

Low Latency Trajectory Predictions For Interaction Aware Highway Driving

Cyrus Anderson¹, Ram Vasudevan², and Matthew Johnson-Roberson³

Abstract—Highway driving places significant demands on human drivers and autonomous vehicles (AVs) alike due to high speeds and the complex interactions in dense traffic. Merging onto the highway poses additional challenges by limiting the amount of time available for decision-making. Predicting others’ trajectories accurately and quickly is crucial to safely execute maneuvers. Many existing prediction methods based on neural networks have focused on modeling interactions to achieve better accuracy while assuming the existence of observation windows over 3 s long. This paper proposes a novel probabilistic model for trajectory prediction that performs competitively with as little as 400 ms of observations. The proposed model extends a deterministic car-following model to the probabilistic setting by treating model parameters as unknown random variables and introducing regularization terms. A realtime inference procedure is derived to estimate the parameters from observations in this new model. Experiments on dense traffic in the NGSIM dataset demonstrate that the proposed method achieves state-of-the-art performance with both highly constrained and more traditional observation windows.

I. INTRODUCTION

Merging in dense traffic necessitates cooperating with other drivers. Successful cooperation in turn relies on predicting others’ actions. Predicting a vehicle’s trajectory, however, is complicated by its possible interactions with surrounding vehicles [1], [2]. Recent works based on deep neural networks (DNNs) have proven effective at modeling these interactions [3]–[12]. These methods utilize a fixed number of observations of surrounding vehicles to infer which trajectories are likely. A drawback to this is the duration of time needed to collect observations before making predictions, ranging from 3 s up to 5 s. This window of time determines the minimum delay between first seeing a vehicle and predicting its trajectory. While observation windows of 3 s may be tenable in low speed environments, the high speeds in highway driving call for faster reaction times. Occlusions and sensor limitations such as maximum range also impact the quality of any observations of the surrounding vehicles. This motivates the need for predictions that can be made in short time and with few observations. In this work we aim to strike a balance between the richness of interactions modeled and the number of observations

This work was supported by a grant from Ford Motor Company via the Ford-UM Alliance under award N022884.

¹C. Anderson is with the Robotics Institute, University of Michigan, Ann Arbor, MI 48109 USA andersct@umich.edu

²M. Johnson-Roberson is with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 USA mattjr@umich.edu

³R. Vasudevan is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA ramv@umich.edu

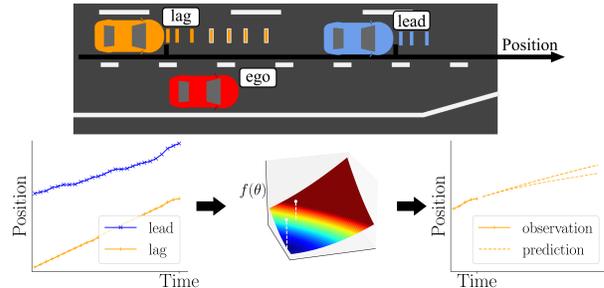


Fig. 1: Overview of prediction method. The ego vehicle (red) seeks to predict the trajectories of the front and rear (lead and lag) vehicles to enable a safe merge between them. Observations of both vehicles (blue and orange marks) are used to define a likelihood function over possible controllers θ for the lag vehicle. Solving a convex problem yields an estimate that is used to sample realistic trajectories.

needed to make predictions. Merging onto the highway in particular poses a challenge to autonomous vehicles (AVs). In addition to the delay induced by observation windows, limited ramp length will further restrict the time available to make decisions. Due to the heightened need of fast reaction times when merging onto the highway, we focus on these scenarios. The main contributions of this work are: 1) a low-latency probabilistic highway interaction model capable of predicting with as little as 400 ms of observations and 2) a realtime and consistent inference procedure that enables the estimation of model parameters from observations rather than by manual specification. Evaluation on merge scenarios in the real-world NGSIM dense highway traffic dataset [13] show that the proposed method achieves state-of-the-art performance. We first extend the deterministic car-following model proposed by Wei et al. [14] to the probabilistic setting. Instead of choosing model parameters by hand, we treat them as unknown random variables and estimate them from observed velocities of the lead and lag vehicles depicted in Figure 1. Though the resulting estimation problem is nonconvex, we prove that it is equivalent to a semidefinite program and solve it in realtime with an off-the-shelf solver. The estimate for the model parameters is then used to sample realistic trajectories.

The paper is organized as follows. Section II describes related works in interaction-based trajectory prediction for traffic participants in general scenarios and those focused on ramp merging. Section III describes the interaction model we use to make predictions and the inference procedure

used to determine the probabilities of different outcomes. We evaluate our model on the NGSIM dataset and perform an ablation study in Section IV before concluding in Section V.

II. RELATED WORK

We first describe prediction methods that may operate on highly restricted observation windows, but do not take into account interactions. Methods that focus on modeling interactions, but operate on longer observation windows, are described in the next section. In the last section we describe methods designed specifically to account for the interactions and restricted observation windows in ramp merging scenarios.

A. Single Agent Prediction

Classical methods specify a simple kinematic model to predict trajectories, such as constant velocity or constant yaw rate and acceleration [15]. Other methods have employed learning based approaches such as Gaussian mixture models [16], or hidden Markov models to ensemble simple kinematic models [17]. More similar to the method proposed in this paper, Houdenou et al. [18] combine predictions from a simple kinematic model and weight these with penalty terms on future accelerations. These methods, however, do not account for interactions between different vehicles on the road. This can lead to inconsistent predictions in common scenarios such as a fast vehicle needing to slow down for a vehicle in front.

