PRECOG: PREdictions Conditioned On Goals in Visual Multi-Agent Scenarios

Nicholas Rhinehart [1] **Rowan McAllister** [2] **Kris Kitani** [1] **Sergey Levine** [2]

Abstract

For autonomous vehicles (AVs) to behave appropriately on roads populated by human-driven vehicles, they must be able to reason about the intentions and stochastic decisions of other drivers given rich perceptual information. Towards these capabilities, we present a deep probabilistic forecasting model of future intentions and interactions of multiple agents. We design it to perform both standard forecasting and conditional forecasting with respect to the AV's goals. Conditional forecasting reasons about how all agents will likely respond to specific decisions of a controlled agent. We train our model on real and simulated data to forecast vehicle trajectories given past positions and LIDAR. Our evaluation shows that our model is substantially more accurate in multi-agent driving scenarios compared to existing state-of-the-art, even when restricting its inputs to past positions alone.

1 Introduction

Autonomous driving requires reasoning about the future behaviors of nearby agents, e.g. at stop signs, roundabouts, crosswalks, or when parking. In multi-agent settings, each agent's behavior affects the behavior of others. Motivated by people's ability to reason in these settings, we present a method to forecast multi-agent interactions from perceptual data, such as images and LIDAR. Beyond forecasting the behavior of all agents, we want our model to *conditionally forecast* how other agents are likely to respond to different decisions each agent could make. When planning a robot to a goal, we want to forecast what other agents may do in response. This reasoning is essential for agents to make good decisions in multi-agent environments: they must reason how their future decisions could affect the multi-agent system. Examples of forecasting (Fig. 1) and conditional forecasting (Fig. 2) on test data are shown. Videos available at `https://sites.google.com/view/precog`.

[1]Carnegie Mellon University [2]University of California, Berkeley. Correspondence to: Nicholas Rhinehart <nrhineha@cs.cmu.edu>.

First workshop on *Invertible Neural Networks and Normalizing Flows* (ICML 2019), Long Beach, CA, USA

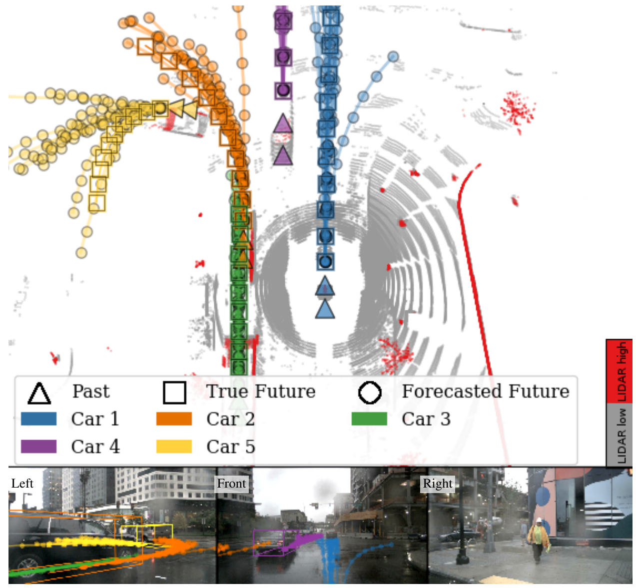


Figure 1. Forecasting on nuScenes (Caesar et al., 2019). The input to our model is a high-dimensional LIDAR observation, which informs a distribution over all agents' future trajectories.

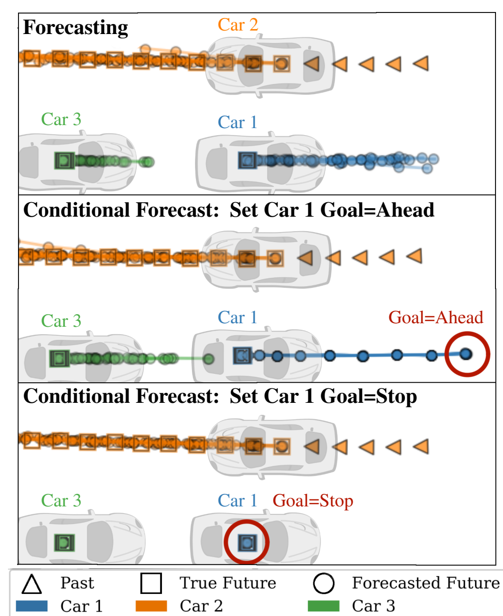


Figure 2. Conditioning the model on different Car 1 intentions produces different predictions: here it forecasts Car 3 to move if Car 1 yields space, or stay stopped if Car 1 stays stopped.

