

Learning Car-Following Behaviors Using Bayesian Matrix Normal Mixture Regression

Chengyuan Zhang¹, *Graduate Student Member, IEEE*, Kehua Chen^{2,3}, *Graduate Student Member, IEEE*, Meixin Zhu^{4,5}, *Member, IEEE*, Hai Yang³, and Lijun Sun¹, *Senior Member, IEEE*,

Abstract—Learning and understanding car-following (CF) behaviors are crucial for microscopic traffic simulation. Traditional CF models, though simple, often lack generalization capabilities, while many data-driven methods, despite their robustness, operate as “black boxes” with limited interpretability. To bridge this gap, this work introduces a Bayesian Matrix Normal Mixture Regression (MNMR) model that simultaneously captures feature correlations and temporal dynamics inherent in CF behaviors. This approach is distinguished by its separate learning of row and column covariance matrices within the model framework, offering an insightful perspective into the human driver decision-making processes. Through extensive experiments, we assess the model’s performance across various historical steps of inputs, predictive steps of outputs, and model complexities. The results consistently demonstrate our model’s adeptness in effectively capturing the intricate correlations and temporal dynamics present during CF. A focused case study further illustrates the model’s outperforming interpretability of identifying distinct operational conditions through the learned mean and covariance matrices. This not only underlines our model’s effectiveness in understanding complex human driving behaviors in CF scenarios but also highlights its potential as a tool for enhancing the interpretability of CF behaviors in traffic simulations and autonomous driving systems.

I. INTRODUCTION

Car-following (CF) maneuver involves the interactive response of the following vehicles (FV) dictated by the states of the leading vehicles (LV) in the longitudinal direction [1]. As a vital component in microscopic traffic analysis, accurately modeling CF behaviors benefits traffic management and the development of autonomous driving. Therefore, researchers have dedicated efforts to CF models over the past several decades [2]–[4].

Early studies developed models based on traffic theory and human behaviors, such as Gipps’ model [5], Newell model [6], and Intelligent Driver Model (IDM) [7]. Although

traditional models have clearly defined formulas with high interpretability, these models have low complexity and cannot capture complex temporal dynamics. With the thriving of data-driven methods, a number of studies utilized advanced deep learning techniques to model CF behaviors, such as deep reinforcement learning [8] and Recurrent Neural Networks (RNNs) [9]. While the aforementioned data-driven approaches attain satisfactory accuracy, it is important to note that they function as black-box models with limited interpretability, and lack the capability to quantify uncertainties.

To capture the stochasticity of human behaviors while maintaining reasonable interpretability, some studies modeled CF behaviors from Bayesian perspectives, such as using the Gaussian Mixture Model (GMM) [10]–[12]. GMM is widely adopted since it explicitly models the uncertainty in the parameters, providing a better understanding of the model’s confidence in its predictions. This is crucial in applications with safety-critical driving tasks, where understanding the uncertainty of the model can help human drivers make better decisions. Besides, apart from GMM capturing stochasticity, it is also good at depicting nonlinear human behaviors with explainable modalities. For instance, Angkititakul et al. [10] employed GMM and piecewise auto-regressive exogenous algorithms to model CF behaviors. Afterward, the authors enhance the model structure, enabling non-parametric Bayesian modeling of CF behaviors [12]. Lefevre et al. [11] conducted a comparison between parametric and non-parametric methods for CF modeling. They observed that the relative performance of the various models exhibited significant variations depending on the prediction horizon. Recently, Chen et al. [13] used a Bayesian GMM to achieve probabilistic modeling of CF. In their study, they developed an inference algorithm using Gibbs sampling and provided an analysis of model interpretability. Based on GMM, Zhang et al. [1] proposed a framework to quantify the interactions between the FV and LV.

Nonetheless, vanilla GMM learns the full covariance matrices given historical information, while there often exist explicit correlations among the variables or in the temporal space. This provides us an insight that a separable covariance structure of the proposed model can effectively reduce the model parameters, speed up computations, and improve practical interpretability. With these motivations, we develop a probabilistic CF model with good interpretability using Bayesian Matrix Normal Mixture Regression (MNMR). Specifically, we first learn a joint distribution in terms of the Bayesian Matrix Normal Mixture Model (MNMM) [14],

(Corresponding author: Meixin Zhu)

[†] Chengyuan Zhang and Kehua Chen contributed equally to this work.

¹C. Zhang and L. Sun are with the Department of Civil Engineering, McGill University, Montreal, QC H3A 0C3, Canada. E-mail addresses: enzozcy@gmail.com (C. Zhang), lijun.sun@mcgill.ca (L. Sun).

²K. Chen is with Division of Emerging Interdisciplinary Areas (EMIA), Academy of Interdisciplinary Studies, The Hong Kong University of Science and Technology, Hong Kong, China.

³K. Chen and H. Yang are also with Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.

⁴M. Zhu is with Intelligent Transportation Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China.

⁵M. Zhu is also with Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things, Guangzhou, China.

[15], then make predictions based on conditional distributions. The learned row and column covariance matrices indicate the correlations among features and temporal dynamics, respectively. Compared with the previous data-driven models, our proposed MNMR provides an intuitive understanding of complex CF dynamics. We conduct extensive experiments using the HighD dataset and present a case study to illustrate the interpretability of our model.

