Physica A 639 (2024) 129671

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Parameter estimation for Gipps' car following model in a Bayesian framework

Samson Ting [a,*], Thomas Lymburn [a,b], Thomas Stemler [a], Yuchao Sun [a,b], Michael Small [a,c]

[a] The Complex Systems Group, Department of Mathematics and Statistics, The University of Western Australia, Perth, Western Australia, Australia
[b] Planning and Transport Research Centre (PATREC), The University of Western Australia, Perth, Western Australia, Australia
[c] Mineral Resources, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Kensington, Western Australia, Australia

ARTICLE INFO

Keywords:
Model calibration and validation
Bayesian inference
Uncertainty quantification
Adaptive Markov chain Monte Carlo

ABSTRACT

Car following model is an important part in traffic modelling and has attracted a lot of attentions in the literature. As the proposed car following models become more complex with more components, reliably estimating their parameters becomes crucial to enhance model predictive performance. While most studies adopt an optimisation-based approach for parameters estimation, we present a statistically rigorous method that quantifies uncertainty of the estimates. We present a Bayesian approach to estimate parameters using the popular Gipps' car following model as demonstration, which allows proper uncertainty quantification and propagation. Since the parameters of the car following model enter the statistical model through the solution of a delay-differential equation, the posterior is analytically intractable so we implemented an adaptive Markov Chain Monte Carlo algorithm to sample from it. Our results show that predictive uncertainty using a point estimator versus a full Bayesian approach are similar with sufficient data. In the absence of adequate data, the former can make over-confident predictions while such uncertainty is more appropriately incorporated in a Bayesian framework. Furthermore, we found that the congested flow parameters in the Gipps' car following model are strongly correlated in the posterior, which not only causes issues for sampling efficiency but more so suggests the potential ineffectiveness of a point estimator in an optimisation-based approach. Lastly, an application of the Bayesian approach to a car following episode in the NGISM dataset is presented.

1. Introduction

The use of traffic models to make predictions and hence inform decision-making has become ubiquitous. Car following models in particular have gained a lot of research interests due to their foundational role in traffic flow. The recently proposed car following models [1–3] have grown in complexity and included more and more components in attempt to produce more realistic behaviours. Their predictive performance often depend crucially on appropriate parameter values, which raises the importance of parameters estimation. Such parameter estimation task is more commonly referred to as model calibration and validation in the transport research community. Although conceptually similar, there are subtle differences between estimating parameters for a model and calibrating a model.

* Corresponding author.
E-mail address: samson.ting@research.uwa.edu.au (S. Ting).

https://doi.org/10.1016/j.physa.2024.129671
Received 11 December 2023; Received in revised form 25 February 2024
Available online 12 March 2024

Let $D = (X, Y) \sim R = p(X, Y)$ denote the data which comes from the data generating process R that we call *reality* where X and Y are the random vectors denoting *predictor* and *prediction* respectively. We will inevitably treat R as a stochastic process regardless of whether it is inherently stochastic (aleotoric uncertainty) or we as modellers simply do not have completely knowledge over it (epistemic uncertainty). Let $\hat{Y} \sim F(X, \theta)$ denote the model prediction for Y given X made by a model $F(\cdot)$ parametrised by θ with feasible parameter space Θ. In statistical inference, parameters estimation usually assumes the data generating process belongs to the parametric family of the model, i.e. $\exists \theta^* \in \Theta : F(x, \theta^*) = p(Y|X = x)$ so one can speak of the *true parameter values*. On the other hand, model calibration in transport modelling implicitly assumes the structurally imperfect $F(\cdot)$ and hence aims to find best-fitting parameter values. Indeed, one can view model calibration as estimating $\theta^* := \text{argmin}_{\theta \in \Theta} \mathbb{E}_{(X,Y)}[\|F(X, \theta) - Y\|]$ which is the parameter values that minimise the *generalised error* based on some distance metric.

The most common approach in the transport literature for model calibration is then to construct an *extremum estimator* as the solution to an optimisation problem [4–12]. Such point estimators often take the form $\hat{\theta} = \text{argmin}_{\theta \in \Theta} \frac{1}{|D|} \sum_{(x_i, y_i) \in D} \|F(x_i, \theta) - y_i\|$ to minimise the training error, which is essentially an estimate of θ^* with finite samples. Under regularity conditions, the weak law of large number guarantees $\hat{\theta} \to \theta^*$ in probability as $|D| \to \infty$. From this perspective, such an optimisation-based approach is sensible.

However, there are several issues associated with point estimators which are often overlooked in existing literature. Firstly, subsequent model predictions on new data treat the estimate $\hat{\theta}$ as if it was θ^*. This not only neglects the fact that the training/validation error is only an estimate of the generalised error, but also the Monte Carlo error when numerically solving the typically intractable optimisation problem. As a result, it ignores the uncertainty in $\hat{\theta}$ when making subsequent predictions and hence predictive uncertainty is typically underestimated. This can be somewhat addressed with suitable regularisation and cross-validation to ensure the predictive performance conditioning on $\hat{\theta}$ is still acceptable. The second and more important issue is that for principle-driven models, the parameters might be assigned physical meanings so their estimated values are used for subsequent inference. For instance, one might draw further conclusions about driver's aggressiveness based on their estimated minimum headway to their preceding vehicle during car following. This is a deeper statistical question which we discussed in detail in a previous work [13]. Strictly speaking, imperfect models by definition do not fully reflect the reality so model parameters are not necessarily informative about the true data generating process. Even if one assumes $F(X, \theta)$ is a good approximation of $p(\cdot|X)$ and hence the assigned physical meaning of θ is close to its counterpart in reality, inference based on a point estimate without proper uncertainty quantification is potentially misleading.

In this research, a Bayesian approach is adopted which is a flexible and statistically coherent framework for uncertainty quantification and propagation. We performed Bayesian inference for a generalisation of the popular Gipps' car following model where the integration time step is not necessarily the reaction time. Since the likelihood depends on the car following parameters through the solution of a delayed differential equation, the posterior is analytically intractable so we implemented an adaptive Markov chain Monte Carlo algorithm to sample from it. We demonstrate the advantages of a Bayesian approach in handling uncertainty compared to an optimisation-based approach, at the cost of computations. Furthermore, we found strong correlation between the congested flow behavioural parameters in the Gipps' car following model, which not only reduces efficiency of the sampler but more so suggests the need for inference beyond simple point estimators. Lastly, we applied the methods to real data using the NGISM dataset. While signs of model mis-specification emerge, the model fit appears satisfactory with reasonable predictive uncertainty on validation data. The contribution in this paper is two-fold, namely (1) the demonstration of Bayesian inference as a useful framework for parameters estimation in transport modelling, and (2) its application to estimate parameters for the Gipps' car following model which entails implementation of an adaptive Markov chain Monte Carlo algorithm to sample from the posterior.

The paper is structured as follows: Section 2 introduces the Gipps' car following model and details our implementation of the corresponding delay-differential equation solver, Section 3 provides some background on model calibration and validation, with an emphasis on Bayesian inference and its computations, Section 4 details the statistical model and the adaptive MCMC sampler, Section 5 presents the results and discussions, Section 6 demonstrates an application to real dataset, and Section 7 concludes the paper.

2. Gipps' car following model

Car following describes the longitudinal movement of a pair of vehicles called the leader and follower as illustrated in Fig. 1, where the states of the vehicles are the usual kinematic variables namely displacement $x(t)$, velocity $v(t)$ and acceleration $a(t)$. It is one of the most important components in traffic micro-simulator. The Gipps' car following model originally proposed by [14], belongs to the class of collision avoidance model and is arguably one of the most popular car following models. It assumes the follower chooses a safe speed should its leader brake abruptly. The model updates the follower's velocity in one reaction time according to the following equations:

$$v_f(t + \tau) = \min(v^{\text{ff}}(t + \tau), v^{\text{cf}}(t + \tau)) \tag{1}$$

$$v^{\text{ff}}(t + \tau) = v_f(t) + 2.5 a_{\max} \tau \left(1 - \frac{v_f(t)}{V_{\max}}\right) \sqrt{0.025 + \frac{v_f(t)}{V_{\max}}} \tag{2}$$

$$v^{\text{cf}}(t + \tau) = b_{\max} \tau + \sqrt{b_{\max}^2 \tau^2 - b_{\max} \left[2(x_\ell(t) - l_c - x_f(t)) - v_f(t)\tau - \frac{v_l^2(t)}{\hat{b}_\ell}\right]} \tag{3}$$

where τ is the reaction time; $a_{\max}, b_{\max}(<0)$ and $V_{\max}$ are the follower's maximum desired acceleration, deceleration and velocity respectively; l_c is the clearance length for the leader; $\hat{b}_\ell$ is the follower's estimate of the leader's maximum desired deceleration

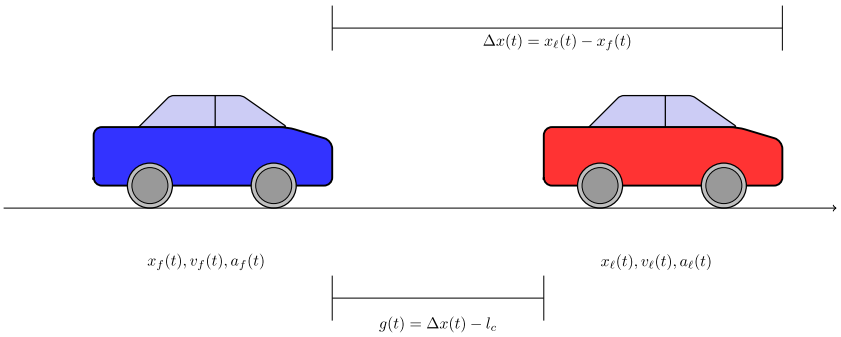

Fig. 1. Illustration of car following, red vehicle is the leader.

usually assumed to be the leader's actual maximum desired deceleration scaled by the sensitivity parameter ψ i.e. $\hat{b}_\ell = \psi b_{\max,\ell}$. The model assumes the follower will be either in free flow described by (2) where it is not impeded by the leader, or in congested flow described by (3) where the opposite is true, whichever is more restrictive. Note that the original formulation included an additional safety term $\tilde{\theta}$ when calculating $v^{\text{cf}}(\cdot)$ [14,15]. The formulation in (3) is a specific case of $\tilde{\theta} = \tau/2$ which is commonly adopted in the literature.

2.1. Solving Gipps' model

The general case of the Gipps' update rules (1)–(3) combined with the usual kinematic relationships $\dot{x}_f(t) = v_f(t)$ form a delayed differential equation (DDE) [16] that describes the evolution of the follower's state $(x_f(t), v_f(t))$. In Gipps' original paper [14] and most implementations in the literature, the integration time step Δt is chosen to be the same as the reaction time τ. In such case, the solution simplifies to that of a difference equation which performs discrete updates. While this is erroneous for chaotic systems as it ignores dynamics at a time scale smaller than τ, from a modelling perspective and the definition of τ it appears to be a sensible choice. We implemented a generalisation of the Gipps' car following model where Δt can be a submultiple of τ, i.e. $\tau = M\Delta t, M \in \mathbb{N}$. Furthermore, an initial interval $\{x_f(t), v_f(t) | 0 \le t < \tau\}$ is technically required as opposed to just the initial conditions $x_f(0), v_f(0)$ in solving ordinary differential equation.

We implemented a simplified DDE solver as shown in algorithm 1. Let $f_{\text{Gipps}}(\cdot)$ denote the aforementioned follower velocity update rule (1) with calibration parameter γ. The leader's displacement $x_\ell(t)$ and velocity $v_\ell(t)$ are viewed as stimulus to the system that describes the follower's state. For simplicity, instead of estimating the entire initial interval, we assumed it is determined by the initial condition and interpolated the initial interval assuming constant acceleration. Subsequent integration in time is simply performed with the method of steps.

Algorithm 1 Gipps' car following model solver

Input: follower velocity update rules $f_{\text{Gipps}}(\cdot)$ as per Gipps' car following model, initial conditions $x_f(0), v_f(0)$, integration time step Δt, simulation period T, leader's trajectory as stimulus $\{x_\ell(t), v_\ell(t) | 0 \le t \le T\}$, calibration parameters γ
Output: The follower's trajectory as system response $\{x_f(t), v_f(t) | 0 \le t \le T\}$
Calculate $v_f(\tau) = f_{\text{Gipps}}(x_f(0), v_f(0); x_\ell(0), v_\ell(0), \gamma)$
Interpolate the initial interval assuming constant acceleration:
$a = \frac{v_f(\tau) - v_f(0)}{\tau}$
$v_f(t) = v_f(t - \Delta t) + a\Delta t, \quad x_f(t) = x_f(t - \Delta t) + \frac{\Delta t}{2}(v_f(t) + v_f(t - \Delta t)) \quad \text{for} \quad t \in (0, \tau]$
Numerically integrate forward in time using method of steps:
for $t > \tau$ **do**
    $v_f(t + \Delta t) = f_{\text{Gipps}}(x_f(t + \Delta t - \tau), v_f(t + \Delta t - \tau); x_\ell(t + \Delta t - \tau), v_\ell(t + \Delta t - \tau), \gamma)$
    $x_f(t + \Delta t) = x_f(t) + \frac{\Delta t}{2}(v_f(t) + v_f(t + \Delta t))$
    $a_f(t) = \frac{v_f(t+\Delta t) - v_f(t)}{\Delta t}$ ▷ assumes constant acceleration during Δt

In synthetic data generation, we assumed the leader also follows the Gipps' free flow component, with random perturbations imposed in the form of forced acceleration/deceleration. Otherwise, the leader and follower will simply approach a steady state in the absence of perturbations. A sample trajectory simulated with Gipps' car following model is shown in Fig. 2. In the case of $\Delta t = \tau$, we recover the usual implementation of Gipps' model as a discrete update equation. A comparison between the two is given in Appendix A.

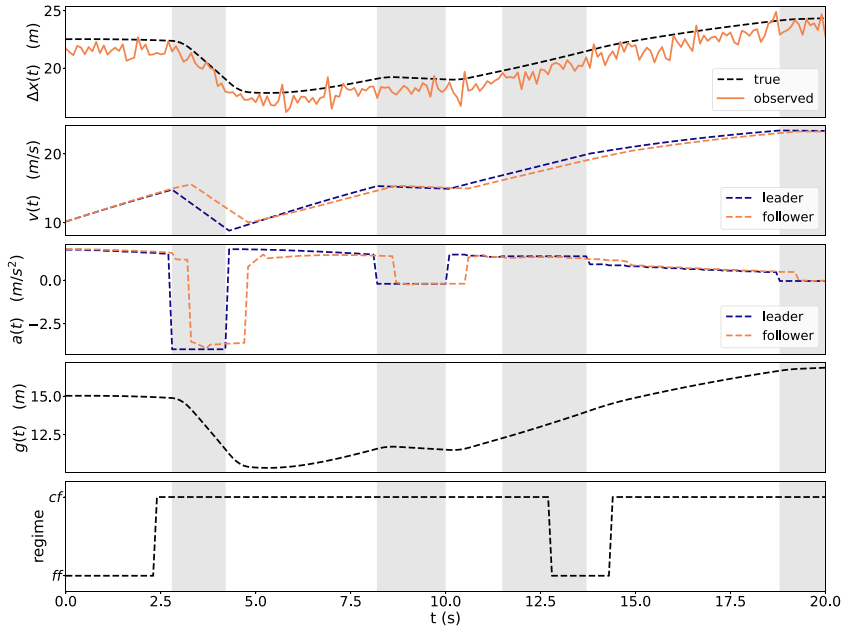

Fig. 2. A sample trajectory simulated with Gipps' car following model using the DDE solver described in algorithm 1. The dashed lines describe the corresponding latent state variables, the solid line in the top panel is the noisy displacement time series which is the only observable quantity. The fourth row describes the gap between the vehicles $g(t) = \Delta x(t) - l_c$. The fifth row describes the follower regime as either in congested flow (cf) or free flow (ff). The shaded panels indicate random perturbations imposed on the leader's acceleration/deceleration.