We propose an efficient factorized generative model of joint behavior conditioned on rich observations of the environment. Our contributions follow:

1. **State-of-the-art multi-agent forecasting:** We develop a multi-agent forecasting model that uses *exact* likelihood inference (unlike VAEs or GANs) to outperform three state-of-the-art forecasting methods in real (nuScenes (Caesar et al., 2019)) and simulated (CARLA (Dosovitskiy et al., 2017)) datasets.
2. **Goal-conditioned multi-agent forecasting:** Ours is the first generative multi-agent forecasting method that can condition on agent goals or intentions. Given our model's learned coupling of agent interactions, conditioning on one agent's intentions causes the predictions of other agents to change. Our *factorized*-latent variables enable modelling decoupled agent intentions even though agent dynamics are coupled.
3. **Multi-agent imitative planning objective:** We derive a data-driven objective for safe motion planning in multi-agent environments. It balances the likelihood of reaching a goal with the probability that expert demonstrators would execute the same plan. We use this objective for offline planning to known goals, which further improves forecasting performance.

2 Related Work

Game-theoretic planning: Traditionally, multi-agent planning and game theory approaches explicitly model multiple agents' policies or internal states, usually by generalizing Markov decision process (MDP) to multiple decisions makers (Claus & Boutilier, 1998; Tan, 1993). These frameworks facilitate reasoning about collaboration strategies, but suffer from "state space explosion" intractability except when interactions are sparse (Melo & Veloso, 2011).

Imitative forecasting: Data-driven approaches have been applied to forecast complex interactions between pedestrians (Alahi et al., 2016; Bartoli et al., 2017), vehicles (Deo & Trivedi, 2018; Lee et al., 2017; Park et al., 2018), and athletes (Le et al., 2017; Lee & Kitani, 2016). These "imitative forecasting" methods generalize from previously observed interactions to predict multi-agent behavior in new situations. While these data-driven methods forecast multi-agent scenarios as observers, controlled vehicles knowingly affect the multi-agent system, and must condition their forecasts on their known controls.

Forecasting for control and planning: Generative models for multi-agent forecasting and control have been proposed. (Schmerling et al., 2018) which uses a conditional VAE encoding of the joint states of multiple agents to predict future human actions. Our work differs by 1) using contextual information to generalize to new scenes, and 2) modelling interactions between more than two vehicles *jointly*.

3 Deep Multi-Agent Forecasting

We consider scenarios in which the model may control one of the agents (a "robot"). By modeling co-influence, our robot's trajectory are conditional on the (uncertain) future human trajectories, and therefore future robots states are necessarily uncertain.

3.1 Notation

We consider A agents (vehicles) that interact over T time steps. We model all agent positions at time t as $\mathbf{S}_t \in \mathbb{R}^{A\times D}$, where $D=2$. $\mathbf{S}_t^a$ represents agent a's (x,y) coordinates on the ground plane. We assume there is one "robot agent" (e.g. the autonomous vehicle that our model can control) and $A-1$ "human agents" (e.g. human drivers that our model cannot control). For convenience, we define $\mathbf{S}_t^r \doteq \mathbf{S}_t^1 \in \mathbb{R}^D$ to index the robot state, and $\mathbf{S}_t^h \doteq \mathbf{S}_t^{2:A} \in \mathbb{R}^{(A-1)\times D}$ to index the human states. Random variables are capitalized and $t=0$ defines the current time. Perception is given by $\phi \doteq \{\mathbf{s}_{-\tau:0}, \boldsymbol{\chi}\}$, where τ is the number of past multi-agent positions we condition on and $\boldsymbol{\chi}$ is a high-dimensional observation of the scene. LIDAR is provided as $\boldsymbol{\chi} = \mathbb{R}^{200\times 200\times 2}$, with $\boldsymbol{\chi}_{ij}$ representing a 2-bin histogram of points above and at ground level in $0.5\mathrm{m}^2$ cells.