To sum up, the contributions of this work are as follows:

- We present a novel model named Bayesian Matrix Normal Mixture Regression (MNMR) to authentically capture car-following dynamics while appropriately accounting for stochasticity.
- We perform extensive experiments using the HighD dataset to demonstrate and analyze the impact of different historical steps, prediction steps, and model complexity in the regression task.
- We present a case study to demonstrate the outperforming interpretability of our model, showcasing that the learned mean and covariance matrices of different components represent distinct human driving behaviors.

II. PRELIMINARIES AND PROBLEM DEFINITION

The primary objective of this work is to develop a comprehensive statistical model based on Bayesian learning to accurately represent, interpret, and analyze CF behaviors. This model aims to capture the complex, dynamic interactions between a follower-leader pair in a traffic stream, with a particular focus on the following aspects: (1) The variability in car-following patterns under different traffic situations; and (2) The temporal dynamics of car-following, considering factors like speed, distance, and relative acceleration.

In a typical CF task, an FV adjusts its driving speed based on the actions of the leading vehicle (LV). Given a continuous period (e.g., the past T time steps) of observations, this work aims to predict the FV's actions in the next ΔT time steps. Here we denote $s_{(\cdot)}(t)$, $v_{(\cdot)}(t)$, and $a_{(\cdot)}(t)$ as the position, speed, and acceleration of vehicles at a specific time t , where $(\cdot)$ can be either the FV or LV. In the setting of a regression task, we consider the following regressors for CF modeling at each time step t : the speed of FV (i.e., $v_{\text{FV}}(t)$), the speed difference between FV and LV (i.e., $\Delta v = v_{\text{FV}}(t) - v_{\text{LV}}(t)$), and the longitudinal spacing between FV and LV (i.e., $\Delta s = s_{\text{FV}}(t) - s_{\text{LV}}(t) - \text{vehicle length}$), denoted as $\mathbf{x}(t) \in \mathbb{R}^{D_x=3}$. We select the action of FV (i.e., $a_{\text{FV}}(t)$) as the response variable $y(t) \in \mathbb{R}^{D_y=1}$. In general, the CF regression task is to learn a mapping $f: \mathbf{x}(t) \in \mathbb{R}^{D_x} \mapsto y(t) \in \mathbb{R}^{D_y}$, but note that in our setting $D_y = 1$, indicating that y is a scalar.

In literature, Gaussian mixture regression (GMR) provides an elegant method to estimate the distribution of output data given input data [16]. The elemental assumption of GMR is that all the data points are generated from a GMM with unknown parameters. Therefore, it encodes the joint distribution of input and output using a GMM

$$\begin{bmatrix} \mathbf{x}(t) \\ y(t) \end{bmatrix} \sim \sum_{k=1}^K \pi_k \mathcal{N}(\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}), \quad (1)$$

where $\boldsymbol{\mu}_k^{(t)} \in \mathbb{R}^{(D_x+D_y)}$ and $\boldsymbol{\Sigma}_k^{(t)} \in \mathbb{R}^{(D_x+D_y) \times (D_x+D_y)}$. Then it exploits the Gaussian conditioning theorem to build the mapping from input to output (i.e., $p(y(t)|\mathbf{x}(t))$). In practice, although it is efficient to implement GMR at each time step, the need to estimate the full covariance matrices $\boldsymbol{\Sigma}_k^{(t)}$ would become complicated when dealing with time series. For instance, once we have observed $\mathbf{x} \in \mathbb{R}^{(D_x \times T)}$ from the past T time steps, we then want to predict the future driving behaviors $\mathbf{y}' \in \mathbb{R}^{(D_y \times \Delta T)}$. Correspondingly, the parameters of GMM are adapted to $\boldsymbol{\mu}_k^{(t)} \in \mathbb{R}^{(D_x \times T + D_y \times \Delta T)}$ and $\boldsymbol{\Sigma}_k^{(t)} \in \mathbb{R}^{(D_x \times T + D_y \times \Delta T) \times (D_x \times T + D_y \times \Delta T)}$. As aforementioned, learning the full covariance matrices requires more computational resources with longer time series.

To relieve the computational burdens from learning the full covariance matrices, in this work, we model the observations as a matrix variable instead of a vector-form and leverage the trick of the Kronecker product to decompose a full covariance matrix into two smaller covariance matrices.

III. METHOD

Accurately model and analyze human driving data is inherently complex due to the high dimensionality and intricate temporal dynamics. Traditional models often struggle to capture the full scope of the correlations and nonlinear dynamics. In this section, we introduce a Bayesian Matrix Normal Mixture Regression (MNMM) and a Matrix Normal Mixture Regression (MNMR) to effectively model such data.

A. Matrix Normal Mixture Model (MNMM)

First, as aforementioned, we model the observations as a matrix-variable instead of a vector one. Here, we assume that spatiotemporal driving data $\mathcal{X} \in \mathbb{R}^{N \times D \times \tau}$ can be modeled by the matrix normal distribution. For the notation conciseness, we have denoted $D = D_x + D_y$ and $\tau = T + \Delta T$. Let the matrix-variate symbol $\mathbf{X}$ be a $D \times \tau$ random matrix following a matrix normal distribution with mean matrix $\mathbf{M} \in \mathbb{R}^{D \times \tau}$ to represent the data trends, row covariance matrix $\mathbf{U} \in \mathbb{R}^{D \times D}$ to capture the correlations among variables, and column covariance matrix $\mathbf{V} \in \mathbb{R}^{\tau \times \tau}$ to depict the temporal dynamics, whose dual structure allows for a more explicit understanding of the driving behavior trends. Then we have,

