

An Empirical Bayes Analysis of Object Trajectory Representation Models

Yue Yao, Daniel Goehring, Joerg Reichardt

Abstract—Linear trajectory models provide mathematical advantages to autonomous driving applications such as motion prediction. However, linear models’ expressive power and bias for real-world trajectories have not been thoroughly analyzed. We present an in-depth empirical analysis of the trade-off between model complexity and fit error in modelling object trajectories. We analyze vehicle, cyclist, and pedestrian trajectories. Our methodology estimates observation noise and prior distributions over model parameters from several large-scale datasets. Incorporating these priors can then regularize prediction models. Our results show that linear models do represent real-world trajectories with high fidelity at very moderate model complexity. This suggests the feasibility of using linear trajectory models in future motion prediction systems with inherent mathematical advantages.

I. INTRODUCTION

The prediction of other traffic participants’ future trajectories is an important input to the motion planning systems of autonomous agents. Recently, a number of large datasets of real-world traffic scenarios, e.g. Argoverse Motion Forecasting v1.1 (A1) [1], Argoverse 2 Motion Forecasting (A2) [2] and Waymo Open Motion (WO) [3], have been made available and corresponding prediction challenges have sparked a large interest in designing trajectory prediction systems.

As one of the most widely used criteria of prediction accuracy in these challenges, the displacement error measures the difference between the predicted object positions and the *noisy observations* of the object positions over the prediction horizon.

Figure 1 shows how we can decompose the total displacement error produced by any prediction system into three independent components:

- *Observation noise*: Deviation between ground truth and observations that is intrinsic to the data(set) and its collection process.
- *Representation error*: Deviation between ground truth and predictions due to a trajectory modelling choice.
- *Prediction error*: Deviation between ground truth and predictions due to the inadequacies of the prediction system and missing input information such as agents’ intentions.

We denote *fit error* as the sum of representation error and observation noise to describe how much a trajectory representation model deviates from observations.

Yue Yao is with Continental AG and Ph.D. candidate at Freie Universität Berlin.

Daniel Goehring 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.

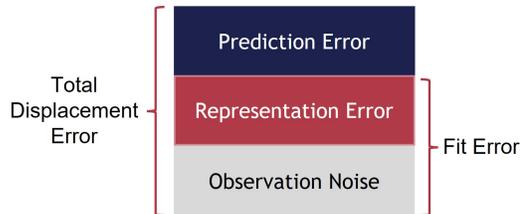


Fig. 1. The decomposition of the total displacement error from prediction system.

Linear trajectory models, i.e. linear combination of basis functions, such as polynomials, are one of the model-based approaches for representing predicted trajectories in recent works [4], [5]. Linear models bring many advantages (cf. Section III) into the prediction task and real applications [6]. As an example, instead of predicting future states sequentially, the prediction task can be simplified as predicting the polynomial parameters of trajectories. However, linear models have limited expressive power depending on model complexity and introduce representation error (bias) into the total displacement error.

Though linear trajectory models are used in recent works, the following questions have not been studied extensively:

- What is the optimal model complexity for representing trajectories with different timescales?
- How much representation error is introduced by linear models of trajectories?

The public datasets provide noisy observations rather than ground truth, thus we will answer the second question by analyzing the fit error as the upper bound of representation error as visualized in Figure 1.

In this work, we study the general class of linear models with polynomial basis functions and investigate the trade-off between model complexity, i.e., flexibility, and model fit in a principled manner and on a large scale. Our focus is the general characterization of linear trajectory models *independent* of any particular prediction method. Hence, we deliberately do not analyze the performance difference of any prediction models integrating linear trajectory models. This avoids conflating representation error and prediction error in our work. Our key contributions are as follows:

- An extensive empirical investigation of the fit error of polynomial trajectory representations for different classes of traffic participants in public datasets.
- Estimation of prior distributions over observation noise and model parameters using the Empirical Bayes method.
- Estimation of the optimal model complexity from noisy trajectory data.

To the best of our knowledge, this work is the first that estimates noise levels in trajectory prediction datasets in a grounded manner.

Our discussion will proceed as follows: We will first summarize some of the theoretical advantages of polynomial representations. We introduce 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 average fit error between the trajectory representation and the data. In three large public datasets and for three different classes of traffic participants, we investigate this trade-off for trajectories of various lengths and determine the optimal model complexities. Compared to the estimated total displacement error reported for state-of-the-art prediction systems, we show that the fit error is indeed small. From this, we conclude that model-based representations currently do not limit prediction performance and argue for the feasibility of their use in prediction systems.

We release our demo for analyzing WO in [7].

II. RELATED WORK



Fig. 2. An example of expression flexibility of sequence-based representations and model-based representations.

