Uncertainty Quantification of Car-following Behaviors: Physics-Informed Generative Adversarial Networks

Zhaobin Mo
zm2302@columbia.edu
Department of Civil Engineering and Engineering Mechanics, Columbia University
New York, USA

Xuan Di*
sharon.di@columbia.edu
Department of Civil Engineering and Engineering Mechanics, Columbia University
New York, USA

ABSTRACT

Uncertainty Quantification (UQ) of human driver car-following (CF) behavior is crucial for reliable and robust prediction, given various sources of uncertainty in driving behaviors. There is a growing trend using the physics-informed deep learning (PIDL) for the UQ problems. However, existing studies assume that the uncertainty arises from noisy measurement of external environments, while ignoring intrinsic randomness in the underlying physics. Thus, existing PIDL methods cannot be directly applied to the UQ of CF behavior. To tackle this problem, we propose a novel PIDL model called DoubleGAN, which encodes the stochastic physics into the PIDL structure. We first construct a generative adversarial network (GAN) that captures the uncertain from vehicle trajectory data. Further, we introduce a stochastic CF model to inform the generator with prior physics information. An auxiliary discriminator is used to measure the distributional discrepancy between the GAN prediction and the physics prediction. We demonstrate the effectiveness of our approach through both numerical experiments and a real-world dataset, the Next Generation SIMulation (NGSIM) data. Results demonstrate that the proposed DoubleGAN outperforms the baseline models in terms of both data efficiency and estimation accuracy. Also, the robustness of DoubleGAN has been evaluated through abundant ablation studies.

KEYWORDS

Generative Adversarial Networks (GAN); Physics-informed Deep Learning (PIDL); Uncertainty Quantification (UQ)

1 INTRODUCTION

Characterizing and understanding human car-following (CF) behavior is a fundamental research area that is involved in many urban-computing tasks, such as trajectory prediction [20, 40], traffic signal control [3, 29], and travel time prediction [15, 21]. The existing CF models mainly bifurcate into two categories: *physics-based* [7, 30] and *data-driven* models [11, 34, 35]. Physics-based models approximate the CF behavior based on known physics laws or empirical rules. Those models are easy to interpret, while they may fail to learn complex human behavior due to ideal assumptions. Data-driven models can learn underlying patterns directly from data without prior knowledge or assumptions. However, this type of model may not produce interpretable and physically-consistent results, and is also data-hungry. To increase the interpretability and data efficiency of the data-driven model, the *physics-informed deep learning* (PIDL) model [18–20, 26, 27] previous two. It aims to train a data-driven component that is consistent with the physics. The physics-supervision ensures an improved prediction performance for the data-driven component, especially when the data size is small.

The main challenge for modeling the CF behavior lies in various sources of uncertainty, including internal uncertainty like driver heterogeneity and external uncertainty like measurement noise. Uncertainty quantification (UQ) aims to characterize the CF behavior in a stochastic manner. The most widely used UQ methods include Bayesian approximation [10, 25, 31–33], ensemble methods [8, 14, 23], and generative models like the variational autoencoder [2, 13] and generative adversarial networks (GAN) [22, 28]. There is a growing trend in applying PIDL to UQ. One branch of studies apply physics-informed GANs (PhysGAN) to approximate solutions of partial differential equations (PDE) [4, 38, 39]. However, all these methods assume that the randomness arises from initial conditions or inputs while neglecting stochasticity in parameters associated with inherent physics or behaviors. The other branch of studies apply PhysGAN to solve stochastic differential equations [36, 37]. Although those studies assume that experimental data is generated from stochastic differential equations, they still use deterministic equations to calculate the physics discrepancy. Moreover, they demonstrate the results using only numerical data, and it remains a question whether those models can be applied to real-world cases. To this end, existing PhysGAN models may fail to capture the uncertainty arising from heterogeneity of drivers, which we believe is a major source of randomness when it comes to CF behavior prediction.

To bridge this research gap, we propose a novel PIDL model called DoubleGAN to incorporate stochastic physics into GAN. DoubleGAN contains two generator-discriminator pairs, one (primal GAN) to capture the data uncertainty, and the other (auxiliary GAN) to encode the stochastic physics. In the auxiliary GAN, an auxiliary discriminator is constructed to measure the distributional discrepancy between the generator prediction and the physics prediction. To speed up the convergence of DoubleGAN, we apply the moment-matching technique, which compares the statistical difference between the generator prediction and the physics prediction. This statistical difference provides additional information to the generator as a regularization term to speed up the convergence.

We first evaluate DoubleGAN using numerical data, by which the underlying physics is known and we can ablate the noise type of human behavior. We then apply DoubleGAN to a real-world dataset, the Next Generation SIMulation (NGSIM) dataset, where the underlying physics is unknown. The results demonstrate that

*This author is also affiliated with the Data Science Institute of Columbia University, New York, USA.

DoubleGAN outperforms other baselines in terms of both data efficiency and estimation accuracy. Further, abundant ablation studies are conducted to evaluate the robustness of DoubleGAN.

To the best of our knowledge, we are the first to apply PIDL to the UQ of CF behavior. We are also the first to encode stochastic physics into the PIDL structure. Specifically, the main contributions of this paper include:

- We propose DoubleGAN that encodes stochastic physics to inform the training of GAN.
- We propose to use the moment-matching to speed up the convergence of DoubleGAN.
- We apply DoubleGAN to learn human car-following (CF) behavior. The effectiveness of DoubleGAN is experimentally evaluated using both numerical and real-world data with abundant ablation studies.

2 RELATED WORK AND BACKGROUND

In this section, we first introduce the physics-based CF models, and then the related work and overview of two key concepts that are related to our proposed DoubleGAN, i.e., GAN-based UQ and physics-informed GAN.

2.1 Physics-based CF models

CF models describe a mapping from drivers' states (e.g., spacing headway, velocity difference, and velocity) to actions (e.g., acceleration and target velocity). Denote $\mathcal{S} \subseteq R^{|\mathcal{S}|}$ and $\mathcal{A} \subseteq R$ as the state and action spaces, respectively, where $|\mathcal{S}|$ stands for the cardinality of $\mathcal{S}$. A physics-based CF model f_λ learns the mapping from the state s to the action a, $f : s \rightarrow a$, where $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Examples of physics-based models include the intelligent driving model (IDM) [30] and the Helly model [7]. The IDM model uses the acceleration as its action, and its equation is depicted as:

$$\begin{aligned} &a(t+\Delta t) \\ &= a_{max}\left[1-(v(t)/v_0)^4-\left(s_0+vT_0+\frac{v(t)\Delta v(t)}{2\sqrt{a_{max}b}}\right)^2/\Delta x(t)^2\right] \\ &= f_\lambda(s(t)), \end{aligned} \quad (1)$$

which have 5 parameters: v_0 is the desired velocity, T_0 is the desired time headway, s_0 is the minimum spacing in congested traffic, a_{max} is the maximum acceleration allowed, and b is the comfortable deceleration. The state s is a vector $[\Delta x, \Delta v, v]$. Δx, Δv and v are spacing headway, velocity difference and velocity, respectively.