2.2. Parameters to be estimated

In the Gipps' model, the follower's response $(x_f(t), v_f(t))^T$ is fully deterministic given the stimulus $(x_l(t), v_l(t))^T$, the calibration parameters γ and the initial conditions. Uncertainty arises in the observation process where the follower's response is observed with noise. We assumed the follower's displacement time series is observed with additive Gaussian noise with mean μ_x and variance σ_x^2, which is conditionally i.i.d. given latent follower's displacement $x_f(t)$. The parameters to be estimated are the main driver behavioural parameters $a_{\max}$, $V_{\max}$ and $b_{\max}$, the initial conditions $x_f(0)$ and $v_f(0)$, and the noise parameters μ_x and σ_x^2. For notational convenience, we will refer to $\gamma = (a_{\max}, V_{\max}, b_{\max}, x_f(0), v_f(0))$ as the calibration parameters, and $\theta = (a_{\max}, V_{\max}, b_{\max}, x_f(0), v_f(0), \mu_x, \sigma_x^2)$ as the full parameters to be estimated. Other unobservable quantities such as τ and ψ are technically also parameters to be estimated. We omitted τ because it introduces further complication in the numerical integration scheme. While the sampler described in Section 4.2 can be easily extended to incorporate ψ and still converge asymptotically, we found that the mixing of the chains is slow so convergence of the sampler might take many more iterations. This issue and potential solutions are discussed more extensively in Section 5.3. Hence we restrict ourselves to only a subset of the parameters. While this is a limitation of our study, we believe it still serves the purpose of demonstrating the applicability of a Bayesian approach.

3. Model calibration and validation

3.1. Optimisation-based approach

The most common approach in the literature to estimate the parameters in car following model is to formulate it as an optimisation problem [4,7,10,11,17–20]. Most notably, the estimator $\hat{\theta}$ is the minimiser of some error metric $\|\cdot\|$ between model output and observed output, i.e.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|F(x, \theta) - y\| \tag{4}$$

where (x, y) is the data realisation and $F(\cdot)$ is the statistical model. Strictly speaking, $F(\cdot)$ should model the data generating process and hence should include both the underlying dynamics described by car following model and the observation process. However most literature seems to treat $F(\cdot)$ as only the car following model, with some exception such as [18] who showed that neglecting observation noise can induce bias in the estimate. This is however not the most apparent weakness of an optimisation-based approach, as one could easily include an observation noise model by replacing the objective function in (4) with the likelihood which gives the *maximum likelihood estimator* (MLE).

Instead, a major weakness of an optimisation-based approach is that it only constructs a point estimator, and it is usually not obvious how one can obtain statistically coherent uncertainty quantification. In such approach, $\hat{\theta}$ is effectively treated as known

and used for all subsequent predictions. Although one might conduct cross-validation to assess whether $\hat{\theta}$ generalises to new data set, such training-validation pipeline successfully applied in machine learning is however not fully transferable to traffic models for several reasons. Firstly, traffic models are not *universal approximators*. One usually does not know what the minimum achievable error is, and hence what an acceptable error threshold should be. It is quite possible that for a given error threshold, there are no parameter values where both training error and validation error fall below this threshold. In addition, the parameters in traffic models usually have physical meanings. Aside from only monitoring the training and validation error as in machine learning, in model calibration one is usually also interested in monitoring whether the estimated parameter values converge to plausible values.

Even if cross-validation shows that $\hat{\theta}$ as a point estimate is useful for prediction, one has to be careful before drawing further inferences based on $\hat{\theta}$ about reality. Such interpretability is fundamentally why one might choose a principle-driven traffic model over a machine learning model to begin with. However, the validity of any subsequent inference depends on how reliable the estimates are. Hence, there is often a need for proper uncertainty quantification about the estimated parameter values.

3.2. Bayesian inference

Statistical inference, and Bayesian inference in particular, provides a flexible and statistically coherent framework for uncertainty quantification. Bayesian statistics differ from classical statistics in their interpretation of probability as degree of belief, so uncertainty is associated with a lack of information. Bayesian inference treats parameters as random variables, and describes uncertainty about them with probability distributions. For an excellent and detailed account of Bayesian statistics, see [21].

Let θ denote the parameters and D denote the data, the driving principle is the Bayes' rule:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{5}$$

$$\propto p(D|\theta)p(\theta) \tag{6}$$

where $p(\theta)$ is the *prior* which represents the uncertainty about the parameters before one observes the data; $p(D|\theta)$ is the *likelihood* which is the statistical model of the data generating process and describes the relative plausibility of observing the data given different values of θ; and $p(\theta|D)$ is the *posterior* which represents the remaining uncertainty about the parameters after one has observed the data. The intuition is that observing D hopefully provides information and allows us to update our uncertainty about θ.

The posterior distribution $p(\theta|D)$, which is central in Bayesian inference, contains all the information for almost every question one can ask about θ given D. One might then be interested in computing various summaries of the posterior such as:

- Point estimators such as the *posterior mean*: $\mathbb{E}[\theta|D] = \int_{\theta\in\Theta} \theta p(\theta|D)d\theta$
- Interval estimators[1] such as a $100\alpha\%$ *credible interval*: $[a, b]$ s.t. $\int_a^b p(\theta|D)d\theta = \alpha$
- or equivalently, posterior probability $P(a \le \theta \le b) = \int_a^b p(\theta|D)d\theta$.

Furthermore, the uncertainty in θ can be propagated into the observation space via the *posterior predictive distribution*

$$p(\tilde{y}|D) = \int_\Theta p(\tilde{y}|\theta)p(\theta|D)d\theta \tag{7}$$

when making prediction for a new observation $\tilde{y}$. It can be interpreted as using all possible values of θ weighted by its posterior density. An optimisation-based approach that constructs a point estimator $\hat{\theta}$ essentially approximates $p(\theta|D)$ with a point mass $\delta_{\hat{\theta}}$.

In Bayesian inference, the full posterior distribution $p(\theta|D)$ is the only estimator of θ, unlike the numerous possible combinations of distance metric in an optimisation-based approach. The assumptions made in a Bayesian model are explicit in the choice of prior and likelihood, from which the posterior must be a logical consequence.

3.2.1. Bayesian computation and MCMC

In likelihood based inference where the likelihood $p(D|\theta)$ can still be evaluated pointwise, the posterior $p(\theta|D)$ is often only available up to a normalising constant. This is because the denominator $p(D) = \int_\Theta p(D|\theta)p(\theta)d\theta$ in (5), also called the *evidence*, is often expensive if not impossible to compute. Furthermore, most quantities of interest take the form

$$J = \mathbb{E}_{\pi(\theta)}[h(\theta)] \tag{8}$$

$$= \int_{\theta\in\Theta} h(\theta)\pi(\theta)d\theta \tag{9}$$

where $\pi(\theta)$ is usually the posterior in Bayesian inference. For example: $h(\theta) = \theta$ for posterior mean; $h(\theta) = I(a \le \theta \le b)$ for credible interval and posterior probability where $I(\cdot)$ is the indicator function; and $h(\theta) = p(\tilde{y}|\theta)$ for the posterior predictive distribution. Such integrals are often analytically intractable and numerical integration using quadrature rules quickly becomes infeasible as the computational cost scales exponentially with the dimension of θ. Instead, most Bayesian computational methods are usually concerned with constructing a Monte Carlo estimator $\hat{J}_n = \frac{1}{n}\sum_{i=1}^n h(\theta_i)$ where $\theta_i \overset{\text{iid}}{\sim} \pi(\theta)$, which converges to J in probability as

[1] For multivariate θ, this would be a credible region. However, the more common and simpler practice is to estimate the marginal credible interval for each univariate component.

$n \to \infty$ by the (weak) law of large number (LLN). More importantly, such convergence is not explicitly dependent on the dimension of θ so the Monte Carlo method usually scales better. Hence, it turns a question of how to integrate into one of how to sample from a target distribution $\pi(x)$.

This motivates *Markov chain Monte Carlo* (MCMC), which comprises a broad class of algorithms to sample from an arbitrary distribution, with only a fairly non-restrictive assumption of being able to evaluate an unnormalised target density. Recall that this is particularly attractive for Bayesian inference because the posterior $p(\theta|D)$ given by (5) is often only known up to a normalising constant. The general idea of MCMC is to implicitly construct a Markov chain with $\pi(x)$ as its limiting distribution so one simply iterates the Markov chain forward to eventually obtain samples from $\pi(x)$. By definition the samples will be correlated but the Markov chain LLN and central limit theorem still provide theoretical guarantees on the validity of using them for $\hat{J}_n$.

The literature on Markov chain theory and state-of-the-art MCMC algorithm is vast and far beyond the scope of this paper. We will only describe the two simplest and foundational MCMC algorithms namely the *Metropolis–Hastings algorithm* and the *Gibbs sampler* due to their relevance to this research. Readers are referred to more specialised sources such as [22–24] for a more thorough review on MCMC. The Metropolis–Hastings (MH) algorithm [25,26] given in algorithm 2 is used when one cannot sample directly from $\pi(\theta)$ but can still evaluate its unnormalised density pointwise. A separate proposal distribution $q(\theta^*|\theta^{[j]})$ that depends on the current state of the chain $\theta^{[j]}$ is used to generate a candidate θ^*, which is accepted with probability $\alpha(\theta^*, \theta^{[j]})$. The Gibbs sampler [27] given in algorithm 3 is used when θ is multivariate and it is easier to sample from the full conditional than the joint distribution.[2] For a p-dimensional θ, the full conditional for the ith component θ_i is $\pi(\theta_i|\theta_{-i})$ where $\theta_{-i} = (\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_p)$. In essence, Gibbs sampling decomposes the difficult problem of sampling from the joint distribution into easier sub-components of sampling from lower dimensional conditional distributions. During a Gibbs cycle, a MH step can be applied to sample from a full conditional if it cannot be sampled directly.

Algorithm 2 Metropolis–Hastings Algorithm

Input: (unnormalised) target density $\pi(\theta)$, a proposal distribution $q(\theta^*|\theta^{[j]})$, number of proposals N
Output: samples from $\pi(\theta)$
Initialise $\theta^{[0]}$ randomly in the support of $\pi(\theta)$
for $j = 0$ to $N - 1$ **do**
    Draw a candidate $\theta^* \sim q(\theta^*|\theta^{[j]})$
    Calculate $\alpha(\theta^*, \theta^{[j]}) = \min(1, \frac{\pi(\theta^*)q(\theta^{[j]}|\theta^*)}{\pi(\theta^{[j]})q(\theta^*|\theta^{[j]})})$
    Draw $u \sim \text{Unif}(0, 1)$
    if $u < \alpha(\theta^*, \theta^{[j]})$ **do**
        $\theta^{[j+1]} = \theta^*$
    else do
        $\theta^{[j+1]} = \theta^{[j]}$

Algorithm 3 Gibbs sampling

Input: target distribution $\pi(\theta)$ where $\theta = (\theta_1, \theta_2, ..., \theta_p)^T$ is a p-dimensional parameter vector, the full conditionals $\pi(\theta_i|\theta_{-i})$, number of iterations N
Output: samples from $\pi(\theta)$
Initialise $\theta^{[0]}$ randomly in the support of $\pi(\theta)$
for $j = 0$ to $N - 1$ **do**
    Sample $\theta_1^{[j+1]} \sim \pi(\cdot|\theta_2^{[j]}, \theta_3^{[j]}, ..., \theta_{p-1}^{[j]}, \theta_p^{[j]})$
    Sample $\theta_2^{[j+1]} \sim \pi(\cdot|\theta_1^{[j+1]}, \theta_3^{[j]}, ..., \theta_{p-1}^{[j]}, \theta_p^{[j]})$
    ...
    Sample $\theta_{p-1}^{[j+1]} \sim \pi(\cdot|\theta_1^{[j+1]}, \theta_2^{[j+1]}, ..., \theta_{p-2}^{[j+1]}, \theta_p^{[j]})$
    Sample $\theta_p^{[j+1]} \sim \pi(\cdot|\theta_1^{[j+1]}, \theta_2^{[j+1]}, ..., \theta_{p-2}^{[j+1]}, \theta_{p-1}^{[j+1]})$

3.2.2. MCMC diagnostics

Although Markov chain theory guarantees that the constructed chain approaches $\pi(x)$ asymptotically, in practice one can only perform a finite number of iterations. Hence several diagnostic tools have been proposed to assess the (lack of) convergence of the chain to its limiting distribution. Some common diagnostic tools are:

- Visual checks
  This includes: (1) trace plot which simply plots the state of the chain $\theta^{[j]}$ against the iteration number and should show no obvious pattern, (2) running mean plot which plots the cumulative mean $\frac{1}{j}\sum_{i<j}\theta^{[i]}$ against iteration j and should stabilise as

[2] θ need not be decomposed into univariate components but rather can be blocked in whatever ways. Furthermore, Gibbs sampling need not cycles through every component in fixed order, and can even do so probabilistically which is referred to as *random scan Gibbs sampling*.

a consequence of LLN, and (3) auto-correlation function (acf) which plots the auto-correlation of the chain at each lag and should ideally drop to zero relatively quickly.

- Gelman–Rubin $\hat{R}$ [28]
  The $\hat{R}$ statistic, also called the potential scale reduction factor, is largely similar to the analysis of variance. A number of chains with different initialisation are run in parallel and some burn-in period is removed. The within-chain variance W and the combined chain variance V are computed and $\hat{R} = \sqrt{V/W}$. The idea is that the chains are initialised over-dispersed relative to the posterior so $W < V$ and $\hat{R} > 1$. However once all chains have converged to the limiting distribution, W is comparable to V so $\hat{R}$ should be close to 1.
- Effective sample size n_{eff}
  The correlated samples produced in MCMC effectively incurs a loss of information. The effective sample size can be interpreted as the equivalent number of i.i.d. samples that hold the same amount of information, and is formally defined as

$$n_{\text{eff}} := \frac{N}{1 + 2\sum_{k=1}^{\infty} \rho_k} \tag{10}$$

  where N is the number of MCMC samples and ρ_k is the auto-correlation at lag k.
- Monte Carlo standard error (MCSE)
  Since the posterior mean $\mathbb{E}[\theta|D] = \int_{\Theta} \theta p(\theta|D) d\theta$ is estimated by the empirical mean of MCMC samples, the MCSE (of the posterior mean) is a measure of accuracy defined as the standard deviation of such Monte Carlo estimator. Markov chain central limit theorem implies MCSE $= \sigma/\sqrt{n_{\text{eff}}}$ where σ is the standard deviation of the posterior. The method of batch means is commonly used to estimate the MCSE.

The exact implementation details for the various diagnostic statistics are given in Appendix B. However it should be noted that all diagnostic statistics provide only necessary but not sufficient conditions for convergence.