Many prediction systems employ sequence-based representation for the output, e.g., as a sequence of positions [8]–[10] or control states (acceleration and turn rate) [11], [12] at fixed time-points over the prediction horizon. Sequence-based approaches appear, at first glance, as very rational trajectory representations, having the same format as observations in datasets. Sequence-based approaches are examples of unbiased representations that have no representation error by design. They can express even random, erratic trajectories as in Figure 2. The downside of unbiasedness for any estimator is the high variance which typically requires more training data and the fact that sequence-based representations amalgamate the observation uncertainty and the actual motion of the objects being modelled. Additionally, the computational requirements of these approaches scale with the length and temporal resolution of the prediction horizon.

On the other hand, parametric, or model-based, trajectory representations are used less as output representations in recent works. Scholler et al. [13] for example argue that predicting pedestrian trajectories as straight lines, i.e. polynomials of degree 1, yields competitive prediction performance. Vehicle trajectories are represented as polynomials of at most degree 4 in [4], [5]. Su et al. [5] highlight the temporal continuity, i.e. the ability to provide arbitrary temporal resolution, of this representation. Reichardt [6] argues for

the use of polynomial representations to integrate trajectory tracking and prediction into a filtering problem. Model-based trajectory representations place restrictions on the kind of trajectories that can be predicted and introduce bias into prediction systems. This limited flexibility is generally associated with lower demands for training data and greater computational efficiency. The deliberate choice of a model, its parameters and associated prior distributions can directly encode knowledge about the system behavior that serves as a regularizer for prediction models.

III. LINEAR MODELS OF TRAJECTORIES

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 $\mathbf{c}(\tau) \in \mathbb{R}^d$ at (rescaled) time $\tau = \frac{t-t_0}{T}$ can be expressed as a linear combination of $n + 1$ *fixed* basis functions of time $\phi_k(\tau) : \mathbb{R} \rightarrow \mathbb{R}$ and parameters $\omega_k \in \mathbb{R}^d$:

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

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 having strong preferences for smooth motion [14]–[16], we can choose a small number $n + 1$ of smooth basis functions and model deviations from smooth behavior as random noise.

We now discuss the benefits of linear models for representing trajectories:

- 1) Parameters ω_k have spatial semantics. Taking the *Bernstein* polynomial as an example, the curve \mathbf{c} is the Bezier curve and the parameters ω_k are control points. This allows the formulation of prior distributions based on spatial information, which is advantageous to regularize and inform predictions of traffic participants' future motion [17]. Without loss of generality, we can constrain the range of $\phi_k(\tau)$ to the interval $[0, 1]$ and this automatically constrains $\mathbf{c}(\tau)$ to the convex hull of the ω_k .
- 2) The transformation of $\mathbf{c}(\tau)$ is equivalent to applying the same transformation to the ω_k and vice versa, which significantly simplifies applications with moving sensors.
- 3) Temporal derivatives only affect the basis functions $\dot{\mathbf{c}}(\tau) = \sum_{k=0}^n \dot{\phi}_k(\tau) \omega_k$, i.e. positions, velocities, accelerations and higher derivatives share the same parameterization!
- 4) Spatio-temporal distributions over the kinematic state of an object at any point in time can be readily derived from distributions over the parameters ω_k .
- 5) We may estimate the parameters ω_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 $n = 5$ we only need position,

velocity and acceleration at two time-points τ and $\tau + \Delta\tau$ as 6 constraints in order to fully determine all 6 trajectory parameters ω_k .

- 6) The linear combination of basis functions enables a common trajectory representation in tracking, filtering and prediction applications, e.g. parameters of the past trajectory can be tracked and parameters of the future trajectory will be predicted. It even allows the formulation of the trajectory prediction problem as a filtering problem [6]. In these applications, the trajectory parameters ω_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 ω_k .

IV. ESTIMATION OF FIT ERROR

We summarized the characteristics of 3 large-scale datasets in Table I. Datasets for training and evaluating trajectory prediction methods 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 trajectory models to the data in order to measure the fit 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 centroids, which is what we focus on here. Figure 3 shows a typical example.

TABLE I
DATASETS CHARACTERISTICS

Dataset (training split)	A1 [1]	A2 [2]	WO [3]
#scenarios, #ego trajectories	206K	200K	487K
#agent (vehicle) trajectories	206K	176K	1.84M
#agent (cyclist) trajectories	-	3K	63K
#agent (pedestrian) trajectories	-	14K	232K
maximal time horizon [s]	5	11	9
#cities	2	6	6
sampling rate	10 Hz		
trajectory	position	2D	3D
	velocity	-	2D
information	orientation	-	✓
	timestamp	✓	✓