2.2 GAN based UQ

Suppose (s, a) is an input-output pair, where a follows a conditional probability distribution $p(a|s)$ given its input s. The conditional GAN [17] learns to generate fake outputs that resemble the real ones. In a conditional GAN model, the generator G_θ learns the mapping from the input s and a random noise z to the output a, $G : (s, z) \rightarrow a$. The objective of the generator G_θ is to fool an adversarially trained discriminator D_ϕ. The learning objective of the conditional GAN can be depicted as a min-max game :

$$\begin{aligned} \min_{G_\theta}\max_{D_\phi} &\mathbb{E}_{q(s)p(z)}\left[\log D_\phi(s, G_\theta(s,z))\right] \\ &+\mathbb{E}_{q(s,a)}\left[\log(1-D_\phi(s,a))\right], \end{aligned} \quad (2)$$

where θ and ϕ are the parameters of the generator and the discriminator, respectively. $q(s, a)$ is the joint distribution of inputs and outputs, and $q(s)$ is the marginal distribution of inputs. $p(z)$ is the distribution for a random noise, which is a standard normal distribution in the conditional GAN model. For simplicity, "conditional GANs" are referred to as "GANs" in the remainder of this paper.

2.3 Physics-informed GAN

Let $\{(s_o^{(i)}, a_o^{(i)})\}_{i=1}^{N_o}$ be a labeled set of input-output pairs and $\{s_c^{(j)}\}_{j=1}^{N_c}$ be an unlabeled set, where N_o and N_c are their sizes. We aim to train the generator G_θ to minimize Eq. 2 on the labeled set; we also want G_θ's prediction $\hat{a}_c$ to be close to the prediction of a (deterministic) physics-based model $\tilde{a}_c = f_\lambda(s_c)$ on the unlabeled set. The *physics loss* is defined as the discrepancy between the neural network prediction $\hat{a}_c$ and the physics-based model prediction $\tilde{a}_c$. Existing studies quantify the discrepancy between $\hat{a}_c$ and $\tilde{a}_c$ using the mean squared error (MSE):

$$\mathcal{L}_c(\theta, \lambda) = \frac{1}{N_c}\sum_{j=1}^{N_c}\left|\hat{a}_c^{(j)} - \tilde{a}_c^{(j)}\right|^2. \quad (3)$$

Note that $\hat{a}_c = G_\theta(s_c, z)$ is a random variable conditioned on s_c, while $\tilde{a}_c$ is deterministically determined by s_c. Minimizing Eq. 3 makes the G_θ's prediction distribution $p_\theta(\hat{a}_c|s_c)$ be centered around $\tilde{a}_c$ with the variance approaching zero, which is the so-called mode collapse (proof in the appendix).

The loss function of the generator considering the physics loss can then be depicted as:

$$\begin{aligned} \mathcal{L}_G(\theta, \lambda) &= \alpha \cdot \mathcal{L}_o(\theta) + (1-\alpha)\cdot\mathcal{L}_c(\theta,\lambda) \\ &= \frac{\alpha}{N_o}\sum_{i=1}^{N_o} D_\phi(s_o^{(i)}, G_\theta(s_o^{(i)}, z_o^{(i)})) + (1-\alpha)\mathcal{L}_c(\theta,\lambda), \end{aligned} \quad (4)$$

where, $\alpha \in [0, 1]$ is a hyper-parameter that balances the data loss $\mathcal{L}_o$ and physics loss. Note that the physics loss is calculated on the unlabeled set, where the neural network prediction $\hat{a}_c$ is compared to the physics-based model prediction $\tilde{a}_c$ instead of the ground-truth value.

In the remainder of this paper, we keep the above subscripts to distinguish between inputs from different sets (e.g. s_o for labeled inputs and s_c for unlabeled inputs). In addition, we use $\hat{()}$ and $\tilde{()}$ to represent predictions of the neural network and the physics-based model, respectively.

3 METHODOLOGY

3.1 Problem statement

UQ of CF behavior is illustrated in Fig. 1. A red car is following a blue car along the horizontal axis, and the vertical axis is time. It is assumed that a driver obeys an underlying stochastic policy $\pi(a|s)$ that maps from driving states $s \in \mathcal{S}$ to a distribution over actions $a \in \mathcal{A}$. A CF model learns a surrogate policy $\pi_\theta(a|s)$ that approximates the ground-truth policy $\pi(a|s)$. At time step t, the red car samples its action a given its current state s, which leads to the true position (solid red car) at time step $t + \Delta t$. Meanwhile, a surrogate model π_θ predicts the action distribution and sample an action $\hat{a}$, which leads to the estimated position (transparent red car) at time step $t + \Delta t$. The key problem is to quantify the uncertainty of prediction $\hat{a}$ and its discrepancy with regards to the true action a.

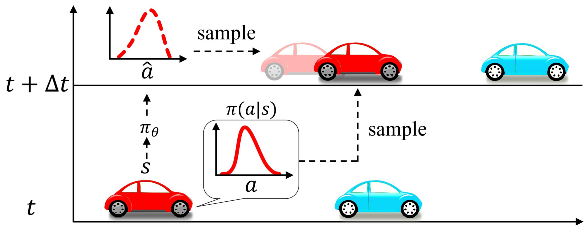


Figure 1: An illustration of UQ for CF behavior

3.2 Double-GAN for encoding stochastic physics

In the existence of stochastic underlying physics, the aforementioned physics model f_λ in Eq. 1 becomes a stochastic one. Without loss of generality, we represent this stochastic model by $\tilde{a}_c = f_\lambda(s_c, z)$, where the random noise z represents either the parametric (internal) noise in λ or the measurement (external) noise in $\tilde{a}_c$. Thus, the physics loss quantifies the discrepancy between two distributions $p_\theta(\hat{a}_c|s_c)$ and $p_\lambda(\tilde{a}_c|s_c)$. We propose to construct an auxiliary discriminator to distinguish between those two distributions, which is the main idea of the proposed DoubleGAN.

The structure of the DoubleGAN is illustrated in Fig. 2. It consists of two parts: the left half (blue) contains the primal GAN and relevant variables; the right half (red) contains the auxiliary GAN and relevant variables. We will explain the left part first and then the right part. The primal GAN consists of the generator G_θ and the primal discriminator D_ϕ. Labeled states s_o are fed into the generator together with random noise z. The predicted state-action pairs $(s_o, \hat{a}_o)$ and the labeled state-action pairs (s_o, a_o) are judged by the primal discriminator D_ϕ, and the data loss is thus computed as $\mathcal{L}_o(\theta) = \frac{1}{N_o}\sum_{i=1}^{N_o} D_\phi(s_o^{(i)}, \hat{a}_o^{(i)})$, which is the same as in Eq. 4.

