An Empirical Bayes Analysis of Vehicle Trajectory Models

Yue Yao, Daniel Goehring, Joerg Reichardt

Abstract—We present an in-depth empirical analysis of the trade-off between model complexity and representation error in modelling vehicle trajectories. Analyzing several large public datasets, we show that simple linear models do represent real-world trajectories with high fidelity over relevant time scales at very moderate model complexity. This finding allows the formulation of trajectory tracking and prediction as a Bayesian filtering problem. Using an Empirical Bayes approach, we estimate prior distributions over model parameters from the data that inform the motion models necessary in the trajectory tracking problem and that can help regularize prediction models. We argue for the use of linear models in trajectory prediction tasks as their representation error is much smaller than the typical epistemic uncertainty in this task.

I. INTRODUCTION

Safe and comfortable driving necessitates anticipatory planning. Drivers must form expectations about the future development of the traffic scene and adapt their behavior accordingly. Other traffic participants’ (TPs) motion plans are unobservable, and their intents are often not signalled unequivocally, even when required by law. The expectation on the future trajectory of TPs is thus uncertain and must be modelled probabilistically. Past trajectories are equally uncertain, albeit to a lesser degree, as observations are generally noisy or may be missing entirely due to occlusions.

In order to formulate probabilistic statements about traffic scenarios, both past and future trajectories must be considered and modelled as random variables. This naturally leads to the question of how to best represent a trajectory for prediction and tracking applications. In this contribution, we present an empirical investigation into the usefulness of one representation that is mathematically particularly appealing: linear combinations of basis functions.

Our discussion will proceed as follows: We will first summarize some of the theoretical advantages and disadvantages of sequence-based trajectory representations and compare them with classical linear combinations of basis functions. We proceed to investigate the expressive power of this trajectory representation by analyzing a number of publicly available datasets of vehicle trajectories. Specifically, we employ the Empirical Bayes method to estimate prior distributions over model parameters and observation noise. We characterize the trade-off between model complexity, i.e. the number of basis functions used, and data-fit. For this, we employ information theoretic measures and mean absolute spatial error between the trajectory representation and the data. We investigate this trade-off for trajectories of various lengths and determine the optimal model complexities before concluding with a short summary of our findings.

II. RELATED WORK

A. Sequence Models of Trajectories

Trajectory prediction systems must take representations of past object trajectories and the current static environment in a scene as input, and output some representation of a predicted future trajectory over a finite time horizon $T$.

Past information can be input as a sequence of observations at fixed temporal or spatial resolution. One popular approach for items in such an input sequence is rasterization, which renders past object trajectories and the static environment into a stack of images from a “bird’s-eye” view. Rasterizing the environment is powerful and comprehensive in representing arbitrary information as a multi-channel image. However, the memory usage and computational cost need to scale with desired spatial and temporal resolution and the discrete nature of this approach is inadequate in representing continuous physical states.

Alternatively, the input sequence contains object trajectories as lists of kinematic measurements (position, heading, velocity, etc.). The contextual static environment can also be represented in form of lists of static features such as road boundary positions or in form of a single ‘bird’s-eye’ rendering of the scene. The sequence-wise representation is a more compact form compared to rasterization. However, this approach still represents discrete physical states and the memory usage scales with the resolution of this approach.

In symmetry with the sequential nature of the input, many prediction systems employ sequence representation also for the output, e.g., as a sequence of positions or control states (acceleration and turn rate) at fixed time-points over the prediction horizon. The computational requirements of these approaches scale with the length and temporal resolution of the prediction horizon.

Sequence-based approaches are examples of unbiased models that offer great flexibility of expression. They can express even random, erratic trajectories. The downside of unbiasedness of any estimator is the high variance which in turn generally requires larger amounts of training data and the fact that sequence-based representations amalgamate the observation uncertainty and the underlying system behavior to be modelled.