In order to formulate the regression, we will introduce some definitions for notational convenience. We form a parameter vector $\omega \in \mathbb{R}^{(n+1)d}$ as $\omega = [\omega_0, \dots, \omega_n]^\top$. Then we can express a Gaussian prior over parameters $p(\omega) = \mathcal{N}(\omega|0, \Sigma_\omega)$. The prior has zero mean due to symmetry and a full covariance Σ_ω that allows expressing correlations between spatial dimensions. We form a vector of basis functions $\phi(\tau) \in \mathbb{R}^{n+1}$ by concatenation $\phi(\tau) = [\phi_0(\tau), \dots, \phi_n(\tau)]^\top$. In this notation, equation (1) is then written as $\mathbf{c}(\tau) = (\phi^\top(\tau) \otimes \mathbf{I}_d)\omega$, where \mathbf{I}_d is a $d \times d$ identity matrix and the Kronecker product \otimes distributes the

basis functions over the d spatial dimensions. Data for a single object i consists of m sample points at discrete time-points $\tau_{i,1}, \dots, \tau_{i,m}$. We can arrange all modelled trajectory points of the object i into a vector $\mathbf{c}_i \in \mathbb{R}^{md}$ as $\mathbf{c}_i = [\mathbf{c}_i(\tau_{i,1}), \dots, \mathbf{c}_i(\tau_{i,m})]^\top$ noting that the sample times, in general, are different for different objects. By introducing the $(n+1)d \times md$ matrix $\Phi_i = [\phi(\tau_{i,1}) \otimes \mathbf{I}_d, \dots, \phi(\tau_{i,m}) \otimes \mathbf{I}_d]$ we can write most compactly $\mathbf{c}_i = \Phi_i^\top \omega$.

We denote the observed trajectory as \mathbf{c}_i^{ob} for the i^{th} object and $\mathbf{c}_{i,j}^{\text{ob}}$ for j^{th} sample point to distinguish them from our model notation. We assume additive 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 : $\mathbf{c}_{i,j}^{\text{ob}} = \mathbf{c}_{i,j} + \eta$, $\eta \sim \mathcal{N}(0, \Sigma_{o,i,j})$

Assuming statistically independent noise along a single trajectory, we form the observation noise covariance for \mathbf{c}_i as an $md \times md$ block diagonal matrix $\Sigma_{o,i}$ where the j^{th} diagonal block is given by $\Sigma_{o,i,j}$. The left part of Figure 3 illustrates the time and trajectory dependence of the observation noise. Note how it rotates and scales due to the angular and distance dependence of the observation noise to the sensors on the ego vehicle.

Using this notation, the posterior estimate of model parameters for a single trajectory \mathbf{c}_i is given in closed form [18, pp. 232–234]:

$$\begin{aligned} \Sigma_{\omega,i}^{\text{post}} &= (\Sigma_\omega^{-1} + \Phi_i \Sigma_{o,i}^{-1} \Phi_i^\top)^{-1} \\ \omega_i^{\text{post}} &= \Sigma_{\omega,i}^{\text{post}} \Phi_i \Sigma_{o,i}^{-1} \mathbf{c}_i^{\text{ob}} \end{aligned} \quad (2)$$

With this improved parameter estimate from (2), we can then compute the *average fit error* (AFE) along each trajectory in an entire dataset:

$$\text{AFE} = \frac{1}{Nm} \sum_i^N \sum_j^m \|(\phi^\top(\tau_{i,j}) \otimes \mathbf{I}_d)\omega_i^{\text{post}} - \mathbf{c}_{i,j}^{\text{ob}}\|_2 \quad (3)$$

For vehicles, the projection of the AFE onto the longitudinal and lateral direction of motion are of particular interest for driving applications. We denote these projections as AFE_{lon} and AFE_{lat} . The object heading is either provided directly in the data (A2 and WO) or is inferred from a Rauch-Tung-Striebel (RTS) smoothing of the data (A1, cf. Section VI-A).

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

V. EMPIRICAL BAYES METHOD

The Empirical Bayes approach [19] 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 \mathbf{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.