4. Methodology

4.1. Synthetic data and statistical model

Let $F(x, \gamma)$ denote the solution to Gipps' car following model which predicts the entire follower's displacement time series, where x is the predictor that consists of the leader's trajectory and all other arguments to $F(\cdot)$ aside from γ, and $\gamma = (a_{\max}, V_{\max}, b_{\max}, x_f(0), v_f(0))$ is the calibration parameters. The prediction denoted by y is the observed follower's displacement time series. The parameters to be estimated are $\theta = (a_{\max}, V_{\max}, b_{\max}, x_f(0), v_f(0), \mu_x, \sigma_x^2)$.

Synthetic data is generated by adding i.i.d. Gaussian noise to the follower's displacement time series predicted by Gipps' model. Fig. 3 illustrates the training data and an instance of validation data respectively, both generated with the same parameter values which serve as ground truth, but with different leader and hence different predictor. Both consist of $T = 100$ s data, with $\tau = 0.5$ s; $l_c = 7.5$ m; $\psi = 1.05$; and integration time step Δt = observation time step = 0.1 s. The true values of the parameters to be estimated are listed in Section 5.1.

The full statistical model including the priors is given below:

$$a_{\max} \sim \text{Unif}(a_L, a_U) \tag{11}$$
$$b_{\max} \sim \text{Unif}(b_L, b_U) \tag{12}$$
$$V_{\max} \sim \text{Unif}(V_L, V_U) \tag{13}$$
$$x_f(0) \sim \text{Unif}(x_L, x_U) \tag{14}$$
$$v_f(0) \sim \text{Unif}(v_L, v_U) \tag{15}$$
$$\mu_x \sim \mathcal{N}(\mu_0, \sigma_0^2) \tag{16}$$
$$\sigma_x^2 \sim \text{Inv-Gamma}(\alpha_0, \beta_0) \tag{17}$$
$$x_f|x, \gamma = F(x, \gamma) \tag{18}$$
$$\epsilon \sim \mathcal{N}(\mu_x \mathbf{1}_{K+1}, \sigma_x^2 \mathbf{I}_{K+1}) \tag{19}$$
$$y = x_f + \epsilon \tag{20}$$

where $x_f = (x_f(0), x_f(\Delta t), x_f(2\Delta t), \ldots, x_f(K\Delta t))^T$ is the follower displacement time series observed at interval Δt with $T = K\Delta t$, $\mathbf{1}_{K+1}$ is the all ones vector of dimension $K+1$, and $\mathbf{I}_{K+1}$ is the identity matrix of size $K+1$. All calibration parameters are in SI units and given a weakly informative uniform prior over their physically plausible range of values. A diffused normal prior and inverse-gamma prior are used for μ_x and σ_x^2 respectively because they are the *conjugate* prior for the likelihood so direct sampling is possible, see Section 4.2. For the statistical model concerned, the implemented sampler does not depend on the mathematical form of the car following model so it is also applicable to other deterministic car following models. Appendix C demonstrates an application to the intelligent driver model.

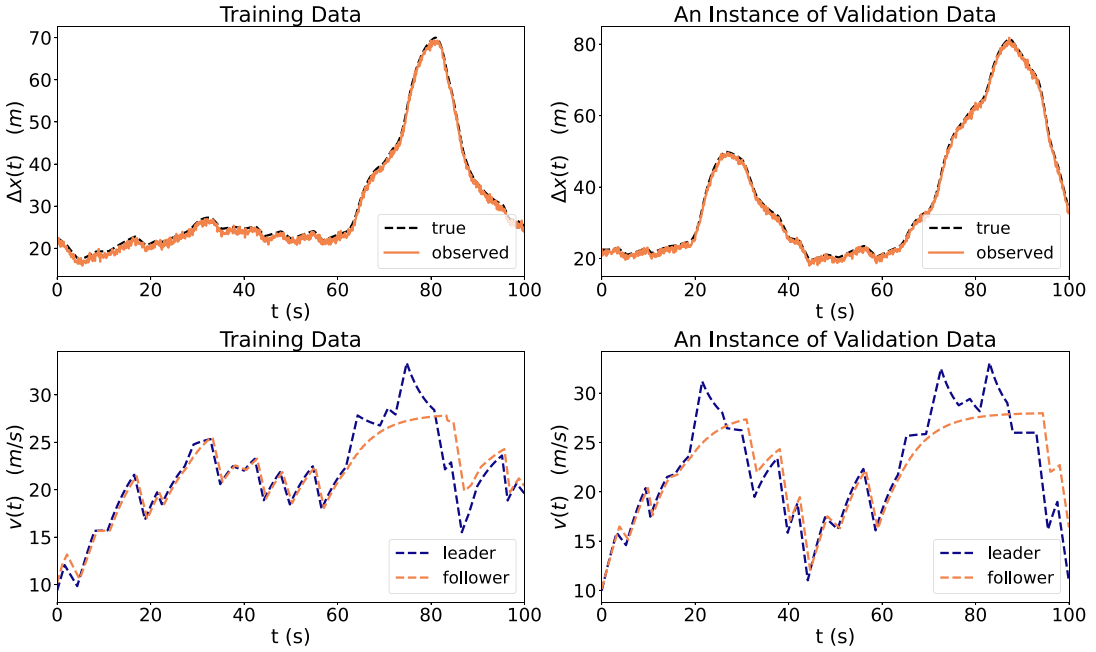

Fig. 3. Synthetic data generated. The dashed and solid lines in the top panels are the unobserved relative displacement and its observed noisy counterpart respectively. The bottom panels show the latent follower's velocity when responding to its leader.

Let $D = (x, y)$ denote the data. The likelihood for the model is hence

$$p(D|\theta) = \frac{\exp[-\frac{1}{2}(y-(x_f+\mu_x\mathbf{1}_{K+1}))^T(\sigma_x^2\mathbf{I}_{K+1})^{-1}(y-(x_f+\mu_x\mathbf{1}_{K+1}))]}{\sqrt{(2\pi)^{K+1}|\sigma_x^2\mathbf{I}_{K+1}|}} \tag{21}$$

where the dependence on γ is through x_f. Since all computations are performed on the log scale for numerical stability, the log-likelihood (up to some additive constant C) can be simplified to

$$\log p(D|\theta) = -\frac{1}{2\sigma_x^2}\|y - x_f - \mu_x\mathbf{1}_{K+1}\|_2^2 - \frac{K+1}{2}\log(\sigma_x^2) + C \tag{22}$$

where $\|\cdot\|_2$ is the ℓ^2 norm.

We will compare the predictive performance between the two common point estimators namely

- Maximum likelihood estimator (MLE)

$$\hat{\theta}_{\text{MLE}} = \underset{\theta}{\text{argmax}} \quad \log p(D|\theta) \tag{23}$$

- Maximum a posterior estimator (MAP)

$$\hat{\theta}_{\text{MAP}} = \underset{\theta}{\text{argmax}} \quad \log p(D|\theta) + \log p(\theta) \tag{24}$$

with the full Bayesian approach. It is easy to see that both the MLE and MAP are specific cases of an optimisation-based approach where the error metrics are the negative log-likelihood and negative log-posterior respectively.

4.2. Sampler description

The posterior is only known up to a normalising constant so MCMC algorithms are used to sample from it. The specific implementation of the sampler is detailed below. In each Gibbs cycle:

1. Sample the calibration parameter γ from its full conditional. Since γ enters the statistical model through the solution of a delay-differential equation, its full conditional will not be a standard distribution so a MH step is used. In particular, a single component MH step is performed for each component $\gamma_i \in \{a_{\max}, V_{\max}, b_{\max}, x_f(0), v_f(0)\}$ in turn. The logarithm of the target density $\pi(\cdot)$ is hence the unnormalised full conditional which is simply given by

$$\log p(\gamma_i|D, \gamma_{-i}, \mu_x, \sigma_x^2) \propto -\frac{1}{2\sigma_x^2}\|y - x_f - \mu_x\mathbf{1}_{K+1}\|_2^2 + \log p(\gamma_i). \tag{25}$$

2. Sample the systematic noise component μ_x directly from its full conditional. By choice of normal conjugate prior, the full conditional posterior will also be a normal distribution

$$\mu_x|D,\gamma,\sigma_x^2 \sim \mathcal{N}(\mu_p,\sigma_p^2) \tag{26}$$

where $\mu_p = \sigma_p^2\left(\frac{\mu_0}{\sigma_0^2} + \frac{(y-x_f)^T\mathbf{1}_{K+1}}{\sigma_x^2}\right)$ and $\sigma_p^2 = \left(\frac{1}{\sigma_0^2} + \frac{K+1}{\sigma_x^2}\right)^{-1}$.

3. Sample the random noise component σ_x^2 directly from its full conditional. By choice of inverse gamma conjugate prior, the full conditional posterior will also be an inverse gamma distribution

$$\sigma_x^2|D,\gamma,\mu_x \sim \text{Inv-Gamma}(\alpha_p,\beta_p) \tag{27}$$

where $\alpha_p = \alpha_0 + \frac{K+1}{2}$ and $\beta_p = \beta_0 + \frac{1}{2}\|y - x_f - \mu_x\mathbf{1}_{K+1}\|_2^2$.

Since the support of the uniform prior for γ is bounded, its posterior will also have bounded support. This is perhaps a reasonable approximation for parameters with physical interpretations as in the case for the car following model. Although this does not raise issue on the correctness of the asymptotic properties of the chain, it concerns the efficiency of the algorithm. In other words, the chain will still converge to the correct stationary distribution as long as it is initialised in the support of the posterior, but if there is significant posterior mass near the boundary, many MH proposals will step outside the support and hence be rejected. We reparametrised each component of γ with a logit transform and perform random walk in an unbounded space. Recall $\text{logit}(x) = \log\frac{x}{1-x}$, which is a monotonic transformation and hence its inverse exists. Let $\gamma_i \in (L_i, U_i)$ denote the original bounded variable, we perform the change of variable $u = g(\gamma_i) = \text{logit}\frac{\gamma_i - L_i}{U_i - L_i} \in \mathbb{R}$. Let $|J(u')|$ denote the absolute value of the determinant of the Jacobian of $g^{-1}(u)$ evaluated at u' and $q(u^*|u^{[j]})$ denote the random walk proposal in the u-space. The slight change to the MH step is that the acceptance probability

$$\alpha(u^*,u^{[j]}) = \min\left(1, \frac{\pi(g^{-1}(u^*))q(u^{[j]}|u^*)|J(u^*)|}{\pi(g^{-1}(u^{[j]}))q(u^*|u^{[j]})|J(u^{[j]})|}\right) \tag{28}$$

now includes the Jacobian term, and is computationally equivalent to using an asymmetric proposal in the γ_i-space.

The more important step in improving the efficiency of the sampler is to choose appropriate hyperparameter values in the proposal distribution. For a random walk proposal $q_\lambda(u^*|u^{[j]}) = \mathcal{N}(u^{[j]},\lambda^2)$, the hyperparameter λ controls the proposal step size. If the step size is too small, the sampler frequently accepts but crawls around the posterior; if the step size is too large, the sampler rejects frequently and stays at the same state for long period. In both cases the sampler will explore the posterior slowly. Although the theory guarantees all resulting chains converge asymptotically to the correct stationary distribution, the rate of convergence can differ drastically. Manual trial and error searching for appropriate hyperparameter value proves difficult and time consuming, which calls for the need of an adaptive MCMC algorithm that automatically tunes the proposal distribution. We briefly state the subtle issue of adaptive MCMC as a cautionary reminder.

The adaptation implies the chain no longer satisfies the Markov property because the distribution $p(\theta^{[j+1]}|\theta^{[j]},\theta^{[j-1]},\ldots,\theta^{[0]})$ now depends on the entire history as opposed to just $\theta^{[j]}$, which means it can converge to the wrong stationary distribution. More formally, let Θ denote the state space with σ-algebra $\mathcal{F}$ and q_γ denote a proposal distribution with hyperparameters indexed by γ. It is well-known in Markov chain theory that if π is the stationary distribution of q_γ and the chain is irreducible and aperiodic, then the Markov chain generated by iterating q_γ will be *ergodic* to π, namely

$$\lim_{n\to\infty}\|q_\gamma^n(\theta,A) - \pi(A)\| = 0 \quad \forall\theta\in\Theta,\ A\in\mathcal{F} \tag{29}$$

where $\|\cdot\|$ is the total variation distance. Adaptive MCMC replaces γ with a random variable Γ_n. Suppose $\mathcal{Y}$ is a family of adaptation index where all $\gamma\in\mathcal{Y}$ induces π-ergodic Markov chains. Even if $\Gamma_n\in\mathcal{Y}$ $\forall n$, this does not guarantee q_{Γ_n} will be ergodic to π.

We refer readers to more specialised text such as [29,30] for sufficient conditions on the ergodicity of adaptive MCMC. Our implementation as shown in algorithm 4 utilises parallel chains similar to [31], and is based on finite adaptation which is the simplest yet safest adaptation scheme. The hyperparameters are updated only during an initial adaptation phase and are fixed in the subsequent sampling phase. In particular, λ is updated by monitoring the difference between the empirical acceptance rate $\hat{\alpha}$ and the optimal acceptance rate $\alpha^* \approx 0.44$, where the latter was derived under certain assumption [32] and served as a rule of thumb.

5. Results and discussions

5.1. Inference results

Without loss of generality, we set $x_\ell(0) = 0$. The parameters for the prior are chosen as: $a_L = 0.5, a_U = 3.5, V_L = 15, V_U = 35, b_L = -6.0, b_U = -1.0, x_L = -60, x_U = -10, v_L = 5, v_U = 25, \mu_0 = 0, \sigma_0^2 = 9, \alpha_0 = 1$ and $\beta_0 = 3$, which are weakly informative. The hyperparameters of the adaptive sampler are $M = 4, B = 50, R = 20$ and $\alpha^* = 0.4$ and determined by trial and error. Fig. 4 illustrates the inference result after running MCMC for the statistical model with only the first 30 s of training data presented in

Algorithm 4 Finite Interchain Batch Adaptation

1: **Input**: (unnormalised) target density $\pi(\theta)$, random walk proposal $\mathcal{N}(\theta^*|\theta^{[j]}, (\lambda^{[r]})^2)$, number of parallel chains M, batch size B, adaptation rounds R, initial hyperparameters $\lambda^{[0]}$, optimal acceptance rate α^*
2: **Output**: suitable hyperparameters value $\lambda^{[R]}$
3: Initialise $\theta^{[0]}$ randomly in the support of $\pi(\theta)$ for each of the M chain
4: **for** $r = 0$ to $R - 1$ **do**
5: Perform B iterations from M chains with proposal $q_{\lambda^{[r]}}(\cdot)$ separately
6: Update the hyperparameters with

$$\log(\lambda^{[r+1]}) = \log(\lambda^{[r]}) + \frac{1}{\log(r+1)}\text{sign}(\hat{\alpha} - \alpha^*)$$

where

$$\hat{\alpha} = \frac{\sum_{m=1}^{M}\sum_{j=Br+1}^{B(r+1)} I(\theta^{[j],m} \text{ is accepted})}{MB}$$

Table 1
Estimates and MCMC diagnostic statistics for 30 s synthetic data.