The auxiliary GAN consists of the generator G_θ and the auxiliary discriminator D'_η. On one hand, unlabeled states s_c and random noise z are fed into the physics equation to generate physics predictions $\tilde{a}_c$. On the other hand, s_c and z are fed into the generator to get predictions $\hat{a}_c$. The auxiliary discriminator D'_η is trained to distinguish the generator-predicted state-action pairs $(s_c, \hat{a}_c)$ from the physics-predicted state-action pairs $(s_c, \tilde{a}_c)$, from which we can revise the physics loss in Eq. 3 as

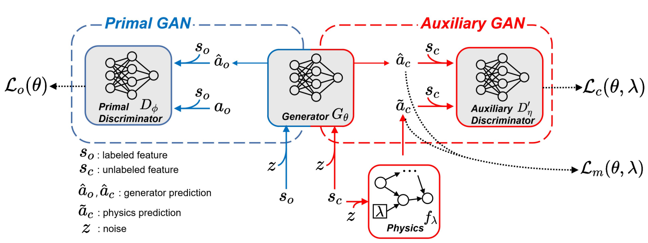


Figure 2: Structure of the DoubleGAN

$$\mathcal{L}_c(\theta, \lambda) = \frac{1}{N_c}\sum_{j=1}^{N_c} D'_\eta(s_c^{(j)}, \hat{a}_c^{(j)}). \tag{5}$$

For simplicity, the loss functions of the discriminators D_ϕ and D'_η are not shown in Fig. 2. As physics is not incoporated into the discrimimantors, the loss functions of D_ϕ and D'_η keep the original forms as in Eq. 2, i.e.,

$$\begin{aligned}\mathcal{L}_{\mathcal{D}}(\phi) &= -\frac{1}{N_o}\sum_{i=1}^{N_o}[\log D_\phi(s_o^{(i)}, \hat{a}_o^{(i)}) + \log(1 - D_\phi(s_o^{(i)}, a_o^{(i)}))], \\ \mathcal{L}_{\mathcal{D}'}(\eta) &= -\frac{1}{N_c}\sum_{j=1}^{N_c}[\log D'_\eta(s_c^{(j)}, \hat{a}_c^{(j)}) + \log(1 - D'_\eta(s_c^{(j)}, \tilde{a}_c^{(j)}))].\end{aligned} \tag{6}$$

3.3 Moment-matching for faster convergence

Although Eq. 5 encodes the stochastic physics that better captures the real-world noise, it incurs an additional adversarial loss, which may hinder the model convergence. To tackle this problem, we propose to use the moment-matching technique to speed up the convergence of the DoubleGAN without incurring the model collapse. The moment-matching loss is depicted as:

$$\mathcal{L}_m(\theta, \lambda) = \rho\mathbb{E}_{q(s_c)}\left[\frac{\mu(\tilde{a}_c) - \mu(\hat{a}_c)}{|\mu(\tilde{a}_c)| + |\mu(\hat{a}_c)|}\right]^2 + (1-\rho)\mathbb{E}_{q(s_c)}\left[\frac{\sigma(\tilde{a}_c) - \sigma(\hat{a}_c)}{|\sigma(\tilde{a}_c)| + |\sigma(\hat{a}_c)|}\right]^2, \tag{7}$$

where the first and the second terms measure the discrepancies of the mean and the standard deviation between the neural network predictions and the physics predictions, respectively. μ is the operator for the mean, and σ is the operator for the standard deviation. To mitigate the effect of the scale difference between the first and the second moments, the sum of the absolute values of each moment is added as a normalization term. $\rho \in [0, 1]$ is the ratio of each constraint component. Details of calculating the moment-matching loss are shown in Algorithm 1.

3.4 Posterior Estimator

Posterior estimators are employed to mitigate the mode collapse of GANs [16]. The basic idea is to use a posterior estimator $Q(s, \hat{a})$, which is an additional neural network, to learn the mapping from the generated state-action pair to the posterior probability of the latent variable $z : (s, \hat{a}) \mapsto p(z|s, \hat{a})$. A *reconstruction error* is defined as the expectation of the negative log-likelihood, which is depicted as below:

$$\mathcal{L}_r(\theta) = -\mathbb{E}_{q(s_o)p(z)}[\log Q_\xi(s_o, \hat{a}_o)] - \mathbb{E}_{q(s_c)p(z)}[\log Q'_\psi(s_c, \hat{a}_c)], \tag{8}$$

where, Q_ξ and Q'_ψ are two posterior estimators for the labeled and unlabeled sets, respectively. ξ and ψ are their parameters.

3.5 Joint Estimation

The physics model is jointly trained along with other networks by minimizing the physics loss $\mathcal{L}_c(\theta, \lambda)$ in Eq. 5 with regard to both generator parameters θ and physics parameters λ on the unlabeled

Algorithm 1 Calculation of the moment-matching loss of DoubleGAN.

Require: generator G_θ; physics model f_λ

Initialization: the number of samples for Monte Carlo approximation $n_s = 50$; hyperparameter ρ

Input: unlabeled set $\{s_c^{(j)}\}_{j=1}^{N_c}$

Output: moment-matching loss $\mathcal{L}_m(\theta, \lambda)$

```
for j = 1 : N_c do
    for k = 1 : n_s do
        sample z ~ N(0,1), â^(j,k) = G_θ(s_c^(j), z)
        sample z ~ N(0,1), ã^(j,k) = f_λ(s_c^(j), z)
    end for
    μ(â)^(j) = 1/n_s Σ_{k=1}^{n_s} â^(j,k)
    μ(ã)^(j) = 1/n_s Σ_{k=1}^{n_s} ã^(j,k)
    σ(â)^(j) = sqrt( 1/(n_s-1) Σ_{k=1}^{n_s} (â^(j,k) - μ(â^(j)))^2 )
    σ(ã)^(j) = sqrt( 1/(n_s-1) Σ_{k=1}^{n_s} (ã^(j,k) - μ(ã^(j)))^2 )
end for
```

$$\mathcal{L}_m(\theta,\lambda) = \rho \frac{1}{N_c}\sum_{j=1}^{N_c}\left[\frac{\mu(\tilde{a})^{(j)}-\mu(\hat{a})^{(j)}}{|\mu(\tilde{a})^{(j)}|+|\mu(\hat{a})^{(j)}|}\right]^2 + (1-\rho)\frac{1}{N_c}\sum_{j=1}^{N_c}\left[\frac{\sigma(\tilde{a})^{(j)}-\sigma(\hat{a})^{(j)}}{|\sigma(\tilde{a})^{(j)}|+|\sigma(\hat{a})^{(j)}|}\right]^2$$

data. We illustrate the joint estimation in Fig. 3. The line colors are associated with different types of data: the blue for the labeled data and the red for the unlabeled data. The solid lines indicate how generator G_θ and physics f_λ are trained: the generator G_θ is trained by both the labeled data and the samples of the physics f_λ, and the physics f_λ is trained by the samples of the generator G_θ. The dashed line indicates that the physics could be pre-trained by the labeled data prior to the joint estimation.

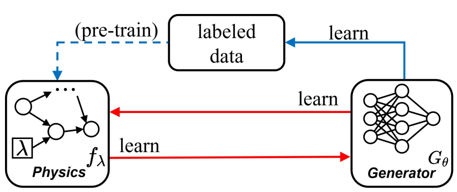


Figure 3: Joint estimation: training the physics model and the generator simutaneously.