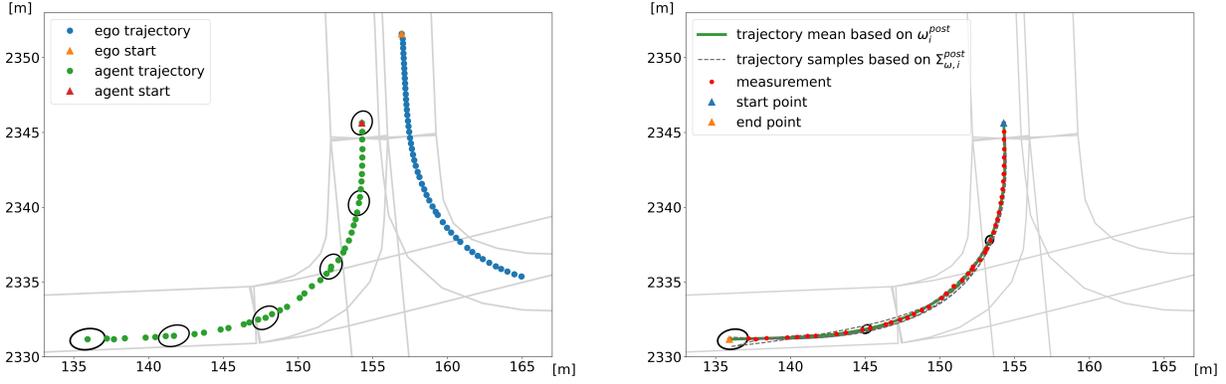


Fig. 3. **Left:** A typical scene from a trajectory prediction dataset, here Argoverse Motion Forecasting v1.1 [1]. Data is gathered by a moving sensor platform (ego vehicle) and subsequently transformed into a fixed world coordinate frame. As distance and angle between sensor and agent change during 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. **Right:** The same agent trajectory as in the left figure but fitted with a 5-degree polynomial trajectory representation estimated via eq. (2). The resulting posterior covariances for agent positions are enlarged by a factor of 8 for better visibility.

In an ideal world, the only prior parameter to be estimated would be the covariance matrix Σ_ω . The observation noise covariance matrices $\Sigma_{o,i,j}$ would be derived from the ego vehicle’s sensor setup and given with the dataset. Unfortunately, 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 $\Sigma_{o,i,j} = \Sigma_o(\theta)$.

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 observation point is expressed as:

$$\Sigma_{o,i,j}^{\text{ego,world}} = \Sigma_o^{\text{ego,world}} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \quad (4)$$

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

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 (σ_α^2) and variable distance resolution ($\sigma_{r,i,j}^2$):

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

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

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

Clearly, this is not perfect since we are using observations to parameterize the uncertainty of these very observations. But it is the best approach we have and it is safe to assume $\sigma_{r,i,j}^2$ 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 $\Sigma_{o,i,j}^{\text{agent,ego}}$ based on [20, p. 77].

Finally, we rotate $\Sigma_{o,i,j}^{\text{agent,ego}}$ from the *ego* frame to the *world* frame with:

$$\Sigma_{o,i,j}^{\text{agent,world}} = \mathbf{R}_{i,j}(\Sigma_{o,i,j}^{\text{agent,ego}} + \sigma_c^2 \mathbf{I}_d) \mathbf{R}_{i,j}^\top \quad (7)$$

where $\mathbf{R}_{i,j}$ denotes the rotation matrix at j^{th} timestamp in the i^{th} agent trajectory. The additional diagonal term σ_c^2 is necessary to model errors resulting from timing instability of the tracking system especially at close ranges.

The observation covariance $\Sigma_{o,i}(\theta_{\text{agent}})$ for one complete agent trajectory is then an $md \times md$ block diagonal matrix where the j^{th} block is given by $\Sigma_{o,i,j}^{\text{agent,world}}$. Hence, we have only 5 parameters to estimate: $\theta_{\text{agent}} = [\sigma_\alpha, \beta_0, \beta_1, \beta_2, \sigma_c]$. We illustrate one example with our observation covariance model in Figure 3 with covariance marked as black ellipses.

C. Estimating Prior Parameters via Empirical Bayes

Finally, we are in the position to formulate the type-II likelihood. Following [18, pp. 172–176], we obtain:

$$\begin{aligned} p(\mathbf{C} | \Sigma_o(\theta), \Sigma_\omega) &= \prod_{i=1}^N \int \mathcal{N}(\mathbf{c}_i^{\text{ob}} | \Phi_i^\top \omega, \Sigma_{o,i}(\theta)) \\ &\quad \times \mathcal{N}(\omega | \mathbf{0}, \Sigma_\omega) d\omega \\ &= \prod_{i=1}^N \mathcal{N}(\mathbf{c}_i^{\text{ob}} | \mathbf{0}, \Sigma_{o,i}(\theta) + \Phi_i^\top \Sigma_\omega \Phi_i) \end{aligned} \quad (8)$$

where \mathbf{C} denotes all N trajectories in the dataset. We maximize the log of (8) with respect to Σ_ω and θ for ego and agent trajectories separately using gradient descent method [21]. These optima represent the prior parameters estimated from the dataset:

$$\hat{\theta}, \hat{\Sigma}_\omega = \underset{\theta, \Sigma_\omega}{\text{argmax}} \log(p(\mathbf{C} | \Sigma_o(\theta), \Sigma_\omega)) \quad (9)$$

For any model complexity n , we can thus estimate Bayesian optimal trajectory representations from given data by plugging the estimated prior parameters $\hat{\Sigma}_\omega$ and $\hat{\Sigma}_{o,i}(\hat{\theta})$ into (2).

