e,k}$ are MLPs with parameters denoted by ω and χ respectively. We denote the shaped reward function of agent v_j by $\mathbf{f}_{\xi, \omega, \psi, \chi}(\mathbf{x}^t, \mathbf{a}^t, \mathbf{x}^{t+1}, \mathbf{z})$, which equals to the left hand side of Eqn. (4) but with r_ξ^n and $r_{\psi k}^{e,k}$ substituted by the augmented rewards. The shaped reward function, together with the policy model, defines the discriminator which distinguishes τ^G from τ^E :

$$\begin{aligned} & \mathcal{D}_{\xi, \omega, \psi, \chi, \eta}(\mathbf{x}^t, \mathbf{a}^t, \mathbf{x}^{t+1}, \mathbf{z}) \\ &= \frac{\exp \{ \mathbf{f}_{\xi, \omega, \psi, \chi}(\mathbf{x}^t, \mathbf{a}^t, \mathbf{x}^{t+1}, \mathbf{z}) \}}{\exp \{ \mathbf{f}_{\xi, \omega, \psi, \chi}(\mathbf{x}^t, \mathbf{a}^t, \mathbf{x}^{t+1}, \mathbf{z}) \} + \pi_\eta(\mathbf{a}^t|\mathbf{x}^t, \mathbf{z})}. \end{aligned}$$

B. Training

We aim to train the three modules simultaneously. Consequently, we incorporate the encoder model $q_\phi(\mathbf{z}|\tau^E)$ into the objective function of AIRL, resulting in the optimization problem (8), as shown at the bottom of the next page. The encoder is integrated into the minimization problem because the reward function directly depends on the latent space. The model is then trained by solving the problem (8) in an adversarial scheme: we alternate between training the encoder and reward for the minimization problem and training the policy for the maximization problem. Specifically, the objective for the encoder and reward is the following minimization problem given fixed η :

$$\begin{aligned} & \min_{\xi, \omega, \psi, \chi, \phi} \quad \mathcal{J}(\xi, \omega, \psi, \chi, \phi, \eta) \\ & \text{s.t. } \mathbb{E} \{ D_{KL} [q_\phi(\mathbf{z}|\tau^E) || p(\mathbf{z})] \} \leq I_c. \end{aligned} \quad (7)$$

The objective for the policy is maximizing $\mathcal{J}(\xi, \omega, \psi, \chi, \phi, \eta)$ with fixed ξ, ω, ψ, χ and ϕ .

The objective function in the problem (8) is essentially the expectation of the objective function in the problem (3) over the inferred posterior distribution $q_\phi(\mathbf{z}|\tau^E)$ and the demonstration distribution $\pi^E(\tau)$. The constraint enforces an upper bound I_c on the KL-divergence between $q_\phi(\mathbf{z}|\tau^E)$ and the prior distribution $p(\mathbf{z})$. A sparse prior is chosen to encourage sparsity in $\mathcal{G}_{\text{interact}}$. It has a similar regularization effect as the D_{KL} term in ELBO. We borrow its format from variational discriminator bottleneck (VDB) [28]. VDB improves adversarial training by constraining the information

flow from the input to the discriminator. The KL-divergence constraint is derived as a variational approximation to the information bottleneck [38]. Although having different motivations, we adopt it for two reasons. First, the proposed model is not generative because our goal is not synthesizing trajectories from the prior $p(\mathbf{z})$, but inferring the posterior $p(\mathbf{z}|\tau^E)$. Therefore, regularization derived from information bottleneck is more sensible compared to ELBO. Second, the constrained problem (7) can be relaxed by introducing a Lagrange multiplier β . During training, β is updated through dual gradient descent as follows:

$$\beta \leftarrow \max (0, \alpha_\beta (\mathbb{E} \{ D_{KL} [q_\phi(\mathbf{z}|\tau^E) || p(\mathbf{z})] \} - I_c)) \quad (9)$$

We find the adaptation scheme particularly advantageous. The model can focus on inferring \mathbf{z} for reward learning after satisfying the sparsity constraint because the magnitude of β decreases towards zero once the constraint is satisfied. However, it is worth noting that our framework does not rely on the bottleneck constraint to induce a semantically meaningful latent space as in [39]. In contrast, GRI relies on structured reward functions to ground the latent space into semantic interactive behaviors. The bottleneck serves as a regularization to find out the minimal interaction graph to represent the interactions. In fact, we trained the baseline NRI models with the same constraints and weight update scheme. The experimental results show that the constraint itself is not sufficient to induce a sparse interaction graph.

In general, when the dynamics \mathcal{T} is unknown or non-differentiable, maximum entropy RL algorithms [40] are adopted to optimize the policy. In this work, we assume known and differentiable dynamics, which is a reasonable assumption for the investigated scenarios. It allows us to directly back-propagate through the trajectory for gradient estimation, which simplifies the training procedure.

VI. EXPERIMENTS

We evaluate the proposed GRI model on a synthetic dataset as well as a naturalistic traffic dataset. The synthetic data are generated using policy models trained given the ground-truth reward function and interaction graph. We intend to verify if GRI can induce a semantically meaningful relational latent

space and infer the underlying relations precisely. The naturalistic traffic data are extracted from the NGSIM dataset. We aim to validate if GRI can model real-world traffic scenarios effectively with the grounded latent space. Unlike synthetic agents, we do not have the privilege to access the ground-truth graphs governing human drivers' interactions. Instead, we construct hypothetical graphs after analyzing the segmented data. The hypotheses reflect humans' understanding of traffic scenarios. Moreover, the hypothetical graphs are built upon a set of interactive behavior whose characteristics are described by the designed reward functions. We would like to see if the reward functions can incorporate the semantic information into the latent space, and let GRI model real-world interactive systems in the same way as humans. In each setting, we consider two scenarios, car-following, and lane-changing.

A. Baselines

The main question of interest is whether GRI can induce semantically meaningful interaction graphs. To answer the question, the most important baseline model for comparison is NRI, because GRI shares the same prior distribution of latent variables with NRI. Comparing the posterior distributions provides insights into whether the structured reward functions can ground the latent space into semantic interactive behaviors. In each experiment, the baseline NRI model has the same encoder and policy decoder as the GRI model. Besides, as stated in Sec. V, the same bottleneck constraint and the weight update scheme in Eqn. (9) were applied as regularization for minimal representation.

Another model for comparison is a supervised policy decoder. We assume that ground-truth graphs or human hypotheses are available. Therefore, we can directly train a policy decoder in a supervised way. The ground-truth graph is fed to the policy decoder as a substitute for the interaction graph sampled from the encoder output $q_\phi(\mathbf{z}|\tau^E)$. The training of the decoder becomes a simple regression problem. We used mean square error as the loss function to train it.

As additional information is granted, it is unfair to directly compare the performance of GRI with the supervised policy model. Since the supervised model is trained with the ground-truth interaction graphs governing the systems, it is expected to achieve smaller reconstruction errors. However, the supervised baseline provides some useful insights. In the naturalistic traffic scenarios, the supervised model gives us some insights into whether the human hypotheses are reasonable. If the supervised model can reconstruct the trajectories precisely, it will justify our practice to adopt graph accuracy as one of the evaluation metrics.

More importantly, in Sec. VI-E, we demonstrate that GRI's latent space still maintains its semantic meaning under some perturbations to the initial states, whereas the decoders of baseline models fail to synthesize those behaviors under the same perturbations, including the supervised policy decoder which is trained with the ground-truth interaction graphs. It supports our argument that direct supervision via interaction labels is not sufficient to guide the policy to synthesize behaviors with correct semantic meaning.

There exist other alternatives for the purpose of trajectory reconstruction. However, it is not our goal in this paper to find an expressive model for accurate reconstruction. Therefore, we do not consider other baselines from this perspective. For the task of grounding the latent space into semantic interactive driving behaviors, we did not find any exact alternatives in the literature. For the specific scenarios studied in this paper, we may design some rule-based approaches to directly infer the interaction graph. However, it is difficult to decide the parameters that best describe the interactive behaviors, because there is a spectrum in how people follow the rules [41]. In this paper, we are interested in a data-driven module that can be incorporated into an end-to-end learning model and has the potential to be generalized to complicated driving scenarios and systems in other domains. Apart from GRI, a potential alternative solution could be adopting a differentiable logic module. For instance, Leung et al. [41] proposed a differentiable parametric Signal Temporal Logic formula (pSTL) which could be learned from data. We will investigate this direction in our future works.

B. Evaluation Metrics

To evaluate a trained model, we sample a τ^E from the test dataset and extract the maximum posterior probability (MAP) estimate of edge variables, $\hat{\mathbf{z}}$, from $q_\phi(\mathbf{z}|\tau^E)$. Afterward, we obtain a single sample of trajectories $\hat{\tau}$ by executing the mean value of the policy output. The root mean square errors (RMSE) of states and the accuracy of $\mathcal{G}_{\text{interact}}$ are selected as the evaluation metrics, which are computed based on $\hat{\mathbf{z}}$, $\hat{\tau}$, τ^E , and the ground truth or hypothetical latent variables denoted by \mathbf{z}^E :

$$\begin{aligned} \text{RMSE}_\epsilon &= \sqrt{\frac{1}{(N-1)T} \sum_{j=1}^N \sum_{t=0}^{T-1} (\epsilon_j^{E,t} - \hat{\epsilon}_j^t)^2}, \\ \text{Accuracy} &= \frac{\sum_{i=1}^N \sum_{j=1, j \neq i}^N \mathbf{1}(z_{i,j}^E = \hat{z}_{i,j})}{N(N-1)}. \end{aligned}$$

$$\begin{aligned} \max_{\eta} \min_{\xi, \omega, \psi, \chi, \phi} \quad & \mathcal{J}(\xi, \omega, \psi, \chi, \phi, \eta) = \mathbb{E}_{\tau^E \sim \pi^E(\tau)} \left\{ \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\tau^E)} \left[- \sum_{t=0}^{T-1} \log \mathcal{D}_{\xi, \omega, \psi, \chi, \eta}(\mathbf{x}^{E,t}, \mathbf{a}^{E,t}, \mathbf{x}^{E,t+1}, \mathbf{z}) \right. \right. \\ & \left. \left. - \mathbb{E}_{\tau^G \sim \pi_\eta(\tau|\mathbf{z})} \sum_{t=0}^{T-1} \log \left(1 - \mathcal{D}_{\xi, \omega, \psi, \chi, \eta}(\mathbf{x}^{G,t}, \mathbf{a}^{G,t}, \mathbf{x}^{G,t+1}, \mathbf{z}) \right) \right] \right\}, \\ \text{s.t.} \quad & \mathbb{E}_{\tau^E \sim \pi^E(\tau)} \{ D_{KL}[q_\phi(\mathbf{z}|\tau^E) || p(\mathbf{z})] \} \leq I_c, \end{aligned} \quad (8)$$

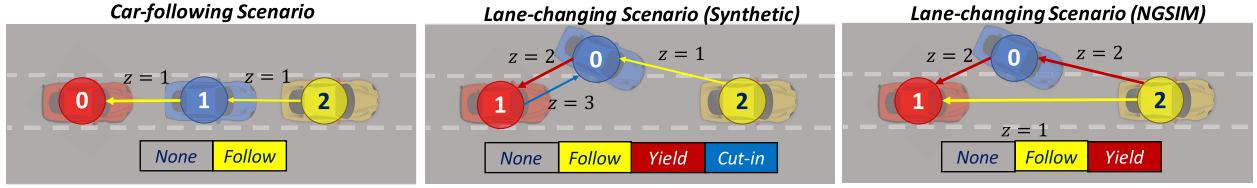


Fig. 3. Test scenarios with the underlying interaction graphs. In the synthetic scenarios, the graphs are the ground-truth ones governing the synthetic experts. In the naturalistic traffic scenarios, the graphs are human hypotheses reflecting humans' understanding of the traffic scenarios.