3.6 Training algorithm

The loss of the generator G_θ is :

$$\mathcal{L}_G(\theta,\lambda) = \alpha \cdot \mathcal{L}_o(\theta) + (1-\alpha)\cdot \mathcal{L}_c(\theta,\lambda) + \beta \cdot \mathcal{L}_m(\theta,\lambda) + \gamma \cdot \mathcal{L}_r(\theta), \tag{9}$$

where $\beta, \gamma \in [0, +\infty)$ are hyperparameters that control the weights of the moment-matching and reconstruction losses, respectively. The Adam optimizer [12] is used for training, and the details of the training process is summarized in Algorithm 2.

4 EXPERIMENT RESULTS

In this section, we will first introduce the experiment setting, including the dataset, baselines, evaluation metrics, hyperparameters, and the computation platform. Then we will show the experiment results and discussions.

Algorithm 2 Training process of DoubleGAN.

Require: generator G_θ; primal discriminator D_ϕ; auxiliary discriminator D'_η; posterior estimators Q_ξ and Q'_ψ; Adam optimizer

Initialization: Pre-trained physics parameters λ^0; Initialized networks parameters θ^0, ϕ^0, η^0, ξ^0, and ψ^0; Epochs *epochs*; Batch size m; Learning rate lr; Clipping parameter c; Weights of loss functions α, β, and γ

Input: labeled set $\{(s_o^{(i)}, a_o^{(i)})\}_{i=1}^{N_o}$ and unlabeled set $\{s_c^{(j)}\}_{j=1}^{N_c}$.

```
for iter ∈ {1, ..., epochs} do
    Sample batches {(s_o^(i), a_o^(i))}_{i=1}^m and {s_c^(j)}_{j=1}^m from the labeled and unlabeled sets, respectively
    Sample noises {z^(i)}_{i=1}^m and {z^(j)}_{j=1}^m from a standard normal distribution
    // update the primal and auxiliary discriminators
    Calculate L_D and L_D' by Eq. 6
    ϕ ← ϕ − lr · Adam(ϕ, ∇_ϕ L_D)
    η ← η − lr · Adam(η, ∇_η L_D')
    // update the physics
    Calculate L_c by Eq. 5
    λ ← λ − lr · Adam(λ, clip(∇_λ L_c, −c, c))
    // update the generator and posterior estimators
    Calculate L_m by Algorithm 1
    Calculate L_G by Eq. 9
    θ ← θ − lr · Adam(θ, ∇_θ L_G)
    ξ ← ξ − lr · Adam(ξ, ∇_ξ L_G)
    ψ ← ψ − lr · Adam(ψ, ∇_ψ L_G)
end for
```

Dataset. DoubleGAN is evaluated using both numerical and real-world CF data. Numerical data is used because its underlying physics parameters are known and controllable. This helps to ablate the noise types and also to evaluate whether DoubleGAN can discover the underlying physics while training, which cannot be done using the real-world dataset. We will then apply DoubleGAN to a real-world dataset to demonstrate its performance for the real-world scenario where the underlying physics is unknown.

The numerical data is generated from a known IDM equation as Eq. 1. To simulate the internal and external uncertainties, we artificially add Gaussian noise to the ground-truth model parameters along with the model outputs. We also add the log-normal noise to the model outputs as an alternative. The distributions for parameters and model outputs are depicted as:

$$s \sim \text{Uniform}(l, u); \lambda \sim \mathcal{N}(\mu_\lambda, \Sigma_\lambda),$$
$$\begin{cases} \tilde{a} \sim \mathcal{N}(f_\lambda(s|\lambda), \sigma_a^2), \\ \text{or} \\ \tilde{a} \sim \text{Lognormal}(f_\lambda(s|\lambda), \sigma_a^2), \end{cases} \tag{10}$$

where l and u are the lower bound and upper bound of the state $s = [\Delta x, \Delta v, v]$. $\lambda = [v_0, T_0, s_0, a_{max}, b]$ is the parameter vector of the IDM model f_λ, which follows a Gaussian distribution. Σ_λ is a diagonal matrix as we assume no correlation for different parameters for simplicity. We set $l = [10m, -8m/s, 2m/s]$ and

$u = [50m, 8m/s, 10m/s]$ to cover different traffic regimes, such as decelerating, cruising and accelerating. The mean values and covariance of parameters are set to be $\mu_\lambda = [10, 1, 1, 1, 1]$ and $\Sigma_\lambda = diag(1, 0.01, 0.01, 0.01, 0.01)$. The standard deviation for the model output is $\sigma_a = 0.05$. Accelerations smaller than $-2m/s^2$ are clipped to avoid unrealistic braking.

The real-world data is from the Next Generation SIMulation (NGSIM) dataset[24], which is an open dataset that collects vehicle trajectories every 0.1 second. We focus on the US Highway 101.

Baselines. We compare the proposed PhysGANs with 3 baselines: a GAN model, a PhysGAN using a deterministic physics loss as Eq. 3, and a deterministic physics-informed neural network with a Monte Carlo (MC)-Dropout [14], which is denoted as PINN-Drop.

Evaluation metrics. We use two metrics to measure the deviation of every test point from the mean of our model prediction: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). In addition, we use two metrics to evaluate the difference between the prediction distribution and the sample distribution: Kullback–Leibler (KL) divergence and Negative Log Predictive Density (NLPD) computed using the Parzen window. To mitigate the randomness in both training and test procedures, all models are trained for 3 rounds, and each trained model is evaluated 10 times with new test data. The mean of each metric is recorded.

Hyperparameters and platforms. The generator, discriminator, and posterior estimator share the same network complexity, which consists of 4 layers with 20 neurons in each layer. The He uniform initializer [6] is used. All models are trained using an Adam optimizer with a learning rate of 0.001 and other hyperparameters as default. Each model is trained for 5000 epochs, and the batch size is 128. The clipping parameter $c = 1$. Sizes of the labeled (N_o) and unlabeled (N_f) data are 500 and 1000, respectively. We randomly generate another 200 labeled data as test data. Experiments were conducted on a local workstation with 16 Intel Xeon W-2145 CPUs and an NVIDIA Titan RTX GPU with 24 GB memory in Ubuntu 18.04.3.

4.1 Numerical Data

4.1.1 Performance comparison. The results of DoubleGAN and baselines when the training size $N_o = 500$ are summarized in Table. 1. The first column is the name of the proposed model and baselines. “Moment-only” means the DoubleGAN with only the moment-matching loss, i.e., $\alpha = 1$ and $\beta > 0$ in Eq. 9; “DoubleGAN$^-$” means the DoubleGAN without using the moment-matching loss, i.e., $a < 1$ and $\beta = 0$ in Eq. 9. Other columns record the mean of each metric. The best and the worst scores are bolded and underscored, respectively. We interpret these results from the following two perspectives.

Comparing our methods to baselines. All of our methods outperform baselines in terms of all metrics in general (except for some specific metrics, which will be discussed later). Although the PhysGAN achieves a compelling KL for the normal noise data, it achieves the worst KL among all methods for the log-normal noise data. In comparison, the superior performance of our methods is consistent for both noise types of data.

Comparison among our methods. Among our methods, DoubleGAN achieves the best performance for both noise types of data.