3.2 Factorized Multi-Agent Forecasting

We propose a data-driven likelihood-based generative model of multi-agent interaction to probabilistically predict T-step dynamics of a multi-agent system: $\mathbf{S} \sim q(\mathbf{S}|\phi;\mathcal{D})$, where $\mathcal{D}$ is training data of observed multi-agent state trajectories. Our model is generative, and learns to map latent variables $\mathbf{Z}$ via an invertible function f to generate multi-agent state trajectories conditioned on ϕ. f's invertibility induces $q(\mathbf{S}|\phi)$, a *pushforward distribution* (McCann et al., 1995), also known as an *invertible generative model* (Dinh et al., 2016). Invertible generative models can compute the probability of joint multi-agent trajectories, critical to our goal of *planning*. $\mathbf{S}$ is sampled from q as follows:

$$\mathbf{S} = f(\mathbf{Z};\phi) \in \mathbb{R}^{T\times A\times D},\ \mathbf{Z} \sim \mathcal{N}(\mathbf{0},\mathbf{I}) \in \mathbb{R}^{T\times A\times D}. \quad (1)$$

Our model is related to the R2P2 single-agent generative model (Rhinehart et al., 2018), which we extend to the uncertain multi-agent setting:

$$\mathbf{S}_t^a \;=\; \mu_\theta^a(\mathbf{S}_{1:t-1},\phi) + \sigma_\theta^a(\mathbf{S}_{1:t-1},\phi)\cdot \mathbf{Z}_t^a \;\in \mathbb{R}^D, \quad (2)$$

where $\mu_\theta^a(\cdot)$ and $\sigma_\theta^a(\cdot)$ are neural network functions (with trainable weights θ) outputting a one-step mean prediction $\boldsymbol{\mu}_t^a \in \mathbb{R}^D$ and standard-deviation matrix $\boldsymbol{\sigma}_t^a \in \mathbb{R}^{D\times D}$ of agent a, defining the system's transition function q:

$$q(\mathbf{S}_t|\mathbf{S}_{1:t-1},\phi) = \prod_{a=1}^{A} \mathcal{N}(\mathbf{S}_t^a; \boldsymbol{\mu}_t^a, \boldsymbol{\sigma}_t^a \boldsymbol{\sigma}_t^{a\top}), \quad (3)$$

$$q(\mathbf{S}|\phi) = \prod_{t=1}^{T} q(\mathbf{S}_t|\mathbf{S}_{1:t-1},\phi). \quad (4)$$

3.3 Model Implementation

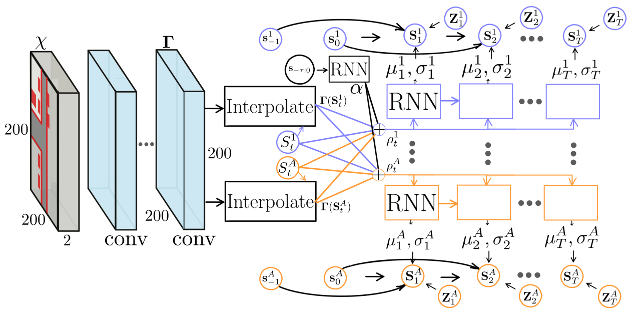


Figure 3. We use latent variable $\mathbf{Z}^a_{t+1}$ to represent variation in agent a's *plausible* scene-conditioned reactions to all agents $\mathbf{S}_t$, causing uncertainty in every agents' future states $\mathbf{S}$ because they interact. Variation exists because of unknown driver goals and different recorded driving styles.