$$\mathbf{X} \sim \mathcal{MN}(\mathbf{M}, \mathbf{U}, \mathbf{V}), \quad (2)$$

$$\Rightarrow \text{vec}(\mathbf{X}) \sim \mathcal{N}(\text{vec}(\mathbf{M}), \mathbf{V} \otimes \mathbf{U}), \quad (3)$$

where $\text{vec}(\cdot)$ is the vectorization operator that stacks all the columns in a matrix into a tall column vector. To distinguish the matrix normal distribution and multivariate normal distribution, we represent them using $\mathcal{MN}$ and $\mathcal{N}$, respectively. In addition, the symbol $\otimes$ denotes the Kronecker product, defined as

$$\mathbf{V} \otimes \mathbf{U} = \begin{bmatrix} v_{11}\mathbf{U} & \cdots & v_{1\tau}\mathbf{U} \\ \vdots & \ddots & \vdots \\ v_{\tau 1}\mathbf{U} & \cdots & v_{\tau\tau}\mathbf{U} \end{bmatrix} \in \mathbb{R}^{(D \times \tau) \times (D \times \tau)}. \quad (4)$$

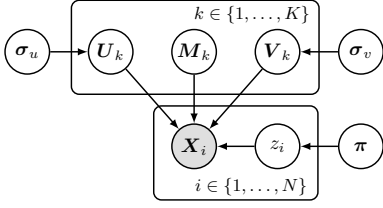


Fig. 1. Probabilistic graphical model of MNMM.

Note that $\forall \xi \neq 0$, a property that Kronecker product holds is $\mathbf{V} \otimes \mathbf{U} = (\mathbf{V}/\xi) \otimes (\xi \mathbf{U})$. Therefore, to avoid the non-identifiable inference issue, we put a strong prior on one covariance matrix and an uninformative prior on another.

Here we define a mixture model of matrix normal distributions with K components [14]:

$$\mathbf{X} \sim \sum_{k=1}^K \pi_k \mathcal{MN}(\mathbf{M}_k, \mathbf{U}_k, \mathbf{V}_k), \quad (5)$$

where $0 \leq \pi_k \leq 1$ denote the mixing weight of the k -th matrix normal distribution, and $\sum_{k=1}^K \pi_k = 1$.

In a Bayesian framework, we develop a MNMM as follows

For component $k = 1, \dots, K$:

$$\mathbf{M}_k \sim \mathcal{MN}(\mathbf{M}_0, \mathbf{U}_0, \mathbf{V}_0), \quad (6)$$

$$\mathbf{V}_k \sim \text{LKJCholeskyCov}(\eta, \boldsymbol{\sigma}_v), \quad (7)$$

$$\mathbf{U}_k \sim \text{LKJCholeskyCov}(\eta, \boldsymbol{\sigma}_u), \quad (8)$$

For sample $i = 1, \dots, N$:

$$z_i \sim \text{Cat}(\boldsymbol{\pi}), \quad (9)$$

$$\mathbf{X}_i \sim \mathcal{MN}(\mathbf{M}_{z_i}, \mathbf{U}_{z_i}, \mathbf{V}_{z_i}), \quad (10)$$

where ‘LKJ’ represents the LKJ distribution [17], and the priors are defined as $\boldsymbol{\pi} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, $\boldsymbol{\sigma}_v \sim \text{Exp}(\lambda_v)$ and $\boldsymbol{\sigma}_u \sim \text{Exp}(\lambda_u)$, in which $\boldsymbol{\alpha}$, $\mathbf{M}_0$, $\mathbf{U}_0$, $\mathbf{V}_0$, η , λ_v , λ_u , are all hyperparameters to tune. The corresponding model structure is illustrated as a probabilistic graphical model, see Fig. 1.

B. Matrix Normal Mixture Regression (MNMR)

In an interactive driving scenario, human drivers make decisions and perform driving maneuvers based on observed information from several past seconds [18]. This insight reveals that it is significant to incorporate the past observations into the future predictions. Therefore, based on this insight, we develop a Matrix Normal Mixture Regression (MNMR) based on MNMM in the following.

In a sequence-to-sequence case, we are interested in learning a mapping $f: \tilde{\mathbf{X}} = [\mathbf{X}, \mathbf{X}'] \in \mathbb{R}^{D_x \times (T+\Delta T)} \mapsto \tilde{\mathbf{Y}} = [\mathbf{Y}, \mathbf{Y}'] \in \mathbb{R}^{D_y \times (T+\Delta T)}$, where $\mathbf{X} \in \mathbb{R}^{D_x \times T}$, $\mathbf{X}' \in \mathbb{R}^{D_x \times \Delta T}$, $\mathbf{Y} \in \mathbb{R}^{D_y \times T}$, and $\mathbf{Y}' \in \mathbb{R}^{D_y \times \Delta T}$. Then the MNMM can be rewritten as:

$$\begin{bmatrix} \tilde{\mathbf{X}} \\ \tilde{\mathbf{Y}} \end{bmatrix} = \begin{bmatrix} \mathbf{X} & \mathbf{X}' \\ \mathbf{Y} & \mathbf{Y}' \end{bmatrix} \sim \sum_{k=1}^K \pi_k \mathcal{MN}(\mathbf{M}_k, \mathbf{U}_k, \mathbf{V}_k), \quad (11)$$