	True	MLE	MAP	$\mathbb{E}[\cdot\|D]$	MCSE	$q_{0.025}$	$q_{0.5}$	$q_{0.975}$	$\hat{R}$	n_{eff}
$a_{\max}$	1.797	3.106	3.120	2.522	0.0149	1.488	2.548	3.445	1.0023	654.3
$V_{\max}$	28.000	25.262	25.155	26.453	0.0429	24.516	26.000	30.801	1.0038	597.7
$b_{\max}$	−3.566	−3.548	−3.551	−3.550	0.0003	−3.575	−3.550	−3.525	1.0052	873.6
$x_f(0)$	−22.521	−22.570	−22.528	−22.756	0.0090	−23.492	−22.744	−22.098	1.0017	634.1
$v_f(0)$	10.148	9.170	9.191	9.850	0.0210	8.746	9.746	11.815	1.0004	620.3
μ_x	0.800	1.062	1.028	1.038	0.0037	0.740	1.039	1.335	1.0050	899.7
σ_x^2	0.250	0.257	0.275	0.281	0.0001	0.239	0.280	0.331	1.0000	55 591.4

Section 4.1. Four chains are run in parallel marked by the four different colours, for a total of 50 000 iterations. Throughout this paper, we adopted the rule of thumb of discarding the first half as burn-in.

The first column in Fig. 4 shows the marginal prior and posterior density for each parameter. The prior density is easily available and a kernel density estimate is performed for the posterior using the MCMC samples. Compared to the diffuse prior, the posterior generally concentrates onto a smaller region and the true parameter values marked by the black vertical lines are contained in the bulk of the posterior mass with the notable exception of $a_{\max}$ and $V_{\max}$. The posterior for them is still rather diffused, this is because the 30 s trajectory data does not contain enough information to reliably estimate them. Furthermore, $b_{\max}$ and the noise parameters are somewhat overestimated. The next three columns in Fig. 4 illustrate the trace plot, running mean plot and the auto-correlation function respectively. The chains appear to be mixing reasonably well as there is no obvious pattern or strong sign of stickiness in the trace plot. The running mean all converged and although there does exist noticeable auto-correlation for most of the parameters, it drops off sufficiently fast and does not raise any major concern.

Let $\bar{\theta}_{i,m}^{[j]}$ denote the running mean for the ith component of θ at jth iteration for chain m, with across chain mean $\bar{\theta}_{i,\cdot}^{[j]} = \frac{1}{m}\sum_m \bar{\theta}_{i,m}^{[j]}$ and standard deviation $s_i^{[j]} = \sqrt{\frac{1}{m}\sum_m(\bar{\theta}_{i,m}^{[j]} - \bar{\theta}_{i,\cdot}^{[j]})^2}$ estimated with the usual sample statistics. The coefficient of variation $s_i^{[j]}/|\bar{\theta}_{i,\cdot}^{[j]}|$ for all components stabilised below 0.05 at iteration ≈26 989. The sampling algorithm took ∼1.5 h on an Intel Xeon processor 3.10 GHz running Windows Server 2019, with multiprocessing implemented to sample the four chains in parallel.

Convergence of running mean however, is guaranteed by the weak law of large number and the more important assessment is the convergence of the Markov chain to the posterior. Table 1 lists the various point estimates such as the MLE, MAP and the posterior mean $\mathbb{E}[\cdot|D]$, and MCMC diagnostic statistics to formally investigate any potential issues of mixing. The $\hat{R}$ statistic falls below 1.1 for all parameters, indicating that further iterating the chain is unlikely to improve the mixing of the chains. The effective sample size n_{eff} for all parameters is greater than 100, hence inference is deemed acceptable at least for the posterior mean. The effect of auto-correlation is clear as the effective sample size for some parameters is roughly two orders of magnitude lower than the number of proposals. The MLE and MAP in this case are noticeably different for $a_{\max}$ and $V_{\max}$ due to the insufficiently informative data reason mentioned above.

The benefit of the computationally more intensive Bayesian approach lies in its ability to properly quantify uncertainty. Table 1 also lists the posterior quantiles q_α such that $p(\theta_i \le q_\alpha|D) = \alpha$ for $\alpha = 0.025, 0.5, 0.975$. The true values are contained in the central 95% credible interval for all parameters. The credible intervals help to summarise the shape of the posterior and are convenient in reporting different level of uncertainty. Indeed, one advantage of a Bayesian approach is that the meaning of credible (as opposed to confidence) interval, agrees with common intuition. Namely there is probability α that the true parameter value falls in a $100\alpha\%$ credible interval.

We then performed inference with the full 100 s training data. The coefficient of variation stabilised around iteration 27 287 and the algorithm took ∼4 h, as expected because the car following model integration period is roughly three times longer. The MCMC results are shown in Fig. 5. With a longer trajectory data which is more informative, the posterior mass concentrates more

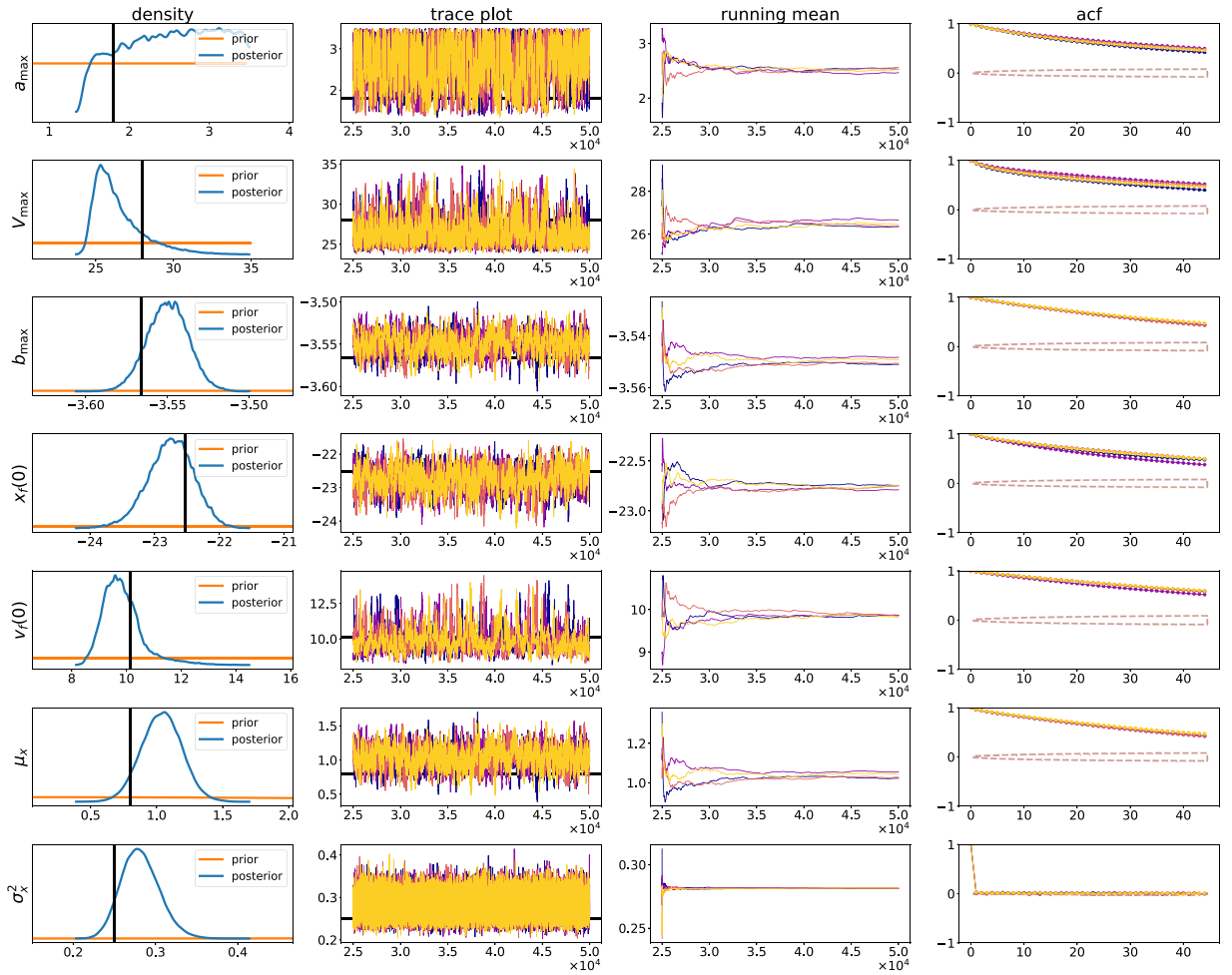

Fig. 4. MCMC results for 30 s synthetic data. Column 1 shows the marginal prior and posterior density for each parameter. The vertical black solid line in the column 1 and the horizontal black solid line in the column 2 mark the true parameter values used to generate the data. Column 2, 3 and 4 show the trace plot, running mean plot and the auto-correlation function respectively. Each colour corresponds to one of the four chains which were run in parallel. The trace plot and the running mean plot are plotted against the sampler iterations after burn in. The auto-correlation function is plotted against different lag.

Table 2
Estimates and MCMC diagnostic statistics for 100 s synthetic data.

	True	MLE	MAP	$\mathbb{E}[\cdot\|D]$	MCSE	$q_{0.025}$	$q_{0.5}$	$q_{0.975}$	$\hat{R}$	n_{eff}
a_{max}	1.797	1.818	1.816	1.817	0.0004	1.792	1.817	1.842	1.0018	421.6
V_{max}	28.000	27.967	27.968	27.968	0.0006	27.931	27.968	28.005	1.0018	437.5
b_{max}	−3.566	−3.557	−3.557	−3.559	0.0002	−3.569	−3.559	−3.548	1.0001	416.5
$x_f(0)$	−22.521	−22.685	−22.628	−22.780	0.0069	−23.398	−22.752	−22.273	1.0020	930.9
$v_f(0)$	10.148	10.109	10.041	10.359	0.0139	9.794	10.198	11.931	1.0026	810.2
μ_x	0.800	0.960	0.958	0.934	0.0028	0.759	0.933	1.112	1.0001	439.1
σ_x^2	0.250	0.240	0.242	0.246	0.0000	0.225	0.246	0.269	1.0001	87 554.0

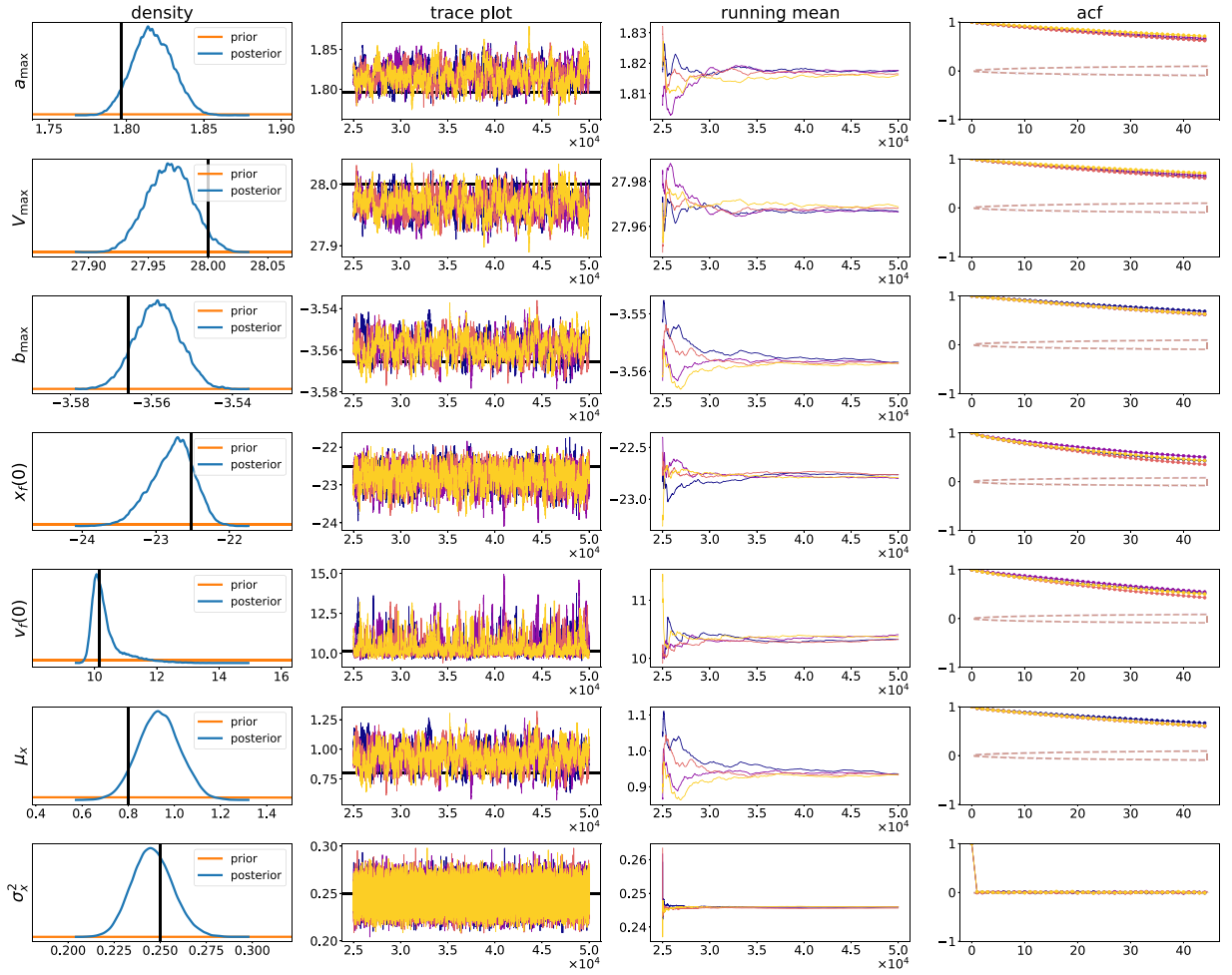

Fig. 5. MCMC results for the full 100 s synthetic data.

and the majority of the posterior mass is close to the true parameter values. Table 2 shows the estimates, the diagnostic statistics and the posterior quantiles. The MLE and MAP returned very similar results compared to the posterior mean, and all are quite close to the true parameter values. In this case, the data is sufficient and quite informative about the parameters. There is noticeably less uncertainty about the parameter values so using point estimators in this case is perhaps sufficient.

5.2. Predictive uncertainty

We then investigate the effect of propagating the uncertainty in the parameter estimates to predictions, in particular we consider the following predictive distributions:

$$\text{MLE}: \int_{\Theta} p(\tilde{y}|\tilde{x},\theta)\delta_{\hat{\theta}_{\text{MLE}}} d\theta \tag{30}$$

$$\text{MAP}: \int_{\Theta} p(\tilde{y}|\tilde{x},\theta)\delta_{\hat{\theta}_{\text{MAP}}} d\theta \tag{31}$$

$$\text{posterior predictive}: \int_{\Theta} p(\tilde{y}|\tilde{x},\theta)p(\theta|D)d\theta \tag{32}$$

where MLE and MAP are the common point estimators described in Section 4.1. The notation $\delta_{\hat{\theta}}$ denotes an atom with a probability mass of one at $\hat{\theta}$.