If multiple edge types exist, we test all the possible permutations of edge types and report the one with the highest graph accuracy for NRI.

It is worth noting that the graph accuracy on the naturalistic traffic dataset merely quantifies the divergence between the inferred graphs and the hypotheses we construct. We anticipate that GRI can attain a higher accuracy than NRI. It will imply that we can incorporate human domain knowledge into GRI and induce a semantic relational latent space consistent with the hypotheses built upon the same domain knowledge. However, a low graph accuracy does not necessarily mean that humans cannot interpret the inferred graphs well. The hypothetical graphs represent one perspective to interpret the interactive scenes. It is possible that NRI may find another sensible way to categorize and interpret the interactions, which can also be understood by humans.

To further study the explainability of the learned latent spaces, we want to look into the inferred graphs and have a qualitative comparison between the latent spaces learned by the two models. For each setting, we compute the distribution of estimated edge variables $\hat{\mathbf{z}}$ over the test dataset. As in [12], we visualize the results in multiple adjacency matrices corresponding to different edge types. In the adjacency matrix corresponding to the k^{th} type of interaction, the element $A_{i,j}$ indicates the relative frequency of $\hat{z}_{j,i} = k$, where $\hat{z}_{j,i}$ is the latent variable for the edge from node j to node i . In other words, $A_{i,j}$ equals the ratio of test samples where the model infers $\hat{z}_{j,i} = k$. By inspecting the edge-type distributions, we can get some extra insights into the explainability of the two models beyond the quantitative metrics.

C. Synthetic Scenes

As mentioned above, we designed two synthetic scenarios, car-following and lane-changing. The two scenes and their underlying interaction graphs are illustrated in Fig. 3. In both scenarios, we have a leading vehicle whose behavior does not depend on the others. Its trajectory is given without the need for reconstruction. We assume it runs at constant velocity. The other vehicles interact with each other and the leader in different ways. In the car-following scene, we model the system with two types of edges: $z_{i,j} = 1$ means that Vehicle j follows Vehicle i ; $z_{i,j} = 0$ means that Vehicle j does not interact with Vehicle i . In the lane-changing scene, two additional edge types are introduced: $z_{i,j} = 2$ means that Vehicle j yields to Vehicle i ; $z_{i,j} = 3$ means that Vehicle j cuts in front of Vehicle i .

The MDPs for the tested scenarios are specified as follows. In the car-following scene, since the vehicles mainly interact

in the longitudinal direction, we only model their longitudinal dynamics to simplify the problem. For all $j \in \{1, 2, 3\}$, the state vector of Vehicle j consists of three states: $\mathbf{x}_j^t = [x_j^t \ v_j^t \ a_j^t]^T$, where x_j^t is the longitudinal coordinate, v_j^t is the velocity, and a_j^t is the acceleration. There is only one control input which is the jerk. We denote it as δa_j^t . The dynamics are governed by a 1D point-mass model:

$$\begin{aligned} x_j^{t+1} &= x_j^t + v_j^t \Delta t + \frac{1}{2} a_j^t \Delta t^2, \\ v_j^{t+1} &= v_j^t + a_j^t \Delta t, \\ a_j^{t+1} &= a_j^t + \delta a_j^t \Delta t, \end{aligned}$$

where Δt is the sampling time. Note that the dynamics of x_j^t is discretized with zero-order hold discretization so that the longitudinal acceleration can directly affect x_j^t without delay. In practice, we find GRI benefits from reducing the control delay as the longitudinal coordinates under different policies will diverge at earlier time steps. It is then easier for GRI to infer the latent interaction types. In the lane-changing scene, we consider both longitudinal and lateral motions. The state vector consists of six states instead: $\mathbf{x}_j^t = [x_j^t \ y_j^t \ v_j^t \ \theta_j^t \ a_j^t \ \omega_j^t]^T$. The three additional states are the lateral coordinate y_j^t , the yaw angle θ_j^t , and the yaw rate ω_j^t . There is one additional action which is the yaw acceleration, denoted by $\delta \omega_j^t$. We model the vehicle as a Dubins' car:

$$\begin{aligned} x_j^{t+1} &= x_j^t + v_j^t \cos(\theta_j^t) \Delta t, \\ y_j^{t+1} &= y_j^t + v_j^t \sin(\theta_j^t) \Delta t, \\ v_j^{t+1} &= v_j^t + a_j^t \Delta t, \\ \theta_j^{t+1} &= \theta_j^t + \omega_j^t \Delta t, \\ a_j^{t+1} &= a_j^t + \delta a_j^t \Delta t, \\ \omega_j^{t+1} &= \omega_j^t + \delta \omega_j^t \Delta t. \end{aligned}$$

The structured reward functions were designed based on expert domain knowledge (e.g. transportation studies [35], [42]). We mainly referred to [14] and [16] in this paper. For the car-following behavior, its reward function is defined as follows:

$$\begin{aligned} r_{\psi_1}^{e,1} (\mathbf{x}_i^t, \mathbf{x}_j^t) &= -(1 + \exp(\psi_{1,0})) g_{\text{IDM}}(\mathbf{x}_i^t, \mathbf{x}_j^t) \\ &\quad - (1 + \exp(\psi_{1,1})) g_{\text{dist}}(\mathbf{x}_i^t, \mathbf{x}_j^t) \\ &\quad - (1 + \exp(\psi_{1,2})) g_{\text{lat}}(\mathbf{x}_i^t, \mathbf{x}_j^t), \end{aligned}$$

where the features are defined as:

$$g_{\text{IDM}}(\mathbf{x}_i^t, \mathbf{x}_j^t) = \left(\max(x_i^t - x_j^t, 0) - \Delta x_{i,j}^{\text{IDM}, t} \right)^2, \quad (10)$$

TABLE I
PERFORMANCE COMPARISON ON SYNTHETIC DATASET

Model	Car Following ($\Delta t = 0.2s, T = 20$)			Lane Changing ($\Delta t = 0.2s, T = 30$)			
	RMSE _x (m)	RMSE _v (m/s)	Accuracy(%)	RMSE _x (m)	RMSE _y (m)	RMSE _v (m/s)	Accuracy(%)
GRI	0.241 \pm 0.125	0.174 \pm 0.068	100.00 \pm 0.00	0.529 \pm 0.230	0.207 \pm 0.046	0.303 \pm 0.128	99.95 \pm 0.01
NRI	0.047 \pm 0.024	0.056 \pm 0.015	66.70 \pm 0.00	0.109 \pm 0.045	0.155 \pm 0.038	0.061 \pm 0.016	55.9 \pm 7.98
Supervised	0.039 \pm 0.016	0.050 \pm 0.009	-	0.062 \pm 0.027	0.145 \pm 0.035	0.048 \pm 0.011	-

¹ The data is presented in form of mean \pm std.

$$g_{\text{dist}}(\mathbf{x}_i^t, \mathbf{x}_j^t) = \exp \left(-\frac{\left(\max \left(x_i^t - x_j^t, 0 \right) \right)^2}{\zeta^2} \right),$$

$$g_{\text{lat}}(\mathbf{x}_i^t, \mathbf{x}_j^t) = \left(y_j^t - g_{\text{center}}(y_i^t) \right)^2. \quad (11)$$

The feature g_{IDM} suggests a spatial headway $\Delta x_{i,j}^{\text{IDM},t}$ derived from the intelligent driver model (IDM) [35]. The feature f_{dist} ensures a minimum collision-free distance. We penalize the following vehicle for surpassing the preceding one with the help of $x_{i,j}^{\text{IDM},t}$ in Eqn. (10) and Eqn. (11). The last feature g_{lat} exists only in lane-changing. It regulates the following vehicle to stay in the same lane as the preceding one with the help of g_{center} , which determines the lateral coordinate of the corresponding centerline based on the position of the preceding vehicle. Altogether, the features define the following behavior as staying in the same lane as the preceding vehicle whereas keeping a safe longitudinal headway.

The reward function for yielding is defined as:

$$r_{\psi_2}^{e,2} \left(\mathbf{x}_i^t, \mathbf{x}_j^t \right) = -(1 + \exp(\psi_{2,0})) g_{\text{yield}}(\mathbf{x}_i^t, \mathbf{x}_j^t) \\ - (1 + \exp(\psi_{2,1})) g_{\text{dist}}(\mathbf{x}_i^t, \mathbf{x}_j^t).$$

The feature g_{dist} is defined in Eqn. (11). The other feature g_{yield} suggests an appropriate spatial headway for yielding:

$$g_{\text{yield}}(\mathbf{x}_i^t, \mathbf{x}_j^t) = \mathbf{1} \left(g_{\text{center}}(y_j^t) = g_{\text{center}}(y_i^t) \right) g_{\text{IDM}}(\mathbf{x}_i^t, \mathbf{x}_j^t) \\ + \mathbf{1} \left(g_{\text{center}}(y_j^t) \neq g_{\text{center}}(y_i^t) \right) g_{\text{goal}}(\mathbf{x}_i^t, \mathbf{x}_j^t),$$

$$g_{\text{goal}}(\mathbf{x}_i^t, \mathbf{x}_j^t) = \left(\max \left(x_i^t - x_j^t - \Delta x_{i,j}^{\text{yield}}, 0 \right) \right)^2. \quad (12)$$

The suggested headway is set to be a constant value, $\Delta x_{i,j}^{\text{yield}}$, when the other vehicle is merging, and switches to $\Delta x_{i,j}^{\text{IDM},t}$ once the merging vehicle enters into the same lane, where its behavior becomes consistent with car-following. We follow [14] to adopt different reward functions depending on the lanes where the vehicles are located. Merging occurs during a short period of time. Therefore, we assume the driver sets a fixed short-term goal distance as in [14] and then transits to the following behavior afterward.