where

$$\begin{aligned} \mathbf{M}_k &= \begin{bmatrix} \mathbf{M}_{\tilde{\mathbf{X}}, k} \\ \mathbf{M}_{\tilde{\mathbf{Y}}, k} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{\mathbf{X}, k} & \mathbf{M}_{\mathbf{X}', k} \\ \mathbf{M}_{\mathbf{Y}, k} & \mathbf{M}_{\mathbf{Y}', k} \end{bmatrix}, \\ \mathbf{V}_k &= \begin{bmatrix} \mathbf{V}_{11, k} \in \mathbb{R}^{T \times T} & \mathbf{V}_{12, k} \in \mathbb{R}^{T \times \Delta T} \\ \mathbf{V}_{21, k} \in \mathbb{R}^{\Delta T \times T} & \mathbf{V}_{22, k} \in \mathbb{R}^{\Delta T \times \Delta T} \end{bmatrix}, \\ \mathbf{U}_k &= \begin{bmatrix} \mathbf{U}_{11, k} \in \mathbb{R}^{D_x \times D_x} & \mathbf{U}_{12, k} \in \mathbb{R}^{D_x \times D_y} \\ \mathbf{U}_{21, k} \in \mathbb{R}^{D_y \times D_x} & \mathbf{U}_{22, k} \in \mathbb{R}^{D_y \times D_y} \end{bmatrix}. \end{aligned}$$

This form is equivalent to the vectorized form

$$\begin{bmatrix} \text{vec}(\tilde{\mathbf{X}}) \\ \text{vec}(\tilde{\mathbf{Y}}) \end{bmatrix} \sim \sum_{k=1}^K \pi_k \mathcal{N}\left(\begin{bmatrix} \text{vec}(\mathbf{M}_{\tilde{\mathbf{X}}, k}) \\ \text{vec}(\mathbf{M}_{\tilde{\mathbf{Y}}, k}) \end{bmatrix}, \mathbf{V}_k \otimes \mathbf{U}_k\right), \quad (12)$$

and for $\mathbf{V}_k \otimes \mathbf{U}_k$ we have

$$\mathbf{V}_k \otimes \mathbf{U}_k = \begin{bmatrix} \mathbf{V}_{11, k} \otimes \mathbf{U}_k & \mathbf{V}_{12, k} \otimes \mathbf{U}_k \\ \mathbf{V}_{21, k} \otimes \mathbf{U}_k & \mathbf{V}_{22, k} \otimes \mathbf{U}_k \end{bmatrix}. \quad (13)$$

The target of the matrix normal mixture regression (MNMR) model is deriving $p(\tilde{\mathbf{Y}}|\tilde{\mathbf{X}})$, therefore, the joint distribution can be decomposed into two parts

$$\text{vec}(\tilde{\mathbf{X}}) \sim \sum_{k=1}^K \pi_k \mathcal{N}\left(\text{vec}(\mathbf{M}_{\tilde{\mathbf{X}}, k}), \mathbf{V}_{11, k} \otimes \mathbf{U}_k\right), \quad (14)$$

$$\text{vec}(\tilde{\mathbf{Y}})|\text{vec}(\tilde{\mathbf{X}}) \sim \sum_{k=1}^K w_k \mathcal{N}\left(\text{vec}(\mathbf{M}_{\tilde{\mathbf{Y}}|\tilde{\mathbf{X}}, k}), \boldsymbol{\Sigma}_{\tilde{\mathbf{Y}}|\tilde{\mathbf{X}}, k}\right), \quad (15)$$

where we have:

$$w_k = \frac{\pi_k \mathcal{N}\left(\text{vec}(\tilde{\mathbf{X}})|\text{vec}(\mathbf{M}_{\tilde{\mathbf{X}}, k}), \mathbf{V}_k \otimes \mathbf{U}_{11, k}\right)}{\sum_{k=1}^K \pi_k \mathcal{N}\left(\text{vec}(\tilde{\mathbf{X}})|\text{vec}(\mathbf{M}_{\tilde{\mathbf{X}}, k}), \mathbf{V}_k \otimes \mathbf{U}_{11, k}\right)}, \quad (16)$$

$$\begin{aligned} \text{vec}(\mathbf{M}_{\tilde{\mathbf{Y}}|\tilde{\mathbf{X}}, k}) &= \text{vec}(\mathbf{M}_{\tilde{\mathbf{Y}}, k}) + \\ &\quad (\mathbf{V}_{21, k} \otimes \mathbf{U}_k)(\mathbf{V}_{11, k}^{-1} \otimes \mathbf{U}_k^{-1})(\text{vec}(\tilde{\mathbf{X}}) - \text{vec}(\mathbf{M}_{\tilde{\mathbf{X}}, k})), \end{aligned} \quad (17)$$

$$\begin{aligned} \boldsymbol{\Sigma}_{\tilde{\mathbf{Y}}|\tilde{\mathbf{X}}, k} &= \mathbf{V}_{22, k} \otimes \mathbf{U}_k - \\ &\quad (\mathbf{V}_{21, k} \otimes \mathbf{U}_k)(\mathbf{V}_{11, k}^{-1} \otimes \mathbf{U}_k^{-1})(\mathbf{V}_{12, k} \otimes \mathbf{U}_k). \end{aligned} \quad (18)$$