The full Bayesian approach to making prediction integrates over the posterior distribution of the unknown parameter θ, resulting in the posterior predictive distribution. On the contrary, point estimators effectively put all posterior mass at a point when making prediction. Since the statistical model is relatively simple, the predictive distribution is available in closed form conditioning on point estimators and can be computed exactly. Let $\hat{\theta} = (\hat{\gamma}, \hat{\mu}_x, \hat{\sigma}_x^2)$ denote a parameters point estimate, then the central $100(1-\alpha)\%$

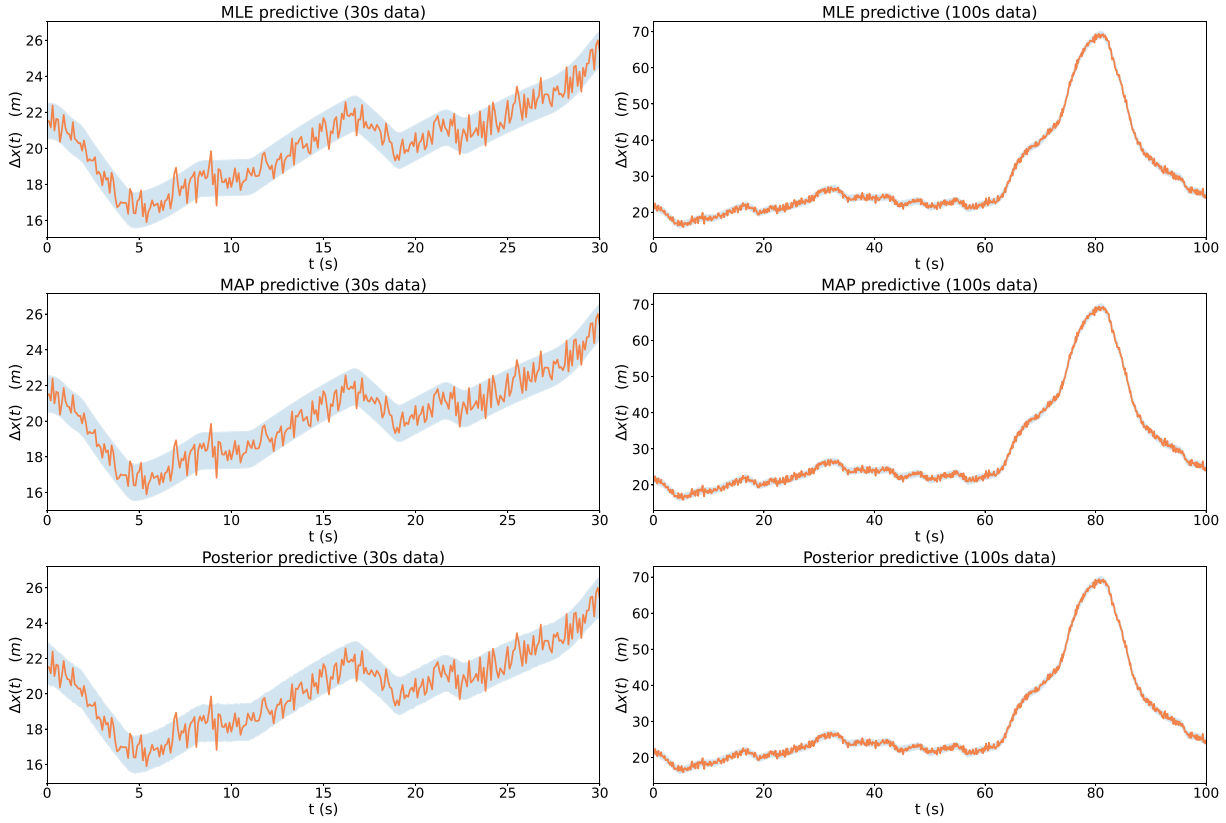

Fig. 6. The central 95% pointwise predictive intervals on training data, for MLE, MAP and posterior predictive, and for 30 s and 100 s training data.

pointwise predictive interval is given by $[F(\tilde{x},\hat{\gamma}) + \hat{\mu}_x + z_{\alpha/2}\hat{\sigma}_x, F(\tilde{x},\hat{\gamma}) + \hat{\mu}_x + z_{1-\alpha/2}\hat{\sigma}_x]$ where $z_{\alpha/2}$ is the $\alpha/2$ quantile of a standard normal. The posterior predictive distribution is estimated using 5000 random posterior samples.

Fig. 6 illustrates the central 95% pointwise predictive distributions on the 30 s and 100 s training data, for the MLE, MAP and posterior predictive. In both cases, the MLE, MAP and posterior predictive intervals are visually similar and contain the training data roughly 95% of the times. Since the data was used to fit the model, the close agreement between the predictive intervals and the observed data is unsurprising.

Fig. 7 illustrates the predictive distributions on an instance of validation data described in Section 4.1, based on parameters estimated from the training data. For the 100 s training data case, the estimated parameters were close to the true parameter values so the predictive distributions match the new data well. In this case of sufficient data, a full Bayesian approach and optimisation approaches return similar results, so it is acceptable to approximate the posterior by a point estimator. For the 30 s training data case, the data is not as informative so the point estimates were not as reliable. As a result the MLE and MAP predictive do not match the validation data. The predictive intervals are too narrow, which implies the model conditioned on point estimators is over-confident in its prediction. The posterior predictive incorporates the parameter uncertainty, and results in much wider predictive intervals which contain the validation data.

In order to assess the accuracy of the uncertainty quantification, we generated 100 instances of validation data and calculated the coverage. Let $\tilde{D}_i = (\tilde{x}_i, \tilde{y}_i)$ denote the ith validation data, then the coverage probability $\hat{p}$ is calculated as the relative frequency that the predictive intervals contain the validation data

$$\hat{p} = \frac{1}{K+1}\sum_{k=0}^{K} I\left(\tilde{y}_i(k\Delta t) \in 100(1-\alpha)\% \text{ predictive interval}\right) \tag{33}$$

where $K\Delta t = T$ is the duration of the validation data, for a given α. Although the Bayesian credible intervals will depend on the prior and generally do not need to have the correct frequentist coverage, we use this as a metric to assess how well the uncertainty is calibrated. The results for central 95% predictive intervals are shown in Fig. 8. In the case of the 30 s training data, the MLE and MAP noticeably underestimates the predictive uncertainty. The observed coverage is less than the nominal 95%, and varies noticeably over different validation data. The full Bayesian approach however, tends to overestimate the predictive uncertainty and the observed coverage is more than the nominal. In the case of 100 s training data, the posterior concentrates sufficiently to the true parameters so the observed coverage are all quite in line with the nominal coverage. The full Bayesian approach using

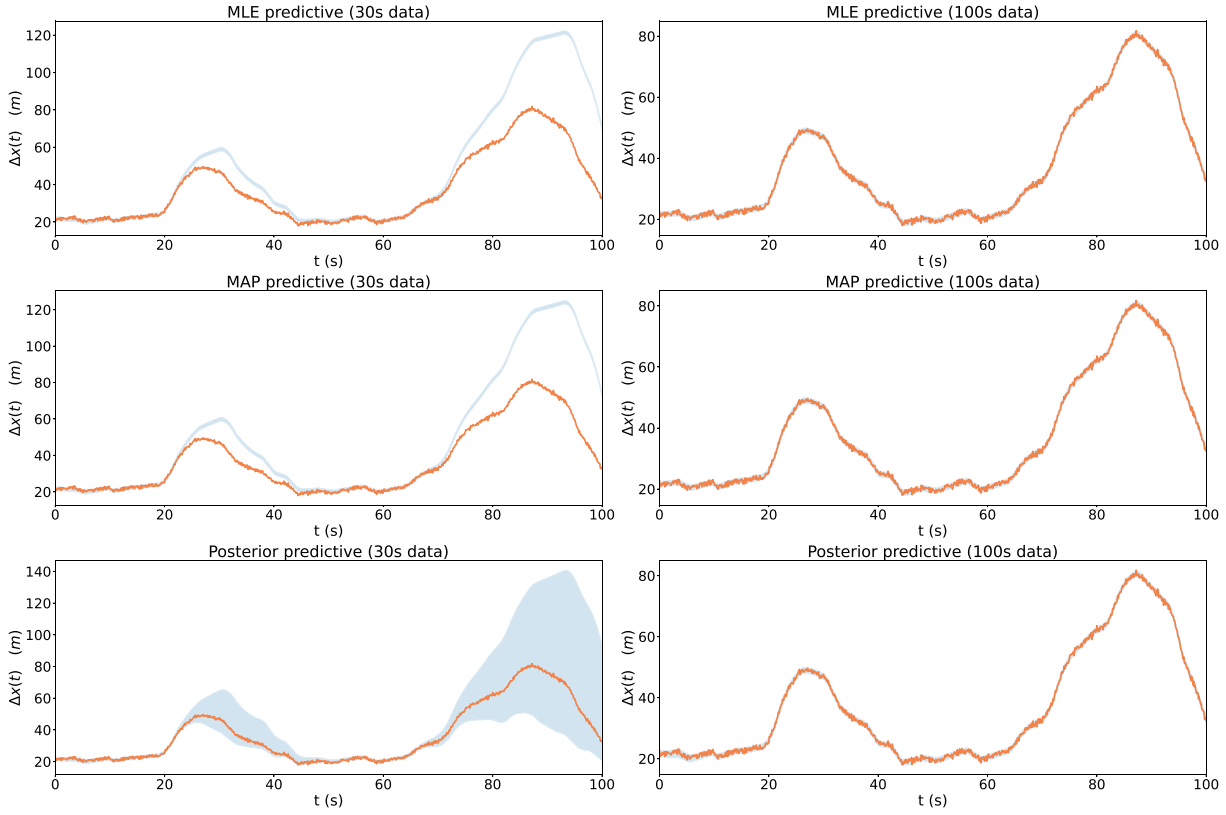

Fig. 7. The central 95% pointwise predictive intervals on validation data, for MLE, MAP and posterior predictive, and for 30 s and 100 s training data.

the posterior predictive distribution exhibits less variability and is more centred around the nominal coverage, indicating a more accurate uncertainty quantification.

5.3. Correlated parameters

We also attempted to perform estimation for other quantities in Gipps' car following model that could be considered parameters. For instance, one might be interested in estimating the sensitivity ψ that also governs the congested flow behaviour along with $b_{\max}$. For illustration purpose, the parameters of interest in this section is restricted to $\theta = (b_{\max}, \psi, \mu_x, \sigma_x^2)$ and all the remaining parameters are set to their true values.

The corresponding MCMC results are shown in Fig. 9. The implemented adaptive sampler does not perform as well in this case. Even with four times more iterations than the previous case, the chains for $b_{\max}$ and ψ appear sticky with clear patterns in the trace plot. The auto-correlation remains high even for a large lag, which confirms the stickiness of the chains. The coefficient of variation stabilised at iteration ≈100 016, and sampling took ∼10 h. We tested with different hyperparameter values of B, R and α^* for the sampler. We also attempted a blocked MH step where a multivariate normal distribution is used to generate proposal for the vector $(b_{\max}, \psi)$, and adaptation proceeds by estimating the covariance of the proposal with sample covariance. Neither of these significantly improve the performance of the sampler.

Similarly, Table 3 illustrates the point estimates and MCMC diagnostic statistics. The MLE and MAP are again quite similar to the posterior mean. The effect of high auto-correlation is clear since the effective sample size is now much lower compared to the number of proposals. Nevertheless, the diagnostics suggest the inference result is perhaps still somewhat acceptable when estimating the posterior mean with $n_{\text{eff}} > 100$. Indeed, the posterior means for all parameters are close to their true values.

We performed a grid search on the log-posterior to identify the causes of the inefficient sampler and the result is shown in Fig. 10. The left hand figures illustrate the log-posterior for $(a_{\max}, b_{\max})$ holding all the remaining parameters at their true values. The bottom sub figure is a zoomed in version of the black box in the top sub figure. Similarly, the right hand figures illustrate the log-posterior for $(b_{\max}, \psi)$. The posterior for $a_{\max}$ and $b_{\max}$ is uni-modal and resembles a multivariate normal distribution. There is only weak correlation between them, which is expected because the two parameters govern different aspects of the car-following model. On the other hand, the strong posterior correlation between $b_{\max}$ and ψ becomes evident, which one might expect apriori since they both govern the congested flow behaviour. Most posterior mass is located along a narrow ridge, so for a random walk sampler to maintain a sufficiently high acceptance rate, it is forced to take very small steps which results in the high auto-correlation.

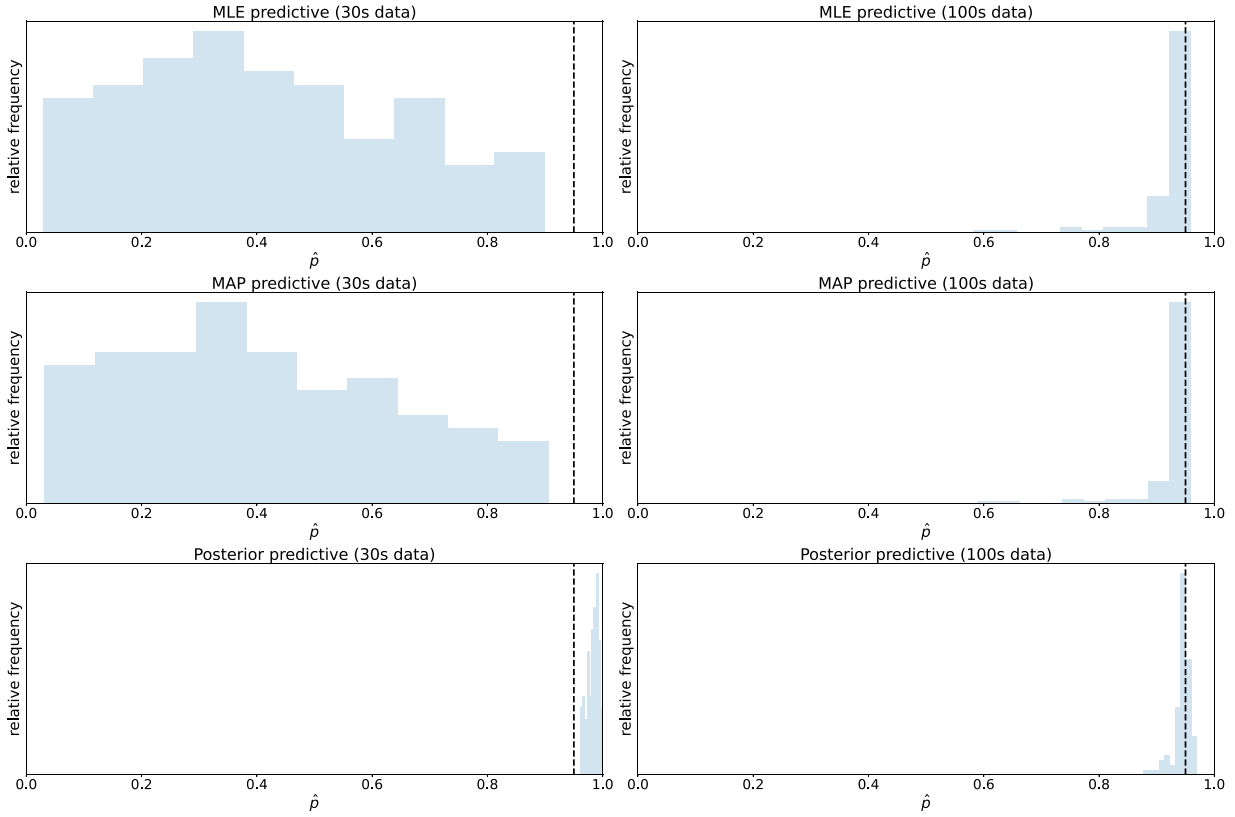

Fig. 8. Coverage probability of predictive distributions for 100 instances of validation data. The dashed line denotes the nominal 95% coverage.