B. General Interaction-Based Trajectory Prediction

More recently, DNNs have been used to model the interactions between multiple vehicles. Early works account for interaction but do not make probabilistic predictions [5], [12]. Predicting occupancy maps offers an alternative approach [3], [4]. Occupancy maps specify the probability of space being occupied, but do not maintain the tracking label for each individual vehicle. Approximation error resulting from discretizing continuous space also limits these methods. Other works have avoided these difficulties by directly predicting the parameters of a normal distribution [6]–[8], [10]. Some works, however, require additional labels for maneuver types [6], [7]. Another method, Traffic [8], requires no such labels, but performs a potentially slow social pooling operation for each agent at each timestep. TrafficPredict [10] uses an attention based mechanism to extract features for all pairwise interactions which scales poorly with the number of potentially interacting agents. Recent works have avoided social pooling and attention mechanisms to reduce computational complexity [9], [11]. Social GAN [9] introduces a permutation invariant pooling layer to account for distant interactions while using a Generative Adversarial Network architecture to predict all timesteps in a single forward pass. Multi-Agent Tensor Fusion [11] instead uses a global pooling layer to avoid pooling for each agent separately, as well as preserve spatial structure. The method proposed in this paper does not account for lane changes, but for ramp merging scenarios most vehicles on the highway will cooperatively

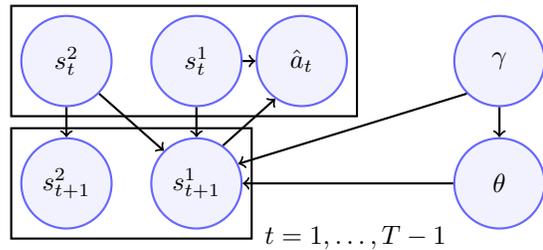


Fig. 2: Proposed interaction model for predicting lag vehicle behavior. The lag vehicle state s^1 depends on the lead vehicle’s state s^2 , its own controller θ , and hyperparameters γ . Noisy estimates of lag vehicle acceleration \hat{a}_t are calculated from state measurements.

merge to an inner lane if at all merging, to avoid interacting with vehicles entering the highway [19], [20]. Other works not based on neural networks have relied on manually defined cost functions to specify vehicles’ behavior [21], or solving integer linear programs [22]. These methods, however, have not performed as well as those based on neural networks.

C. Ramp Merging Trajectory Prediction

In this paper we focus on modeling the interactions between the lag vehicle and lead vehicle shown in Figure 1. A complementary body of research has instead focused on modeling the interactions between the lag vehicle and ego vehicle [14], [23], [24]. In these works the goal is to infer whether or not the lag vehicle will yield to make space for the ego vehicle, or not yield. The lag vehicle is then modeled as following a controller specific to the yield intent. The controllers have been modeled as Markov chains [23] and with the Intelligent Driver Model (IDM) [25] with manually chosen parameters [24].

III. ON-DEMAND TRAJECTORY PREDICTIONS

Here we define our model of interactions between the lag and lead vehicles. We then show how this is used to predict trajectories. Section III-A states the trajectory prediction problem in our probabilistic setting. Section III-B defines the interaction-based controller whose parameters we aim to estimate, and the dynamics of the system. The full probabilistic model with novel regularization terms is defined in Section III-C. The realtime inference procedure for predicting trajectories with this model is described in Section III-D.

A. Problem Statement

We consider the ramp merging scenario in Figure 1 depicting the ego vehicle merging onto the highway. To enable safe merging we are interested in predicting the longitudinal positions of the two (lead, lag) vehicles in the target lane. Let $s_t^i = (x_t^i \ v_t^i)^\top \in \mathbb{R}^2$ denote vehicle state, consisting of longitudinal position x_t^i and velocity v_t^i at timestep t . We denote the state of the lag vehicle by s_t^1 and state of the lead vehicle by s_t^2 . We observe the state of both vehicles

$s_t = (s_t^1 \top s_t^2 \top)^\top$ at timesteps $t = 1, \dots, k$ and predict x_t^1 until the final timestep, $t = k + 1, \dots, T$. Additionally we will use the subscript notation $i : j$ to refer to the set of variables indexed by $i, i + 1, \dots, j$. In making probabilistic predictions this amounts to sampling

$$x_{k+1:T}^1 \sim p(x_{k+1:T}^1 | s_{1:k}). \quad (1)$$

We now explain our focus on the lag vehicle. With several assumptions summarized in the graphical model shown in Figure 2, we decompose the joint prediction of both vehicles into two parts. The first part predicts the lead vehicle's trajectory and the second predicts the lag vehicle's trajectory conditioned on that of the lead vehicle. We start with the assumption of not having observations for the vehicle in front of the lead vehicle. We model the lag vehicle behavior as dependent on the state of the lead vehicle, yet we do not model the state of the lead vehicle as dependent on its own lead. One reason for this inconsistency is that occlusions and limited sensor range may prevent us from obtaining such observations. Aside from this, the proposed model could be extended to account for the lead vehicle's own lead by repeated decomposition, but there is a point at which we cannot see further vehicles ahead. We thus present the simplest model here. Next, assume that the lead vehicle's actions do not depend on the lag vehicle's state as in common car-following models. Furthermore, assume measurements of each vehicle's position and velocity are noiseless, while the acceleration measurement \hat{a}_t of the lag vehicle has zero mean Gaussian noise with a small variance σ_a^2 . This is reasonable when the former measurements have low variances but acceleration is approximated from velocity via finite differences. For example, given velocity measurements with variance σ_v^2 and timestep size Δt , acceleration then has variance $\text{var}(\hat{a}_t) = \text{var}(v_{t+1} - v_t) / \Delta t^2 = 2\sigma_v^2 / \Delta t^2$. The small timestep will magnify the variance as in the NGSIM dataset. Using the independence assumptions in graphical model shown in Figure 2 we may write

$$\begin{aligned} p(s_{k+1:T} | s_{1:k}) &= p(s_{k+1:T}^1 | s_{1:k}, s_{k+1:T}^2) p(s_{k+1:T}^2 | s_{1:k}) \\ &= p(s_{k+1:T}^1 | s_{1:k}, s_{k+1:T}^2) p(s_{k+1:T}^2 | s_{1:k}^2) \end{aligned} \quad (2)$$

which provides the problem decomposition. Throughout the remainder of this paper we focus on the prediction problem for the lag vehicle posed as

$$s_{k+1:T}^1 \sim p(s_{k+1:T}^1 | s_{1:k}, s_{k+1:T}^2) \quad (3)$$

from which we obtain the predicted positions.