Yue Yao is with Continental AG and Ph.D. candidate at Freie Universität Berlin.
Daniel Goering is Juniorprofessor for Autonomous Vehicles at the Dahlem Center for Machine Learning and Robotics of the Freie Universität Berlin.
Joerg Reichardt is with Continental AG.
B. Linear Models of Trajectories

On the other hand, linear combinations of basis functions limit their expressive power to what the basis functions allow - they are the foundation of compact but biased estimators. They may introduce systematic error (bias) but feature less variance and typically reduce the need for training data. We will turn to a detailed discussion of their properties and merits next.

In contrast to paths, i.e. curves in space, trajectories are curves in space and time. Over any finite time horizon $[t_0, t_0 + T]$, the position of an object $c(\tau) \in \mathbb{R}^d$ at (rescaled) time $\tau = (t - t_0)/T$ can be expressed as a linear combination of $n + 1$ fixed basis functions of time $\phi_k(\tau) : \mathbb{R} \rightarrow \mathbb{R}$ and variable points in space $\omega_k \in \mathbb{R}^d$ parameterizing the curve:

$$c(\tau) = \sum_{k=0}^{n} \phi_k(\tau) \omega_k.$$  \hspace{1cm} (1)

It allows us to incorporate prior knowledge about object trajectories via the basis functions $\phi_k(\tau)$. For example, since traffic participants carry mass and underlie physical constraints on how much force they can exert as well as strong preferences for smooth motion [9]–[11], we can choose a small number $n + 1$ of smooth basis functions and model deviations from smooth behavior as random noise. Authors of [12], [13] all use polynomials of at most degree 4 as basis functions.

In contrast to sequence-based trajectory representations, this formulation is temporally continuous, i.e. provides arbitrary temporal resolution as highlighted by Su et al. [13]. Parameters $\omega_k$ have spatial semantics, i.e. they can be interpreted as points in space and this allows to formulate prior distributions based on spatial information, which is advantageous to regularize and inform predictions of TPs future motion [14]. Without loss of generality, we can constrain the range of $\phi_k(\tau)$ to the interval $[0, 1]$ and this automatically constrains $c(\tau)$ to the convex hull of the $\omega_k$.

Parameters $\omega_k$ transform lossless as spatial points under translation and rotation of the coordinate reference frame, which significantly simplifies applications with moving sensors.

Temporal derivatives only affect the basis functions $\dot{c}(\tau) = \sum_{k=0}^{n} \phi_k(\tau) \dot{\omega}_k$, i.e. positions, velocities, accelerations and higher derivatives share the same parameterization! Spatio-temporal distributions over the kinematic state of an object at any point in time can be readily derived from distributions over the parameters $\omega_k$.

We may estimate the parameters $\omega_k$ via the solution of a linear system from either measurements of position at different time-points or measurements of different derivatives at the same time-point, or combinations thereof. For example, with $n = 5$ and $\phi_k(\tau) = \tau^k$, we only need position, velocity and acceleration at two time-points $\tau$ and $\tau + \Delta \tau$ in order to fully determine all 6 trajectory parameters $\omega_k$.

The linear combination of basis functions enables the drop-in replacement of kinematic object states by trajectories over constant time horizons in tracking and filtering applications at minimal computational overhead. It even allows the formulation of the trajectory prediction problem as a filtering problem [15]. In these applications, the trajectory parameters $\omega_k$ replace the conventionally used kinematic state variables for tracking, filtering and prediction. The motion models in such applications can then be informed and regularized by prior distributions over $\omega_k$.

However, in order to reap the benefits of these mathematical merits in practical applications, we must show that linear combinations of a few smooth basis functions can indeed represent real-world trajectories with high fidelity.

Our key contributions are thus as follows: we provide a methodology to i) estimate the optimal model complexity $\hat{n}$ from noisy data, ii) estimate prior distributions over observation noise and model parameters $\omega_k$ to regularize model fits and trajectory predictions, and iii) provide an extensive empirical investigation of the representation error of this trajectory formulation for the case of polynomial basis functions.