A high-level diagram of our implementation shown in Fig. 3. Recall the context $\phi \doteq \{\mathbf{s}_{-\tau:0}, \boldsymbol{\chi}\}$, containing the past positions of all agents, $\mathbf{s}_{-\tau:0}$, and a feature map $\boldsymbol{\chi}$, implemented as LIDAR is mounted on the first agent. We encode $\mathbf{s}_{-\tau:0}$ with a GRU. A CNN processes $\boldsymbol{\chi}$ to $\boldsymbol{\Gamma}$ at the same spatial resolution as $\boldsymbol{\chi}$. Features for each agent's predicted position $\mathbf{S}^a_t$ are computed by interpolating into $\boldsymbol{\Gamma}$.

3.4 PREdictions Conditioned On Goals (PRECOG)

A distinguishing feature of our generative model for multi-step, multi-agent prediction is its latent variables $\mathbf{Z} \doteq \mathbf{Z}^{1:A}_{1:T}$ that factorizes over agents and time. Factorization makes it possible to use the model for highly flexible conditional forecasts. Since robots are not merely passive observers, but one of potentially many agents, the ability to anticipate how they affect others is critical to their ability to plan safely. Human drivers can appear to take highly stochastic actions in part because we cannot observe their intentions. In our model, the source of this uncertainty comes from the latent variables $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. In practical scenarios, the robot knows its own intentions, can choose its own actions, and can plan a course of action to achieve a desired goal. Recall from (2) that *one-step* agent predictions are conditionally independent from each other give the previous multi-agent states. Therefore, certainty in the latent state $\mathbf{Z}^a_t$ corresponds to certainty of the a^{th} agent's *reaction* to the multi-agent system at time t. Different values of $\mathbf{Z}^a_t$ correspond to different ways of reacting to the same information. Deciding values of $\mathbf{Z}^a_t$ corresponds to controlling the agent a. We can therefore implement control of the robot via assigning values to its latent variables $\mathbf{Z}^r \leftarrow \mathbf{z}^r$. In contrast, human reactions $\mathbf{Z}^h_t$ cannot be decided by the robot, and so remain uncertain from the robot's perspective and can only be influenced by their conditioning on the robot's previous states in $\mathbf{S}_{1:t-1}$. Therefore, to generate conditional-forecasts, we simply decide $\mathbf{z}^r$, sample $\mathbf{Z}^h$, concatentate $\mathbf{Z} = \mathbf{z}^r \oplus \mathbf{Z}^h$, and warp $\mathbf{S} = f(\mathbf{Z}, \phi)$.

3.5 Multi-Agent Planning

We perform multi-agent planning by optimizing an objective $\mathcal{L}$ w.r.t. the control variables $\mathbf{z}^r$, which allows us to produce the "best" forecasts under $\mathcal{L}$. First, we chose a "goal likelihood" function that represents the likelihood that a robot reaches its goal $\mathcal{G}$ given state trajectory $\mathbf{S}$. For instance, the likelihood could be a waypoint $\mathbf{w} \in \mathbb{R}^D$ the robot should approach: $p(\mathcal{G}|\mathbf{S}, \phi) = \mathcal{N}(\mathbf{w}; \mathbf{S}^r_T, \epsilon\mathbf{I})$. Second, we combine the goal likelihood with a "prior probability" model of safe multi-agent state trajectories $q(\mathbf{S}|\phi)$, learned from expert demonstrations. Note that unlike many other generative multi-agent models, we can compute the probability of generating $\mathbf{S}$ from $q(\mathbf{S}|\phi)$ exactly, which is critical to our planning approach. This results in a "posterior" $p(\mathbf{S}|\mathcal{G}, \phi)$we seek to maximize:

$$\log \mathbb{E}_{\mathbf{Z}^h}[p(\mathbf{S}|\mathcal{G}, \phi)] \geq \mathbb{E}_{\mathbf{Z}^h}[\log p(\mathbf{S}|\mathcal{G}, \phi)] \tag{5}$$