B. Interaction Model

Here we describe the controller used to model the interactions between the lead and lag vehicle. The controller is based on balancing two goals. The first is to match the speed of the lead vehicle, and the second is to maintain a desired gap to the lead vehicle. Let g_t denote the current gap between the lead and lag vehicles. This gap is calculated from their positions as $g_t = x_t^2 - x_t^1 - l$, where l is the length of the lead vehicle. We denote the desired gap by g_* . Denoting

k_v and k_g as the proportional control gains for the desired speed and desired gap, respectively, we denote the parameters that define this controller by $\theta = (k_v \ k_g \ g_*)$. The controller proposed in [14] sets the lag vehicle's acceleration with

$$h(s_t, \theta) = k_v(v_t^2 - v_t^1) + k_g(g_t - g_*) \quad (4)$$

according to a manually chosen θ . In this work we treat θ as unknown and our main focus is to estimate it. We assume that the parameters are nonnegative, hence $\theta \in \mathbb{R}_+^3$. We will use the notation $0_{m \times n}$ to denote the matrix of zeros with m rows and n columns. Given current state of the lag vehicle and controller parameters θ the next state is given by

$$s_{t+1}^1 = C s_t^1 + \begin{pmatrix} \Delta t^2/2 \\ \Delta t \end{pmatrix} h(s_t, \theta), \quad (5)$$

where

$$C = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}. \quad (6)$$

Given the lead vehicle's states we write the entire system dynamics as

$$\begin{pmatrix} s_t \\ 1 \end{pmatrix} = \begin{pmatrix} A(\theta) \\ 0_{2 \times 5} \\ 0_{1 \times 4} \ 1 \end{pmatrix} \begin{pmatrix} s_{t-1} \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ s_t^2 \\ 0 \end{pmatrix}, \quad (7)$$

where

$$A(\theta) = (C \ 0_{2 \times 3}) + \begin{pmatrix} \Delta t^2/2 \\ \Delta t \end{pmatrix} \begin{pmatrix} -k_g \\ -k_v \\ k_g \\ k_v \\ -k_g(l + g_*) \end{pmatrix}^\top. \quad (8)$$

C. Regularized Prediction

The difficulty in sampling trajectories in (3) stems from not knowing the lag vehicle's controller θ . Direct estimation of θ can assign significant probability to controllers that produce unrealistic behaviors. In this section we define regularization terms to promote more realistic behaviors. We can express the distribution of trajectories in (3) given the known hyperparameters γ , which we define later, as

$$\begin{aligned} p(s_{k+1:T}^1 | s_{1:k}, s_{k+1:T}^2, \gamma) &= \int_{\mathbb{R}_+^3} p(s_{k+1:T}^1 | s_{1:k}, \theta | s_{1:k}, s_{k+1:T}^2, \gamma) d\theta \\ &= \int_{\mathbb{R}_+^3} p(s_{k+1:T}^1 | s_{1:k}, s_{k+1:T}^2, \theta, \gamma) p(\theta | s_{1:k}, s_{k+1:T}^2, \gamma) d\theta^{(9)} \\ &= \int_{\mathbb{R}_+^3} p(s_{k+1:T}^1 | s_{1:k}, s_{k+1:T}^2, \theta, \gamma) p(\theta | s_{1:k}, \gamma) d\theta, \end{aligned}$$

where the last equality follows from the conditional independence expressed in Figure 2. We begin by defining the second term in (9) which can be written using Bayes' rule and calculating the accelerations derived from velocities as

$$p(\theta | s_{1:k}, \gamma) \propto p(\hat{a}_{1:k-1}, s_{2:k} | s_1, \theta, \gamma) p(\theta | s_1, \gamma). \quad (10)$$

For the first term in (10) we impose a recursive structure independent of γ to mirror standard Markov chain structure as

$$p(\hat{a}_{1:k-1}, s_{2:k} | s_1, \theta, \gamma) = \prod_{i=1}^{k-1} p(\hat{a}_i, s_{i+1} | s_i, \theta), \quad (11)$$

which combined with the system dynamics in (7) and the assumed Gaussian noise for acceleration yields

$$\begin{aligned} p(\hat{a}_{1:k-1}, s_{2:k} | s_1, \theta, \gamma) &= \prod_{i=1}^{k-1} p(\hat{a}_i | h(s_i, \theta)) = \\ &= \prod_{i=1}^{k-1} \mathcal{N}(\hat{a}_i; h(s_i, \theta), \sigma_a^2). \end{aligned} \quad (12)$$

Using this term only, we could estimate θ that fits the observed data. We now introduce the hyperparameters to address weaknesses of this initial model. One problem is an occasionally large and unrealistic estimate of the desired gap g_* . For this we regularize g_* to be close to a given mean gap g_0 . Additionally, the car-following model was originally designed for dense traffic where the lead vehicle is near the lag vehicle. The model is thus vulnerable to overfitting the lag vehicle's behavior to a distant lead vehicle's accelerations. To address this, we regularize the proportional parameters to zero more as the distance between the lead and lag vehicles increases. We introduce the scalars α, β to control the precision of the normal priors placed on the desired gap and proportional parameters. Letting $\gamma = (\alpha, g_0, \beta)$, we define the second term in (10) as

$$-\log p(\theta | s_1, \gamma) = \alpha(g_* - g_0)^2 + \beta g_0^2(k_v^2 + k_g^2). \quad (13)$$