Table 1: The CF behavior prediction results of learning numerical data generated from a known IDM model.

Normal Noise				
Model	RMSE	MAE	KL	NLPD
PINN-Drop	0.207	0.1220	1.171	−0.044
GAN	0.237	0.156	2.203	−0.042
PhysGAN	0.137	0.077	0.607	−0.549
Moment-only (ours)	0.138	0.081	0.640	−0.551
DoubleGAN$^-$ (ours)	0.121	0.078	0.580	**−0.836**
DoubleGAN (ours)	**0.117**	**0.075**	**0.543**	−0.832
Log-normal Noise				
Model	RMSE	MAE	KL	NLPD
PINN-Drop	0.228	0.159	1.171	−0.032
GAN	0.205	0.145	1.173	−0.186
PhysGAN	0.117	0.090	1.373	−0.714
Moment-only (ours)	0.114	0.089	0.896	−0.718
DoubleGAN$^-$ (ours)	0.134	0.094	1.103	**−0.733**
DoubleGAN (ours)	**0.109**	**0.084**	**0.758**	−0.729

Although the NLPD of the DoubleGAN$^-$ is slightly better than the DoubleGAN, this difference is very small. Neither Moment-only nor DoubleGAN$^-$ can outperform baselines in terms of all metrics and noise types. For the normal noise data, Moment-only achieves a similar RMSE as PhysGAN does. For the log-normal data, DoubleGAN$^-$ achieves a similar KL as GAN and PINN-Drop do. In contrast, DoubleGAN outperforms other methods for both noise types of data, which demonstrates the robustness of DoubleGAN against noise types.

4.1.2 Ablation of training size and moment-matching. To demonstrate the generalizability of the proposed DoubleGAN in the sparsity of labeled data, Fig. 4 shows the RMSE (left) and KL (right) of all GAN-based methods. We can see that the performance of DoubleGAN degrades the least compared to others as we reduce the training size. This is because DoubleGAN is imposed with physics information from both the auxiliary GAN and the moment-matching, thus better exploiting physics information from unlabeled data. We also notice that mode collapse happens for the PhysGAN. Its RMSE is lower than the GAN's across all training sizes, while its KL is generally much higher than the GAN. A low RMSE with a high KL is an indication that PhysGAN weighs more on learning the mean of the sample distribution despite the shape of the sample distribution.

To evaluate the effectiveness of the moment-matching technique, we present the prediction results of the DoubleGAN$^-$ and DoubleGAN during the training process in Fig. 5, which corresponds to the normal noise data case with the training size $N_o = 500$. The x-axis is the index of the training data points, which is sorted by the value of the acceleration. The y-axis is the acceleration. The blue and green lines are the mean of the ground-truth and the prediction, respectively; the yellow band is the two-standard (2-σ band) of the ground-truth. We can see that, imposed with the moment-matching, DoubleGAN converges much faster than DoubleGAN$^-$. The reason is that, the discriminators are not well-trained at the early-training

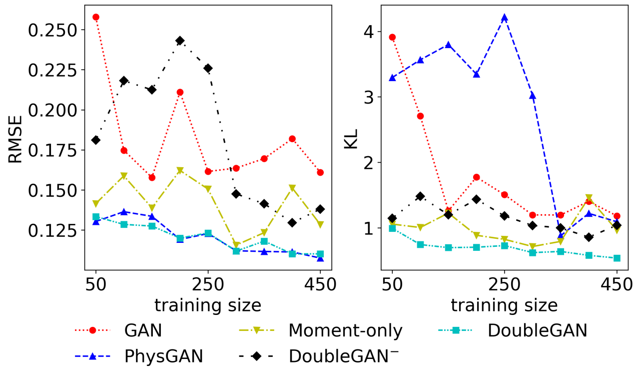
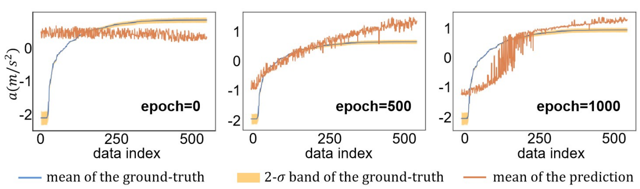
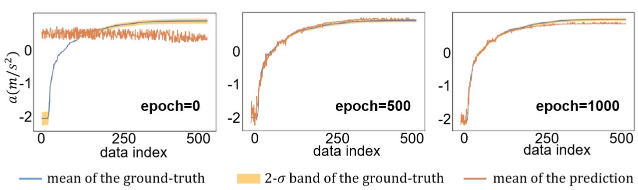


Figure 4: Effect of training size on learning numerical data generated from a known IDM model



(a) Predictions of DoubleGAN⁻ during training



(b) Predictions of DoubleGAN during training

Figure 5: Predictions of DoubleGAN⁻ (top row) and DoubleGAN (bottom row) during the training process. The comparison between the model prediction and the ground-truth is presented at training epochs 0, 500, and 1000

stage and thus cannot supervise the generator very well. In comparison, moment-matching directly computes the moment discrepancy between the generator and the physics, which can assist with the training of the generator throughout the training process.

4.1.3 Joint discovery of physics parameters. The jointly estimated physics parameters of DoubleGAN are shown in Table. 2. The first column is the number of labeled data, and the other columns are the relative errors for each IDM parameter. When the training size increases, the model parameters converge to $a_{max} = 0.964$, $v_0 = 10.270$, $b = 1.036$, $T_0 = 0.935$, and $s_0 = 1.069$, which are close to the ground-truth values. These results show the ability of DoubleGAN for addressing both UQ of CF behavior and parameter discovery simultaneously.

4.1.4 Visualization. We compare the prediction distribution of DoubleGAN to the sample distribution of normal noise data (N_o =

Table 2: Results of the mean of the jointly estimated physics parameters by learning numerical data generated from a known IDM model.

N_o	a_{max}(%)	v_0 (%)	b (%)	T_0 (%)	s_0(%)
50	4.10	4.71	15.45	37.60	32.10
250	3.86	2.89	9.67	13.30	21.75
500	3.63	2.70	3.60	6.65	6.90

The relative errors are computed by comparing to the ground-truth mean parameters $a_{max} = 1$, $v_0 = 10$, $b = 1$, $T_0 = 1$, and $s_0 = 1$.

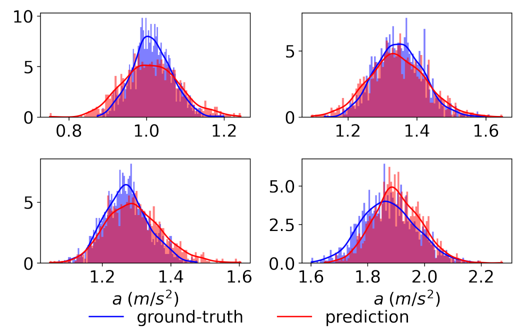


Figure 6: Visualization of predictions of DoubleGAN for learning numerical data generated from a known IDM model

500) at 4 randomly sampled data points in Fig. 6. The blue and red colors represent the ground-truth and the prediction, respectively. We can see that most parts of the predicted and ground-truth distributions overlap with each other, which demonstrates that DoubleGAN can capture the CF uncertainty of the numerical data well.