Note that this choice of basis functions is quite general but not unique and only illustrates the general method of estimating prior distributions and model complexity.

III. ESTIMATION OF REPRESENTATION ERROR

Datasets for training and evaluating trajectory prediction methods such as Argoverse Motion Forecasting v1.1 (A1) [16], Argoverse 2 Motion Forecasting (A2) [17] and Waymo Open Motion (WO) [13] are obtained from a moving sensor platform (ego vehicle) during measurement campaigns. Object detections are tracked and transformed into a world coordinate system and reported in the dataset. Each dataset selects one or multiple objects of interest in one scenario and refers to them as agents. Rather than ground truth, object positions and kinematics in the data represent noisy estimates. Hence, when fitting models to the data in order to measure the approximation error, we need to regularize and take the observation noise into account explicitly, i.e. we need to perform a Bayesian regression. The kinematic variables provided in datasets vary, but all provide position measurements for object center points, which is what we focus on here. Figure 1a shows a typical example.

### Table I

| Dataset | A1 | A2 | WO |
|---------|----|----|----|
| #scenarios, #ego trajectories | 206K | 200K | 487K |
| #agent (vehicle) trajectories | 206K | 176K | 1.84M |
| maximal time horizon [s] | 5 | 11 | 9 |
| #cities | 2 | 6 | 6 |
| sampling rate [Hz] | 10 |

| trajectory information | position | velocity | orientation | timestamp |
|------------------------|----------|----------|-------------|-----------|
|                         | 2D       | -        | ✓           | ✓         |
|                         | 2D       | 2D       | ✓           | ✓         |

The astute reader will recognize the choice of Bernstein Polynomials for $\phi_k(\tau)$ leads to the familiar Bezier curves. In fact, all formulations with polynomial basis functions of degrees $0, ..., n$ are equivalent and can be interchanged via a fixed linear transformation of the model parameters $\omega_k$. 
Then we can express a Gaussian prior over parameters $\omega$ and subsequently transformed into a fixed world coordinate frame. As distance and angle between sensor and agent change during recording, the observation covariance of agent locations stretches and rotates over time. We show all sample points and a few 95% confidence ellipses for agent position, enlarged by a factor of 4 for better visibility. The resulting posterior covariances for agent positions are also shown, enlarged by a factor of 8 for better visibility.

In order to formulate the regression, we will introduce some definitions for notational convenience. We form a vector of basis functions $\phi$ that allows to express correlations between spatial dimensions. We form a parameter vector $\omega$ which consists of $n$ parameters and model complexity $n$ - neither of which is given. We now employ the Empirical Bayes Method to estimate all three quantities.

We assume zero mean Gaussian observation noise with $d \times d$ covariance matrix $\Sigma_{o,i,j}$ depending on the object index $i$ and the sample point $j$. Assuming statistically independent noise along a single trajectory, we form the observation noise covariance for $c_i$ as a $md \times md$ block diagonal matrix $\Sigma_{o,i}$ where the $j$th diagonal block is given by $\Sigma_{o,i,j}$. Figure 1a illustrates the time and trajectory dependence of the noise covariance.

Using this notation, the posterior estimate of model parameters for a single trajectory $c_i$ is given in closed form [19] pp. 232–234):

$$\Sigma_{\omega,i} = (\Sigma_{\omega}^{-1} + \Phi_i \Sigma_{o,i}^{-1} \Phi_i^T)^{-1}$$

$$\omega_i^\text{post} = \Sigma_{\omega,i}^{-1} \Phi_i \Sigma_{o,i}^{-1} c_i$$

(2)

With this improved parameter estimate from [2], we can then compute the representation error by measuring the average distance error (ADE) along each trajectory for every measurement:

$$\text{ADE} = \frac{1}{Nm} \sum_i \sum_j ||(\Phi_i^T \tau_{i,j} \otimes I_d) \omega_i^\text{post} - c_i^\text{ob} ||_2$$

(3)

Since the projection of the ADE onto the longitudinal and lateral direction of motion are of particular interest for driving applications, we also calculate these projections and denote them as $\text{ADE}_{\text{long}}$ and $\text{ADE}_{\text{lat}}$. The object heading is either provided by the measurement (A2 and WO) or is inferred from a Rauch-Tung-Striebel (RTS) smoothing of the data (A1, cf. Section V-A).