Similar to the above procedures, it is easy to derive $p(\text{vec}(\mathbf{Y}')|\text{vec}(\mathbf{Y}), \text{vec}(\tilde{\mathbf{X}}))$ based on $p(\text{vec}(\tilde{\mathbf{Y}})|\text{vec}(\tilde{\mathbf{X}}))$. Furthermore, given that we cannot observe the future information $\mathbf{X}'$ at a specific time step, and we still want to predict the future $\mathbf{Y}'$ given the previous observations, we can therefore marginalize $\mathbf{X}'$ to obtain

$$\begin{aligned} &p(\text{vec}(\mathbf{Y}')|\text{vec}(\mathbf{Y}), \text{vec}(\mathbf{X})) \\ &= \int_{\mathbf{X}'} p(\text{vec}(\mathbf{Y}')|\text{vec}(\mathbf{Y}), \text{vec}(\tilde{\mathbf{X}})) d\mathbf{X}', \end{aligned} \quad (19)$$

which is still a Gaussian mixture model

$$\text{vec}(\mathbf{Y}')|\text{vec}(\mathbf{Y}), \text{vec}(\mathbf{X}) \sim \sum_{k=1}^K \beta_k \mathcal{N}(\boldsymbol{\mu}_{y, k}, \boldsymbol{\Sigma}_{y, k}), \quad (20)$$

where the mean vector $\boldsymbol{\mu}_{y, k} \in \mathbb{R}^{(D_y \times \Delta T)}$ and the covariance matrix $\boldsymbol{\Sigma}_{y, k} \in \mathbb{R}^{(D_y \times \Delta T) \times (D_y \times \Delta T)}$.

TABLE I
REGRESSION PERFORMANCE ON THE ACCELERATION OBSERVATIONS.

		RMSE on the test data / K					MAE on the test data / K					NLL on the training data / K				
ΔT	T	5	10	20	40	60	5	10	20	40	60	5	10	20	40	60
1	1	0.054	0.055	0.054	0.053	0.053	0.047	0.047	0.046	0.046	0.045	-0.213	-0.240	-0.234	-0.247	-0.271
	3	0.013	0.014	0.013	0.013	0.013	0.011	0.011	0.011	0.011	0.011	-1.611	-1.603	-1.665	-1.677	-1.716
	5	0.019	0.012	0.013	0.012	0.011	0.015	0.010	0.011	0.010	0.010	-1.757	-1.877	-2.043	-2.116	-2.084
	7	0.015	0.014	0.020	0.015	0.016	0.013	0.012	0.017	0.012	0.013	-2.006	-2.138	-2.237	-2.255	-2.344
	9	0.025	0.016	0.018	0.017	0.013	0.021	0.014	0.015	0.014	0.011	-2.191	-2.248	-2.327	-2.440	-2.333
3	1	0.240	0.230	0.223	0.219	0.205	0.107	0.104	0.101	0.099	0.093	-1.547	-1.611	-1.648	-1.674	-1.696
	3	0.115	0.089	0.106	0.069	0.069	0.045	0.037	0.043	0.030	0.030	-1.994	-2.007	-2.026	-2.188	-2.172
	5	0.250	0.110	0.107	0.122	0.080	0.097	0.045	0.044	0.048	0.033	-1.861	-2.054	-2.157	-2.256	-2.297
	7	0.189	0.129	0.126	0.095	0.140	0.077	0.053	0.053	0.039	0.055	-2.113	-2.326	-2.333	-2.432	-2.376
	9	0.174	0.148	0.167	0.149	0.120	0.077	0.060	0.071	0.063	0.050	-2.219	-2.302	-2.290	-2.383	-2.360
5	1	0.772	0.788	0.759	0.652	0.675	0.249	0.237	0.250	0.214	0.217	-1.987	-1.831	-2.083	-2.088	-2.114
	3	0.220	0.321	0.376	0.222	0.380	0.068	0.093	0.107	0.068	0.110	-2.399	-2.169	-2.181	-2.339	-2.271
	5	0.361	0.536	0.323	0.241	0.262	0.103	0.130	0.095	0.072	0.077	-2.081	-2.209	-2.100	-2.429	-2.408
	7	0.395	0.248	0.235	0.331	0.451	0.123	0.076	0.135	0.096	0.096	-2.088	-2.226	-2.316	-2.377	-2.381
	9	0.334	0.391	0.349	0.321	0.281	0.097	0.117	0.102	0.093	0.083	-2.314	-2.228	-2.353	-2.444	-2.412
7	1	1.612	1.429	1.018	1.093	0.934	0.452	0.379	0.283	0.309	0.262	-2.162	-2.082	-2.140	-2.187	-2.334
	3	0.632	0.856	0.818	0.490	0.685	0.155	0.222	0.210	0.123	0.161	-2.177	-2.311	-2.381	-2.393	-2.418
	5	0.433	0.708	0.559	0.500	0.451	0.113	0.159	0.135	0.129	0.113	-2.185	-2.330	-2.380	-2.344	-2.380
	7	0.591	0.521	0.483	0.483	0.525	0.152	0.134	0.118	0.123	0.127	-2.328	-2.391	-2.246	-2.447	-2.474
	9	0.549	0.590	0.615	0.503	0.478	0.142	0.143	0.154	0.130	0.121	-2.373	-2.277	-2.427	-2.492	-2.467
9	1	1.194	1.227	1.210	1.281	1.233	0.292	0.299	0.302	0.306	0.306	-2.151	-2.307	-2.253	-2.409	-2.446
	3	0.910	0.875	1.143	1.044	0.849	0.208	0.206	0.251	0.231	0.177	-2.254	-2.259	-2.293	-2.300	-2.290
	5	0.950	0.665	0.683	0.998	0.716	0.208	0.143	0.155	0.216	0.158	-2.185	-2.310	-2.487	-2.445	-2.512
	7	0.761	0.957	0.853	0.716	0.742	0.171	0.200	0.180	0.158	0.163	-2.456	-2.434	-2.475	-2.489	-2.523
	9	0.850	0.731	0.830	1.132	0.811	0.221	0.156	0.178	0.240	0.175	-2.356	-2.515	-2.504	-2.621	-2.577