The reward function for cutting-in is quite similar:

$$r_{\psi_3}^{e,3} \left(\mathbf{x}_i^t, \mathbf{x}_j^t \right) = -(1 + \exp(\psi_{3,0})) g_{\text{goal}}(\mathbf{x}_j^t, \mathbf{x}_i^t) \\ - (1 + \exp(\psi_{3,1})) g_{\text{dist}}(\mathbf{x}_j^t, \mathbf{x}_i^t),$$

where the features are defined as in Eqn. (11) and Eqn. (12), but with the input arguments switched, because the merging vehicle should stay in front of the yielding one.

Apart from the edge rewards, all the agents share the same node reward function. The following one is adopted for lane-changing:

$$r_{\xi}^n(\mathbf{x}_j^t, \mathbf{a}_j^t) = -(1 + \exp(\xi_0)) f_v(\mathbf{x}_j^t) \\ - (1 + \exp(\xi_{1:3}))^T f_{\text{state}}(\mathbf{x}_j^t) \\ - (1 + \exp(\xi_{4:5}))^T f_{\text{action}}(\mathbf{a}_j^t) \\ - (1 + \exp(\xi_6)) f_{\text{lane}}(\mathbf{x}_j^t),$$

where f_{state} and f_{action} take the element-wise square of $[a_j^t \theta_j^t \omega_j^t]$ and $[\delta a_j^t \delta \omega_j^t]$ respectively. It penalizes large control inputs as well as drastic longitudinal and angular motions to induce smooth and comfortable maneuvers. The feature f_v is the squared error between v_j^t and the speed limit v_{lim} . It regulates the vehicles to obey the speed limit. The last term f_{lane} penalizes the vehicle for staying close to the lane boundaries. For car-following, we simply remove those terms that are irrelevant in 1D motion. In all the reward functions, the parameters collected in ψ and ξ are unknown during training and inferred by GRI. We take the exponents of them and add one to the results. It enforces the model to use the features when modeling the corresponding interactions.

With the scenarios defined above, we aim to generate one dataset for each scenario. For each scenario, we randomly sampled the initial states of the vehicles and trained an expert policy given the ground-truth reward functions and the interaction graph. Afterward, we used the trained policy to generate the dataset. The same sampling scheme was used to sample the initial states.

Results: On each dataset, we trained a GRI model with the policy decoder (18)-(20) introduced in Appx. A. The results are summarized in Table I. The NRI model can reconstruct the trajectories with errors close to the supervised policy. However, it learns a relational latent space that is different from the one governing the demonstration; Therefore, the edge variables cannot be interpreted as those semantic interactive behaviors. In contrast, our GRI model interprets the interactions consistently with the domain knowledge inherited in the demonstration and recovers the interaction graph with high accuracy. It has larger reconstruction errors compared to the baseline approaches. However, it still sufficiently recovers the interactive behaviors, and the reconstructed trajectories are sensible (see Appx. B).

We computed the empirical distribution of the estimated edge variables \hat{z} over the test dataset. The results are summarized in Fig. 4. The distribution concentrates on a single interaction graph for both models in both scenarios—as opposed to the case on the naturalistic traffic dataset introduced in the next section—because the synthetic agents have

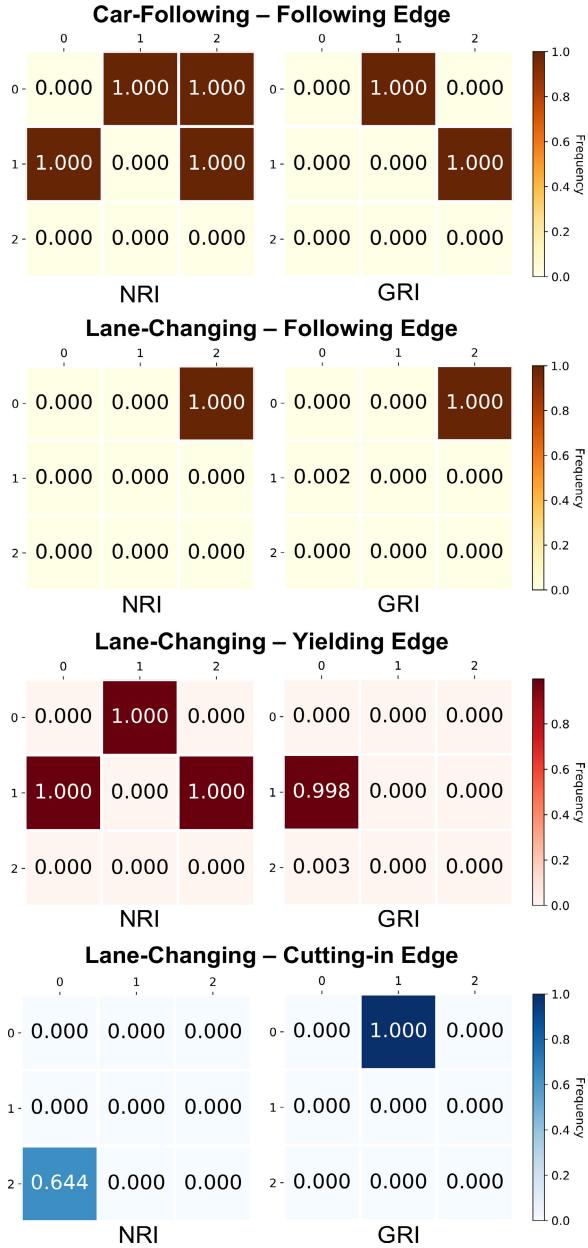


Fig. 4. The empirical distribution of estimated edge variables \hat{z} over the test dataset in the synthetic scenarios. We summarize the results in multiple adjacency matrices corresponding to different edge types. In the adjacency matrix corresponding to the k^{th} type of interaction, the element $A_{i,j}$ indicates the relative frequency of $\hat{z}_{j,i} = k$, where $\hat{z}_{j,i}$ is the latent variable for the edge from node j to node i .

consistent interaction patterns over all the samples. We observe that NRI learns symmetric relations: In both scenarios, the NRI model assigns the same edge types to the edges $e_{0,1}$ and $e_{1,0}$. It is difficult to interpret their semantic meaning because those pairwise interactions are asymmetric in our synthetic scenes. In contrast, the reward functions in our GRI model enforce an asymmetric relational latent space.

D. Naturalistic Traffic Scenes

To evaluate the proposed method in real-world traffic scenarios, we investigated the same scenarios as in the synthetic case, car-following and lane-changing. we segmented data

from the Highway-101 and I-80 datasets of NGSIM. Afterward, we further screened the data to select those interactive samples and ensure that no erratic swerving or multiple lane changes occurred. Unlike synthetic agents, human agents do not have a ground-truth interaction graph that governs their interactions. Instead, we constructed hypothetical $\mathcal{G}_{\text{interact}}$ after analyzing the segmented data. The hypotheses for the two scenarios are depicted in Fig. 3. The one for car-following is identical to the ground-truth interaction graph we designed for the synthetic agents. However, we proposed a different hypothesis for lane-changing. We excluded the cutting-in relation to reduce the number of edge types and therefore simplify the training procedure. Moreover, we differentiated distinct interactions according to the vehicle's lateral position. We say that a vehicle yields to its preceding vehicle if they drive in neighboring lanes, whereas it follows the preceding one if they drive in the same lane.

As in the synthetic scenes, the trajectory of the leading vehicle is given without the need for reconstruction. We feed the ground-truth state of the leading vehicle sequentially to the policy decoder when decoding the trajectories of the other vehicles. This practice enables us to heuristically isolate a small interacting group out of the numerous number of vehicles on the highway. While the leading vehicle's behavior depends on the other vehicles, it is fairly reasonable to assume that the behavior of the modeled following vehicles is independent of other surrounding vehicles on the road after conditioning on the trajectory of the leading vehicle. Even though there may still exist other surrounding vehicles interacting with them, their influence should be subtle. The models should be able to well capture the interactions among the modeled subset while marginalizing those subtle effects.

The node dynamics are the same as in the synthetic scene for car-following. For lane-changing, since we did not have accurate heading information, we adopted a 2D point-mass model instead. Since the behavior of human drivers is much more complicated than the synthetic agents, we designed reward functions with larger model capacities using neural networks. In car-following, the reward functions are defined as follows:

$$\begin{aligned} r_{\psi_1}^{e,1} \left(\mathbf{x}_i^t, \mathbf{x}_j^t \right) &= -(1 + \exp(\psi_{1,0})) g_v^{\text{NN}}(\mathbf{x}_i^t, \mathbf{x}_j^t) \\ &\quad - (1 + \exp(\psi_{1,1})) g_s^{\text{NN}}(\mathbf{x}_i^t, \mathbf{x}_j^t), \\ r_{\xi}^n \left(\mathbf{x}_i^t, \mathbf{a}_j^t \right) &= -(1 + \exp(\xi_0)) f_v^{\text{NN}}(\mathbf{x}_j^t) \\ &\quad - (1 + \exp(\xi_1)) f_{\text{acc}}(\mathbf{x}_j^t) \\ &\quad - (1 + \exp(\xi_2)) f_{\text{jerk}}(\mathbf{x}_j^t, \mathbf{a}_j^t), \end{aligned}$$

where the features are defined as:

$$\begin{aligned} f_v^{\text{NN}}(\mathbf{x}_j^t) &= \left(v_j^t - h_1(\mathbf{x}_j^t) \right)^2, \\ g_v^{\text{NN}}(\mathbf{x}_i^t, \mathbf{x}_j^t) &= \left(v_i^t - h_2(\mathbf{x}_i^t, \mathbf{x}_j^t) \right)^2, \\ g_s^{\text{NN}}(\mathbf{x}_i^t, \mathbf{x}_j^t) &= \text{ReLU} \left(h_3 \left(\mathbf{x}_i^t, \mathbf{x}_j^t \right) - x_i^t + x_j^t \right)^2. \end{aligned}$$