Table 3
Estimates and MCMC diagnostic statistics for $\theta = (b_{\max}, \psi, \mu_x, \sigma_x^2)$.

	True	MLE	MAP	$\mathbb{E}[\cdot\|D]$	MCSE	$q_{0.025}$	$q_{0.5}$	$q_{0.975}$	$\hat{R}$	n_{eff}
$b_{\max}$	−3.566	−3.540	−3.539	−3.538	0.0005	−3.567	−3.538	−3.511	1.0026	151.3
ψ	1.051	1.044	1.044	1.044	0.0002	1.035	1.044	1.052	1.0026	152.1
μ_x	0.800	0.858	0.856	0.859	0.0004	0.760	0.859	0.957	1.0001	11 554.7
σ_x^2	0.250	0.239	0.244	0.245	0.0000	0.225	0.245	0.268	1.0000	384 324.7

Although the huge drop of computational efficiency is clear, the MCMC result is still acceptable for point estimator such as the posterior mean. However one should be more cautious when interpreting the full posterior distribution and the associated uncertainty. We believe such strong posterior correlation is common for parameters of a dynamic model such as the car following model, which poses notable challenges for the sampler to fully explore the posterior. More iterations can be performed if one wishes to increase the effective sample size and hence the accuracy of inference. However a more promising solution is to adopt more advanced samplers such as Hamiltonian Monte Carlo methods [33,34] which require the gradient of log-posterior w.r.t. the parameters. This usually means the car following model should be implemented in a framework such as JAX [35] that supports automatic differentiation.

More importantly, the narrow ridge in the bottom right sub figure in Fig. 10 will have an even more adverse impact on model calibration approaches based on optimisation. There is a region in the parameter space where the log-posterior is effectively the same given the finite amount of data, which implies potential identifiability issue as many combinations of parameters result in seemingly equally good fit. This means reporting only a point estimator as in the case for an optimisation-based approach is often inappropriate and potentially strongly biased. Since a Bayesian approach explores the full posterior distribution as opposed to just finding a local optimum, one can be more aware of such issues as alarmed by the diagnostic statistics.

6. Application to NGSIM dataset

In this section, we applied Bayesian inference to real data with the NGSIM dataset [36]. Inference for real data is subjected to model imperfection error, contrary to the case of synthetic data. In this case, the data generating process is most likely different from the statistical model so there is no ground truth θ^* to assess how well each method can recover the parameters. If the parameters

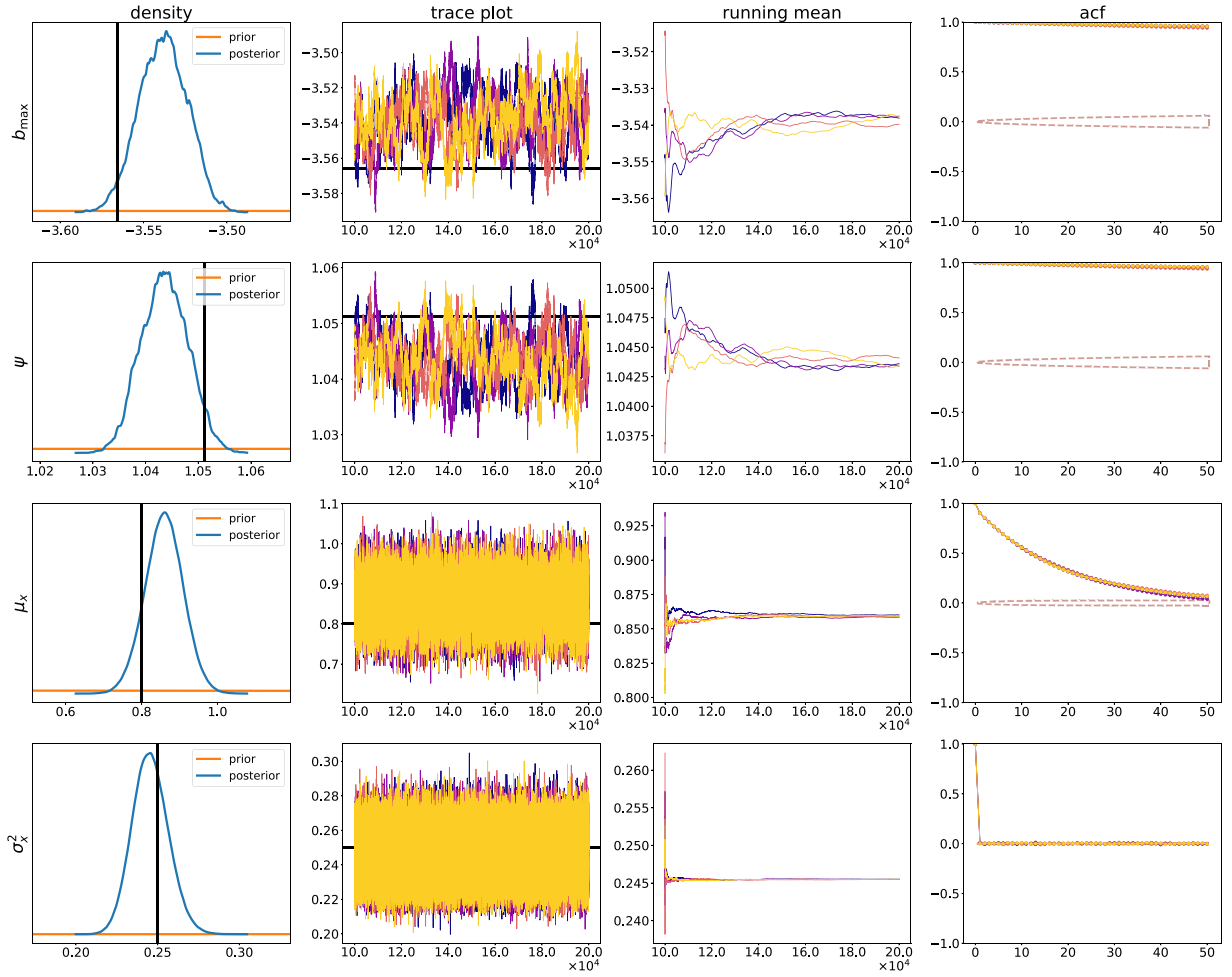

Fig. 9. MCMC results for $\theta = (b_{\max}, \psi, \mu_x, \sigma_x^2)$ using all 100 s training data.

estimates with real data are used to make further inference, the coherent uncertainty quantification enabled by a Bayesian approach compared to a point estimator is of particular relevance.

We focused on a specific car following episode in the NGSIM dataset at the I-80 freeway, where the leader and follower vehicle IDs are 1260 and 1267 respectively, as shown in Fig. 11. The car following episode lasted for 92.5 s and is collected at 0.1 s interval. The acceleration is noticeably more sporadic than the synthetic data. We adopted the same statistical model in Section 4.1 and the same sampler setup to estimate the parameters based on the first 50 s trajectory data. Since parameters such as the leader's maximum desired deceleration is also unknown, we assumed a sensible estimate of $\tau = 0.5$ s, $l_c = 7.5$ m and $\hat{b}_\ell = -3.4$ m/s^2 based on the parameters' physical meaning. We adopted the same weakly informative prior as that for inference with synthetic data in Section 5.1, with the exception that $V_L = 40$ instead. This is because the inference result exhibited significant posterior mass for $V_{\max}$ near the upper bound, which we suspected might be due to the prior support being too narrow. The upper bound is changed to $V_L = 40$ m/s to more accurately reflect the prior knowledge of possible values of $V_{\max}$. We believe the careful trajectory extraction process for NGSIM dataset means the initial conditions are relatively well known so we also fixed the initial conditions $(x_f(0), v_f(0))$ to their observed values.

The inference results are shown in Fig. 12 and Table 4. Both visual inspection and diagnostic statistics did not indicate any notable issues with mixing. There is still noticeable posterior mass for $V_{\max}$ close to the boundary of the support. Based on the physical meaning of $V_{\max}$ which we believe is apriori unlikely to exceed 40 m/s, this is more likely a potential sign of model mis-specification instead. Furthermore, the observational noise variance σ_x^2 is much higher than expected which also suggests the statistical model does not match the data particularly well so the unexplained variations are assigned to high observational noise. The MLE and MAP are numerically very similar. However since $V_{\max}$ is not reliably estimated, these point estimates can be misleading so inspecting the whole posterior distribution in this case might be more appropriate. A more thorough analysis can be performed to improve the statistical model to better match the data generating process, our aim was solely to demonstrate the applicability of the Bayesian approach to real data and highlight the subtle issues associated with model imperfections. Fig. 13 illustrates the posterior predictive

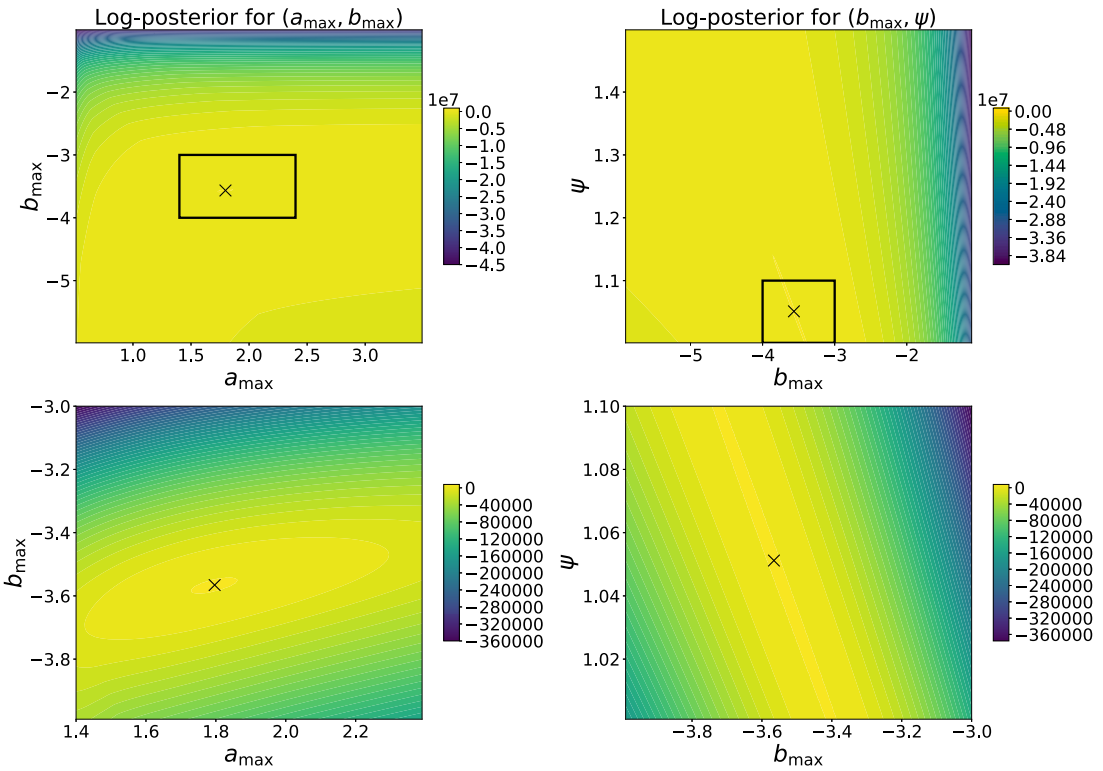

Fig. 10. Grid search on the log-posterior, with all the remaining parameters set to their true values. The cross marks the true values. The sub figures in the bottom row are the zoomed in version of the black box in the top row respectively.

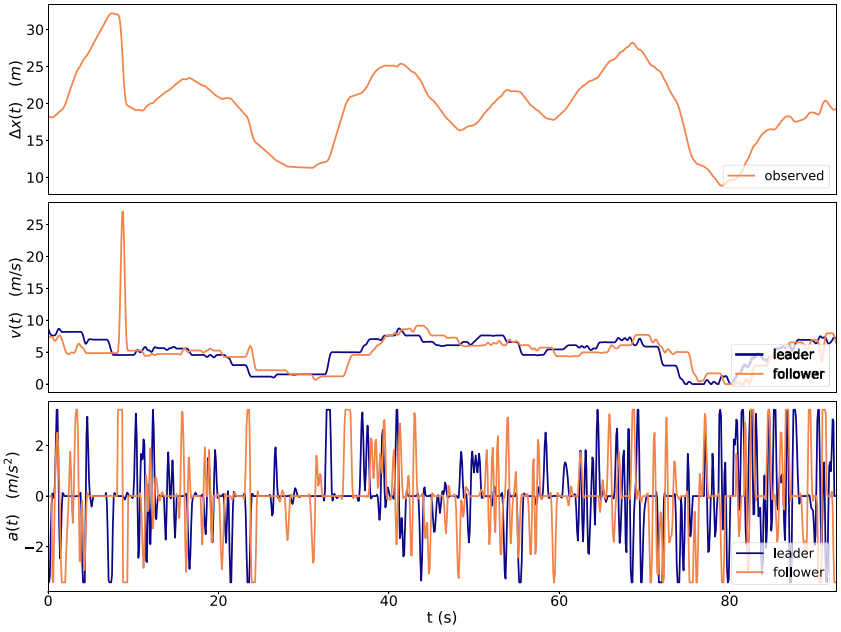

Fig. 11. A car following episode from the NGSIM dataset at the I-80 freeway, with leader ID = 1260 and follower ID = 1267.

distribution on the first 50 s of training data and the future data held out for validation. The model fit is satisfactory and appears to capture the predictive uncertainty on validation data reasonably well so is perhaps still somewhat useful for the purpose of prediction.