With increasing n we will be able to achieve smaller AFE 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) [22] and Bayesian Information Criterion (BIC) [23] 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:

$$\begin{aligned} \text{AIC} &= \frac{\log(p(\mathbf{C}|\Sigma_o(\theta), \Sigma_\omega))}{N} - \text{dof}(\theta, \Sigma_\omega) \\ \text{BIC} &= \frac{\log(p(\mathbf{C}|\Sigma_o(\theta), \Sigma_\omega))}{N} - \frac{\text{dof}(\theta, \Sigma_\omega)}{2} \log(m) \end{aligned} \quad (10)$$

where $\text{dof}(\theta, \Sigma_\omega) = \text{dof}(\theta) + d(n+1)(d(n+1)+1)/2$ denotes the degrees of freedom in the observation covariance and model parameter covariance. For ego and agent trajectories, $\text{dof}(\theta_{\text{ego}}) = 2$ and $\text{dof}(\theta_{\text{agent}}) = 5$, 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.

VI. EXPERIMENTS

A. Preprocessing: Data Selection and Outlier Detection

For comparability, we only analyze the training set and limit our analysis to vehicle trajectories in A1 and additionally consider cyclist and pedestrian trajectories in 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 vehicle trajectories in WO, we limit our analysis to a random sample of 300k (without outlier) from 1.84 million vehicle trajectories to reduce computational cost.

Timestamps do not exactly follow the sampling rate. 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 , 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 double integrator based on [24, p. 48] and adjust the parameters according to [24, p. 59]. We discard trajectories for which the RTS-Smoother estimates positions more than 2m away from observation or the estimated longitudinal

acceleration (deceleration) exceeds a $6\frac{m}{s^2}$ ($-10\frac{m}{s^2}$) threshold for vehicles [25], $2\frac{m}{s^2}$ ($-4\frac{m}{s^2}$) threshold for cyclists [26], and $2\frac{m}{s^2}$ ($-3\frac{m}{s^2}$) threshold for pedestrians. 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 [27]. 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 $(0, 0)$. WO flags these trajectory points as non-valid.

TABLE II
PERCENTAGE OF OUTLIERS FOR $T = 5s$ TRAJECTORY

Datasets		A1	A2	WO
ego	time	22.81	0	0.02
	static	23.95	20.66	25.41
	out of view	0	0	0
	RTS	0	1.18	0
	total	42.95	21.84	25.42
agent vehicle	time	22.81	0	0.02
	static	0	4.95	1.70
	out of view	0	0	19.40
	RTS	6.81	0.86	19.45
	total	28.11	5.81	20.83
agent cyclist	time	-	0	0.02
	static	-	1.04	1.43
	out of view	-	0	24.12
	RTS	-	3.91	24.58
	total	-	4.94	25.62
agent pedestrian	time	-	0	0.02
	static	-	0.19	1.32
	out of view	-	0	32.20
	RTS	-	0.76	32.25
	total	-	0.93	33.12

B. 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 vehicle 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 for vehicle trajectories due to its offline tracking algorithm [3].

2) *Model Complexity and Fit Error:* Figure 4 shows Box-plots for the longitudinal and lateral fit error for vehicle trajectory samples with $T \in [3s, 5s, 8s]$. Figure 5 presents the fit error for cyclist and pedestrian trajectories with the same time horizons. 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.

Figure 4, Figure 5 and Table IV clearly show that the 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. Trajectories of various object classes require