The features f_{acc} and f_{jerk} penalize the squared magnitude of acceleration and jerk to induce smooth and comfortable maneuver. The functions h_1 , h_2 , and h_3 are neural networks

TABLE II
PERFORMANCE COMPARISON ON NATURALISTIC TRAFFIC DATASET

Model	Car Following ($\Delta t = 0.2s, T = 30$)			Lane Changing ($\Delta t = 0.2s, T = 40$)			
	RMSE _x (m)	RMSE _v (m/s)	Accuracy(%)	RMSE _x (m)	RMSE _y (m)	RMSE _v (m/s)	Accuracy(%)
GRI	1.700 \pm 1.005	0.721 \pm 0.363	100.00 \pm 0.00	7.118 \pm 3.647	0.764 \pm 0.336	4.320 \pm 2.392	98.55 \pm 0.06
NRI	1.436 \pm 0.880	0.650 \pm 0.328	64.09 \pm 0.08	6.532 \pm 3.822	0.330 \pm 0.181	4.291 \pm 2.544	28.98 \pm 0.08
Supervised	1.482 \pm 0.938	0.665 \pm 0.344	-	5.897 \pm 3.651	0.323 \pm 0.223	4.307 \pm 2.435	-

¹ The data is presented in form of mean \pm std.

with ReLU output activation. The feature g_s^{NN} is the critical component that shapes the car-following behavior. It learns a non-negative reference headway and penalizes the following vehicle for violating it. The feature g_v^{NN} and f_v^{NN} suggest reference velocities considering interaction and merely itself respectively. The edge reward function has a large modeling capacity because we let it learn adaptive reference headway and velocity from data. Nevertheless, it still defines the fundamental characteristic of the following behavior, which is always staying behind the preceding vehicle.

In lane-changing, the node reward function and the edge reward function for the following behavior are similar to those in the car-following scenario. The node reward function has an additional term for the lateral position, which encourages the vehicles to drive on the target lane, i.e., the lane where the leading vehicle is driving. It also has additional terms to penalize the magnitude of lateral velocity and acceleration to induce comfortable maneuvers. To design the yielding reward, we define a collision point of two vehicles based on their states. We approximate the vehicles' trajectories as piecewise-linear between sequential timesteps and compute the collision point as the intersection between their trajectories (Fig. 5). We threshold the point if it exceeds a hard-coded range of interest (e.g. if it is behind the vehicles or greater than a certain distance). Afterward, we define the distance-to-collision (d_{poc}) as the longitudinal distance from the vehicle to the collision point, and the time-to-collision (T_{col}) as the time to reach the collision point calculated by dividing d_{poc} with the velocity of the vehicle. Then the yielding reward function is defined as follows:

$$r_{\psi_2}^{e,2} \left(\mathbf{x}_i^t, \mathbf{x}_j^t \right) = - \left(1 + \exp(\psi_{2,0}) \right) g_{\text{spatial}}^{\text{NN}}(\mathbf{x}_i^t, \mathbf{x}_j^t) - \left(1 + \exp(\psi_{2,1}) \right) g_{\text{time}}^{\text{NN}}(\mathbf{x}_i^t, \mathbf{x}_j^t),$$

where

$$g_{\text{spatial}}^{\text{NN}}(\mathbf{x}_i^t, \mathbf{x}_j^t) = \text{ReLU} \left((x_j - x_{\text{poc}}) - h_{\text{d}_{\text{poc}}}(\mathbf{x}_i^t, \mathbf{x}_j^t) \right)^2,$$

$$g_{\text{time}}^{\text{NN}}(\mathbf{x}_i^t, \mathbf{x}_j^t) = \text{ReLU} \left(h_{T_{\text{col}}}(\mathbf{x}_i^t, \mathbf{x}_j^t) - (T_{\text{col},i} - T_{\text{col},j}) \right)^2.$$

The functions $h_{\text{d}_{\text{poc}}}$ and $h_{T_{\text{col}}}$ are neural networks with ReLU output activation. The g_{spatial} term learns a spatial aspect of the yield behavior and compares the agent's distance from the estimated collision point with the NN-learned *safe* reference within which the lane-changing maneuver can be done. The second term g_{time} adds a temporal aspect, by enforcing the vehicle to ensure a minimum *safe* time headway. We adopt g_{time} because time-to-collision is an important measure in traffic safety assessment [43]. The intuition behind this is to

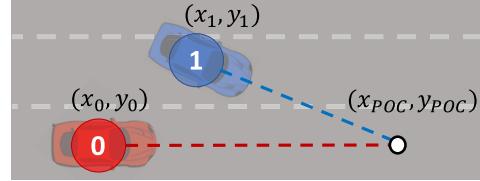


Fig. 5. Collision point diagram. At every timestep, the heading vector of the agents' can be calculated approximating the motion as linear. The intersection between these vectors is taken to be the collision point where the agents would collide if a yield action is not taken.

ensure that the vehicles do not occupy the same position at the same time.

Results: For each scenario, we trained a GRI model with the recurrent policy decoder (21)-(24) in Appx. A. The results are summarized in Table II. In car-following, the NRI model still performs better on trajectory reconstruction, but the GRI model achieves comparable RMSE on NGSIM dataset. In lane-changing, their comparison is consistent: The NRI model slightly outperforms our model in trajectory reconstruction; Our model dominates the NRI model in graph accuracy.

We visualize the interaction graphs in Fig. 6. One interesting observation is that the graphs inferred by NRI have more edges in general. We want to emphasize that both models are trained under the same sparsity constraint. The results imply that we could guide the model to explore a clean and sparse representation of interactions by incorporating relevant domain knowledge, whereas the sparsity regularization itself is not sufficient to serve the purpose. Moreover, the NRI model assigns the same edge type to both edges between a pair of agents. It makes the graphs less interpretable because the vehicles ought to affect each other in different ways. On the other hand, even if different from the hypotheses, our GRI model tends to infer sparse graphs with directional edges.

For the supervised policy, it has the lowest reconstruction error in lane-changing. It implies that the human hypothesis is reasonable because it is capable to model the interactions among human drivers. For the car-following case, its reconstruction error is slightly higher than NRI. Since we cannot assure that our hypothesis is the ground-truth interaction graph underlying the interacting system—In fact, as we mentioned before, we never meant to treat it as the ground-truth—it is possible that the NRI model can find a latent space that can effectively model the interactions in an unsupervised manner. However, as shown in Fig. 6, it is difficult to interpret the graphs inferred by NRI. Considering the sparse and semantic nature of the hypothesis as well as the fact that the supervised policy's reconstruction error is on par with the NRI model, we think the chosen hypothesis is a valid one.

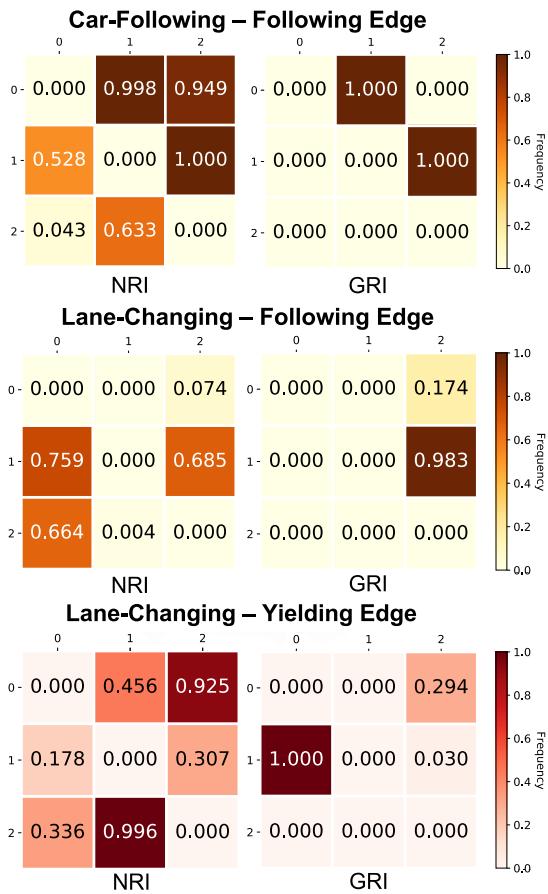


Fig. 6. The empirical distribution of estimated edge variables \hat{z} over the test dataset in the naturalistic traffic scenarios. We summarize the results in multiple adjacency matrices corresponding to different edge types. In the adjacency matrix corresponding to the k^{th} type of interaction, the element $A_{i,j}$ indicates the relative frequency of $\hat{z}_{j,i} = k$, where $\hat{z}_{j,i}$ is the latent variable for the edge from node j to node i .

E. Semantic Meaning of Latent Space

The above experimental results show that our GRI model can recover the ground-truth interaction graphs in the synthetic scenarios with high accuracy, and infer interaction graphs that are consistent with human hypothesis on the NGSIM dataset. However, as we argue in Sec. I, accurate interaction inference alone is not sufficient to show that the model learns a semantically meaningful latent space that is consistent with human domain knowledge. Given an edge, the policy decoder should also synthesize the corresponding semantic interactive behavior indicated by its edge type. It is difficult to verify whether the policy decoder is able to synthesize semantically meaningful interaction simply by monitoring the reconstruction error. Small reconstruction error on in-distribution data could be achieved by imitating demonstration without modeling the correct interaction [30], [44]. To study the semantic meaning of latent space, we design a set of out-of-distribution tests² by adding increasing perturbation to the initial states.

²For clarification, the models used in this section are the same as those introduced in Sec. VI-C. We merely designed additional out-of-distribution cases for testing.

We then enforce the same edge types as in the in-distribution case and run those different policy decoders to generate the trajectories. We are curious about whether the policy decoders can consistently synthesize the correct semantic interactive behavior under distribution shift. If so, we claim the latent space indeed possesses the semantic meaning that is consistent with human domain knowledge.