We now define the first term in (9) to regularize the future behavior of the controller, in contrast to (12) which focuses on the fit to observations. Let $\chi_{\{v \leq 0\}}(v)$ denote the characteristic function which equals zero for the real vector v having all positive components and equals infinity elsewhere. The negative log-likelihood is defined to be

$$-\log p(s_{k+1:T}^1 | s_{1:k}, s_{k+1:T}^2, \theta, \gamma) = \chi_{\{v \leq 0\}}(v), \quad (14)$$

where $v = (v_{k+1}^1, \dots, v_T^1)^\top$. Collecting the likelihoods specified in (12)-(14), the negative log-likelihood for a given θ in (9) is

$$f(\theta) = \frac{1}{2\sigma_a^2} \sum_{i=1}^{k-1} (\hat{a}_i - h(s_i, \theta))^2 + \alpha(g_* - g_0)^2 + \beta g_0^2(k_v^2 + k_g^2) + \chi_{\{v \leq 0\}}(v). \quad (15)$$

The σ_a^2 term contributes only to the weighting of the data fit term relative to the regularization terms. We thus may set $\sigma_a^2 = 1$ for convenience and determine the other hyperparameters relative to this. Given parameters θ' we can obtain a predicted trajectory via the dynamics given in equation (7). This trajectory has exact probability equal to

$$\frac{\exp(-f(\theta'))}{\int_{\mathbb{R}_+^3} \exp(-f(\theta)) d\theta} \quad (16)$$

under the model. Calculating this probability by numerical integration, however, is problematic due to the large number of function evaluations. Sampling can be used to obtain a consistent approximation, but we cannot sample θ directly from $\exp(-f(\theta))$ because it does not correspond to a distribution for which efficient samplers exist. In the next section we construct a consistent and efficient sampler for the likelihood specified by (15).

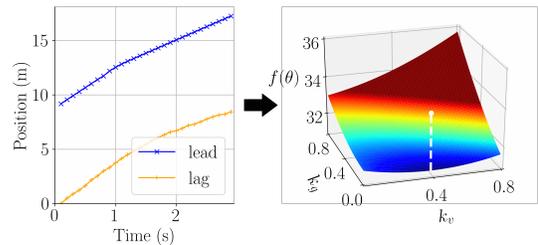


Fig. 3: Likelihood surface of merging scenario. The projection $f(\theta)$ onto k_v and k_g appears smooth and unimodal along these parameters. The proposed inference procedure finds the global minimum (white) at $\hat{\theta} = (0.39, 0, 8.63)$.

D. Efficient Sampling

Our approach to sampling from (15) has two main steps. We first solve an optimization problem to find a set of parameters $\hat{\theta}$ that has high likelihood. We then employ importance sampling to sample from (15).

1) *Importance Sampling Given $\hat{\theta}$* : We sample parameters via $\theta \sim q(\theta; \hat{\theta})$ in the higher likelihood region around $\hat{\theta}$ where q is a distribution chosen to have support over \mathbb{R}_+^3 and admit efficient samplers. These samples are then weighted with their importance weights $w = \exp(-f(\theta))/q(\theta; \hat{\theta})$ and normalized by the sum of the weights to complete the importance sampling. Since q has support over \mathbb{R}_+^3 , the importance sampling is consistent, and depending on the choice of q this procedure may also be efficient for sampling all the high-likelihood θ . Figure 3 shows a typical cost surface of $f(\theta)$ using observations from NGSIM. The smooth surface and unimodality admit efficient sampling.

2) *Optimizing to Find $\hat{\theta}$* : There are multiple possible optimization problems that could be solved to find a high-likelihood $\hat{\theta}$ from $f(\theta)$. Minimizing $f(\theta)$ over $\theta \in \mathbb{R}_+^3$ directly is one choice but we see that $\forall t > k$ s_{t+1} depends on θ through both $A(\theta)$ and s_t in (7) since we only observe up to timestep k . This implies that including the future behavior regularizer (14) produces a nonconvex problem. To ensure our predictions can be made in realtime, we instead optimize over all other terms in $f(\theta)$. Optimizing over these yields

$$\hat{\theta} = \operatorname{argmin}_{\theta \in \mathbb{R}_+^3} \frac{1}{2} \sum_{i=1}^{k-1} (k_v(v_i^2 - v_i^1) + k_g(g_i - g_*) - \hat{a}_i)^2 + \alpha(g_* - g_0)^2 + \beta g_0^2(k_v^2 + k_g^2) \quad (17)$$

$$= \operatorname{argmin}_{\theta \in \mathbb{R}_+^3} \frac{1}{2} \|D(\theta^\top, k_g g_*)^\top - b\|_2^2 \quad (18)$$

$$= \operatorname{argmin}_{\theta \in \mathbb{R}_+^3, u \in \mathbb{R}_+} \frac{1}{2} \|D(\theta^\top, u)^\top - b\|_2^2 \quad \text{s.t. } k_g g_* = u \quad (\text{NC})$$

where we collect the variables and write the optimization as a least squares problem with the rewritten known terms being $D \in \mathbb{R}^{k+2,4}$ and $b \in \mathbb{R}^{k+2}$. For the last equality we use a nonconvex quadratic constraint with a dummy variable to make the vector of decision variables linear. Let $x = (\theta; u) \in \mathbb{R}_+^4$. We also rewrite the nonconvex constraint $k_g g_* = u$ in terms of x . Let $E \in \mathbb{S}^4$ and $c \in \mathbb{R}^4$ be such that