The above estimation of representation error requires the specification of observation covariance $\Sigma_o$, prior covariance $\Sigma_{\omega}$ and model complexity $n$ - neither of which is given. We now employ the Empirical Bayes Method to estimate all three quantities.

IV. Estimation of Observation Noise, Prior Parameters and Model Complexity via Empirical Bayes

The Empirical Bayes approach [20] allows us to bootstrap prior distributions over model parameters if many independent samples of the same phenomenon are observed, such as the object trajectories in our datasets. The idea is to formulate the likelihood of all observed trajectories $C$ as a function of the prior parameters alone. This can be achieved by marginalizing the actual model parameters. Optimal prior parameters maximize the resulting, so-called type-II, likelihood.

In an ideal world, the only prior parameter to be estimated would be the covariance matrix $\Sigma_o$. The observation noise covariance matrices $\Sigma_{o,i,j}$ would be derived from the ego vehicle’s sensor setup and given with the dataset. Unfortu-
nately, the $\Sigma_{o,i,j}$ are not provided in any dataset and so we have to reverse engineer, i.e. estimate, them from the data.

Clearly, we cannot estimate individual $\Sigma_{o,i,j}$ for every trajectory and every time-point. Instead, we provide a structured parameterization in the form $\Sigma_{o,i,j} = \Sigma_{\Theta}(\vartheta)$. Since ego trajectories and agent trajectories result from different sensor setups, we also differentiate the parameterization of their noise models.

A. Observation Noise Covariance for Ego Trajectories

We model the observation noise covariance for ego trajectories in world coordinates, assuming constant observation noise in $x$ and $y$ direction for all sample points. The covariance for one measurement point is expressed as:

$$\Sigma_{ego,world} = \Sigma_{\Theta} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$  (4)

with $\sigma_x^2 = \sigma_x^2$, $\sigma_y^2 = \sigma_y^2$, and $\sigma_{xy} = \sigma_{xy}$. We can extend it to the covariance of one complete ego trajectory via $\Sigma_{ego}(\vartheta_{ego}) = I_m \otimes \Sigma_{ego,world}$ and thus for all ego trajectories, we only have two parameters to estimate $\vartheta_{ego} = [\sigma_{diag}, \sigma_{cov}]$.

B. Observation Noise Covariance for Agent Trajectories

Since the ego vehicles used in the datasets feature a suite of LIDAR sensors, the observation uncertainty for agents is best expressed in polar coordinates. We assume a constant angular resolution ($\sigma_\omega$) and a variable distance resolution ($\sigma_{r,i,j}$):

$$\Sigma_{agent,polar} = \begin{bmatrix} \sigma_{r,i,j}^2 & 0 \\ 0 & \sigma_{\omega}^2 \end{bmatrix}$$  (5)

We model $\sigma_{r,i,j}$ as an increasing function of the measured distance $r_{i,j}$ between agent and ego of the $j$th sample point $(j \in [1,2,...,m])$ as well as the parameters $[\beta_0, \beta_1, \beta_2] \in \mathbb{R}^+$:

$$\sigma_{r,i,j} = \beta_0 + \beta_1 r_{i,j} + \beta_2 r_{i,j}^2$$  (6)

Clearly, this is not perfect since we are using measurements to parameterize the uncertainty of these very measurements. But it is the best approach we have and it is safe to assume $\sigma_{r,i,j}$ varies only slightly within the uncertainty of $r_{i,j}$. Next, we transform the observation covariance from the polar frame to the Cartesian ego frame $\Sigma_{agent,ego}$ based on [21, p. 77].