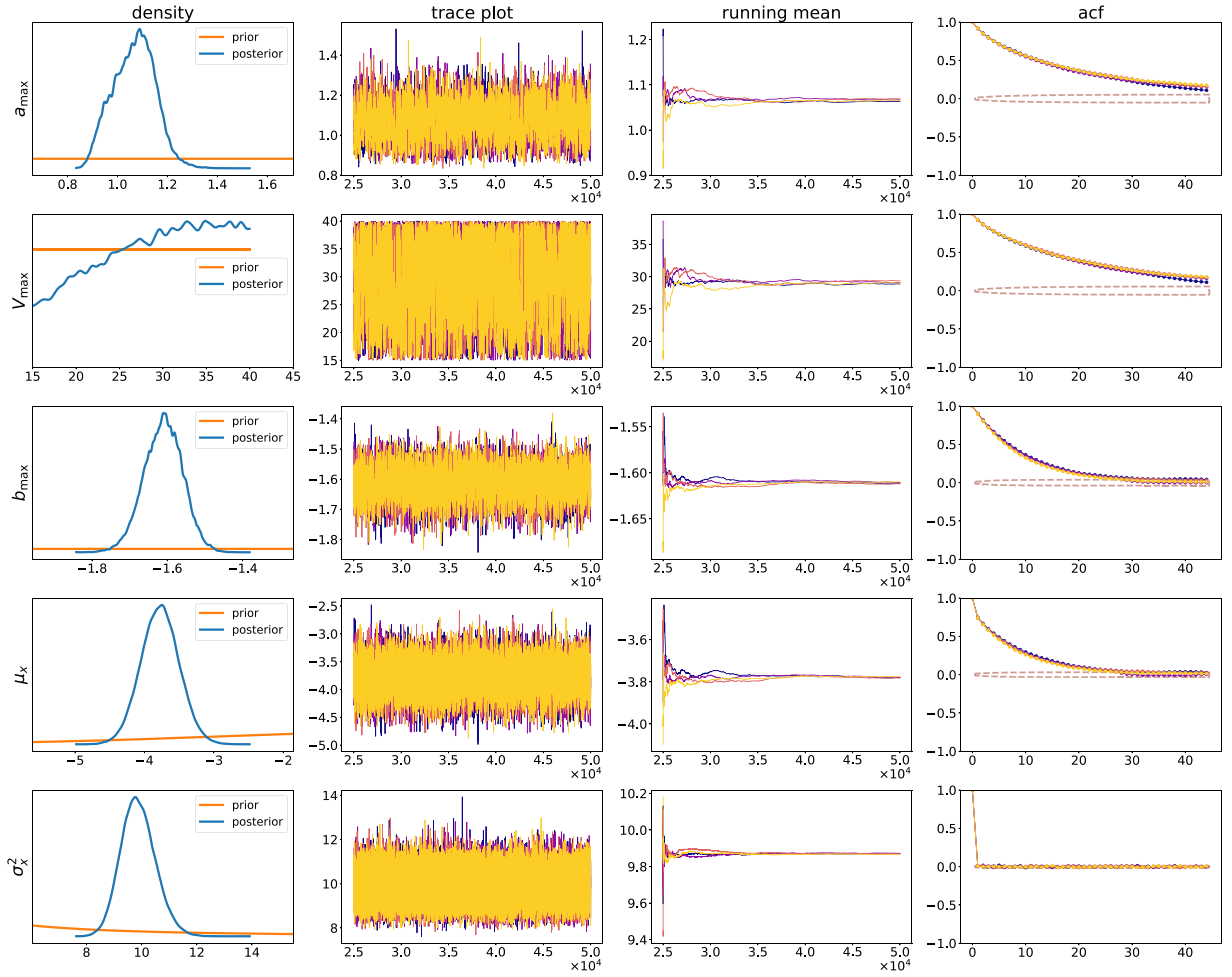

Fig. 12. MCMC results for NGSIM data.

Table 4
Estimates and MCMC diagnostic statistics for NGSIM data.

	MLE	MAP	$\mathbb{E}[\cdot\|D]$	MCSE	$q_{0.025}$	$q_{0.5}$	$q_{0.975}$	$\hat{R}$	n_{eff}
$a_{\max}$	1.144	1.144	1.066	0.0014	0.909	1.070	1.224	1.0005	1769.2
$V_{\max}$	38.317	38.317	29.095	0.1178	16.205	29.678	39.471	1.0005	1701.5
$b_{\max}$	−1.608	−1.608	−1.611	0.0006	−1.715	−1.610	−1.513	1.0002	5266.9
μ_x	−3.771	−3.771	−3.779	0.0027	−4.317	−3.779	−3.236	1.0000	6459.2
σ_x^2	9.762	9.762	9.870	0.0016	8.717	9.840	11.172	1.0000	96 174.5

7. Conclusions and future research

The optimisation-based approach adopted in most traffic model calibration and validation literature can be justified as a finite sample approximation of the parameter values that minimise the generalised error $\theta^* := \text{argmin}_{\theta\in\Theta}\, \mathbb{E}_{(X,Y)}[\|F(X,\theta)-Y\|]$. Although such approach resembles the successful training and validation pipeline in machine learning, there exists differences between the two contexts so its effectiveness in machine learning might not be transferable to traffic model calibration. Traffic models are often structurally rigid with physically meaningful parameters. Aside from predictions, further inferences are often drawn based on the estimated parameter values, so it is crucial to quantify how reliable such estimates are.

Bayesian inference provides a flexible and coherent framework for uncertainty quantification and propagation. In this research, we investigated the application of Bayesian inference to conduct parameter estimation, using the Gipps' car following model as example. We showed how the posterior $p(\theta|D)$ quantifies the uncertainty of the parameter estimates and hence provides a basis for assessing the reliability of inferences drawn based on the estimated parameter values.

Furthermore, we investigated the effect of incorporating parameter uncertainty in subsequent predictions. We found that for our statistical model, applying a full Bayesian approach with the posterior predictive $p(\tilde{y}|D)$ which integrates over the uncertainty in

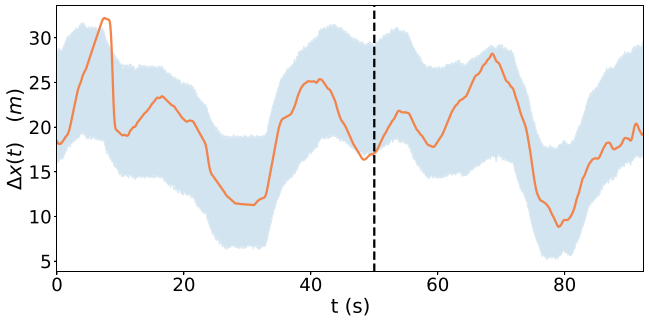

Fig. 13. The central 95% pointwise predictive intervals for the NGSIM data, based on the posterior predictive distribution. The dashed line denotes the 50 s mark that separated the training and validation data.

the parameters, and conditioning on a point estimator $p(\tilde{y}|D,\hat{\theta})$, result in fairly similar predictive uncertainty when given sufficient training data. However, if we reduced the data used to fit the model, the parameters cannot be estimated reliably. Hence conditioning on an unreliable point estimator can result in wrong prediction that is over-confident. In this case, a full Bayesian approach can incorporate parameter uncertainty when making predictions to avoid underestimation of predictive uncertainty, although this does not generally guarantee the correct coverage. Despite the advantages in handling parameter uncertainty, the lack of wide adoption of a Bayesian framework to traffic model calibration is arguably due to both its computationally intensive nature and the requirement of a stronger statistical background. For instance, unlike adaptive optimisers with little restriction on their implementation, an adaptive sampler requires more caution to have the correct ergodic properties.

In addition, we found that the congested flow behavioural parameters in the Gipps' car following model, such as $b_{\max}$ and ψ which both control the congested flow behaviour, are strongly correlated in the posterior. Aside from severely impeding the efficiency of the sampler, the more pressing issue is that such correlation results in a narrow ridge of local optima in the posterior which are all equally good fit, so simply reporting a point estimator can be misleading.

Lastly, we performed Bayesian inference for real data set using a car following episode in the NGISM dataset. In such imperfect model case, there is no ground truth parameters. The full Bayesian approach provides a more informative description of the inference results compared to point estimators, which allows an assessment of the reliability of the parameters estimates. Although the results exhibit some signs of potential model mis-specifications, the model fit appears satisfactory with reasonable predictive performance.

This research emphasises the connection of model calibration and validation to statistical inference, and demonstrates the potential usefulness of a Bayesian approach. For transport researchers unfamiliar with Bayesian inference, this paper is intended to also serve as an introduction to its philosophy and computations. Future work includes investigating the use of more efficient and state-of-the-art samplers which are less impeded by the geometry of the posterior. Another interesting direction is to conduct more extensive investigation on the issue of model mis-specification when fitting to real trajectory, as human driving behaviour is complex and might not be fully described by any existing car following models.

CRediT authorship contribution statement

Samson Ting: Writing – review & editing, Writing – original draft, Visualization, Software, Investigation, Formal analysis, Data curation, Conceptualization. **Thomas Lymburn:** Writing – review & editing. **Thomas Stemler:** Writing – review & editing, Supervision. **Yuchao Sun:** Writing – review & editing, Supervision, Funding acquisition. **Michael Small:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

S.T. was supported by an Australian Government Research Training Program Scholarship at the University of Western Australia, a University Club of Western Australia Research Travel Scholarship, a PATREC Intelligent Transport Systems Postgraduate Research Scholarship, and an iMOVE CRC Postgraduate Research Scholarship funded by iMOVE CRC and supported by the Cooperative Research Centres program, an Australian Government initiative. M.S. acknowledges the support of the Australian Research Council through the Centre for Transforming Maintenance through Data Science (grant number IC180100030), funded by the Australian Government. M.S. is also supported by the ARC Discovery Grant (DP200102961), funded by the Australian Government.

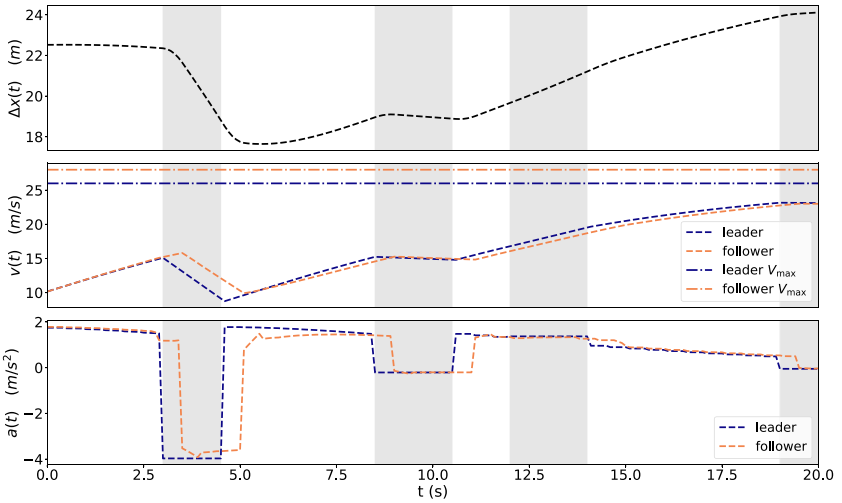

Fig. A.14. Sample trajectory when implemented as a delayed differential equation.

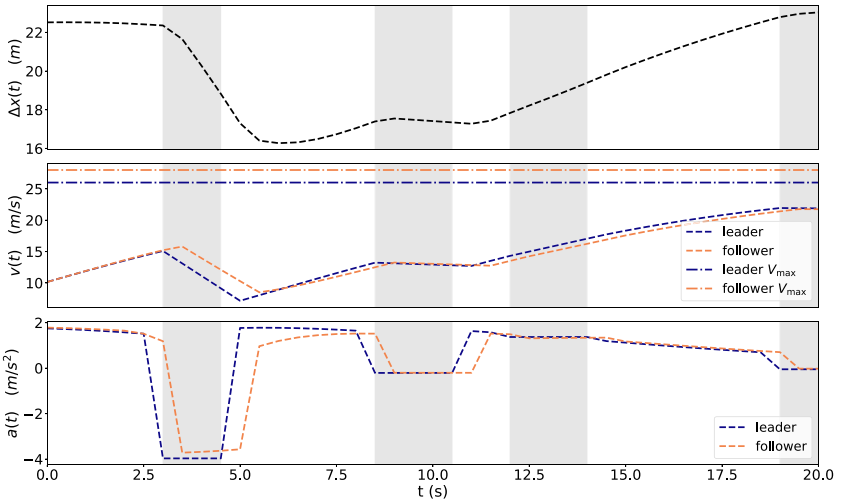

Fig. A.15. Sample trajectory when implemented as a difference equation.

Appendix A. Comparison between Gipps's car following model solver

This appendix compares the two numerical integration schemes for solving the Gipps' car following model, namely the implementation of Gipps' model as a DDE or a difference equation. In the case of $\Delta t = \tau$, the DDE implementation recovers the usual implementation of Gipps' model as a discrete update equation. Figs. A.14 and A.15 compare the two integration schemes which returned fairly similar results. The shaded panels indicate random perturbations imposed on the leader's acceleration/deceleration. In both cases, the reaction time is $\tau = 0.5$ s, with $\Delta t = 0.1$ s for DDE and $\Delta t = \tau = 0.5$ s for difference equation.

Appendix B. MCMC diagnostics

Estimating $\hat{R}$ statistic. Our implementation is motivated by [28] but we also found several slight variations in the literature. Assuming there are M parallel chains each of length N after burn-in, $\hat{R}$ is then estimated as follows:

1. Compute the empirical mean $\bar{\theta}_m$ and variance s_m^2 for each chain $m = 1, \ldots, M$

$$\bar{\theta}_m = \frac{1}{N} \sum_{n=1}^{N} \theta_{m,n} \tag{B.1}$$

$$s_m^2 = \frac{1}{N-1}\sum_{n=1}^{N}(\theta_{m,n} - \bar{\theta}_m)^2 \tag{B.2}$$

2. Compute the overall mean $\bar{\theta}$

$$\bar{\theta} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_m \tag{B.3}$$

3. Compute the within-chain variance W, the between-chain variance B and the pooled variance V

$$W = \frac{1}{M}\sum_{m=1}^{M}s_m^2 \tag{B.4}$$

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{\theta}_m - \bar{\theta})^2 \tag{B.5}$$

$$V = \frac{N-1}{N}W + \frac{1}{N}B \tag{B.6}$$

4. $\hat{R}$ is then given by

$$\hat{R} = \sqrt{\frac{V}{W}} \tag{B.7}$$

Estimating the effective sample size n_{eff}. Likewise, there exists several variations in the literature. Our implementation closely follows [23]:

1. Compute the variogram $\hat{v}_k$ at lag k

$$\hat{v}_k = \frac{1}{M(N-k)}\sum_{m=1}^{M}\sum_{n=k+1}^{N}(\theta_{m,n} - \theta_{m,n-k})^2 \tag{B.8}$$

2. Estimate the lag k auto-correlation by

$$\hat{\rho}_k = 1 - \frac{\hat{v}_k}{2V} \tag{B.9}$$

where V is the pooled variance in step 3 for estimating $\hat{R}$.

3. Estimate the effective sample size by

$$\hat{n}_{\text{eff}} = \frac{MN}{1 + 2\sum_{k=1}^{K}\hat{\rho}_k} \tag{B.10}$$

where K is the first odd positive integer such that $\hat{\rho}_{K+1} + \hat{\rho}_{K+2} < 0$.

Estimating the Monte Carlo standard error (MCSE). The interest is in estimating the MCSE for the posterior mean $\mu = \mathbb{E}[\theta^{[i]}]$ where $\theta^{[i]} \sim p(\theta^{[i]}|D)$ using the method of batch means. A batch is simply a subsequence of the iterates from a Markov chain. Let b denote the batch length and suppose there is a total of $N = ab$ iterations which can hence be divided into a non-overlapping batches each of length b. The method of batch means is as follows:

1. Compute the usual Monte Carlo estimate of μ for each batch, $k = 1, \ldots, a$

$$\hat{\mu}_k = \frac{1}{b}\sum_{j=b(k-1)+1}^{bk}\theta^{[i]} \tag{B.11}$$

2. Compute the empirical variance of the batch means

$$s_b^2 = \frac{1}{a}\sum_{k=1}^{a}(\hat{\mu}_k - \hat{\mu})^2 \tag{B.12}$$

where $\hat{\mu} = \frac{1}{N}\sum_{j=1}^{N}\theta_j^{[i]}$ is the overall mean. By Markov chain CLT, this is can be used to estimate σ^2/b where σ^2 is the variance of $p(\theta^{[i]}|D)$.