$$r(x) = x^\top E x + c^\top x = k_g g_* - u, \quad (19)$$

Algorithm 1: Probabilistic Trajectory Prediction for Ramp Merging

Input: $s_{1:k}, s_{k+1:T}^2, \gamma, n$

Output: $s_{k+1:T}^{1,(i)}, p^{(i)}$, for $i = 1, \dots, n$

```

1 Solve convex problem (P1) for  $\hat{\theta}$ 
2 foreach  $i = 1, \dots, n$  do
3   Sample  $\theta_i \sim q(\theta; \hat{\theta})$ 
4   Generate  $s_{k+1:T}^{1,(i)}$  via (7)
5    $w_i \leftarrow$ 
       $\exp(-f(\theta_i, s_{1:k}, s_{k+1:T}^{1,(i)}, s_{k+1:T}^2, \gamma)) / q(\theta_i; \hat{\theta})$ 
      via (15)
6 end
7  $\forall i = 1, \dots, n$   $p^{(i)} \leftarrow w_i / \sum_{i=1}^n w_i$ 

```

whereby the constraint may be written as $r(x) = 0$. To find an approximate solution to (NC), we remove the constraint $x \succeq 0$, denoting the new problem (NC1). By Lemma 1 in the Appendix, solving the convex relaxation of (NC1) for x given by

$$\begin{aligned}
 & \underset{\substack{X \in \mathbb{S}^4 \\ x \in \mathbb{R}^4}}{\text{minimize}} && \frac{1}{2} \text{tr}(D^\top D X) - b^\top D x + \frac{1}{2} b^\top b \\
 & \text{s.t.} && X \succeq x x^\top \\
 & && \text{tr}(E X) + c^\top x = 0
 \end{aligned} \tag{P}$$

is equivalent to solving (NC1) for x . The nonconvex constraint with $r(x)$ has been replaced by convex constraints. Provided the solution satisfies $x \succeq 0$, it is also the global minimum of (NC). Moreover, the $\hat{\theta}$ found from the solution is the global minimum of the entire negative log-likelihood (15) whenever it satisfies the nonconvex constraint on future velocities (14). To approximate (NC), we thus solve (P) with the additional constraint $x \succeq 0$, denoted (P1). The proposed method for sampling trajectories is summarized in Algorithm 1. Lemma 1 also tells us that we need sufficiently many observations to ensure D is full rank. This occurs at a minimum of two observations, and in practice we find four to be sufficient.

IV. EXPERIMENTS

To evaluate the proposed method’s ability to predict trajectories in dense traffic for ramp merging, we test it on the NGSIM dataset [13]. The NGSIM dataset includes full trajectory data recorded at 10 Hz for two highways, I-80 and US-101, during peak usage. Since our focus is ramp merging for AVs, we extract relevant pairs of lead and lag vehicles. These pairs are those between which a vehicle entering the highway has merged, or the pair behind such a pair. We are most interested in predicting the behavior of the lag vehicle at the most crucial moment—when it can see the potentially merging ego vehicle. For each pair the start of the prediction window $t = k + 1$ begins when the merging vehicle first passes the lag vehicle. The end of the prediction window $t = T$ occurs either when the ego vehicle passes the lead vehicle or first enters the target lane in case of a merge. All

pairs are observed for 3.2s before the prediction window. This choice of observation window allows us to compare to methods that use more traditional window lengths. We extract 420 pairs from the I-80 data and 292 pairs from the US-101 data.

A. Model Specifications

We set g_0 equal to the mean of the observed gaps. For the precision values, we found that values in $[0.5, 2]$ achieve a good balance between performance and probability calibration. Following this, we set α and β to 1 for all experiments. For importance sampling we define $q(\theta; \hat{\theta})$ as the multivariate normal distribution $\mathcal{N}(\hat{\theta}, I_3)$ truncated to \mathbb{R}_+^3 and draw 1,000 samples. This variance was found to be sufficiently large to sample effectively. The convex problem (P1) is solved with CVXOPT v1.2.3 [26], an open-source solver for convex optimization.

B. Baselines

We compare to state-of-the-art methods for ramp merging and general highway prediction in addition to a simplified version of the proposed model:

- **Constant Velocity (CV):** The average velocity is used to predict future positions.
- **IDM-based (IDM)** [24] : The IDM car-following model [25] is parameterized based on the identified lead vehicle. Unlike other methods it uses the future trajectories of the lead vehicle and the ego vehicle.
- **Social GAN (SGAN)** [9] : Shown to achieve state-of-the-art performance on NGSIM when compared to other neural networks [8] despite originally being designed for joint prediction of pedestrian trajectories.
- **Multi-Agent Tensor Fusion (MATF)** [11] : Achieves state-of-the-art performance on NGSIM using a global pooling layer to capture distant interactions while maintaining spatial structure.
- **No Regularization (Proposed-NR):** The proposed method without regularization terms, corresponding to only the first term of (15).

Each DNN is trained once on each highway dataset. For making predictions on a given scenario, the model that has not seen it during training is used to make predictions. To evaluate these probabilistic predictions from SGAN and MATF we draw 100 samples.

The proposed method uses observations for the lead and lag vehicles, but SGAN and MATF have traditionally been evaluated with observations for all vehicles on the freeway [8], [11]. For comparison we include this standard evaluation, denoted by SGAN* and MATF*. In practice, however, we will accurately detect only nearby vehicles. To reflect this case, we evaluate SGAN and MATF with the same observations as the proposed method, augmented with observations of other vehicles we may reasonably detect from the viewpoint of the ego vehicle. We add observations for three additional vehicles: the lead vehicle’s lead, the ego vehicle, and the ego’s own lead vehicle.