Finally, we rotate $\Sigma_{agent,ego}$ from the ego frame to the world frame with:

$$\Sigma_{agent,world} = R_{i,j}(\Sigma_{agent,ego} + \sigma^2 I_d)R_{i,j}^T$$  (7)

C. Estimating Prior Parameters via Empirical Bayes

Finally, we are in the position to formulate the type-II likelihood. Following [19] pp. 172–176, we obtain:

$$p(C|\Sigma_{\Theta}(\vartheta), \Sigma_{\omega}) = \prod_{i=1}^{N} \int \mathcal{N}(c_i^{|\Theta}| \Phi_i^T \omega, \Sigma_{o,i}(\vartheta)) \times \mathcal{N}(\omega|0, \Sigma_{\omega})d\omega$$

(8)

where $C$ denotes all $N$ trajectories in the dataset. We maximize the log of $p(C|\Sigma_{\Theta}(\vartheta), \Sigma_{\omega})$ with respect to $\Sigma_{\omega}$ and $\Theta$ for ego and agent trajectories separately using gradient descent method [22]. These optima represent the prior parameters estimated from the dataset:

$$\hat{\vartheta}, \Sigma_{\omega} = \arg\max_{\vartheta, \Sigma_{\omega}} \log(p(C|\Sigma_{\Theta}(\vartheta), \Sigma_{\omega}))$$  (9)

For any model complexity $n$, we can thus estimate Bayesian optimal trajectory representations from given data by plugging the estimated prior parameters $\Sigma_{\omega}$ and $\Sigma_{o,i}(\vartheta)$ into (7).

With increasing $n$ we will be able to achieve smaller ADE and estimate lower observation noise at the expense of an increasing number of parameters, i.e. we will start to overfit. We next find the optimal trade-off between data fit and model complexity.

D. Estimating Optimal Model Complexity

The Akaike Information Criterion (AIC) [23] and Bayesian Information Criterion (BIC) [24] characterize the score for a model in terms of how well it fits the data, minus how complex the model is to define. AIC and BIC are defined as:

$$\text{AIC} = \frac{\log(p(C|\Sigma_{\Theta}(\vartheta), \Sigma_{\omega})))}{N} - \text{dof}(\theta, \Sigma_{\omega})$$

$$\text{BIC} = \frac{\log(p(C|\Sigma_{\Theta}(\vartheta), \Sigma_{\omega})))}{N} - \frac{\text{dof}(\theta, \Sigma_{\omega})}{2} \log(m)$$

(10)

where $\text{dof}(\theta, \Sigma_{\omega}) = \text{dof}(\theta) + (dn+d)/(dn+d+1)/2$ denotes the degrees of freedom in the observation covariance and model parameter covariance. For ego and agent trajectories, $\text{dof}(\vartheta_{ego}) = 2$ and $\text{dof}(\theta_{agent}) = 5$, respectively. A maximum of either criterion as a function of $n$ indicates optimal trade-off between data-fit and model complexity. In general, BIC penalizes model complexity higher and tends to pick a simpler model.

V. Experiments

A. Preprocessing: Data Selection and Outlier Detection

Dataset characteristics are summarized in Table I. For comparability, we limit our analysis to vehicle trajectories as this agent class is present in all three datasets A1, A2 and WO. We analyze trajectories for time horizons $T \in [3s, 5s, 8s]$. For $T$ smaller than the maximal observation horizon in the dataset, we select time windows of size $T$ in strides of 1s for A1 and randomly, one from each
trajectory, for A2 and WO. From the much larger set of agent trajectories in WO, we limit our analysis to a random sample of 300k agent trajectories (without outlier) to reduce computational cost.

We consider all samples that are within sensor range of the ego vehicle for $T - 0.5s$ to $T + 0.5s$ in the analysis of time horizon $T$ trajectories but discard all static trajectories with lengths $\leq 0.5m$ as these can be trivially represented with small error.