$$= \mathbb{E}_{\mathbf{Z}^h}[\log q(\mathbf{S}|\phi) \cdot p(\mathcal{G}|\mathbf{S}, \phi)] - \log p(\mathcal{G}|\phi) \tag{6}$$

$$\mathcal{L}(\mathbf{z}^r, \mathcal{G}) \doteq \mathbb{E}_{\mathbf{Z}^h}[\log q(\mathbf{S}|\phi) \cdot p(\mathcal{G}|\mathbf{S}, \phi)] \tag{7}$$

$$= \mathbb{E}_{\mathbf{Z}^h}[\log \underbrace{q(f(\mathbf{Z})|\phi)}_{\text{multi-agent prior}} + \log \underbrace{p(\mathcal{G}|f(\mathbf{Z}), \phi)}_{\text{goal likelihood}}], \tag{8}$$

where (5) follows by Jensen's inequality. (6) follows from Bayes' rule and uses our learned model q as the prior. In (8), we drop $p(\mathcal{G}|\phi)$ because it is constant w.r.t. $\mathbf{z}^r$. Recall $\mathbf{Z} = \mathbf{z}^r \oplus \mathbf{Z}^h$ is the concatenation of robot and human control variables. The robot can plan by optimizing $\mathbf{z}^{r*} = \operatorname{argmax}_{\mathbf{z}^r} \mathcal{L}(\mathbf{z}^r, \mathcal{G})$.

4 Experiments

4.1 Datasets

We generated a realistic dataset for multi-agent trajectory forecasting and planning with the CARLA simulator (Dosovitskiy et al., 2017). We ran the autopilot in `Town01` for over 900 episodes of 100 seconds each in the presence of 100 other vehicles, and recorded the trajectory of every vehicle and the autopilot's LIDAR observation. We randomized episodes to either train, validation, or test sets. We created sets of 60,701 train, 7586 validation, and 7567 test scenes, each with 2 seconds of past and 4 seconds of future position information at 5Hz. We similarly used the recently-released nuScenes data (Caesar et al., 2019), a real-world dataset for multi-agent trajectory forecasting.

4.2 Metric

For sample metrics, we must take care not to penalize the distribution when it generates plausible samples different than the expert trajectory. We extend the "minMSD" metric (Lee et al., 2017; Park et al., 2018; Rhinehart et al., 2018) to measure quality of *joint trajectory samples*. In contrast to the commonly-used average displacement error (ADE) and final displacement error (FDE) metrics that computes the mean Euclidean error from a batch of samples to a *single* ground-truth sample (Alahi et al., 2016; Pellegrini et al., 2009), minMSD has the desirable property of not penalizing plausible samples that correspond to decisions the agents could have made, but did not.

$$\hat{m}_K \doteq \mathbb{E}_{\mathbf{S}^*} \min_{k\in\{1..K\}} ||\mathbf{S}^* - \mathbf{S}^{(k)}||^2/(TA), \mathbf{S}^{(k)} \overset{\text{iid}}{\sim} q(\mathbf{S}|\phi), \quad (9)$$

We denote per-agent error of the best *joint* trajectory with $\hat{m}_K^a \doteq \mathbb{E}_{\mathbf{S}^*} ||\mathbf{S}^{*a} - \mathbf{S}^{a,(k^\dagger)}||^2/T, k^\dagger \doteq \operatorname{argmin}_{k\in\{1..K\}} ||\mathbf{S}^* - \mathbf{S}^{(k)}||^2$.

4.3 Baselines

SocialGAN (Gupta et al., 2018) proposed a conditional GAN multi-agent forecasting model that observes the past trajectories of all modeled agents, but not χ. We used the authors' public implementation. In contrast to SocialGAN, we model joint trajectories and can compute likelihoods.

DESIRE (Lee et al., 2017) proposed a conditional VAE model that observes past trajectories and visual context. We followed the implementation as described.

R2P2 (Rhinehart et al., 2018) proposed a likelihood-based conditional generative forecasting model for single-agents. We extend R2P2 to the multi-agent setting and use it as our No-Influence model which assumes independent (non-interactive) agents: $q(\mathbf{S}|\phi) = \prod_{a=1}^{A} q^a(\mathbf{S}^a|\phi)$.

