Reliable Trajectory Prediction and Uncertainty Quantification with Conditioned Diffusion Models

Marion Neumeier[1] Sebastian Dorn[2,3] Michael Botsch[1] Wolfgang Utschick[4]
[1]Technische Hochschule Ingolstadt, [2]Audi AG,
[3]Technische Hochschule Augsburg, [4]Technische Universität München
{marion.neumeier, michael.botsch}@thi.de, sebastian.dorn@tha.de, utschick@tum.de

Abstract

This work introduces the conditioned Vehicle Motion Diffusion (cVMD) model, a novel network architecture for highway trajectory prediction using diffusion models. The proposed model ensures the drivability of the predicted trajectory by integrating non-holonomic motion constraints and physical constraints into the generative prediction module. Central to the architecture of cVMD is its capacity to perform uncertainty quantification, a feature that is crucial in safety-critical applications. By integrating the quantified uncertainty into the prediction process, the cVMD's trajectory prediction performance is improved considerably. The model's performance was evaluated using the publicly available highD dataset. Experiments show that the proposed architecture achieves competitive trajectory prediction accuracy compared to state-of-the-art models, while providing guaranteed drivable trajectories and uncertainty quantification.

1. Introduction

Vehicle trajectory prediction is a fundamental challenge in the automotive domain [3, 5]. Due to the highly interactive nature of traffic scenarios, model-based approaches are generally not able to capture or represent the underlying complexity and the variety of traffic situations. Many prominent approaches for predicting trajectories in cooperative traffic scenarios apply data-driven algorithms, *e.g.* [1, 13]. While these approaches are capable of successfully modelling driving behaviour, the feasibility and drivability of the predicted trajectories are not guaranteed. Vehicles are non-holonomic systems with restricted movement capabilities, such as the coupling between forward and sideway motion. Most machine learning (ML)-based vehicle motion prediction models do not account for non-holonomic and general physical constraints [20]. As a result, there is no guarantee that predictions of ML models are realistic or consistent with the general constraints of motion [13, 29]. Another shortcoming of data-driven regression models is that they typically lack the ability to quantify the uncertainty in their predictions [2, 15, 25, 47]. The models typically provide point estimates that provide the most likely prediction, but do not account for uncertainty in future trajectories. In safety-critical applications, however, it is crucial to have knowledge of the uncertainties associated with trajectory predictions. This enables intelligent systems to make informed decisions and mitigate potential risks [42]. The aim of this work is to address these limitations by introducing the conditioned Vehicle Motion Diffusion Model (cVMD). cVMD is composed of a classifier-free guided diffusion-based probabilistic model considering the non-holonomic kinematic constraints of vehicles. The trajectory prediction task is regarded as a reverse diffusion process, conditional on an interactive highway traffic scenario. To understand the traffic scenario context, cVMD integrates a Vector Quantized Variational Autoencoder (VQ-VAE) as illustrated in Fig. 1. VQ-VAE effectively discretizes the infinite traffic scenario constellations into distinct representative contexts. The cVMD architecture inherently allows for the quantification of the uncertainty in the model's predictions. This uncertainty quantification is used to make uncertainty-adaptive trajectory predictions.The main contributions are as follows:

- Introduction of the cVMD architecture for the prediction of guaranteed drivable trajectories.
- Proposal of a method to quantify prediction uncertainty and to integrate it into trajectory prediction.
- Leveraging the model's generative capabilities to represent real-world scenario stochasticity.
- Evaluation of prediction performance on publicly available highD dataset.

2. Related Work

Modeling the complex interactions in traffic scenarios and their impact on individual driving behaviors poses a significant challenge. Consequently, many studies on trajectory predictions rely on data-driven methods, *e.g.* [1, 13, 31, 41]. Recently, there has been a growing emphasis on graph-based approaches [8, 14, 17, 32] as they allow to directly model relations and inter-dependencies. However, it has been shown that graph-based approaches come with limitations [34, 36]. As a result of the remarkable achievements in the areas of computer vision [38, 40] and natural language processing [16, 26], diffusion models are gaining popularity in a wide range of fields, including the automotive domain. For example, in the work of [50], the authors propose a diffusion model for controllable traffic scenario generation. The sampling process of the diffusion model is guided by specific scenario conditions, which allow for the generation of diverse yet controllable scenarios. Furthermore, Provnost *et al.* [37] introduce a latent diffusion model [38] that utilizes map data to generate realistic driving scenes. Similarly, the authors of [4] use a conditional latent diffusion model with a temporal constraint for scene prediction. The focus of their work is on scenario generation rather than trajectory prediction. In their work, the authors also reconstruct the map data. This is an additional task on top of the scene prediction, introducing an avoidable level of complexity. Chen *et al.* [9] introduce EquiDiff, a deep generative diffusion model for predicting vehicle trajectories based on historical scenario information. The historical scenario information is embedded using Gated Recurrent Unit [10] and Graph Attention Network [46] and subsequently provided as contextual information for the trajectory generation process. By providing the contextual information the generated trajectory prediction is additionally conditioned on the observed scenario information. The intended effect of additional conditioning is to decrease generative diversity of the trajectory prediction while increasing the likelihood of a trajectory close to the ground truth future trajectory. It is worth noting, however, that the conditioning approach of the diffusion process applied by the authors differs from that proposed by Ho *et al.* [18]. Based on the analysis of the existing literature, several works have used diffusion models to predict vehicle motion. However, a common limitation of these methods is the lack of guaranteed trajectory feasibility. The authors of [43] propose the Human Motion Diffusion Model (MDM), a classifier-free diffusion-based model for the generation of realistic human motion. MDM predicts the samples rather than the noise for each diffusion step, allowing the addition of geometric losses such as foot contact to improve human motion synthesis. The optimization of the Transformer-based [45] MDM ensures that the generative process aligns with both the general abilities of humans and the principles of physics. Despite the extensive research, none of the above studies have incorporated uncertainty quantification of their motion predictions.