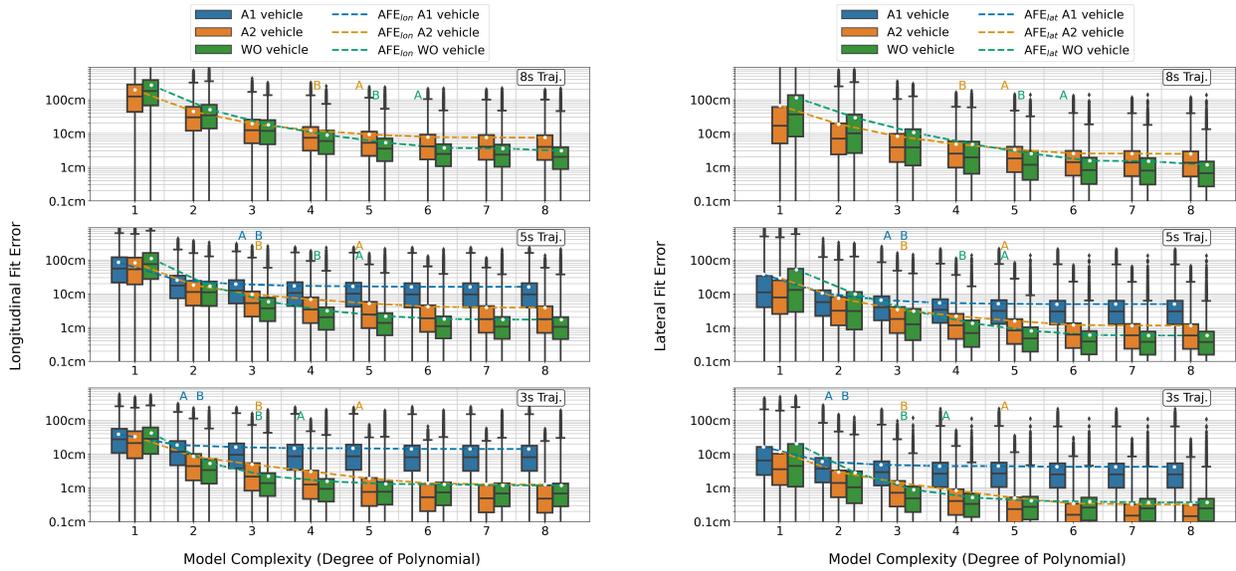


Fig. 4. The longitudinal (left) and lateral (right) fit error of models for vehicle trajectories in A1, A2 and WO with $T \in [3s, 5s, 8s]$. "A, B" denote the model complexity $n = \hat{n}$ that maximizes AIC and BIC, respectively. The upper whisker denotes the 99.9% percentile.

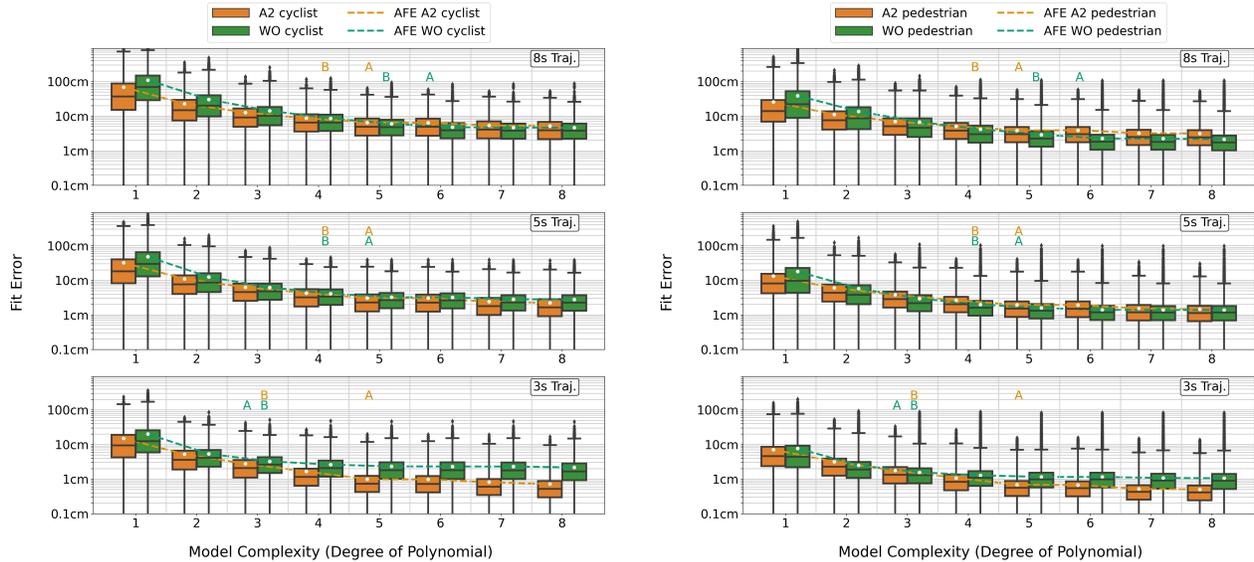


Fig. 5. The fit error of models for cyclist (left) and pedestrian (right) trajectories in A2 and WO with $T \in [3s, 5s, 8s]$. "A, B" denote the model complexity $n = \hat{n}$ that maximizes AIC and BIC, respectively. The upper whisker denotes the 99.9% percentile.

similar \hat{n} according to AIC or BIC in one dataset. Overall, we see how linear models of moderate complexity can represent trajectories with very high fidelity. E.g., the 6th degree polynomial can approximate the 8-seconds vehicle trajectories with 3.7cm longitudinal and 1.6cm lateral AFE 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. We visualize several outliers and random examples from 3 datasets in the appendix of our preprint [28].