4.4 Multi-Agent Forecasting Experiments

Table 1. CARLA and nuScenes multi-agent forecasting evaluation. Mean scores (and their standard errors) of sample quality $\hat{m}_{K=12}$ (9) are shown. Our methods are highlighted gray.

Town01 Test	2 agents	3 agents	4 agents	5 agents
DESIRE	1.656 ± 0.038	1.684 ± 0.031	2.425 ± 0.038	2.599 ± 0.029
SocialGAN	0.842 ± 0.024	1.037 ± 0.030	1.386 ± 0.041	1.464 ± 0.028
No-Influence (R2P2*)	0.430 ± 0.016	0.594 ± 0.015	0.753 ± 0.015	0.843 ± 0.014
Ours: Co-Influ., no LIDAR	0.783 ± 0.022	0.815 ± 0.020	1.096 ± 0.020	1.213 ± 0.019
Ours: Co-Influence	$\mathbf{0.335 \pm 0.013}$	$\mathbf{0.430 \pm 0.013}$	$\mathbf{0.659 \pm 0.013}$	$\mathbf{0.716 \pm 0.012}$
nuScenes Test	2 agents	3 agents	4 agents	5 agents
DESIRE	3.473 ± 0.102	4.421 ± 0.130	5.957 ± 0.162	6.575 ± 0.198
SocialGAN	2.119 ± 0.087	3.033 ± 0.110	3.484 ± 0.129	3.871 ± 0.148
No-Influence (R2P2*)	1.336 ± 0.062	2.055 ± 0.093	2.695 ± 0.100	3.311 ± 0.166
Ours: Co-Influ., no LIDAR	1.496 ± 0.069	2.240 ± 0.084	3.201 ± 0.113	3.442 ± 0.139
Ours: Co-Influence	$\mathbf{1.325 \pm 0.065}$	$\mathbf{1.705 \pm 0.089}$	$\mathbf{2.547 \pm 0.095}$	$\mathbf{3.266 \pm 0.155}$

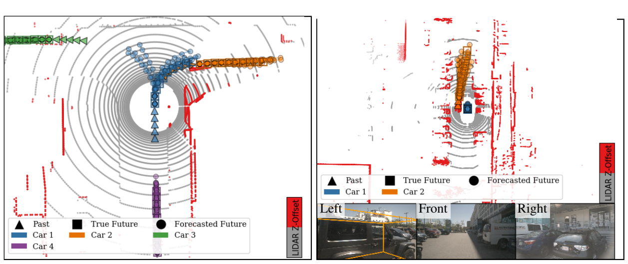


Figure 4. Multi-agent forecasting with our learned Co-Influence model. In each scene, 12 joint samples are shown, and LIDAR colors are discretized to near-ground and above-ground. *Left:* (CARLA) the model predicts Car 1 could either turn left or right. *Right:* (nuScenes) Car 2 is predicted to overtake Car 1.

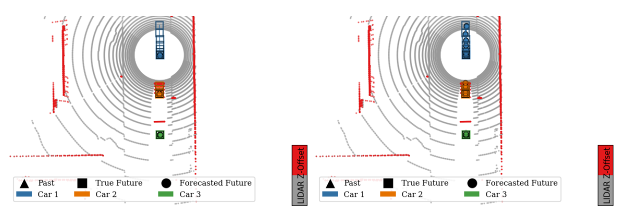
Tab. 1 shows the multi-agent forecasting results. Our models generally achieves outperform prior work. We also ablated our model's access to χ ("Co-Influence, no LIDAR"), for fairer comparison to SocialGAN w.r.t. model inputs. Qualitative examples of our forecasts are shown in Fig. 4: note the model properly predicts diverse plausible paths at intersections, and social behavior where one car will wait for another before accelerating.