We notice a number of outliers in the data due to tracking loss and inconsistent timing, i.e. objects are reported at physically impossible locations for the given timestamps. To automatically detect and discard such trajectories, we employ a Rauch-Tung-Striebel (RTS) smoothing of the data with a simple double integrator based on [25, p. 48] and adjust the parameters according to [25, p. 59]. We discard trajectories for which the RTS-Smoother estimates positions more than 2m away from measurement or the estimated longitudinal acceleration (deceleration) exceeds a $\frac{6\sqrt{m}}{s}$ ($-10\frac{m}{s^2}$) threshold. Table II gives an overview of the percentage of outliers in datasets discarded for trajectories of time horizon $T = 5s$.

We notice the timing issue in A1, where $T$ varies from 4.81s to 25.64s for 50 sample points with 10Hz [26]. The RTS-Smoother detects ego outliers in A2 due to the unstable velocity estimation at the trajectory’s start or end. In WO, we find quantities of outliers detected by RTS-Smoother primarily because the agents leave the sensor range and their positions reset to $(0,0)$.

### Table II

| Datasets | A1 | A2 | WO |
|----------|----|----|----|
| time     | 22.81 | 0 | 0.02 |
| static   | 25.95 | 20.66 | 25.41 |
| out of view | 0 | 0 | 0 |
| RTS      | 0 | 1.18 | 0 |
| total    | 42.95 | 21.84 | 25.42 |
| static   | 22.81 | 0 | 0.02 |
| out of view | 0 | 4.55 | 1.70 |
| RTS      | 6.81 | 0.86 | 19.45 |
| total    | 28.11 | 5.81 | 20.83 |

### Results

1) Estimation of Observation Noise: Table III reports results for $T = 5s$ at the value of $n = \hat{n}$ that maximizes AIC. As expected, ego trajectories exhibit much lower observation noise than agent trajectories in all datasets. The agent trajectories in A1 are significantly noisier than in A2 and WO. WO provides data with the least estimated observation noise due to its offline tracking algorithm [18]. As model misspecification is upper bounded by the estimated observation noise, the very low estimated observation noise indicates the high representation quality.

2) Model Complexity and Representation Error: Figure 2 shows Box-plots for the longitudinal and lateral ADE for agent trajectory samples with $T \in [3s, 5s, 8s]$. We indicate the best model complexity according to AIC and BIC.

Table IV gives the numerical results for both ego and agent trajectories at the model complexity $n = \hat{n}$ that maximizes AIC.

### Table III

| Datasets | A1 | A2 | WO |
|----------|----|----|----|
| time     | 0.024 | 0.012 | 0.008 |
| static   | 2e-4 | 3e-6 | -1e-7 |
| out of view | 0.161 | 0.044 | 0.017 |
| RTS      | 6e-4 | 3e-4 | -1e-7 |
| total    | 0.176 | 0.062 | 0.027 |

Figure 2 and Table IV clearly show the that longer trajectories warrant higher model complexities for their representation, but also that the benefits of higher $n$ are diminishing beyond an optimal model $\hat{n}$ as indicated by AIC or BIC. Overall, we see how simple linear models of moderate complexity can represent trajectories with very high fidelity. E.g., a 6th degree polynomial can approximate the 8-seconds agent trajectories with 3.7cm longitudinal and 1.6cm lateral ADE in WO. However, we also observe large deviations ($>1m$). Inspecting these samples, we find they correspond to physically implausible measurements due to timing jitter or tracking loss that are not excluded by the RTS-Smoother.