In the synthetic scenarios, we focus on the following relation. For both car-following and lane-changing scenes, we keep the two vehicles with the following relation, resulting in interaction graphs merely consisting of the following edges (Fig. 7). We introduce perturbation by decreasing the initial longitudinal headway to values unseen during the training stage. The initial longitudinal headway is defined as $\Delta x = x_1^0 - x_0^0$, namely the longitudinal distance from Vehicle 1 to Vehicle 0 at the first time step. During the training stage, we sampled Δx from uniform distributions: In car-following, $\Delta x \sim \text{unif}(4, 8)$; In lane-changing, $\Delta x \sim \text{unif}(8, 12)$. In the out-of-distribution experiments, we gradually decreased Δx from the lower bound to some negative value, which means Vehicle 0 is placed in front of Vehicle 1. We are curious about if the models can generate trajectories meeting the characteristics of the car-following behavior in these unseen scenarios—scenarios with a different number of vehicles and distorted state distribution. To quantitatively evaluate if the synthesized behavior satisfies the requirement of car-following, we consider three metrics for evaluation:

- Success Rate:

$$\text{SuccessRate} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(\Delta x_i^f \geq \delta_f),$$

where $\Delta x_i^f = x_{1,i}^T - x_{0,i}^T$, (13)

- Collision Rate:

$$\text{CollisionRate} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(d_{\min,i} \leq \delta_c),$$

where $d_{\min,i} = \min_t \sqrt{|x_{1,i}^t - x_{0,i}^t|^2 + |y_{1,i}^t - y_{0,i}^t|^2}$, (14)

- Lateral distance:

$$\Delta y = |y_1^T - y_0^T| - |y_1^0 - y_0^0|. \quad (15)$$

We intend to quantify three typical characteristics of the following behavior with the metrics defined above: 1) staying behind the leading vehicle; 2) maintaining a substantial safe distance from the leading vehicle; 3) keeping in the same lane as the leading vehicle. We consider the following vehicle's maneuver successful if the vehicle manages to keep a substantial positive final headway. And we consider two vehicles colliding if the minimum distance between them is smaller than a safety threshold. Lastly, we expect the following behavior to attain a negative Δy , which means the following vehicle attempts to approach the leading vehicle's lane.

All metrics were applied in the lane-changing scenario, but we only adopted SuccessRate in the car-following scenario.

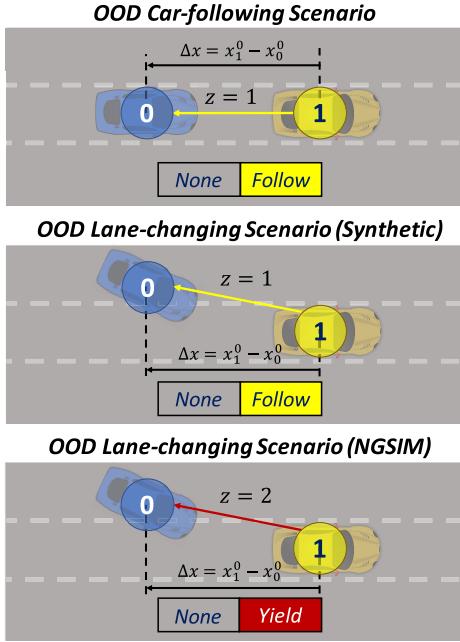


Fig. 7. Out-of-distribution scenarios. We removed one vehicle from the nominal scenes and shifted the initial longitudinal headway Δx to unseen values.

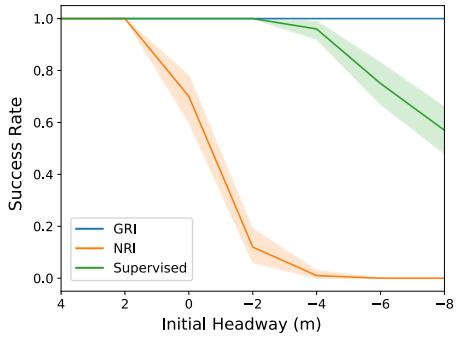


Fig. 8. Results in out-of-distribution synthetic car-following scenario. We plot SuccessRate versus Δx with the error band denoting 95% confidence interval of the indicator, $\mathbf{1}(\Delta x_i^f \geq \delta_f)$. We set $\delta_f = 2\text{m}$.

Since we only model the longitudinal dynamics, Δy is not applicable. For the same reason, if their initial positions are too close or the following vehicle is located ahead of the leading one initially, the following vehicle will inevitably crush into the leading vehicle, which results in $d_{\min} = 0$. Therefore, we only care about the first characteristic.

The results are summarized in Fig. 8 and Fig. 9, where we plot the mean values of the evaluated metrics versus Δx . In the car-following scenario, the NRI policy fails to slow down Vehicle 0 to follow Vehicle 1 when Δx becomes negative. In contrast, the supervised policy and GRI policy maintain high success rates with negative Δx . However, the number of failure cases starts to increase for the supervised policy when Δx becomes substantially negative, whereas the GRI policy maintains a perfect success rate over the tested range of perturbation. We visualize a marginal example in Fig. 12, where both the NRI policy and the supervised one fail to maintain a positive final headway.

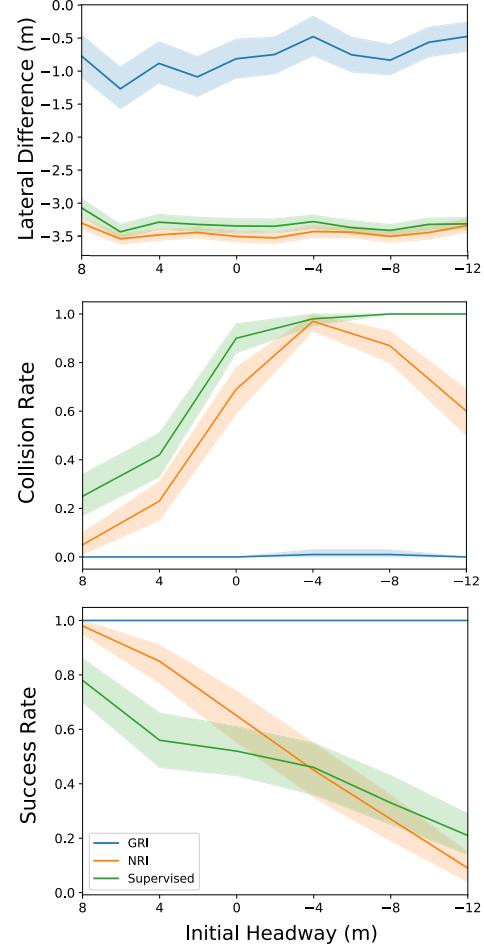


Fig. 9. Results in out-of-distribution synthetic lane-changing scenario. We plot SuccessRate, CollisionRate, and the mean value of Δy versus Δx . The error bands denote 95% confidence interval. For SuccessRate and CollisionRate, the error bands are of the indicator functions. We set $\delta_f = \delta_c = 2\text{m}$.

In the lane-changing scenario, the GRI policy maintains a consistent perfect success rate over all tested values of Δx . For the other two models, the success rates drastically decrease with decreasing Δx . In terms of Δy , all models tend to reduce the lateral distance between the vehicles which is consistent with the second characteristic of the following behavior. However, we found that the GRI policy attains an average Δy with a smaller magnitude and the magnitude decreases with decreasing Δx . It implies that the GRI policy changes its strategy when the initial position of Vehicle 0 is ahead of Vehicle 1. In order to keep a proper safe distance, Vehicle 0 does not change its lane until Vehicle 1 surpasses itself. On the other hand, the lateral behavior is unchanged for the other two models. However, the vehicle cannot maintain a substantial safe distance if it changes its lane too early, which is verified by the plot of collision rate versus Δx . The difference in their strategies is further illustrated by the example visualized in Fig. 12.

We repeat the experiment on the NGSIM datasets. Similar to the case with the synthetic dataset, we remove one vehicle from each scene, resulting in an interaction graph consisting of a single edge (Fig. 7). It is worth noting that removing

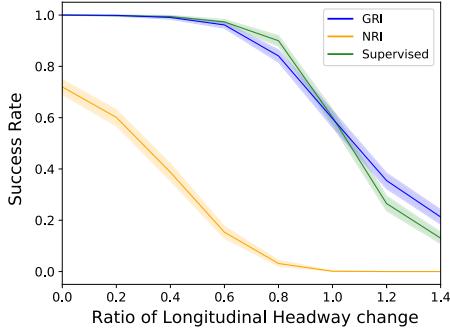


Fig. 10. Results in out-of-distribution naturalistic traffic car-following scenario. We plot SuccessRate versus Δx with the error bands denoting 95% confidence interval of the indicator, $\mathbf{1}(\Delta x_i^f \geq \delta_f)$. We set $\delta_f = 2\text{m}$.

a vehicle from a scene alters the dynamic of the interacting system. It is not fair to expect the models to synthesize the same trajectories in the dataset. Therefore, we do not aim to compare the generated trajectories with the ones in the dataset in this out-of-distribution test. We just check whether the generated trajectories satisfy the desired characteristics of the corresponding interactive behaviors.

In the lane-changing case, the remaining edge has the type of yielding. According to our definition of the yielding relation, we consider the same characteristics and adopt the same metrics defined in Eqn. (13)-(15) for evaluation. Since we do not have control over the data generation procedure, we generate out-of-distribution test samples with different levels of discrepancy by controlling the ratio of longitudinal headway change. Given a sample from the original test dataset, we generate its corresponding out-of-distribution sample by shifting its initial longitudinal headway Δx by a certain ratio, denoted by δ , resulting in a new longitudinal headway $\Delta x'$:

$$\Delta x' = (1 - \delta)\Delta x.$$

We evaluate the models on datasets generated with different values of δ . We are particularly interested in the cases when $\delta \geq 1$, which leads to a negative initial headway. We present the results in Fig. 10 and 11. The comparison is quite consistent with the synthetic scenarios. Compared to the other baselines, our GRI policy can synthesize trajectories that satisfy the desired semantic properties in a larger range of distribution shifts.