4.2 NGSIM Data

Because the underlying physics of the NGSIM dataset is unknown, we need to determine the physics equation to guide our DoubleGAN. We use two CF physics equations, the aforementioned stochastic IDM model in Eq. 10 and a Helly model, for ablation studies. A stochastic version of the Helly model can be depicted as:

$$\begin{cases} \mu_a = \lambda_v \Delta v + \lambda_x (\Delta x - D), \\ a \sim \mathcal{N}(\mu_a, \sigma_a^2), \end{cases} \tag{11}$$

where λ_v, λ_x, and D are parameters of the Helly model that follows Gaussian distributions. The reason to choose the Helly model is that, despite its simple form, it can achieve good performance for the NGSIM dataset. We introduce the comparison of various CF equations and calibration detail in the appendix.

4.2.1 Performance comparison. The results of DoubleGAN and baselines on the NGSIM dataset (training size N_o=500) are shown in Table. 3. The column meaning is the same as Table. 1. We can see that our proposed methods outperform the baselines in terms of all metrics. Among our proposed methods, Moment-only and DoubleGAN-IDM⁻ perform equally worse compared to DoubleGAN-IDM and DoubleGAM-Helly, which demonstrate that moment-matching or auxiliary GAN alone may not learn real-world uncertainty well. Despite that IDM performs better than the Helly model (see appendix), DoubleGAN-Helly outperforms DoubleGAN-IDM in terms of all metrics. One possible explanation is that the complexity of IDM may hinder the training process when it is jointly trained with neural networks.

Table 3: The CF behavior prediction results of DoubleGAN and baselines on NGSIM data.

Model	RMSE	MAE	KL	NLPD
PINN-Drop	1.287	0.989	0.782	1.454
GAN	0.844	0.627	1.487	0.752
PhysGAN	0.793	0.584	1.455	0.752
Moment-only (ours)	0.788	0.589	0.619	0.794
DoubleGAN-IDM$^-$ (ours)	0.786	0.596	0.804	0.796
DoubleGAN-IDM (ours)	0.771	0.583	0.507	0.723
DoubleGAN-Helly (ours)	**0.764**	**0.574**	**0.409**	**0.709**

4.2.2 Ablation of training sizes and physics types. The results under varying training sizes (N_o =50, 100, 250,300, 450, and 500) are shown in Fig. 7, which share the same axes and line specification as Fig. 4. From the left figure, we can see that all models achieve similar RMSEs. While in the right figure, DoubleGAN has a significantly lower KL divergence than other models across all training sizes, which demonstrates that DoubleGAN can better capture the stochastic pattern of the real-world CF data. We notice that when data is sparse ($N_o = 50$), PhysGAN achieves a similar KL as the GAN does, which indicates PhysGAN's deterministic way of encoding physics fails to help the generator to capture the uncertain patterns. We also notice that Moment-only achieves lower KLs than DoubleGAN-IDM$^-$ when $N_o = 50$ and $N_o = 100$. This can be explained by that the auxiliary discriminator of DoubleGAN-IDM$^-$ requires more data to train compared to the moment-matching of Moment-only. When imposed with both moment-matching and auxiliary GAN, DoubleGAN-IDM and DoubleGAN-Helly achieve the lowest KLs compared to other methods, especially when the training size is small. These results demonstrate the superiority of DoubleGAN in terms of both estimation accuracy and data efficiency for real-world CF data.

Comparing DoubleGAN-IDM to DoubleGAN-Helly, we can see that DoubleGAN-Helly outperforms DoubleGAN-IDM across all training sizes, except for $N_o = 300$ that might be an outlier, which indicates that the Helly model can better assist the generator as the physics component. Same as in the previous subsection, this can be explained by that the complexity of the IDM may hinder the training process when it is jointly trained with neural networks. This comparison shows that the most suitable physics model for DoubleGAN may not necessarily be the one with the best performance. The simplicity of physics is also a consideration.

4.2.3 Visualization. We compare the prediction distribution of DoubleGAN with sample distribution at 4 randomly samples data points in Fig. 8 ($N_o = 500$), which share the same axes and line specification as Fig. 6. Most parts of the predicted and ground-truth distributions overlap with each other, which demonstrates that DoubleGAN can capture the CF uncertainty of the real-world data well.

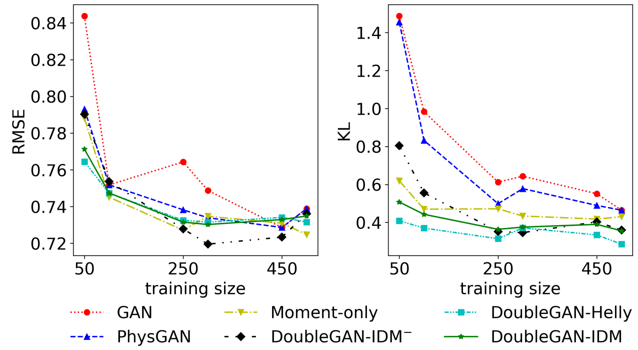


Figure 7: Effect of training size on learning from NGSIM data

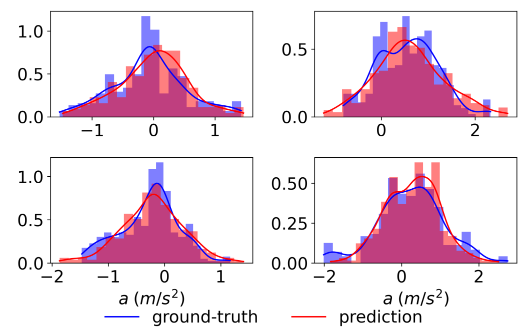


Figure 8: Visualization of predictions of DoubleGAN for the NGSIM data

5 CONCLUSION

In this paper, we propose a novel method called DoubleGAN that quantifies the uncertainty in human driver car-following (CF) behavior. The model encodes the stochastic physics information into the physics-informed generative adversarial network (PhysGAN) without incurring mode collapse. Using numerical data, we evaluate the performance of DoubleGAN under different noise types and training sizes. We further investigate the performance of DoubleGAN on a real-world dataset, the NGSIM dataset, and demonstrate that it outperforms baseline methods under different training sizes. Through ablation studies, we confirm that the moment-matching technique can speed up the model convergence and thus achieve better performance. By comparing DoubleGAN-IDM to DoubleGAN-Helly, we show that the most suitable physics model for DoubleGAN may not necessarily be the one with the best performance. The simplicity of physics is also a consideration.

This work can be further improved in two directions. First, apart from the weighted sum, other approaches to integrating the data loss, moment-matching loss, and reconstruction loss can be proposed. Second, this work can be extended to quantify the uncertainty in sequential behavior, e.g., uncertainty quantification of human driving trajectory prediction.

ACKNOWLEDGEMENTS

This work is partially sponsored by NSF CPS-2038984.