3. Compute the batch means estimate of the MCSE

$$\hat{\sigma}^2 = b \times s_b^2 \tag{B.13}$$

$$\text{MCSE}(\mu) = \frac{\hat{\sigma}}{\sqrt{N}} \tag{B.14}$$

The batch length b should be large enough such that each batch estimator is approximately independent considering auto-correlation in the samples, while maintaining a suitably large number of batches for an accurate estimate of the variance of batch means. We simply followed the heuristics of choosing an integer $b \approx \sqrt{N}$.

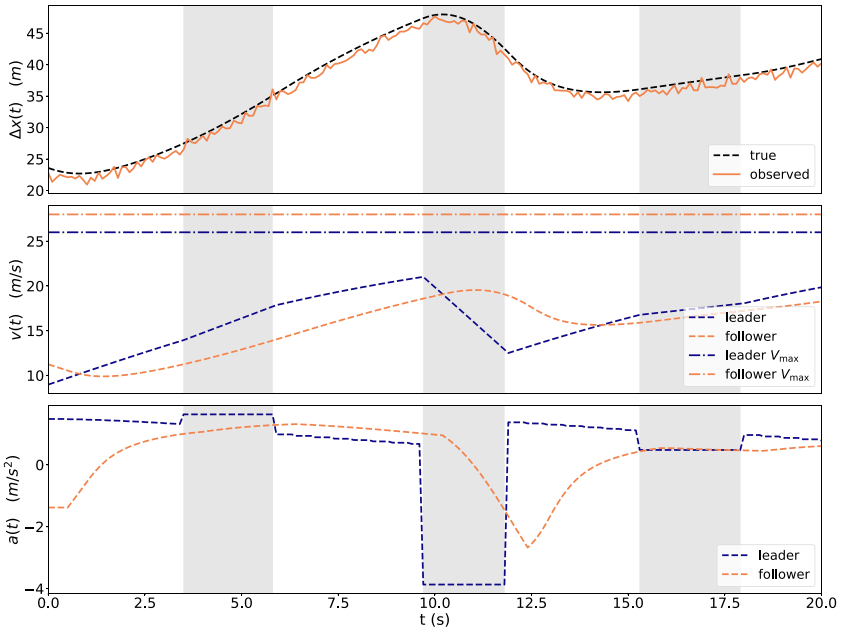

Fig. C.16. Trajectory data simulated with the intelligent driver model, with i.i.d. observational noise. The leader adopts the Gipps' free flow model in the absence of disturbances and the follower adopts the IDM model. The shaded panels indicate random perturbations imposed on the leader's acceleration/deceleration.

Table C.5
Estimates and MCMC diagnostic statistics for IDM.

	True	MLE	MAP	$\mathbb{E}[\cdot\|D]$	MCSE	$q_{0.025}$	$q_{0.5}$	$q_{0.975}$	$\hat{R}$	n_{eff}
$a_{\max}$	1.721	1.710	1.711	1.717	0.0011	1.654	1.716	1.784	1.0114	274.0
$b_{\max}$	3.235	3.189	3.206	3.215	0.0019	3.022	3.213	3.423	1.0040	1460.2
s_0	3.136	3.093	3.086	3.109	0.0057	2.771	3.109	3.437	1.0091	260.1
μ_x	0.800	0.750	0.742	0.749	0.0035	0.497	0.750	1.000	1.0042	428.8
σ_x^2	0.250	0.253	0.279	0.289	0.0001	0.237	0.287	0.352	1.0000	52 435.1

Appendix C. Inference for intelligent driver model

The inference result for the intelligent driver model (IDM) is provided in this appendix, demonstrating the modularity of the proposed method. The IDM proposed by [37] is another popular car following model in the literature, described by ordinary differential equations. Let $\Delta v(t) := v_f(t) - v_l(t)$ denote the velocity difference, $g(t) := x_l(t) - x_f(t) - l_p$ denote the gap between the vehicles with l_p representing the physical length of the leader, then the IDM is given by

$$\frac{dx_f(t)}{dt} = v_f(t) \tag{C.1}$$

$$\frac{dv_f(t)}{dt} = a_{\max}\left[1 - \left(\frac{v_f(t)}{V_{\max}}\right)^{\delta} - \left(\frac{s^*(v_f(t), \Delta v(t))}{g(t)}\right)\right] \tag{C.2}$$

with $s^*(v_f(t), \Delta v(t)) = s_0 + v_f(t)T_{\min} + \frac{v_f(t)\Delta v(t)}{2\sqrt{a_{\max} b_{\max}}}$. The model parameters can be interpreted as follows: $V_{\max}$ is the free flow velocity, s_0 is the minimum gap, $T_{\min}$ is the minimum time headway, $a_{\max}$ is the maximum acceleration, $b_{\max}(>0)$ is the comfortable braking rate and the exponent δ is usually taken as 4. Fig. C.16 illustrates a simulated data with random perturbations imposed on the leader, and with generated parameter values $V_{\max} = 28.0, s_0 = 3.136, T_{\min} = 1.507, a_{\max} = 1.721, b_{\max} = 3.235$. We also introduced reaction time $\tau = 0.5$ and solved the IDM similar to Algorithm 1. The observation model is the same as Section 2.2.

We performed similar Bayesian inference with the IDM, using this 20 s trajectory data with the same adaptive sampler in Section 4.2. For demonstration, we considered only the calibration parameters $\gamma = (a_{\max}, b_{\max}, s_0)$ and noise parameters μ_x and σ_x^2. The MCMC inference results are shown in Fig. C.17 and Table C.5. The MCMC diagnostic statistics did not indicate any issues with mixing and the parameters are recovered well with reasonable uncertainty estimates.

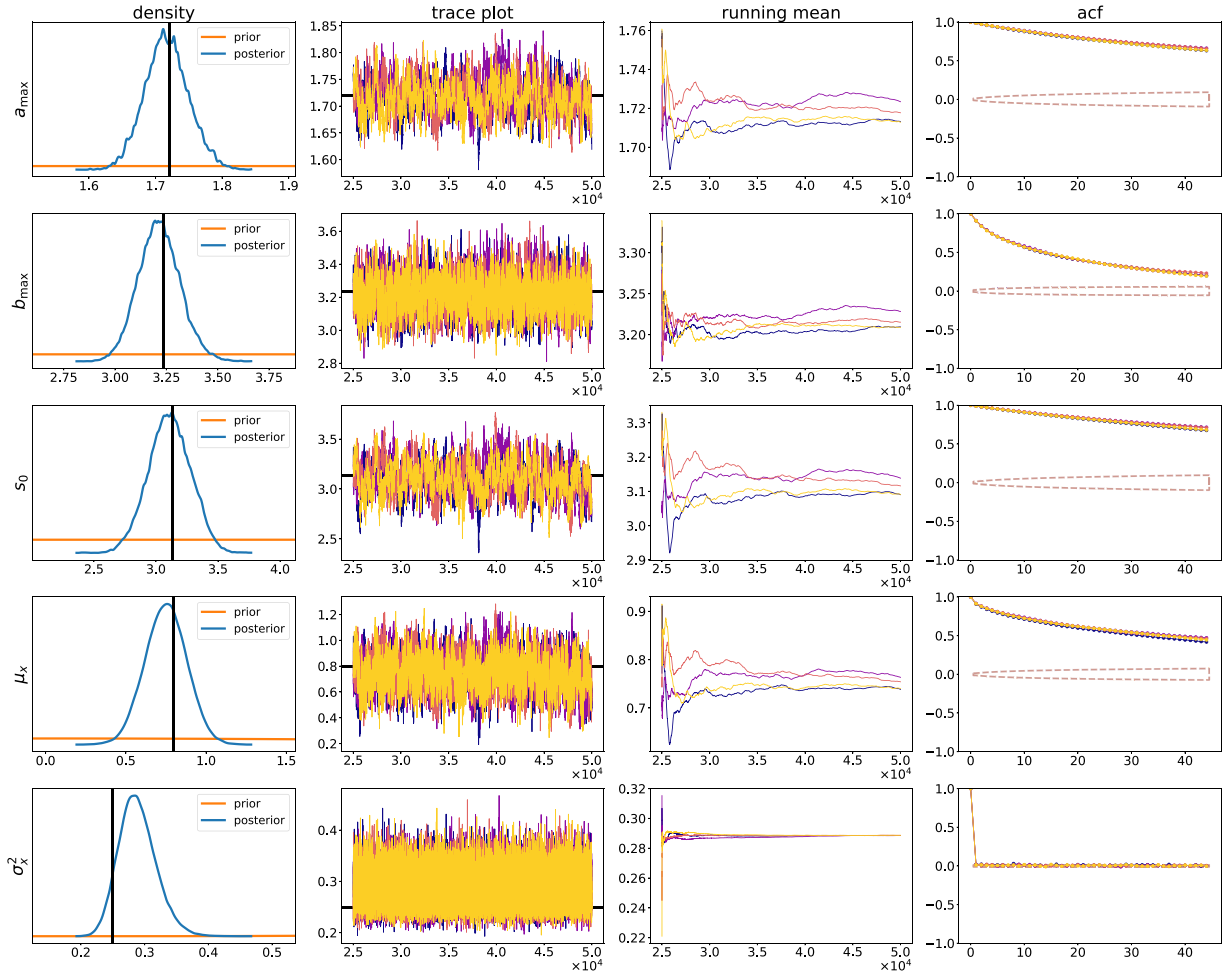

Fig. C.17. MCMC results for IDM model.

References

[1] Y. Peng, S. Liu, Z.Y. Dennis, An improved car-following model with consideration of multiple preceding and following vehicles in a driver's view, Phys. A 538 (2020) 122967.

[2] X. Wang, M. Liu, Y. Ci, L. Wu, Effect of front two adjacent vehicles' velocity information on car-following model construction and stability analysis, Physica A 607 (2022) 128196.

[3] Z. Cui, X. Wang, Y. Ci, C. Yang, J. Yao, Modeling and analysis of car-following models incorporating multiple lead vehicles and acceleration information in heterogeneous traffic flow, Physica A 630 (2023) 129259.

[4] E. Brockfeld, R.D. Kühne, P. Wagner, Calibration and validation of microscopic traffic flow models, Transp. Res. Rec. 1876 (1) (2004) 62–70.

[5] J. Hourdakis, P.G. Michalopoulos, J. Kottommannil, Practical procedure for calibrating microscopic traffic simulation models, Transp. Res. Rec. 1852 (1) (2003) 130–139.

[6] K.-O. Kim, L. Rilett, Simplex-based calibration of traffic microsimulation models with intelligent transportation systems data, Transp. Res. Rec. 1855 (1) (2003) 80–89.

[7] S. Ossen, S.P. Hoogendoorn, Car-following behavior analysis from microscopic trajectory data, Transp. Res. Rec. 1934 (1) (2005) 13–21.

[8] B. Park, H. Qi, Development and evaluation of a procedure for the calibration of simulation models, Transp. Res. Rec. 1934 (1) (2005) 208–217.

[9] Y. Hollander, R. Liu, The principles of calibrating traffic microsimulation models, Transportation 35 (3) (2008) 347–362.

[10] A. Kesting, M. Treiber, Calibrating car-following models by using trajectory data: Methodological study, Transp. Res. Rec. 2088 (1) (2008) 148–156.

[11] V. Punzo, Z. Zheng, M. Montanino, About calibration of car-following dynamics of automated and human-driven vehicles: Methodology, guidelines and codes, Transp. Res. C 128 (2021) 103165.

[12] W. Pan, J. Zhang, J. Tian, F. Cui, T. Wang, Analysis of car–following behaviors based on data–driven and theory–driven car–following models: Heterogeneity and asymmetry, Physica A 632 (2023) 129324.

[13] S. Ting, T. Lymburn, T. Stemler, Y. Sun, M. Small, Model calibration and validation from a statistical inference perspective, 2023, arXiv preprint arXiv:2309.08562.

[14] P.G. Gipps, A behavioural car-following model for computer simulation, Transp. Res. B 15 (2) (1981) 105–111.

[15] L. Lücken, Resolving collisions for the gipps car-following model, 2019, arXiv preprint arXiv:1902.04927.

[16] B. Ciuffo, V. Punzo, M. Montanino, Thirty years of gipps' car-following model: Applications, developments, and new features, Transp. Res. Rec. 2315 (1) (2012) 89–99.

[17] E. Brockfeld, R.D. Kühne, A. Skabardonis, P. Wagner, Toward benchmarking of microscopic traffic flow models, Transp. Res. Rec. 1852 (1) (2003) 124–129.
[18] S. Ossen, S.P. Hoogendoorn, Validity of trajectory-based calibration approach of car-following models in presence of measurement errors, Transp. Res. Rec. 2088 (1) (2008) 117–125.
[19] S. Ossen, S.P. Hoogendoorn, Reliability of parameter values estimated using trajectory observations, Transp. Res. Rec. 2124 (1) (2009) 36–44.
[20] V. Punzo, B. Ciuffo, M. Montanino, Can results of car-following model calibration based on trajectory data be trusted? Transp. Res. Rec. 2315 (1) (2012) 11–24.
[21] G.M. Martin, D.T. Frazier, C.P. Robert, Computing bayes: From then 'til now, Statist. Sci. 1 (1) (2023) 1–17.
[22] G.H. Givens, J.A. Hoeting, Computational Statistics, vol. 703, John Wiley & Sons, 2012.
[23] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin, Bayesian Data Analysis, third ed., Chapman and Hall/CRC, 2013.
[24] R.V. Craiu, J.S. Rosenthal, Bayesian computation via markov chain monte carlo, Annu. Rev. Stat. Appl. 1 (2014) 179–201.
[25] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[26] W.K. Hastings, Monte carlo sampling methods using markov chains and their applications, Biometrika 57 (1) (1970) 97–109.
[27] S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. (6) (1984) 721–741.
[28] A. Gelman, D.B. Rubin, Inference from iterative simulation using multiple sequences, Stat. Sci. (1992) 457–472.
[29] H. Haario, E. Saksman, J. Tamminen, An adaptive metropolis algorithm, Bernoulli (2001) 223–242.
[30] G.O. Roberts, J.S. Rosenthal, Coupling and ergodicity of adaptive markov chain monte carlo algorithms, J. Appl. Probab. 44 (2) (2007) 458–475.
[31] R.V. Craiu, J. Rosenthal, C. Yang, Learn from thy neighbor: Parallel-chain and regional adaptive mcmc, J. Amer. Statist. Assoc. 104 (488) (2009) 1454–1466.
[32] A. Gelman, W.R. Gilks, G.O. Roberts, Weak convergence and optimal scaling of random walk metropolis algorithms, Ann. Appl. Probab. 7 (1) (1997) 110–120.
[33] R.M. Neal, et al., Mcmc using hamiltonian dynamics, in: Handbook of markov chain monte carlo, Vol. 2, 2011, p. 2, (11).
[34] M.D. Hoffman, A. Gelman, et al., The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo, J. Mach. Learn. Res. 15 (1) (2014) 1593–1623.
[35] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: composable transformations of python+numpy programs, 2018, URL http://github.com/google/jax.
[36] U.S. Department of Transportation Federal Highway Administration, Next generation simulation (NGSIM) vehicle trajectories and supporting data, provided by ITS DataHub through data.transportation.gov, 2016, http://dx.doi.org/10.21949/1504477, accessed 2024-02-02 from.
[37] M. Treiber, A. Hennecke, D. Helbing, Congested traffic states in empirical observations and microscopic simulations, Phys. Rev. E 62 (2) (2000) 1805.