The results suggest that even though the NRI model can accurately reconstruct the trajectories, the unsupervised latent space, and the corresponding policies do not capture the semantic meanings behind the interactions. In contrast, the GRI model learns a semantically meaningful latent space that is consistent with human domain knowledge. Another useful insight we draw from the experiment is that interaction labels are not sufficient to induce an explainable model with semantic latent space. Even though the supervised policy utilizes additional information on the ground-truth interaction graph, it fails to synthesize the following behavior in novel scenarios. Although the GRI model still has a considerable gap in reconstruction performance compared to the supervised baseline, it provides a promising and principled manner to incorporate

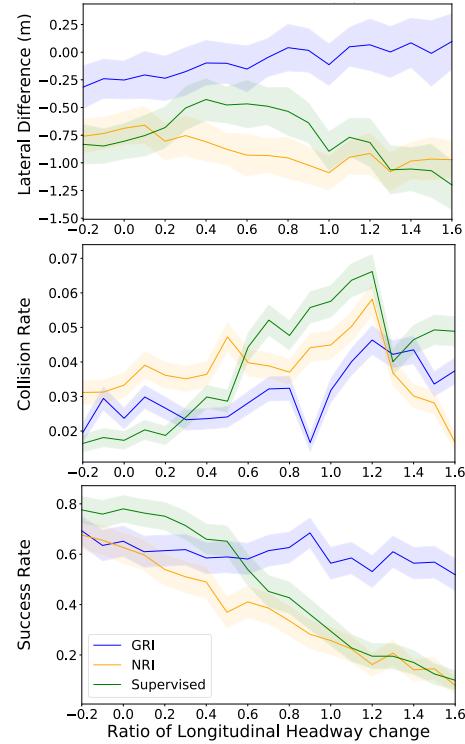


Fig. 11. Results in out-of-distribution naturalistic traffic lane-changing scenario. We plot SuccessRate, CollisionRate, and the mean value of Δy versus Δx . The error bands denote 95% confidence interval. For SuccessRate and CollisionRate, the error bands are of the indicator functions. We set $\delta_f = \delta_c = 2\text{m}$.

domain knowledge into a learning-based autonomous driving system and induce an explainable model.

VII. DISCUSSION AND LIMITATION

A. Application of the Semantic Latent Space

Enabling an explainable model is a crucial step toward trustworthy human interaction. However, it is still unclear how humans may benefit from improved explainability. We would like to have a brief discussion on the potential application of the semantic latent space introduced in GRI. When the autonomous vehicle encounters an unfamiliar situation (e.g., the out-of-distribution scenarios studied in Sec. VI-E), a semantic latent space gives the safety drivers or passengers the privilege to review and override the inferred interaction graph if the model misunderstands the scenario. In contrast, humans can neither understand an interaction graph nor identify the correct edge types, if the learned interactive behaviors do not have explicit semantic meaning. Such kind of safety assurance could help build up safe and trustworthy cooperation between humans and autonomous vehicles.

However, it is impractical to keep the users monitoring the model output in real time. Instead, we can introduce an additional module to detect out-of-distribution scenes [45], [46] and use the estimated epistemic uncertainty to decide when to query the end users. In [45], the authors proposed an adaptive variant of their robust imitative planning algorithm, which incorporates such a unit. It is also a common practice

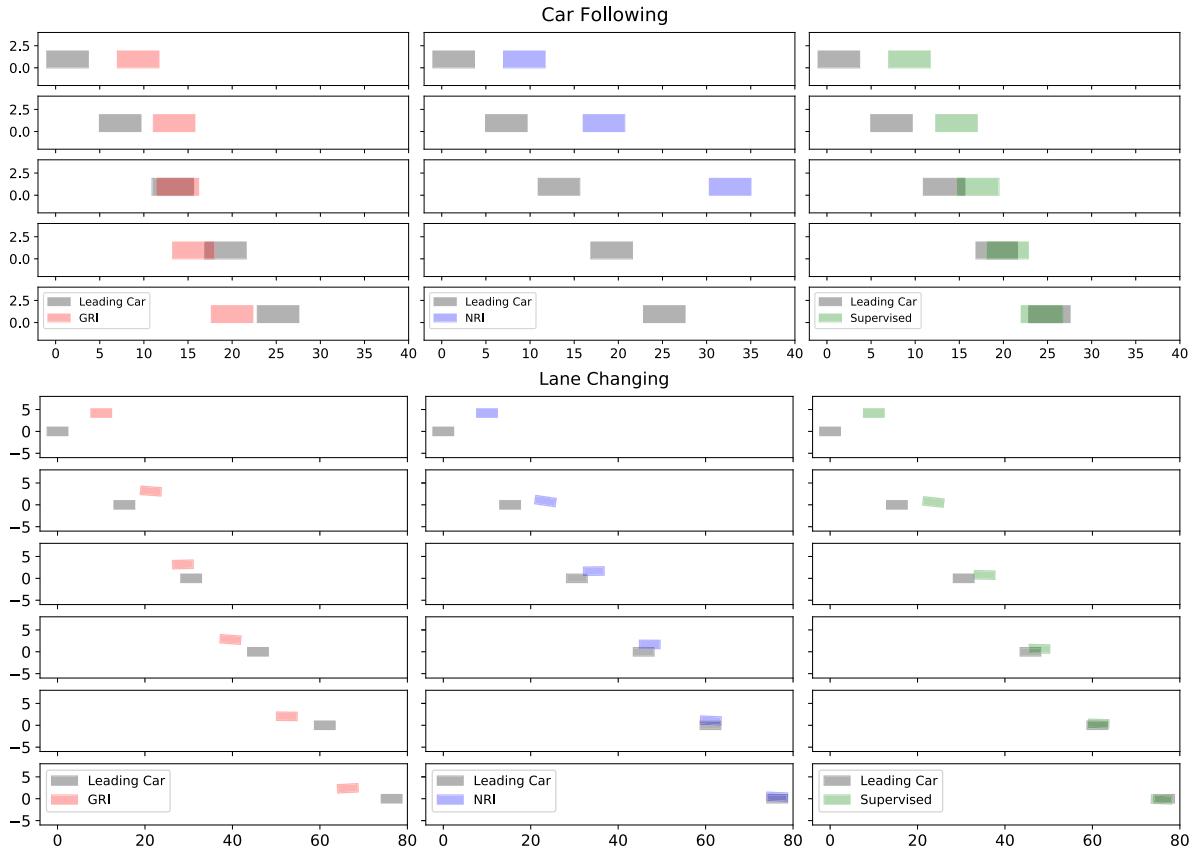


Fig. 12. Examples where the leading car is placed behind the following one at the initial timestep. The trajectories are visualized as sequences of rectangles. Each rectangle represents a vehicle at a specific time step. The vehicles are driving along the positive direction of the x-axis. The GRI policy still prompts the car-following behavior: It slows down the vehicle until the leading one surpasses it. Meanwhile, the NRI policy and the supervised one do not behave as $\mathcal{G}_{\text{interact}}$ suggests.

for current autonomous driving companies to have human assistants for vehicles to query under abnormal situations.

B. Limitation of the Learning Algorithm

In our experiments, GRI always has higher reconstruction errors than NRI, especially on the synthetic dataset. One of the reasons is that reconstruction error is not directly optimized under the AIRL formulation. The objective function of NRI consists of a reconstruction loss, which essentially minimizes the Euclidean distance between the reconstructed trajectory and the ground-truth one. In other words, it directly minimizes the RMSE metrics used in our evaluation. In contrast, GRI adopts the objective function of AIRL, which also minimizes the distance between the trajectory pair. However, the distance is defined by the learned discriminator and is not necessarily equivalent to the Euclidean distance. In Appx. C, we study two AIRL baseline models on the synthetic dataset. The results suggest that none of these AIRL-based approaches achieve the same reconstruction performance as NRI.

Another reason is that the current learning algorithm is not quite stable, because of the adversarial training scheme we introduce when incorporating AIRL into the original NRI model. In typical AIRL settings, we may mitigate this problem by warm-starting the training with a policy network pre-trained through imitation learning or behavior cloning [29], [33]. However, since we aim to learn a semantic latent space,

warm-starting the training with a model with unsupervised latent space is not helpful. Alternatively, we may initialize the policy decoder with the supervised one. One issue is that it will change our current setting where human labels are not required. We will investigate this new setting in our future work, and develop a more stable training scheme to further optimize the performance of GRI.

The structured reward functions also interfere with the stability of the learning procedure. Compared to the variant of GRI studied in Appx. C with semantic reward functions removed, we found GRI is more sensitive to hyperparameters and prone to diverging if not carefully tuned. It is because although the structured reward functions are differentiable, it is not guaranteed that the reward functions can be stably optimized through gradient descent. In our future work, we will explore a more stable and robust learning scheme with those structured reward functions.

C. Scaling to Large-Scale Traffic Scenarios

In our experiments, we tested GRI in three-car interactive scenarios extracted from the NGSIM dataset. Here we discuss how we may extend GRI to model more sophisticated interactive traffic scenarios consisting of a large number of agents. In our current implementation, we assume a complete bi-directed scene graph. The dense graph makes the message-passing operations computationally expensive when

the number of agents N is large. The number of edges grows quadratically with the number of agents. Thus, the message-passing operations involve matrix multiplications over matrices with $O(N^3)$ non-zero elements. To alleviate the computation burden, one practical solution is to construct sparse scene graphs instead. We may leverage heuristics [47] or data-driven methods [48], [49] to coarsely rule out non-interactive agent pairs from the scene graph. Note that we only need a coarse and conservative pre-processing of the scene graphs, and leave it for GRI to infer the remaining non-interaction edges. Thus, we can still avoid the expense and bias introduced by querying human labor to obtain finely-labeled traffic data. It is reasonable to expect that we can obtain a sufficiently sparse scene graph in most cases, since highly interactive scenes are rare in real-world traffic [48]. To further enhance the modeling capacity, we may consider removing some inherent assumptions of GRI on the interactive systems. For instance, the current GRI framework assumes a static interaction graph over the time horizon. We will investigate how to incorporate dynamic graph modeling [50] into GRI in future work. Also, as discussed in Sec. IV, we plan to further generalize GRI to non-cooperative scenarios by removing the cooperative assumption.

VIII. CONCLUSION AND FUTURE WORK

In this work, we propose Grounded Relational Inference (GRI), which models an interactive system's underlying dynamics by inferring the agents' semantic relations. By incorporating structured reward functions, we ground the relational latent space into semantically meaningful behaviors defined with expert domain knowledge. We demonstrate that GRI can model interactive traffic scenarios under both simulation and real-world settings, and generate semantic interaction graphs explaining the vehicle's behavior by their interactions. Although we limit our experiments to the autonomous driving domain, the model itself is formulated without specifying the context. As long as proper domain knowledge is available, the proposed method can be extended naturally to other fields (e.g., human-robot interaction). As discussed in Sec. VII, there are several technical gaps we need to bridge before extending the current framework to more complicated traffic scenarios and interactive systems in other fields. In future work, we will investigate these directions to improve the performance and scalability of GRI while maintaining the advantages of GRI as an explainable model.