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

Figure 6 compares the *fit error* of our linear trajectory

models to the *total displacement error* over predicted vehicle trajectories of state-of-the-art prediction methods with unbiased sequence-based trajectory representation. This comparison indicates to what extent a linear trajectory model may, at the most, impact the performance of a prediction system. We compare to $minADE_k$, i.e. the minimum average displacement error over top k most-likely predicted trajectories. Note that both fit error and total displacement error are dependent on the observation noise level as visualized in Figure 1. If the observation noise level is low, as in WO, the representation error of a linear trajectory formulation, upper bounded by fit error, is negligible in the prediction task. It is still significantly lower than the displacement error of predictions in datasets, where the noise level is likely higher

TABLE III

ESTIMATED OBSERVATION NOISE FOR $T = 5s$ AT \hat{n} MAXIMIZING AIC

Datasets		A1	A2	WO
θ_{ego}	σ_{diag} [m]	0.024	0.012	0.008
	σ_{cov} [m ²]	2e-4	3e-6	-1e-7
θ_{agent} vehicle	σ_{α} [rad]	1e-3	6e-4	3e-4
	σ_c [m]	0.161	0.044	0.017
	σ_r [m]	0.128	0.055	0.019
	$r = 10m, 20m, 40m$	0.176	0.062	0.027
		0.246	0.085	0.042
θ_{agent} cyclist	σ_{α} [rad]	-	2e-4	3e-4
	σ_c [m]	-	0.027	0.027
	σ_r [m]	-	0.014	0.010
	$r = 10m, 20m, 40m$	-	0.022	0.015
		-	0.039	0.026
θ_{agent} pedestrian	σ_{α} [rad]	-	3e-4	2e-5
	σ_c [m]	-	0.018	0.015
	σ_r [m]	-	0.007	4e-4
	$r = 10m, 20m, 40m$	-	0.010	6e-4
		-	0.017	0.001

TABLE IV
FIT ERROR OF EGO AND AGENT TRAJECTORIES [M]
(LON. | LAT.)

T [s]		Dataset			
		A1	A2	WO	
ego	\hat{n}	3	5	4	
	AFE	0.022 0.002	0.004 0.001	0.004 0.001	
	99.9%	0.173 0.120	0.094 0.012	0.066 0.021	
	\hat{n}	5	6	6	
	AFE	0.022 0.004	0.007 0.002	0.005 0.002	
	99.9%	0.157 0.111	0.143 0.030	0.071 0.025	
	\hat{n}	-	7	7	
	AFE	-	0.019 0.006	0.013 0.004	
	99.9%	-	0.251 0.090	0.114 0.069	
	agent vehicle	\hat{n}	2	5	4
		AFE	0.185 0.060	0.019 0.005	0.016 0.005
		99.9%	1.753 0.835	0.312 0.114	0.359 0.056
\hat{n}		3	5	5	
AFE		0.191 0.065	0.051 0.016	0.022 0.008	
99.9%		1.774 0.848	0.735 0.284	0.408 0.090	
\hat{n}		-	5	6	
AFE		-	0.093 0.033	0.037 0.016	
99.9%		-	1.115 0.461	0.483 0.186	
agent cyclist		\hat{n}	-	5	3
		AFE	-	0.008 0.004	0.025 0.016
		99.9%	-	0.098 0.054	0.179 0.130
	\hat{n}	-	5	5	
	AFE	-	0.022 0.016	0.025 0.016	
	99.9%	-	0.216 0.183	0.174 0.123	
	\hat{n}	-	5	6	
	AFE	-	0.042 0.040	0.031 0.028	
	99.9%	-	0.364 0.358	0.217 0.242	
	agent pedestrian	\hat{n}	-	5	3
		AFE	-	0.005 0.004	0.011 0.009
		99.9%	-	0.058 0.046	0.091 0.070
\hat{n}		-	5	5	
AFE		-	0.012 0.012	0.011 0.009	
99.9%		-	0.135 0.126	0.086 0.066	
\hat{n}		-	5	6	
AFE		-	0.023 0.026	0.016 0.012	
99.9%		-	0.257 0.255	0.133 0.108	

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

such as A1. If a prediction method can perfectly predict the parameters of future trajectories, its total displacement error will be close to the fit error of polynomials.

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.

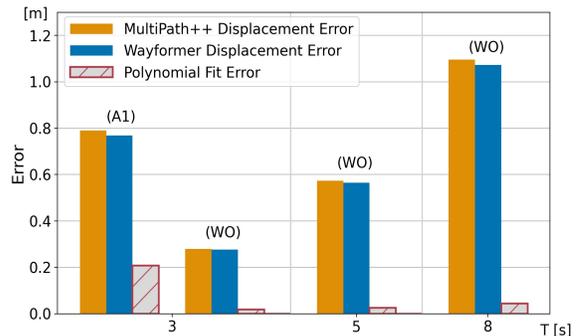


Fig. 6. The average fit error AFE of polynomials at \hat{n} maximizing AIC and the average displacement error $minADE_k$ ($k = 6$) of Wayformer[10] and MultiPath++[12] for vehicle trajectories with $T \in [3s, 5s, 8s]$. Latest prediction results can be found at [29], [30].