IV. EXPERIMENTS

A. Dataset and Preprocessing

Experiments are performed on the HighD dataset that contains high-resolution naturalistic vehicle trajectories extracted from drone videos of German highways. Compared to the commonly used NGSIM dataset [19], the HighD dataset benefits from more reliable data capture methods and advanced computer vision techniques. It features 60 recordings captured at different times of the day, ranging from 8:00 to 17:00, and has a resolution of 25 Hz. In our experiment, the original dataset is downsampled to a smaller set with a sampling frequency of 5 Hz, achieved by uniformly selecting every 5-th sample. HighD dataset provides detailed information on the trajectories, velocities, and accelerations of vehicles, which is essential for developing and evaluating car-following models that accurately capture real-world driving scenarios. In this study, we follow the same data processing procedures as in [20] to transform the data into a new coordinate system. We selected 123 leader-follower pairs where the car-following duration lasted for more than 50 sec. Of these pairs, 93 pairs ($\sim 75\%$) were used to train our model, while the remaining 30 pairs ($\sim 25\%$) were stored as the test set to evaluate the forecasting performance. By using pairs with longer car-following durations, we aim to capture more realistic driving behaviors and enable our model to better handle complex and dynamic traffic situations.

B. Regression Performance Evaluation

Based on the proposed Bayesian structure, we employ Stochastic Variational Inference (SVI) based on Pyro to infer the posterior distributions [21]. The model is trained using the Adam optimizer with a learning rate of 0.01. The experiments

were conducted on a Linux server with one NVIDIA RTX A4000 GPU.

In general, a trade-off in selecting the number of components K arises: with too many components, the learned model may over-fit the data, while a model with too few components may not be flexible enough to represent an underlying distribution of observations. To evaluate the performance impact of factors from different aspects, we conduct extensive experiments using combinations of the predicted steps ΔT , the past steps T , and the number of components K . Specifically, we set $\Delta T = \{1, 3, 5, 7, 9\}$ steps, $T = \{1, 3, 5, 7, 9\}$ steps, and $K = \{5, 10, 20, 40, 60\}$. For each combination of $\{T, \Delta T, K\}$, the regression performance is evaluated using the root mean square error (RMSE) and the mean absolute error (MAE) on the test set, and the negative log-likelihood (NLL) on the training data. These metrics are recognized as effective in diagnosing the variation in the errors in a set of forecasts [22].

Table I presents the evaluation results, in which we mark the relatively good performances with bold text. Note that in this table, the lower the value, the better the results. Generally, the prediction errors decrease with longer historical steps while increasing with longer prediction steps. As for the model complexity, the errors would decrease initially but followed by an increase with the increase of K , which is consistent with the aforementioned patterns: A small value of K is inadequate for capturing the complex dynamics of CF, while a large K tends to overfit the training samples, thereby reducing the model's generalization ability and has worse performance when evaluating on the test data. This point is especially obvious when dealing with longer prediction steps and large K — the performance on the training set seem

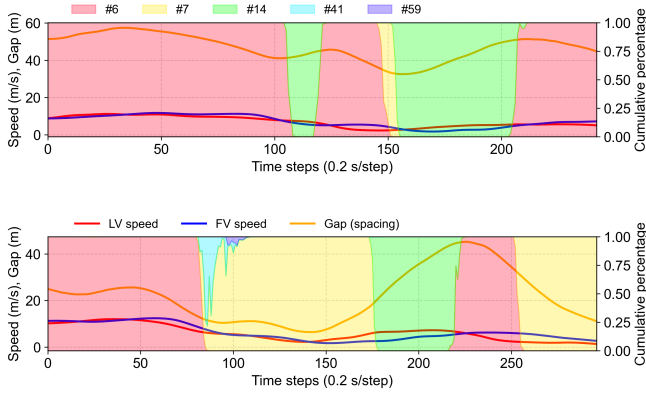


Fig. 2. Two illustrative cases to understand the dominant components (i.e., #6, #7, #14, #41, and #59 in these cases) and the transitioning among them.

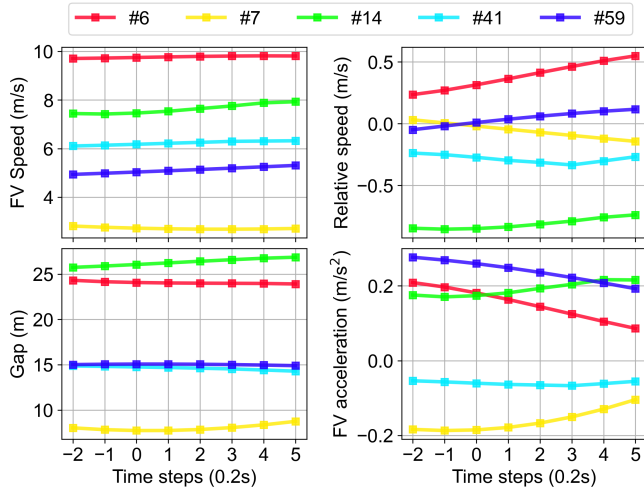


Fig. 3. The mean matrix M_k of the dominant components.

perfect but bad on the test set.