APPENDIX

A. Graph Neural Network Model Details

In terms of model structure, both the encoder and the policy decoder are built based on node-to-node message-passing [37], consisting of a node-to-edge message-passing and an edge-to-node message-passing:

$$v \rightarrow e : \mathbf{h}_{i,j}^l = f_e^l(\mathbf{h}_i^l, \mathbf{h}_j^l, \mathbf{x}_{i,j}), \quad (16)$$

$$e \rightarrow v : \mathbf{h}_j^{l+1} = f_v^l(\sum_{i \in \mathcal{N}_j} \mathbf{h}_{i,j}^l, \mathbf{x}_j), \quad (17)$$

where \mathbf{h}_i^l is the embedded hidden state of node v_i in the l^{th} layer and $\mathbf{h}_{i,j}^l$ is the embedded hidden state of the edge $e_{i,j}$. The features \mathbf{x}_i and $\mathbf{x}_{i,j}$ are assigned to the node v_i and the edge $e_{i,j}$ respectively as inputs. \mathcal{N}_j denotes the set of the indices of v_i 's neighbouring nodes connected by an incoming edge. The functions f_e^l and f_v^l are neural networks for edges and nodes respectively, shared across the graph within the l^{th} layer of node-to-node message-passing.

1) *GNN Encoder*: The GNN encoder is essentially the same as in NRI. It models the posterior distribution as $q_\phi(\mathbf{z}|\boldsymbol{\tau})$ with the following operations:

$$\begin{aligned} \mathbf{h}_j^1 &= f_{\text{emb}}(\mathbf{x}_j), \\ v \rightarrow e : \mathbf{h}_{i,j}^1 &= f_e^1(\mathbf{h}_i^1, \mathbf{h}_j^1), \\ e \rightarrow v : \mathbf{h}_j^2 &= f_v^1 \left(\sum_{i \neq j} \mathbf{h}_{i,j}^1 \right), \\ v \rightarrow e : \mathbf{h}_{i,j}^2 &= f_e^2(\mathbf{h}_i^2, \mathbf{h}_j^2), \\ q_\phi(\mathbf{z}_{i,j}|\boldsymbol{\tau}) &= \text{softmax} \left(\mathbf{h}_{i,j}^2 \right), \end{aligned}$$

where f_e^1 , f_v^1 and f_e^2 are fully-connected networks (MLP) and f_{emb} is a 1D convolutional networks (CNN) with attentive pooling.

2) *GNN Policy Decoder*: The policy operates over $\mathcal{G}_{\text{interact}}$ and models the distribution $\pi_\eta(\mathbf{a}^t|\mathbf{x}^t, \mathbf{z})$, which can be factorized with $\pi_\eta(\mathbf{a}_j^t|\mathbf{x}^t, \mathbf{z})$ as in Eqn. (1). We model π_η as a Gaussian distribution with the mean value parameterized by the following GNN:

$$v \rightarrow e : \tilde{\mathbf{h}}_{i,j}^t = \sum_{k=0}^K \mathbf{1}(z_{i,j} = k) \tilde{f}_e^k(\mathbf{x}_i^t, \mathbf{x}_j^t), \quad (18)$$

$$e \rightarrow v : \mu_j^t = \tilde{f}_v \left(\sum_{i \neq j} \tilde{\mathbf{h}}_{i,j}^t \right), \quad (19)$$

$$\pi_\eta(\mathbf{a}_j^t|\mathbf{x}^t, \mathbf{z}) = \mathcal{N}(\mu_j^t, \sigma^2 \mathbf{I}). \quad (20)$$

Alternatively, the model capacity is improved by using a recurrent policy $\pi_\eta(\mathbf{a}_j^t|\mathbf{x}^t, \dots, \mathbf{x}^1, \mathbf{z})$; Namely, the agents take actions according to the historical trajectories of the system. We follow the practice in [12] and add a GRU unit to obtain the following recurrent model:

$$v \rightarrow e : \tilde{\mathbf{h}}_{i,j}^t = \sum_{k=0}^K \mathbf{1}(z_{i,j} = k) \tilde{f}_e^k(\tilde{\mathbf{h}}_i^t, \tilde{\mathbf{h}}_j^t), \quad (21)$$

$$e \rightarrow v : \tilde{\mathbf{h}}_j^{t+1} = \text{GRU} \left(\sum_{i \neq j} \tilde{\mathbf{h}}_{i,j}^t, \mathbf{x}_j^t, \tilde{\mathbf{h}}_j^t \right), \quad (22)$$

$$\mu_j^t = f_{\text{out}}(\tilde{\mathbf{h}}_j^{t+1}), \quad (23)$$

$$\pi_\eta(\mathbf{a}_j^t|\mathbf{x}^t, \dots, \mathbf{x}^1, \mathbf{z}) = \mathcal{N}(\mu_j^t, \sigma^2 \mathbf{I}), \quad (24)$$

where $\tilde{\mathbf{h}}_i^t$ is the recurrent hidden state encoding the historical information up to the time step $t-1$.

B. Reconstruction Visualization on Synthetic Dataset

In our experiments, we found that GRI has a significantly larger reconstruction error on the synthetic dataset than the NRI baseline. To better understand this performance gap in reconstruction, we looked into the reconstructed trajectories

TABLE III
ABLATION STUDY ON SYNTHETIC DATASET

Model	Car Following ($\Delta t = 0.2s$, $T = 20$)			Lane Changing ($\Delta t = 0.2s$, $T = 30$)			
	RMSE _x (m)	RMSE _v (m/s)	Accuracy(%)	RMSE _x (m)	RMSE _y (m)	RMSE _v (m/s)	Accuracy(%)
GRI	0.241 \pm 0.125	0.174 \pm 0.068	100.00 \pm 0.00	0.529 \pm 0.230	0.207 \pm 0.046	0.303 \pm 0.128	99.95 \pm 0.01
GRI-VAIRL	0.120 \pm 0.054	0.116 \pm 0.039	100.00 \pm 0.00	0.377 \pm 0.201	0.160 \pm 0.038	0.190 \pm 0.058	50.0 \pm 0.00
GRI-AIRL	0.138 \pm 0.068	0.150 \pm 0.043	-	0.304 \pm 0.321	0.198 \pm 0.065	0.173 \pm 0.101	-

¹ The data is presented in form of mean \pm std.

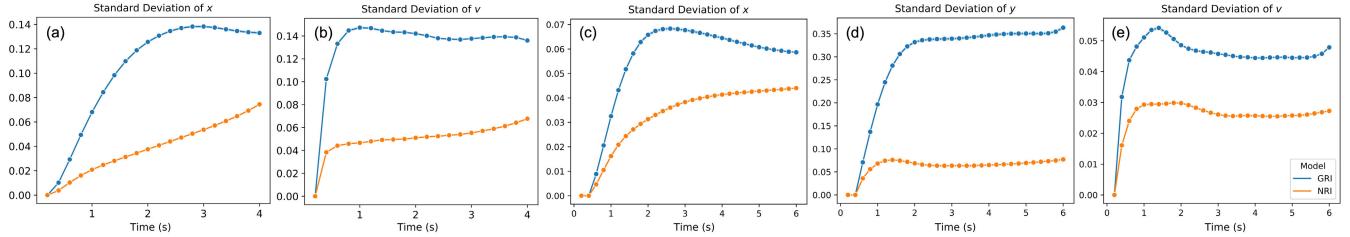


Fig. 13. Average standard deviation of states along the time horizon. (a) and (b) show the standard deviation of x and v in the synthetic car-following scenario. (c)-(e) show the standard deviation of x , y , and v in the synthetic lane-changing scenario.

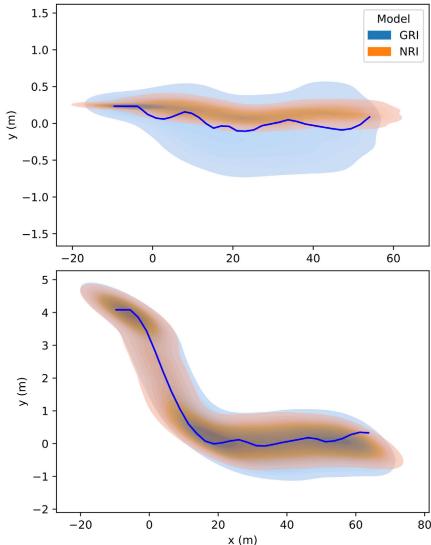


Fig. 14. Visualization of the reconstructed trajectories in a lane-changing scene. (a) and (b) correspond to the trajectories of Car 1 and Car 0 respectively. We visualize the distributions of the reconstructed trajectories estimated using kernel density estimate. The ground-truth trajectories are in blue.

of both models. Instead of executing the mean value of the policy output as we did in our main experiments, we sampled the actions from the policy distribution to estimate the variance of reconstructed trajectories. In Fig. 13, we plot the average standard deviation of reconstructed states along the time horizon. We observed that the policy decoder of GRI tends to have a larger variance. It partially explains the large RMSE values reported in Table I: the metrics were computed with a single reconstructed trajectory. The policy distribution of GRI still has a larger bias than the one of NRI. We visualize the reconstructed trajectories of a lane-changing case in Fig. 14. While the GRI policy induces larger variance, the distribution of the reconstructed trajectories is sensible.

C. AIRL Ablation Study

With the motivation of incorporating semantic meaning into the relational latent space, we developed GRI by introducing

AIRL into relational inference and studied how the semantic reward functions may guide relational latent space learning. Meanwhile, it would be interesting to take a different perspective and study the effects of introducing relational inference and semantic reward functions into AIRL. In this section, we take the synthetic scenarios as examples and conduct an ablation study, where we compare GRI against two variants.

The first one is an AIRL variant, denoted by GRI-AIRL, which is obtained by removing relational inference and semantic reward functions from GRI. Concretely, both the policy and reward decoders operate on a fully-connected interaction graph with homogeneous edge types. And we simply use MLPs to model the reward functions in Eqn. (5) and (6), instead of those semantic reward functions. The objective function then becomes Eqn. (8), but without neither the expectation over \mathbf{z} nor the information bottleneck constraint. The second one is a variational AIRL variant, denoted by GRI-VAIRL, in which we introduce relational inference but do not use the semantic reward functions. In this case, the objective function is identical to the one in GRI, i.e., Eqn. (8).