A MODE COLLAPSE OF PHYSGAN

A.1 Proof of Mode Collapse of PhysGAN

Recall that $\{(s_o^{(i)}, a_o^{(i)})\}_{i=1}^{N_o}$ is a labeled set and $\{(s_c^{(j)}, a_c^{(j)})\}_{j=1}^{N_c}$ is an unlabeled set, where N_o and N_c are their sizes. Predictions of the generator and the physics are denoted as $\hat{a} = G_\theta(s, z)$ and $\tilde{a} = f_\lambda(s, z)$, respectively. The loss function of PhysGAN's generator is shown as :

$$\mathcal{L}_G(\theta, \lambda) = \alpha \mathbb{E}_{q(s_o)p(z)}\left[D_\phi(s_o, \hat{a}_o)\right] + (1-\alpha)\mathcal{L}_c(\theta, \lambda), \tag{12}$$

where the physics loss $\mathcal{L}_c$ carries the form as Eq. 3. We can re-write this physics loss function $\mathcal{L}_c(\theta, \lambda)$ as:

$$\begin{aligned}
&\mathcal{L}_c(\theta, \lambda) = \mathbb{E}_{q(s_c)p(z)}\left[\,|\hat{a}_c - \tilde{a}_c|^2\,\right] \\
&= \mathbb{E}_{q(s_c)p(z)}\left[\hat{a}_c^2 - 2\hat{a}_c\tilde{a}_c + \tilde{a}_c^2\right] \\
&= \mathbb{E}_{q(s_c)}\left\{\mathbb{E}_{p(z)}\left[\hat{a}_c^2\right] - 2\mathbb{E}_{p(z)}\left[\hat{a}_c\tilde{a}_c\right] + \mathbb{E}_{p(z)}\left[\tilde{a}_c^2\right]\right\} \\
&= \mathbb{E}_{q(s_c)}\left\{\mathbb{E}_{p(z)}\left[\hat{a}_c^2\right] - 2\mathbb{E}_{p(z)}\left[\hat{a}_c\right]\mathbb{E}_{p(z)}\left[\tilde{a}_c\right] + \mathbb{E}_{p(z)}\left[\tilde{a}_c^2\right]\right. \\
&\left. - 2Cov_{p(z)}\left(\hat{a}_c, \tilde{a}_c\right)\right\},
\end{aligned} \tag{13}$$

where $Cov_{p(z)}$ stands for the covariance with regards to the latent variable z. As the generator and the physics are two different models without sharing parameters, we assumed their predictions $\hat{a}$ and $\tilde{a}$ are independent. Thus, $Cov_{p(z)}(\hat{a}_c, \tilde{a}_c) = 0$ and we have the following:

$$\begin{aligned}
&\mathbb{E}_{p(z)}\left[\hat{a}_c^2\right] - 2\mathbb{E}_{p(z)}\left[\hat{a}_c\right]\mathbb{E}_{p(z)}\left[\tilde{a}_c\right] + \mathbb{E}_{p(z)}\left[\tilde{a}_c^2\right] \\
&= \left(\mathbb{E}_{p(z)}\left[\hat{a}_c^2\right] - |\mathbb{E}_{p(z)}\hat{a}_c|^2\right) + \\
&\left(|\mathbb{E}_{p(z)}\hat{a}_c|^2 - 2\mathbb{E}_{p(z)}\left[\hat{a}_c\right]\mathbb{E}_{p(z)}\left[\tilde{a}_c\right] + |\mathbb{E}_{p(z)}\tilde{a}_c|^2\right) + \\
&\left(-|\mathbb{E}_{p(z)}\tilde{a}_c|^2 + \mathbb{E}_{p(z)}\left[\tilde{a}_c^2\right]\right) \\
&= Var_{p(z)}(\hat{a}_c) + \left|\mathbb{E}_{p(z)}[\hat{a}_c] - \mathbb{E}_{p(z)}[\tilde{a}_c]\right|^2 + \underbrace{Var_{p(z)}(\tilde{a}_c)}_{independent\ of\ \theta},
\end{aligned} \tag{14}$$

where $Var_{p(z)}$ stands for the variance with regard to the latent variable z. The final form of Eq. 14 has three terms. The first term is the variance of $\hat{a}_c$. The second term is the difference between expectations of $\hat{a}_c$ and $\tilde{a}_c$. The third term is the variance of $\tilde{a}_c$, which is independent of the generator's parameter θ and thus can be dropped. Substituting Eq. 14 into Eq. 13, we have the following:

$$\mathcal{L}_c(\theta, \lambda) = \mathbb{E}_{q(s_c)}\Big[\underbrace{Var_{p(z)}(\hat{a}_c)}_{first\ term} + \underbrace{\left|\mathbb{E}_{p(z)}[\hat{a}_c] - \mathbb{E}_{p(z)}[\tilde{a}_c]\right|^2}_{second\ term}\Big]. \tag{15}$$

A.1.1 Remark. There are two terms in the physics loss function of PhysGAN, as shown in Eq. 15. The first term is the variance of the generator's prediction. The second term is the difference between the expectation of the prediction of the generator and that of the physics. By minimizing $\mathcal{L}_m$ with regard to θ, the first term aims to make the variance of the generator's prediction as small as possible, **which may lead to mode collapse**; the second term aims to make the expectations of the generator's prediction as close to that of the physics as possible.

A.2 Mitigating mode collapse with moment-matching

We replace the first term of Eq. 15 with a square distance between the standard deviation of the generator's prediction and that of the physics. The resulting equation is the moment-matching loss function:

$$\begin{aligned}
&\mathcal{L}_m(\theta, \lambda) \\
&= \mathbb{E}_{q(s_c)}\Big[\underbrace{\left|\sigma_{p(z)}(\hat{a}_c) - \sigma_{p(z)}(\tilde{a}_c)\right|^2}_{first\ term} + \underbrace{\left|\mathbb{E}_{p(z)}[\hat{a}_c] - \mathbb{E}_{p(z)}[\tilde{a}_c]\right|^2}_{second\ term}\Big],
\end{aligned} \tag{16}$$

where $\sigma_{p(z)}$ stands for the standard deviation with regard to the latent variable z. We can see that minimizing the first term of $\mathcal{L}_m$ does not require minimizing the variance of $\hat{a}_c$. Thus, $\mathcal{L}_m$ does not incur mode collapse. Note that in Eq. 7, we omit $p(z)$ for simplicity and normalize the moments.

B COMPARISON AMONG DIFFERENT CF MODELS

There are various CF physics equations, like IDM model, Helly model, Optimal Velocity model (OVM) [1], Full Velocity Difference model (FVDM) [9], and Gazis-Herman-Rothery (GHR) model [5]. We compare the performance of all aforementioned CF equations for the NGSIM data with a 500 training size. The results are shown in Table. 4. The IDM model achieves the best performance, followed by the Helly model. Although the advantage of the Helly model over the FVDM and GHR models is not significant, the Helly model is much simpler. Thus, we choose the IDM and the Helly models as our physics equations.

Table 4: Comparison of different CF models.

	IDM	Helly	OVM	FVDM	GHR
RMSE	0.587	0.695	0.758	0.706	0.696

REFERENCES

[1] Masako Bando, Katsuya Hasebe, Akihiro Nakayama, Akihiro Shibata, and Yuki Sugiyama. 1995. Dynamical model of traffic congestion and numerical simulation. *Physical review E* 51, 2 (1995), 1035.