C. Mixture Components Interpreting

Here we highlight that MNMR, as a mixture model, can provide an intuitive understanding of CF behaviors. Based on the model performance and following the settings in [13], we choose a case with $\{T = 5, \Delta T = 3, K = 60\}$ as a light demonstration. Fig. 2 shows two cases with the variation of five dominant components, in which the LV speed, FV speed, and spacing gap are shown in solid lines; also, the percentage of β_k in Eq. (20) are shown as the shaded area. Fig. 3 illustrates the mean matrix M_k of dominant components. We can see that component #6 exerts dominance in the beginning 110 steps and 80 steps for Case 1 (the upper part) and Case 2 (the lower part), respectively. In these instances, FVs exhibit large speeds, gaps, and relative speeds, which are also shown in the trends in the mean matrix illustrated in Fig. 3. During these stages, the FV slightly accelerates to keep a closer distance to the LV, which leads to the increasing relative speed. Besides, component #14 emerges as a significant component during time steps 110-120 and 160-200 in Case 1. During these periods, the FV's action indicates a typical accelerating stage — the LV rapidly increases its speed (especially when

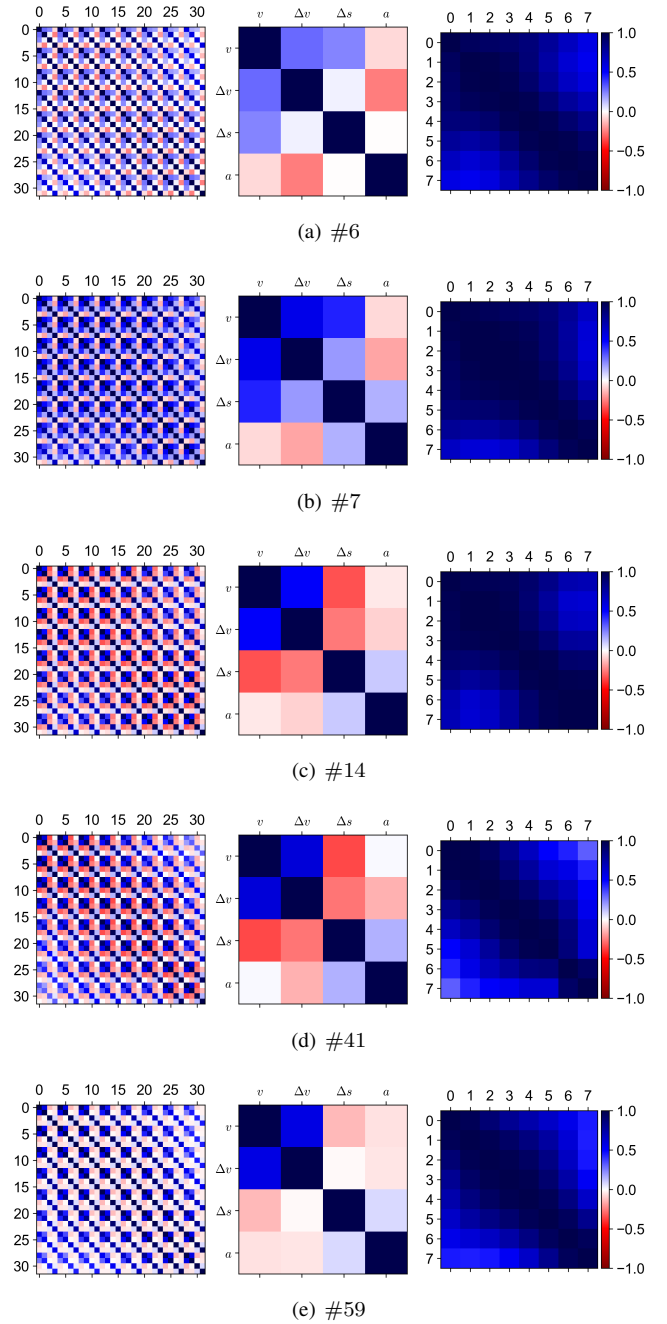


Fig. 4. The correlation matrices of the dominant components.

from a congestion stage to a free-flow stage) and the FV also accelerates to follow up the FV. But due to the action delay, although the FV accelerates hard, it still has a negative relative speed, i.e., $\Delta v = v_{FV}(t) - v_{LV}(t) < 0$, to the LV. Regarding component #7, it dominates during time steps 90-180 and 260-300 in Case 2. Such scenarios reveal the slow-moving stage when the FV and LV are close to each other at a very low speed, and the FV decelerates to keep a safe distance from the LV.

Fig. 4 visualize the learned correlation matrices of the dominant components. Recall that in MNMR, we independently learn the row and column covariance matrices. Here, the left column denotes the full correlation matrix of $V \otimes U$, and the

plots in the second and third columns depict the feature and temporal correlation matrices, respectively. For components #6 and #7, we observe that the speed and relative speed are negatively correlated with acceleration, aligning with our common sense that when speeds and relative speeds are large, FVs are more likely to decelerate. Additionally, gaps are positively correlated to both speed and relative speed since large gaps provide room for FVs to speed up. Conversely, components #14, #41, and #59 suggest that speed and relative speed are negatively correlated with both gaps and acceleration. This observation shows that these two components correspond to operational conditions where FVs are approaching LVs, and FVs tend to decelerate to maintain a safe distance. Concerning the temporal correlation matrix, it is noteworthy that considering neighboring time steps is significant in modeling human driving behaviors, given that the correlations consistently remain positive. Nevertheless, the correlations diminish as the temporal distances increase — according to the results, the most correlated elements are those located within 1 second before and after the diagonal, which is consistent with the results discovered before [23].