The results are summarized in Table III. For the car-following scenario, the reconstruction performance is improved after introducing relational inference into AIRL. It is interesting that the GRI-VAIRL variant is able to recover the ground-truth interaction graph, even without the semantic reward functions. It makes sense because the car-following scenario only consists of a single non-trivial edge type. It is plausible for the model to distinguish non-interaction edges from others because a null reward is enforced for non-interaction edges. In some senses, we may still consider the reward function semantic—it incorporates the semantic meaning of non-interaction into the latent space. However, we cannot guarantee that GRI-VAIRL can distinguish between different non-trivial interactive behaviors, which is verified by the lane-changing case. Fig. 15 shows the inferred interaction graph. The model only adopts a single non-trivial edge type to describe all the interactive behaviors. Compared to the ground-truth graph, the inferred graph has an additional edge $z_{2,1}$ but ignores the edge $z_{1,0}$. Ignoring the edge $z_{1,0}$ limits the

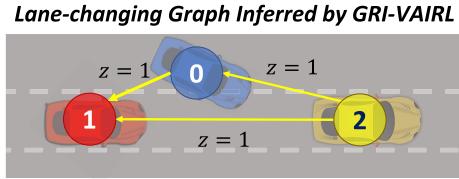


Fig. 15. The interaction graph inferred by the GRI-VAIRL model in the synthetic lane-changing scenario.

modeling capacity of the policy decoder, which could possibly explain why GRI-VAIRL has larger RMSE_x and RMSE_v than GRI-AIRL in the lane-changing case.

In summary, we could improve reconstruction performance by introducing relational inference into AIRL. Even if GRI-VAIRL has larger reconstruction errors in the lane-changing case due to the biased inferred graph, we still observe that GRI-VAIRL converges faster. The learning process becomes more stable and less sensitive to different hyperparameters. We think it is because the model may identify those agents that are not interacting with each other, preventing the reward decoder from fitting a reward function unifying both interactive and non-interactive behaviors. Meanwhile, it is still necessary to incorporate semantic reward functions to differentiate different interactive behaviors and induce a semantically meaningful interaction graph. However, semantic latent space comes at a cost of reconstruction performance. The structured reward functions limit the modeling capacity of the reward decoder. Also, although the structured reward functions are differentiable, it is not guaranteed that they can be well optimized through gradient descent. As a result, they may interfere with the stability of the learning procedure.

REFERENCES

- [1] M. Bojarski et al., “End to end learning for self-driving cars,” 2016, *arXiv:1604.07316*.
- [2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D object detection network for autonomous driving,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.
- [3] C. Tang, Z. Xu, and M. Tomizuka, “Disturbance-observer-based tracking controller for neural network driving policy transfer,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3961–3972, Sep. 2020.
- [4] A. B. Arrieta et al., “Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.
- [5] J. Kim and J. Cann, “Interpretable learning for self-driving cars by visualizing causal attention,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2942–2950.
- [6] M. Bojarski et al., “VisualBackProp: Efficient visualization of CNNs for autonomous driving,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4701–4708.
- [7] 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)*, Jun. 2016, pp. 961–971.
- [8] A. Vemula, K. Muelling, and J. Oh, “Social attention: Modeling attention in human crowds,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–7.
- [9] Y. Hoshen, “VAIN: Attentional multi-agent predictive modeling,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 2701–2711.
- [10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018. [Online]. Available: <https://openreview.net/forum?id=rJXMpikCZ>
- [11] S. Sukhbaatar, A. Szlam, and R. Fergus, “Learning multiagent communication with backpropagation,” in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 2244–2252.
- [12] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, “Neural relational inference for interacting systems,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 2688–2697.
- [13] J. Kim et al., “Textual explanations for self-driving vehicles,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 563–578.
- [14] L. Sun, W. Zhan, and M. Tomizuka, “Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning,” in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2111–2117.
- [15] D. Lee, Y. Gu, J. Hoang, and M. Marchetti-Bowick, “Joint interaction and trajectory prediction for autonomous driving using graph neural networks,” 2019, *arXiv:1912.07882*.
- [16] M. Naumann, L. Sun, W. Zhan, and M. Tomizuka, “Analyzing the suitability of cost functions for explaining and imitating human driving behavior based on inverse reinforcement learning,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Sep. 2020, pp. 5481–5487.
- [17] S. Van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber, “Relational neural expectation maximization: Unsupervised discovery of objects and their interactions,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.
- [18] P. Battaglia et al., “Interaction networks for learning about objects, relations and physics,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 4502–4510.
- [19] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, “Structural-RNN: Deep learning on spatio-temporal graphs,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5308–5317.
- [20] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, “Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 3517–3524.
- [21] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” 2017, *arXiv:1709.04875*.
- [22] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 4565–4573.
- [23] C. Finn, P. Christiano, P. Abbeel, and S. Levine, “A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models,” 2016, *arXiv:1611.03852*.
- [24] L. Yu, J. Song, and S. Ermon, “Multi-agent adversarial inverse reinforcement learning,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.
- [25] R. P. Bhattacharyya, D. J. Phillips, B. Wulfe, J. Morton, A. Kuebler, and M. J. Kochenderfer, “Multi-agent imitation learning for driving simulation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1534–1539.
- [26] R. P. Bhattacharyya, D. J. Phillips, C. Liu, J. K. Gupta, K. Driggs-Campbell, and M. J. Kochenderfer, “Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning,” in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 789–795.
- [27] Z. Wang, J. S. Merel, S. E. Reed, N. de Freitas, G. Wayne, and N. Heess, “Robust imitation of diverse behaviors,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5320–5329.
- [28] X. B. Peng, A. Kanazawa, S. Toyer, P. Abbeel, and S. Levine, “Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.
- [29] L. Yu, T. Yu, C. Finn, and S. Ermon, “Meta-inverse reinforcement learning with probabilistic context variables,” in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2019, pp. 11772–11783.
- [30] P. de Haan, D. Jayaraman, and S. Levine, “Causal confusion in imitation learning,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2019, pp. 11698–11709.
- [31] C. Li, S. H. Chan, and Y.-T. Chen, “Who make drivers stop? Towards driver-centric risk assessment: Risk object identification via causal inference,” 2020, *arXiv:2003.02425*.
- [32] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2008, pp. 1433–1438.
- [33] C. Finn, S. Levine, and P. Abbeel, “Guided cost learning: Deep inverse optimal control via policy optimization,” in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 49–58.
- [34] J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adversarial inverse reinforcement learning,” 2017, *arXiv:1710.11248*.

- [35] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 368, no. 1928, pp. 4585–4605, Oct. 2010.
- [36] R. Bhattacharyya et al., "Modeling human driving behavior through generative adversarial imitation learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 2874–2887, Mar. 2023.
- [37] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learning*, vol. 70, 2017, pp. 1263–1272.
- [38] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
- [39] I. Higgins et al., " β -VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
- [40] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," 2018, *arXiv:1805.00909*.
- [41] K. Leung, N. Aréchiga, and M. Pavone, "Backpropagation for parametric STL," in *Proc. IEEE Intell. Veh. Symp.*, Sep. 2019, pp. 185–192.
- [42] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, no. 2, p. 1805, 2000.
- [43] M. M. Minderhoud and P. H. L. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Anal. Prevention*, vol. 33, no. 1, pp. 89–97, Jan. 2001.
- [44] C. Tang, W. Zhan, and M. Tomizuka, "Exploring social posterior collapse in variational autoencoder for interaction modeling," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 8481–8494.
- [45] A. Filos, P. Tigkas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?" in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3145–3153.
- [46] L. Sun, X. Jia, and A. Dragan, "On complementing end-to-end human behavior predictors with planning," in *Proc. Robot., Sci. Syst.*, Jul. 2021.
- [47] Q. Sun, X. Huang, J. Gu, B. C. Williams, and H. Zhao, "M2i: From factored marginal trajectory prediction to interactive prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6543–6552.
- [48] E. Tolstaya, R. Mahjourian, C. Downey, B. Vadarajan, B. Sapp, and D. Anguelov, "Identifying driver interactions via conditional behavior prediction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 3473–3479.
- [49] W. Luo, C. Park, A. Cornman, B. Sapp, and D. Anguelov, "JFP: Joint future prediction with interactive multi-agent modeling for autonomous driving," in *Proc. Conf. Robot Learn.*, 2023, pp. 1457–1467.
- [50] J. Li, F. Yang, M. Tomizuka, and C. Choi, "Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning," in *Proc. NeurIPS*, 2020, pp. 19783–19794.



Chen Tang (Member, IEEE) received the bachelor's degree in mechanical engineering from The Hong Kong University of Science and Technology (HKUST) and the Ph.D. degree in mechanical engineering from the University of California at Berkeley (UC Berkeley) in 2022. He is currently a Post-Doctoral Fellow with the Department of Computer Science, UT Austin. His research interests lie at the intersection of control, robotics, and learning. The goal of his research is to develop trustworthy and safe autonomous agents interacting with humans. He won second place in the IEEE ITSC 2018 Best Student Paper Award and the AMSE DSCD Rising Star 2022 Award. He was selected as a RSS Pioneer in 2023.



Nishan Srishankar received the B.S. degree in mechanical engineering and the M.Sc. degree in robotics engineering from Worcester Polytechnic Institute, Worcester, MA, USA, in 2015 and 2017, respectively. His current research interests are in multi-robot systems and machine learning namely to develop trustworthy robotic/decision-making systems under realistic learning curriculums.



Sujitha Martin received the B.S. degree in electrical engineering from California Institute of Technology, Pasadena, CA, USA, in 2010, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at San Diego, CA, USA, in 2012 and 2016, respectively. Her current research interests are in human-centered situational awareness and knowledge infused learning for safe, trust inducing and enriching experience between human, and highly autonomous systems.



Masayoshi Tomizuka (Life Fellow, IEEE) received the Ph.D. degree in mechanical engineering from MIT in February 1974. In 1974, he joined as a Faculty Member of the Department of Mechanical Engineering, University of California at Berkeley, where he currently holds the Cheryl and John Neerhout, Jr., Distinguished Professorship Chair. His current research interests include optimal and adaptive control, digital control, signal processing, motion control, control problems related to robotics, precision motion control, and vehicles. He was the Program Director of the Dynamic Systems and Control Program of the Civil and Mechanical Systems Division of NSF from 2002 to 2004. He was a Technical Editor of the *ASME Journal of Dynamic Systems, Measurement and Control (J-DSMC)* from 1988 to 1993 and the Editor-in-Chief of *IEEE/ASME TRANSACTIONS ON MECHATRONICS* from 1997 to 1999. He is a fellow of the ASME and IFAC. He was a recipient of the Charles Russ Richards Memorial Award (ASME) in 1997, the Rufus Oldenburger Medal (ASME) in 2002, and the John R. Ragazzini Award in 2006.