4.5 Goal-Conditional Forecasting Experiments

We also investigate whether robot goal-conditioning generates more plausible joint futures of all agents. Unlike the previous unconditional forecasting scenario, when the robot is using the Co-Influence model for planning, it knows its own goal. We can simulate planning by assuming the goal was the state that the robot actually reached at $t = T$, and then planning a path from the current time step to the goal position. We report results of our planning experiments in Tab. 2. Whilst forecasting performance improve for the controlled (robot) agent is expected ($\hat{m}_K^1$), more interestingly: the forecasting performance of the uncontrolled other agents ($\hat{m}_K^2$ and $\hat{m}_K^3$) also improve. Qualitative examples are shown in Fig. 5.

Table 2. Forecasting evaluation of our model on CARLA Town01 Test data. Planning the robot to a goal position enables better predictions for all agents. Means and standard errors reported.

Data	Approach	Test $\hat{m}_{K=12}$	Test $\hat{m}_{K=12}^{a=1}$	Test $\hat{m}_{K=12}^{a=2}$	Test $\hat{m}_{K=12}^{a=3}$
CARLA	Forecast	0.718 ± 0.012	0.340 ± 0.011	0.759 ± 0.024	0.809 ± 0.025
	PRECOG	$\mathbf{0.640 \pm 0.011}$	$\mathbf{0.066 \pm 0.003}$	$\mathbf{0.741 \pm 0.024}$	$\mathbf{0.790 \pm 0.024}$
nuScenes	Forecast	2.921 ± 0.175	1.861 ± 0.109	2.369 ± 0.188	2.812 ± 0.188
	PRECOG	$\mathbf{2.508 \pm 0.152}$	$\mathbf{0.149 \pm 0.021}$	$\mathbf{2.324 \pm 0.187}$	$\mathbf{2.654 \pm 0.190}$



(a) Forecasted (b) Planned

Figure 5. Examples of *planned* multi-agent forecasting with our learned model in CARLA. By using our planning approach and conditioning the robot on its true final position, our predictions for the robot become more accurate, and often our predictions of the other agent become more accurate.

5 Conclusions

We present a multi-agent forecasting method that outperforms state-of-the-art multi-agent forecasting methods on real (nuScenes) and simulated (CARLA) driving data. Our novel ability to condition forecasts on the robot's intentions demonstrated further improvement.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. Social LSTM: Human trajectory prediction in crowded spaces. In *Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Bartoli, F., Lisanti, G., Ballan, L., and Del Bimbo, A. Context-aware trajectory prediction. *arXiv preprint arXiv:1705.02503*, 2017.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

Claus, C. and Boutilier, C. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998:746–752, 1998.

Deo, N. and Trivedi, M. M. Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs. *arXiv preprint arXiv:1805.05499*, 2018.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. CARLA: An open urban driving simulator. In *Conference on Robot Learning (CoRL)*, pp. 1–16, 2017.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Le, H. M., Yue, Y., Carr, P., and Lucey, P. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pp. 1995–2003, 2017.

Lee, N. and Kitani, K. M. Predicting wide receiver trajectories in american football. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9. IEEE, 2016.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., and Chandraker, M. DESIRE: Distant future prediction in dynamic scenes with interacting agents. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 336–345, 2017.

McCann, R. J. et al. Existence and uniqueness of monotone measure-preserving maps. *Duke Mathematical Journal*, 1995.

Melo, F. S. and Veloso, M. Decentralized MDPs with sparse interactions. *Artificial Intelligence*, 175(11): 1757–1789, 2011.

Park, S., Kim, B., Kang, C. M., Chung, C. C., and Choi, J. W. Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. *arXiv preprint arXiv:1802.06338*, 2018.

Pellegrini, S., Ess, A., Schindler, K., and Van Gool, L. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 261–268. IEEE, 2009.

Rhinehart, N., Kitani, K. M., and Vernaza, P. R2P2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *European Conference on Computer Vision (ECCV)*, September 2018.

Schmerling, E., Leung, K., Vollprecht, W., and Pavone, M. Multimodal probabilistic model-based planning for human-robot interaction. In *International Conference on Robotics and Automation (ICRA)*, pp. 1–9. IEEE, 2018.

Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.