V. CONCLUSION

In the realm of understanding complex human decision-making processes, particularly in car-following (CF) scenarios, the role of generative models is indispensable. These models, e.g., GMM, adept at capturing the joint probability distribution of various inputs and outputs, offer a window into the intricate interplay of causes and effects that define human behavior. This paper introduces a novel approach to modeling human driving behaviors in CF situations through the development of a Bayesian Matrix Normal Mixture Model (MNMM). Our model stands out by employing a Kronecker structure within the covariance matrix, a technique that effectively decouples feature correlations from temporal dynamics. This unique attribute allows for a nuanced representation of the interdependencies inherent in the driving process.

Building on the foundational principles of MNMM, we extend our methodology to encompass Matrix Normal Mixture Regression (MNMR). This progression enables the prediction of future driving actions based on past observations, mirroring the decision-making patterns of human drivers who rely on recently acquired information. Our comprehensive evaluation, conducted using the HighD dataset, demonstrates the model's exceptional capability to encapsulate complex driving dynamics. A case study is presented to further illustrate the superior interpretability of our approach.

Our work has broad implications for advancing planning and decision-making algorithms of autonomous driving by modeling and understanding human behaviors, offering insights to enhance safety and efficiency. It not only contributes a novel theoretical model but also paves the way for practical applications in the field of automotive engineering.

REFERENCES

- [1] C. Zhang, R. Chen, J. Zhu, W. Wang, C. Liu, and L. Sun, "Interactive car-following: Matters but not always," *arXiv preprint arXiv:2307.16127*, 2023.
- [2] M. Saifuzzaman and Z. Zheng, "Incorporating human-factors in car-following models: a review of recent developments and research needs," *Transportation research part C: emerging technologies*, vol. 48, pp. 379–403, 2014.
- [3] M. Treiber and A. Kesting, "Microscopic calibration and validation of car-following models—a systematic approach," *Procedia-Social and Behavioral Sciences*, vol. 80, pp. 922–939, 2013.
- [4] M. Zhu, X. Wang, A. Tarko *et al.*, "Modeling car-following behavior on urban expressways in shanghai: A naturalistic driving study," *Transportation research part C: emerging technologies*, vol. 93, pp. 425–445, 2018.
- [5] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation research part B: methodological*, vol. 15, no. 2, pp. 105–111, 1981.
- [6] G. F. Newell, "A simplified car-following theory: a lower order model," *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 195–205, 2002.
- [7] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.
- [8] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transportation research part C: emerging technologies*, vol. 97, pp. 348–368, 2018.
- [9] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang, "Capturing car-following behaviors by deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 910–920, 2017.
- [10] P. Angkititirakul, T. Ryuta, T. Wakita, K. Takeda, C. Miyajima, and T. Suzuki, "Evaluation of driver-behavior models in real-world car-following task," in *2009 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2009, pp. 113–118.
- [11] S. Lefèvre, C. Sun, R. Bajcsy, and C. Laugier, "Comparison of parametric and non-parametric approaches for vehicle speed prediction," in *2014 American Control Conference*. IEEE, 2014, pp. 3494–3499.
- [12] P. Angkititirakul, C. Miyajima, and K. Takeda, "Stochastic mixture modeling of driving behavior during car following," *Journal of information and communication convergence engineering*, vol. 11, no. 2, pp. 95–102, 2013.
- [13] X. Chen, C. Zhang, Z. Cheng, Y. Hou, and L. Sun, "A bayesian gaussian mixture model for probabilistic modeling of car-following behaviors," *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [14] C. Viroli, "Finite mixtures of matrix normal distributions for classifying three-way data," *Statistics and Computing*, vol. 21, no. 4, pp. 511–522, 2011.
- [15] F. Yin, G. Hu, and W. Shen, "Analysis of professional basketball field goal attempts via a bayesian matrix clustering approach," *Journal of Computational and Graphical Statistics*, vol. 32, no. 1, pp. 49–60, 2023.
- [16] W. Wang, J. Xi, and D. Zhao, "Learning and inferring a driver's braking action in car-following scenarios," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3887–3899, 2018.
- [17] D. Lewandowski, D. Kurowicka, and H. Joe, "Generating random correlation matrices based on vines and extended onion method," *Journal of multivariate analysis*, vol. 100, no. 9, pp. 1989–2001, 2009.
- [18] W. Wang, L. Wang, C. Zhang, C. Liu, L. Sun *et al.*, "Social interactions for autonomous driving: A review and perspectives," *Foundations and Trends® in Robotics*, vol. 10, no. 3–4, pp. 198–376, 2022.
- [19] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (ngsim) program data," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1243–1262, 2011.
- [20] C. Zhang, J. Zhu, W. Wang, and J. Xi, "Spatiotemporal learning of multivehicle interaction patterns in lane-change scenarios," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [21] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep Universal Probabilistic Programming," *Journal of Machine Learning Research*, 2018.
- [22] C. J. Willmott, S. G. Ackleson, R. E. Davis, J. J. Feddema, K. M. Klink, D. R. Legates, J. O'donnell, and C. M. Rowe, "Statistics for the evaluation and comparison of models," *Journal of Geophysical Research: Oceans*, vol. 90, no. C5, pp. 8995–9005, 1985.
- [23] C. Zhang and L. Sun, "Bayesian calibration of the intelligent driver model," *IEEE Transactions on Intelligent Transportation Systems*, 2024.