TABLE I: Predictive performance of each method for the NGSIM dataset (best in **bold** and second best underlined). Average distance error (ADE) and root mean squared error (RMSE) are shown as ADE/RMSE in meters. The proposed method achieves the lowest error for short-term predictions, and outperforms the DNNs when the observations are limited to nearby vehicles.

t (s)	Extra information			3.2s observed				
	IDM	SGAN*	MATF*	CV	SGAN	MATF	Proposed-NR	Proposed
0.8	0.65/0.89	0.46/0.75	0.44/0.68	0.67/0.92	0.67/0.96	0.49/0.75	<u>0.37/0.60</u>	0.33/0.60
1.6	1.95/2.60	1.10/1.63	<u>1.00/1.44</u>	1.47/1.97	1.47/2.01	1.20/1.73	1.08/ <u>1.56</u>	0.95/1.62
2.4	3.47/4.73	1.87/2.60	1.56/2.17	2.34/3.42	2.34/3.13	2.08/2.93	1.99/2.77	<u>1.67/2.47</u>
3.2	4.73/6.20	2.78/3.71	2.04/2.81	3.42/4.44	3.35/4.39	3.01/4.24	3.01/4.14	<u>2.54/3.61</u>
4.0	5.57/7.29	3.81/4.99	2.67/3.60	4.63/5.91	4.46/5.77	4.19/5.87	4.25/5.74	<u>3.54/4.88</u>
4.8	5.97/7.72	4.90/6.26	3.22/4.34	5.94/7.60	5.65/7.27	5.42/7.52	5.63/7.55	<u>4.67/6.31</u>

C. Evaluation Metrics

Let $\hat{x}_{i,t}$ be the random variable corresponding to the probabilistic prediction of the lag vehicle’s longitudinal position at timestep t in the i th scenario. The true position is denoted $x_{i,t}$. Since the time horizon varies between scenarios, we denote N_t as the number of scenarios with time horizon $T \geq t$. To evaluate the accuracy of the probabilistic trajectory predictions we evaluate the following metrics:

- *Average Distance Error (ADE)*: The expected distance between the prediction and the true position, used in [8]–[11]. ADE is calculated at timestep t as:

$$ADE(t) = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbb{E}[|x_{i,t} - \hat{x}_{i,t}|]$$

- *Root Mean Squared Error (RMSE)*: The square root of expected squared error between the prediction and the true position, used in [5]–[8], [11]:

$$RMSE(t) = \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbb{E}[(x_{i,t} - \hat{x}_{i,t})^2]}$$

The ADE tells us how prediction errors are distributed on average. Predicting a distribution over positions that has a mean close to the actual position will result in lower error. The RMSE is similar but assigns more weight to larger errors due to the squared term within the expectation. A method with ADE lower than RMSE suggests that it predicts more extreme cases, but assigns lower probability to these.

D. Performance in Dense Traffic Scenarios

The performance for each method is shown in Table I. The model-based method IDM makes overly conservative predictions about braking which hurt its performance. MATF* achieves the lowest errors for long-term predictions by utilizing positional information of all vehicles on the freeway. SGAN* also achieves low error with the same information. Using the same DNNs to make predictions having observed only the more immediate vehicles, however, results in worse performance. The proposed method is evaluated without knowledge of the lead’s own lead vehicle, or the vehicles in the merge lane, yet still outperforms the DNNs. The drop in DNN performance suggests that rather than learning to predict based on cues in nearby drivers’ behavior, they have learned how traffic waves propagate along highways. Comparing the errors made in the short-term and the long-term,

we observe that while the DNNs perform competitively with CV in the short-term, they are better at capturing behavior in the long-term. Fitting the car-following model directly as Proposed-NR is competitive with MATF, though less so at longer-term predictions. Adding the regularization terms enables the proposed method to outperform the baselines, excepting those with full observations of the freeway. Even compared to these methods, the proposed method predicts the short-term with lower error. This makes sense if MATF* learned to focus on the long-term rather than the short-term, relying heavily on observations of vehicles much farther ahead.

E. Performance with Limited Observations

Previously we compared performance ensuring that each method had sufficiently many observations in each scenario. For the real scenarios that AVs will encounter, however, we cannot guarantee that such time will be available. Even within the NGSIM dataset alone, nearly 8% of the scenarios having between 400 ms and 3.2s of observations are removed from our evaluation to ensure traditional observation windows may be used. In this section we make predictions on the same scenarios as before but limit ourselves to 400 ms of observations. Both SGAN and MATF operate on downsampled data. SGAN operates at 2.5 Hz, so for the limited observation window of 400 ms it sees only one observation. To supply the method with the traditional 3.2s of observations, we extrapolate using the constant velocity model from the 400 ms of original observations at 10 Hz. We also extrapolate for MATF which operates at 5 Hz.

The ADE and RMSE for a subset of the methods are shown in Table II. We see that the performance for both SGAN and MATF remains largely unchanged between using the traditional and limited observation windows. This has also been observed in pedestrian trajectory prediction [27] where only the first few observations were found to significantly affect predictive performance of a DNN. The proposed method’s performance decreases for the longer-term predictions, but still outperforms the baselines.

F. Speed

Sudden and critical scenarios in autonomous driving demand methods that operate in realtime. Table III shows the time taken to make predictions for a single scenario.