### Table IV

| Dataset | A1 | A2 | WO |
|---------|----|----|----|
| time    | 0.004 | 0.001 | 0.004 |
| static  | 0.094 | 0.012 | 0.066 |
| out of view | 0.143 | 0.030 | 0.071 |
| RTS     | 0.071 | 0.025 | 0.025 |
| total   | 0.005 | 0.002 | 0.002 |
| time    | 0.005 | 0.002 | 0.004 |
| static  | 0.114 | 0.009 | 0.014 |
| out of view | 0.071 | 0.030 | 0.069 |
| RTS     | 0.071 | 0.025 | 0.025 |
| total   | 0.004 | 0.002 | 0.002 |
| time    | 0.016 | 0.005 | 0.016 |
| static  | 0.035 | 0.014 | 0.039 |
| out of view | 0.051 | 0.016 | 0.056 |
| RTS     | 0.035 | 0.016 | 0.056 |
| total   | 0.016 | 0.005 | 0.005 |

$\hat{n}$ denotes the best polynomial degree according to AIC. 99.9% means the 99.9 percentile of the representation error.

3) Representation Error vs. Prediction Error: Let us compare the price we pay for the bias introduced by our basis functions to the associated benefits in computational efficiency.

Figure 3 compares the representation error of our linear trajectory models to the prediction error over future trajectories of state-of-the-art unbiased sequence based prediction methods. We compare to $\min ADE_{\theta}$, i.e. the minimum average distance error over top $k$ most-likely predicted trajecto-
Fig. 2. The longitudinal (left) and lateral (right) distance error of models for agent trajectories with $T \in [3s, 5s, 8s]$. "a, b" denote the model complexity $n = \hat{n}$ that maximizes AIC and BIC, respectively. The upper whisker denotes the 99.9% percentile.

Fig. 3. The representation $ADE_k$ of polynomials at $\hat{n}$ maximizing AIC and the prediction $minADE_k$ ($k = 6$) of Wayformer[6] and MultiPath++[8] for agent trajectories with $T \in [3s, 5s, 8s]$. Latest prediction results can be found at [27], [28].

VI. CONCLUSION

Vehicle trajectories can be modelled with high fidelity over timescales relevant for trajectory prediction by simple linear combinations of basis functions. We have characterized the trade-off between model complexity and representation error by an empirical analysis of several large public datasets. Using an Empirical Bayes approach, we have estimated models for observation noise and prior distributions over model parameters. The estimated observation noise parameters can (and should) be considered when training trajectory prediction models with a Gaussian Log-Likelihood loss, particularly when combining different datasets. The prior parameters can inform the motion models of trajectory tracking and filtering models [15] or regularize trajectory prediction models. The representation error of a linear formulation is small compared to prediction errors of current state-of-the-art models. This suggests that the bias inherent in linear models is much smaller than the epistemic uncertainty in the prediction task. The computational benefits, on the other hand, are large. Using linear models of vehicle trajectories, the problem of predicting a trajectory over a finite time horizon $T$ can reduce to predicting the kinematic end state at $t + T$, alone.

ACKNOWLEDGMENT

Authors would like to thank Andreas Philipp for many useful discussions and acknowledge funding from the German Federal Ministry for Economic Affairs and Climate Action within the project “KI Wissen – Automotive AI powered by Knowledge”.

ries. Naturally, any error is larger than the true measurement noise level. If the latter is low, as in WO, the inherent bias of a linear formulation is negligible in the prediction task. It is still significantly lower than the $minADE_k$ of predictions in datasets, where the noise level is likely higher such as A1.

On the other hand, we deem the computational benefits of linear models large. For $n = 5$, a trajectory is specified completely by six kinematic constraints (position, velocity and acceleration) at start and end points. Since three kinematic constraints at the start can be estimated entirely from past observation, the prediction problem is in fact reduced to predicting the kinematics at the end point. Hence, 50% of the accuracy of trajectory prediction is due to exact tracking of initial kinematics and 50% is due to accurate prediction of kinematics at the end of the prediction horizon, only.
REFERENCES

[1] M. Bansal, A. Krizhevsky, and A. Ogale, “Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst,” arXiv preprint arXiv:1812.03079, 2018.

[2] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, “Multimodal trajectory predictions for autonomous driving using deep convolutional networks,” in 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 2090–2096.