VII. CONCLUSION

Object trajectories for various object classes can be modelled with high fidelity by simple linear combinations of polynomial basis functions. *Independent* of any particular prediction method, we have characterized the trade-off between model complexity and fit 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 [6] or regularize trajectory prediction models.

It is important to note that the fit error introduced by a linear formulation is small compared to the total displacement errors of current state-of-the-art prediction models. This indicates that the inherent bias in linear models is much smaller than the epistemic uncertainty in the prediction task. We further suggest using linear trajectory models in future works with inherent mathematical benefits.

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”.

REFERENCES

- [1] M.-F. Chang, J. Lambert, P. Sangkloy, *et al.*, “Argoverse: 3d tracking and forecasting with rich maps,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [2] B. Wilson, W. Qi, T. Agarwal, *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.
- [3] S. Ettinger, S. Cheng, B. Caine, *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.
- [4] T. Buhet, E. Wirbel, A. Bursuc, and X. Perrotton, “Plop: Probabilistic polynomial objects trajectory planning for autonomous driving,” *arXiv preprint arXiv:2003.08744*, 2020.
- [5] Z. Su, C. Wang, H. Cui, N. Djuric, C. Vallespi-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.
- [6] J. Reichardt, “Trajectories as markov-states for long term traffic scene prediction,” in *14-th UniDAS FAS-Workshop*, Berkheim, Germany, 2022, p. 14.
- [7] <https://github.com/continental/kiwissen-empirical-bayes-analysis-of-object-trajectory>.
- [8] M. Liang, B. Yang, R. Hu, *et al.*, “Learning lane graph representations for motion forecasting,” in *European Conference on Computer Vision*, Springer, 2020, pp. 541–556.
- [9] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, “Covernet: Multimodal behavior prediction using trajectory sets,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 074–14 083.
- [10] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Re-faat, and B. Sapp, “Wayformer: Motion forecasting via simple & efficient attention networks,” *arXiv preprint arXiv:2207.05844*, 2022.
- [11] 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.
- [12] B. Varadarajan, A. Hefny, A. Srivastava, *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.
- [13] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, “What the constant velocity model can teach us about pedestrian motion prediction,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.
- [14] C. C. Macadam, “Understanding and modeling the human driver,” *Vehicle system dynamics*, vol. 40, no. 1-3, pp. 101–134, 2003.
- [15] 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.
- [16] I. Bae, J. Moon, J. Jung, *et al.*, “Self-driving like a human driver instead of a robocar: Personalized comfortable driving experience for autonomous vehicles,” *arXiv preprint arXiv:2001.03908*, 2020.
- [17] M. Bahari, S. Saadatnejad, A. Rahimi, *et al.*, “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.
- [18] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [19] B. Efron, *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*. Cambridge University Press, 2012, vol. 1.
- [20] N. Kämpchen, “Feature-level fusion of laser scanner and video data for advanced driver assistance systems,” Ph.D. dissertation, Universität Ulm, 2007.
- [21] J. V. Dillon, I. Langmore, D. Tran, *et al.*, “Tensorflow distributions,” *arXiv preprint arXiv:1711.10604*, 2017.
- [22] H. Akaike, “Information theory and an extension of the maximum likelihood principle,[w:] proceedings of the 2nd international symposium on information, bn petrow, f,” *Czaki, Akademiai Kiado, Budapest*, 1973.
- [23] G. Schwarz, “Estimating the dimension of a model,” *The annals of statistics*, pp. 461–464, 1978.
- [24] A. Philipp, “Perception and prediction of urban traffic scenarios for autonomous driving,” Ph.D. dissertation, Freie Universitaet Berlin (Germany), 2021.
- [25] P. S. Bokare and A. K. Maurya, “Acceleration-deceleration behaviour of various vehicle types,” *Transportation research procedia*, vol. 25, pp. 4733–4749, 2017.
- [26] N. Famiglietti, B. Nguyen, E. Fatzinger, and J. Landerville, “Bicycle braking performance testing and analysis,” *SAE International Journal of Advances and Current Practices in Mobility*, vol. 2, no. 2020-01-0876, pp. 3384–3397, 2020.
- [27] <https://github.com/argoai/argoverse-api/issues/124>.
- [28] Y. Yao, D. Goehring, and J. Reichardt, “An empirical bayes analysis of vehicle trajectory models,” *arXiv preprint arXiv:2211.01696*, 2022.
- [29] <https://eval.ai/web/challenges/challenge-page/454/leaderboard/1279>.
- [30] <https://waymo.com/open/challenges/2022/motion-prediction>.