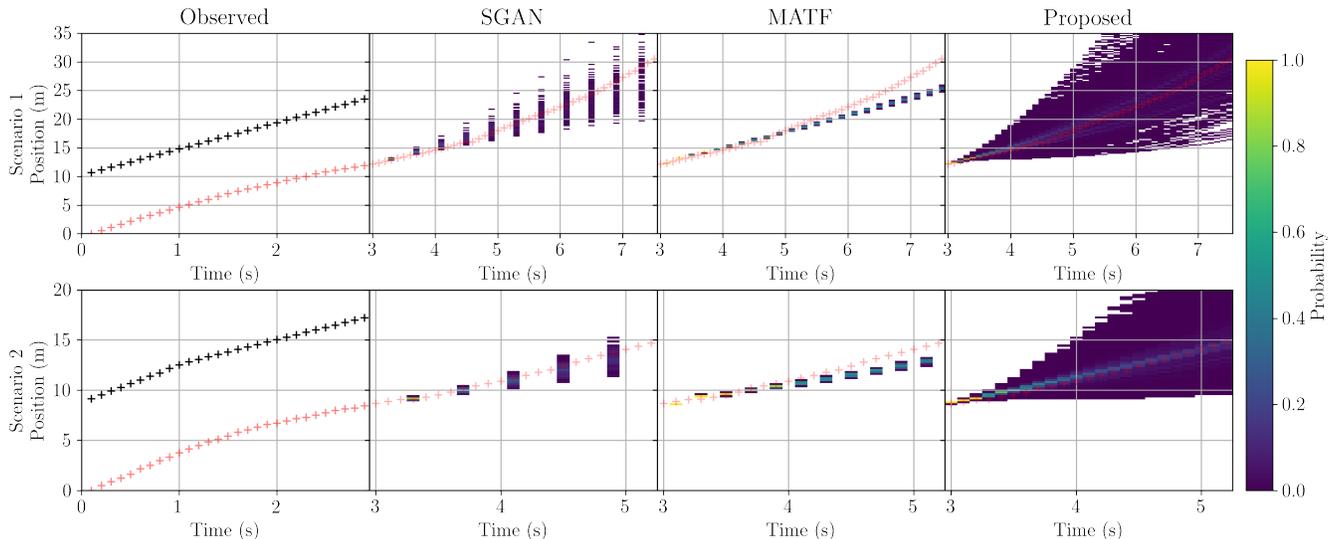


Fig. 4: Predictions of each method on two ramp merging scenarios. The red crosses show the lag vehicle’s position and the black crosses show the observed positions of its lead vehicle. The probabilistic predictions for each method are displayed after the end of the observation window. The color bar (right) provides the probability corresponding to each color. The proposed method predicts the lag vehicle’s positions accurately despite using less information than the other interaction-based methods.

TABLE II: ADE/RMSE with limited observations. CV predicts competitively in the short-term while MATF does so in the long-term. The proposed method retains its performance despite using a short observation window.

t (s)	400 ms observed			
	CV	SGAN	MATF	Proposed
0.8	0.43/0.78	0.65/0.95	0.51/0.82	0.32/0.59
1.6	1.13/1.81	1.46/2.04	1.23/1.84	0.92/1.48
2.4	2.01/2.95	2.35/3.20	2.04/2.92	1.68/2.59
3.2	3.13/4.38	3.34/4.45	2.97/4.19	2.63/3.88
4.0	4.43/6.05	4.40/5.75	4.09/5.74	3.69/5.27
4.8	5.89/7.89	5.60/7.26	5.15/7.24	4.87/6.82

TABLE III: Compute time and probability calibration. The proposed method with 1000 samples and CV predict in realtime, while SGAN and MATF with 100 samples do not. SGAN and the proposed method show calibrated probability estimates, while MATF and CV match in calibration.

	CV	SGAN	MATF	Proposed
Compute time (s)	0.002	0.549	0.908	0.028
Calibration	0.65	0.18	0.65	0.17

SGAN and MATF are benchmarked on GTX 1080 GPU, and the other methods on Intel Core i7-6800K CPU at 3.40 GHz. Although SGAN and MATF do not make realtime predictions with 100 samples, they could do so by reducing the number of samples. This would improve speed at the cost of less accurate probability estimates. The proposed method’s low number of parameters aids in making realtime predictions despite using 1000 samples.

G. Probability Calibration

Probabilistic predictions attach a probability to each predicted outcome and enable planners to calculate risk. The

calculated risk, however, depends on estimates of probability since the true distribution over future outcomes is unknown. The constant velocity model can be viewed as a probabilistic method with degenerate probability estimates. It predicts a single outcome and assigns it full probability. In some sense these probability estimates are not accurate, because vehicles often take different trajectories. We measure this accuracy with a metric for the calibration of regression methods [28]. This measures the squared error between each confidence interval’s probability and the empirical probability of outcomes within the interval being realized. The calibration scores for each method in Table III mirror the predictive distributions in Figure 4. SGAN and the proposed method have calibrated probability estimates, while MATF tends to underestimate the uncertainty in its predictions.

V. CONCLUSION

We propose a novel probabilistic extension for a car-following model and introduce regularization terms to enforce realism in predicted behaviors. Through experiments we demonstrate that these terms lead to increased prediction accuracy for real ramp merging scenarios in dense traffic. Comparing our model to existing methods on the NGSIM dataset shows that it achieves state-of-the-art performance. Furthermore, the proposed model maintains comparable performance when limited to using very few observations. There are multiple limitations to the proposed model that provide grounds for future work. The model considers interactions only between the lag and lead vehicles. Combining this with approaches that consider interactions between the lag and ego vehicles [14], [23], [24] provides one direction for future work. Accounting for lane changes provides another direction.

APPENDIX

We aim to show that the nonconvex problem given in (NC1) is equivalent to the convex reformulation in (P). We first state the dual semidefinite program (SDP) of (P):

$$\begin{aligned} & \underset{s, \mu \in \mathbb{R}}{\text{maximize}} \quad s \\ & \text{s.t.} \quad \begin{pmatrix} \frac{1}{2}D^\top D + \mu E & -b^\top D + \mu c \\ (-b^\top D + \mu c)^\top & \frac{1}{2}b^\top b - s \end{pmatrix} \succeq 0 \end{aligned} \quad (\text{D})$$

We use the following special case of [29, Theorem 6].