[2] Vanessa Böhm, François Lanusse, and Uroš Seljak. 2019. Uncertainty quantification with generative models. *arXiv preprint arXiv:1910.10046* (2019).

[3] K Chandan, Alvaro M Seco, and Ana Bastos Silva. 2017. Real-time traffic signal control for isolated intersection, using car-following logic under connected vehicle environment. *Transportation research procedia* 25 (2017), 1610–1625.

[4] Arka Daw, M. Maruf, and Anuj Karpatne. 2021. PID-GAN: A GAN Framework Based on a Physics-Informed Discriminator for Uncertainty Quantification with Physics. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2021), 237–247.

[5] Denos C Gazis, Robert Herman, and Richard W Rothery. 1961. Nonlinear follow-the-leader models of traffic flow. *Operations research* 9, 4 (1961), 545–567.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[7] Walter Helly. 1959. Simulation of bottlenecks in single-lane traffic flow. (1959).

[8] Ruihan Hu, Qijun Huang, Sheng Chang, Hao Wang, and Jin He. 2019. The MBPEP: a deep ensemble pruning algorithm providing high quality uncertainty prediction. *Applied Intelligence* 49, 8 (2019), 2942–2955.

[9] Rui Jiang, Qingsong Wu, and Zuojin Zhu. 2001. Full velocity difference model for a car-following theory. *Physical Review E* 64, 1 (2001), 017101.

[10] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. 2020. Hands-on Bayesian Neural Networks–a Tutorial for Deep Learning Users. *arXiv preprint arXiv:2007.06823* (2020).

[11] ByeoungDo Kim, Chang Mook Kang, Jaekyum Kim, Seung Hi Lee, Chung Choo Chung, and Jun Won Choi. 2017. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 399–404.

[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[14] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems* 30 (2017).

[15] Jinwoo Lee and Amer Shalaby. 2013. Rule-based transit signal priority control method using a real-time transit travel time prediction model. *Canadian Journal of Civil Engineering* 40, 1 (2013), 68–75.

[16] Chunyuan Li, Jianqiao Li, Guoyin Wang, and Lawrence Carin. 2018. Learning to sample with adversarially learned likelihood-ratio. (2018).

[17] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[18] Zhaobin Mo, Yongjie Fu, and Xuan Di. 2022. Quantifying Uncertainty In Traffic State Estimation Using Generative Adversarial Networks. In *2022 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE (in press), arXiv preprint arXiv:2206.09349.

[19] Zhaobin Mo, Yongjie Fu, Daran Xu, and Xuan Di. 2022. TrafficFlowGAN: Physics-informed Flow based Generative Adversarial Network for Uncertainty Quantification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer (in press), arXiv preprint arXiv:2206.09319.

[20] Zhaobin Mo, Rongye Shi, and Xuan Di. 2021. A physics-informed deep learning paradigm for car-following models. *Transportation research part C: emerging technologies* 130 (2021), 103240.

[21] Simon Oh, Young-Ji Byon, Kitae Jang, and Hwasoo Yeo. 2018. Short-term travel-time prediction on highway: A review on model-based approach. *KSCE Journal of Civil Engineering* 22, 1 (2018), 298–310.

[22] Dhruv V Patel and Assad A Oberai. 2020. GAN-based Priors for Quantifying Uncertainty. *arXiv preprint arXiv:2003.12597* (2020).

[23] Tim Pearce, Felix Leibfried, and Alexandra Brintrup. 2020. Uncertainty in neural networks: Approximately bayesian ensembling. In *International conference on artificial intelligence and statistics*. PMLR, 234–244.

[24] Vincenzo Punzo, Maria Teresa Borzacchiello, and Biagio Ciuffo. 2011. On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data. *Transportation Research Part C: Emerging Technologies* 19, 6 (2011), 1243–1262.

[25] Mizanur Rahman, Mashrur Chowdhury, Taufiquar Khan, and Parth Bhavsar. 2015. Improving the efficacy of car-following models with a new stochastic parameter estimation and calibration method. *IEEE Transactions on Intelligent Transportation Systems* 16, 5 (2015), 2687–2699.

[26] Rongye Shi, Zhaobin Mo, and Xuan Di. 2021. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 540–547.

[27] Rongye Shi, Zhaobin Mo, Kuang Huang, Xuan Di, and Qiang Du. 2021. A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation. *IEEE Transactions on Intelligent Transportation Systems* (2021).

[28] Vinicius LS Silva, Claire E Heaney, and Christopher C Pain. 2021. GAN for time series prediction, data assimilation and uncertainty quantification. *arXiv preprint arXiv:2105.13859* (2021).

[29] Tie-Qiao Tang, Zhi-Yan Yi, and Qing-Feng Lin. 2017. Effects of signal light on the fuel consumption and emissions under car-following model. *Physica A: Statistical Mechanics and its Applications* 469 (2017), 200–205.

[30] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical review E* 62, 2 (2000), 1805.

[31] Chris P IJ van Hinsbergen, Hans WC van Lint, Serge P Hoogendoorn, and Henk J van Zuylen. 2009. Bayesian calibration of car-following models. *IFAC Proceedings Volumes* 42, 15 (2009), 91–97.

[32] Hao Wang and Dit-Yan Yeung. 2016. Towards Bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering* 28, 12 (2016), 3395–3408.

[33] Kuan-Chieh Wang, Paul Vicol, James Lucas, Li Gu, Roger Grosse, and Richard Zemel. 2018. Adversarial distillation of bayesian neural network posteriors. In *International Conference on Machine Learning*. PMLR, 5190–5199.

[34] Xiao Wang, Rui Jiang, Li Li, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. 2017. Capturing car-following behaviors by deep learning. *IEEE Transactions on Intelligent Transportation Systems* 19, 3 (2017), 910–920.

[35] Da Yang, Liling Zhu, Yalong Liu, Danhong Wu, and Bin Ran. 2018. A novel car-following control model combining machine learning and kinematics models for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems* 20, 6 (2018), 1991–2000.

[36] Liu Yang, Sean Treichler, Thorsten Kurth, Keno Fischer, David Barajas-Solano, Josh Romero, Valentin Churavy, Alexandre Tartakovsky, Michael Houston, Mr Prabhat, et al. 2019. Highly-scalable, physics-informed GANs for learning solutions of stochastic PDEs. In *2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*. IEEE, 1–11.

[37] Liu Yang, Dongkun Zhang, and George Em Karniadakis. 2020. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing* 42, 1 (2020), A292–A317.

[38] Yibo Yang and Paris Perdikaris. 2019. Adversarial uncertainty quantification in physics-informed neural networks. *J. Comput. Phys.* 394 (2019), 136–152.

[39] Dongkun Zhang, Lu Lu, Ling Guo, and George Em Karniadakis. 2019. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *J. Comput. Phys.* 397 (2019), 108850.

[40] Xiaohui Zhang, Jie Sun, Xiao Qi, and Jian Sun. 2019. Simultaneous modeling of car-following and lane-changing behaviors using deep learning. *Transportation research part C: emerging technologies* 104 (2019), 287–304.