[3] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, “Vectornet: Encoding hd maps and agent dynamics from vectorized representation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11525–11533.

[4] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Biejbom, and E. M. Wolff, “Governet: Multimodal behavior prediction using trajectory sets,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14074–14083.

[5] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, “Learning lane graph representations for motion forecasting,” in European Conference on Computer Vision, Springer, 2020, pp. 541–556.

[6] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, “Wayformer: Motion forecasting via simple & efficient attention networks,” arXiv preprint arXiv:2207.05844, 2022.

[7] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data,” in Computer Vision – ECCV 2020, 2020, pp. 683–700.

[8] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, et al., “Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction,” in 2022 International Conference on Robotics and Automation (ICRA), IEEE, 2022, pp. 7814–7821.

[9] C. C. Macadam, “Understanding and modeling the human driver,” Vehicle system dynamics, vol. 40, no. 1-3, pp. 101–134, 2003.

[10] H. Hayati, D. Eager, A.-M. Pendrill, and H. Alberg, “Jerk within the context of science and engineering—a systematic review,” Vibration, vol. 3, no. 4, pp. 371–409, 2020.

[11] I. Bae, J. Moon, J. Jung, H. Suk, T. Kim, H. Park, J. Cha, J. Kim, D. Kim, and S. Kim, “Self-driving like a human driver instead of a robocar: Personalized comfortable driving experience for autonomous vehicles,” arXiv preprint arXiv:2001.03908, 2020.

[12] T. Buhet, E. Wirbel, A. Bursuc, and X. Perrotton, “Plop: Probabilistic polynomial objects trajectory planning for autonomous driving,” arXiv preprint arXiv:2003.08744, 2020.

[13] Z. Su, C. Wang, H. Cui, N. Djuric, C. Vallespí-Gonzalez, and D. Bradley, “Temporally-continuous probabilistic prediction using polynomial trajectory parameterization,” in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2021, pp. 3837–3843.

[14] M. Bahari, S. Saadatnejad, A. Rahimi, M. Shaverdikondori, A. H. Shahidzadeh, S.-M. Moosavidezfouli, and A. Alahi, “Vehicle trajectory prediction works, but not everywhere,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17 123–17 133.

[15] J. Reichardt, “Trajectories as markov-states for long term traffic scene prediction,” in 14-th UniDAS FAS-Workshop, Berkheim, Germany, 2022, p. 14.

[16] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, “Argoverse: 3d tracking and forecasting with rich maps,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[17] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, et al., “Argoverse 2: Next generation datasets for self-driving perception and forecasting,” in Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.

[18] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, et al., “Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9710–9719.

[19] K. P. Murphy, Machine learning: a probabilistic perspective. MIT press, 2012.

[20] B. Efron, Large-scale inference: empirical Bayes methods for estimation, testing, and prediction. Cambridge University Press, 2012, vol. 1.

[21] N. Kämpchen, “Feature-level fusion of laser scanner and video data for advanced driver assistance systems, Ph.D. dissertation, Universität Ulm, 2007.

[22] J. V. Dillon, I. Langmore, D. Tran, E. Brevedo, S. Vatsudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, “Tensorflow distributions,” arXiv preprint arXiv:1711.10604, 2017.

[23] H. Akaike, “Information theory and an extension of the maximum likelihood principle,” in Proceedings of the 2nd international symposium on information, bn petrov, i,” Czaki, Akademiai Kiado, Budapest, 1973.

[24] G. Schwarz, “Estimating the dimension of a model,” The annals of statistics, pp. 461–464, 1978.

[25] A. Philipp, “Perception and prediction of urban traffic scenarios for autonomous driving,” Ph.D. dissertation, Freie Universitaet Berlin (Germany), 2021.
[26] https://github.com/argoai/argoverse-api/issues/124
[27] https://eval.ai/web/challenges/challenge-page/454/leaderboard/1279
[28] https://waymo.com/open/challenges/2022/motion-prediction