Corollary 1. *Let $r : \mathbb{R}^n \rightarrow \mathbb{R}$ be defined as (19). Suppose there exist vectors $x_1, x_2 \in \mathbb{R}^n$ such that $r(x_1) < 0 < r(x_2)$. If the nonconvex problem (NC1) has value that is bounded below, the dual SDP (D) always has an optimal solution (s^*, μ^*) with optimal value equal to the infimum of (NC1). Furthermore the infimum of (NC1) is attained when the dual SDP possesses a feasible set that is not a singleton.*

Proof. This follows immediately from [29, Theorem 6]. \square

Remark 1. *Such x_1, x_2 can easily be found by taking $x_1 = (\frac{1}{2}, \frac{1}{2}, 2, 2)$ and $x_2 = (\frac{1}{2}, \frac{1}{2}, 2, \frac{1}{2})$, yielding $r(x_1) = (\frac{1}{2})2 - 2 < 0 < (\frac{1}{2})2 - \frac{1}{2} = r(x_2)$.*

We now aim to show that the feasible μ are not unique to obtain equivalency.

Lemma 1. *If $D \in \mathbb{R}^{m,n}$ with $m \geq n$ as defined in (D) has full rank, then the formulations (NC1) and (P) are equivalent and the optimal solution is attained.*

Proof. First note that D being full rank implies $D^\top D \succ 0$. For sufficiently small $u \in \mathbb{R}$, $\frac{1}{2}D^\top D + uI \succ 0$. For these u , $\frac{1}{2}D^\top D + \frac{u}{\|E\|_2}E \succ 0$ so the interior of $\{\mu \in \mathbb{R} : \frac{1}{2}D^\top D + \mu E \succeq 0\}$ is nonempty. Since there exist $s \in \mathbb{R}$ such that these μ are feasible for (D), and by the previous remarks, we can apply Corollary 1 to obtain the desired result. \square

REFERENCES

- [1] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, "Interactive scene prediction for automotive applications," in *Proc. IEEE Intell. Vehicles Symp. (IV)*. IEEE, 2013, pp. 1028–1033.
- [2] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech J.*, vol. 1, no. 1, pp. 1–14, 2014.
- [3] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.* IEEE, 2017, pp. 399–404.
- [4] S. Hoermann, M. Bach, and K. Dietmayer, "Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2018, pp. 2056–2063.
- [5] L. Xin, P. Wang, C.-Y. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.* IEEE, 2018, pp. 1441–1446.
- [6] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit. Workshops*, 2018, pp. 1468–1476.
- [7] Y. Hu, W. Zhan, and M. Tomizuka, "Probabilistic prediction of vehicle semantic intention and motion," in *Proc. IEEE Intell. Vehicles Symp. (IV)*. IEEE, 2018, pp. 307–313.
- [8] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Trophic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proc. IEEE Conf. Comput. Vis. Pat. Recognit.*, 2019, pp. 8483–8492.
- [9] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pat. Recognit.*, 2018, pp. 2255–2264.
- [10] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6120–6127.
- [11] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pat. Recognit.*, 2019, pp. 12 126–12 134.
- [12] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE Conf. Comput. Vis. Pat. Recognit.*, 2017, pp. 336–345.
- [13] US Department of Transportation. (2008) Ngsim - next generation simulation. Accessed on: 2019-06-30. [Online]. Available: <http://www.ngsim.fhwa.dot.gov/>
- [14] J. Wei, J. M. Dolan, and B. Litkouhi, "Autonomous vehicle social behavior for highway entrance ramp management," in *Proc. IEEE Intell. Vehicles Symp. (IV)*. IEEE, 2013, pp. 201–207.
- [15] A. Berthelot, A. Tamke, T. Dang, and G. Breuel, "Handling uncertainties in criticality assessment," in *Proc. IEEE Intell. Vehicles Symp. (IV)*. IEEE, 2011, pp. 571–576.
- [16] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *Proc. IEEE Intell. Vehicles Symp. (IV)*. IEEE, 2012, pp. 141–146.
- [17] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. Dietmayer, "Imm object tracking for high dynamic driving maneuvers," in *Proc. IEEE Intell. Vehicles Symp. (IV)*. IEEE, 2004, pp. 825–830.
- [18] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2013, pp. 4363–4369.
- [19] M. Sarvi and M. Kuwahara, "Microsimulation of freeway ramp merging processes under congested traffic conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 470–479, 2007.
- [20] A. Kondyli and L. Elefteriadou, "Modeling driver behavior at freeway-ramp merges," *Transportation Research Record*, vol. 2249, no. 1, pp. 29–37, 2011.
- [21] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr, "A combined model-and learning-based framework for interaction-aware maneuver prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1538–1550, 2016.
- [22] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? a unified framework for maneuver classification and motion prediction," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 129–140, 2018.
- [23] C. Dong, J. M. Dolan, and B. Litkouhi, "Smooth behavioral estimation for ramp merging control in autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*. IEEE, 2018, pp. 1692–1697.
- [24] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, "A belief state planner for interactive merge maneuvers in congested traffic," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.* IEEE, 2018, pp. 1617–1624.
- [25] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.
- [26] M. S. Andersen, J. Dahl, and L. Vandenberghe. (2019, 2) Cvxopt. Accessed on: 2019-08-14. [Online]. Available: <https://cvxopt.org/>
- [27] C. Schöllér, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1696–1703, Apr. 2000.
- [28] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *Int. Conf. Mach. Learn.*, 2018, pp. 2796–2804.
- [29] Y. Xia, S. Wang, and R.-L. Sheu, "S-lemma with equality and its applications," *Math. Program.*, vol. 156, no. 1-2, pp. 513–547, 2016.