3. Preliminaries

3.1. Denoising Diffusion Probabilistic Model

Denoising diffusion probabilistic models (DDPMs) [19] are generative models aiming to learn the underlying data distribution $p(\boldsymbol{x})$ by reversing a forward diffusion process. The training of DDPMs consists of two phases: the forward phase and the reverse phase. During the forward phase, DDPMs transform the initial data $\boldsymbol{x}_0$ into Gaussian noise $p(\boldsymbol{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ using a predefined noising procedure. This noising procedure, also known as a noise scheduler, systematically adds Gaussian noise $\boldsymbol{\epsilon}$ at each diffusion step $t = 1, \dots, T$ until it converges to a standard normal Gaussian noise for $T \to \infty$. The noised data $\boldsymbol{x}_t$ at diffusion step t is defined as

$$\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{1}$$

The distribution of a noised data sample $\boldsymbol{x}_t$ can be represented by $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\mathbf{I})$ with mean vector $\boldsymbol{\mu}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0$ and covariance matrix $\mathbf{\Sigma}_t = (1-\overline{\alpha}_t)\mathbf{I}$. The parameter $\overline{\alpha}_t$ results from the noise scheduler and indicates the noise level at diffusion step t. Although various noise scheduling strategies exist, the cosine noise scheduler introduced by Nichol *et al.* [35] has shown particularly good performance. It is defined as

$$\overline{\alpha}_t = \frac{f(t)}{f(0)} \qquad f(t) = \cos^2\left(\frac{t/T+s}{1+s}\cdot\frac{\pi}{2}\right), \tag{2}$$

where $s \in \mathbb{R}^+$ is a small offset, *e.g.* $s = 0.008$, to prevent $\overline{\alpha}_t$ from being too small near $t = 0$. The offset s improves noise prediction in the early timesteps [35]. In the reverse phase, a neural network $p_\theta(\boldsymbol{x})$ is trained to gradually undo the transformation that occurs in the forward phase. At each step of the reverse phase, the model takes a noisy input and learns to reduce the level of noise by recovering some of the obscured information. Hence, DDPMs learn to approximate the conditional distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, t)$ by optimizing the model parameters θ. By repeating the statistical independent denoising step using

$$p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T)\prod_{T}^{t=1} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, t), \tag{3}$$

the original data can effectively be recovered from the noisy data. $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, t) = \mathcal{N}(\boldsymbol{\mu}_\theta(x_t, t), \mathbf{\Sigma}_\theta(\boldsymbol{x}_t, t))$ denotes the denoising transition step. The covariance $\mathbf{\Sigma}_\theta(\boldsymbol{x}_t, t)$ can either be learned or set to the variance determined by the forward diffusion $\mathbf{\Sigma}_\theta(\boldsymbol{x}_t, t) = \mathbf{\Sigma}_t$, where $\mathbf{\Sigma}_t = \sigma^2(t)\mathbf{I}$ and

$$\sigma^2(t) = \frac{(1-\alpha_t)(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}, \tag{4}$$

where $\alpha_t = \frac{\overline{\alpha}_t}{\overline{\alpha}_{t-1}}$. Instead of predicting the denoised data $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$, the authors of [19] found that predicting the noise terms $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is more stable. The commonly used simplified training objective results in

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t\sim[1,T]}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)||_2^2\right]. \quad (5)$$

The goal of DDPMs is to learn the noise that needs to be removed in each denoising step from distorted data in order to recover the original data. Once the training has converged, new data can be generated by repeatedly computing

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha_t}}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) + \boldsymbol{\Sigma}_t\boldsymbol{\epsilon}, \quad (6)$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

3.2. Classifier-Free Guidance

In diffusion models, the term guidance refers to controlling the generation process by incorporating additional conditions or modalities. The classifier-free guidance proposed in [18] suggests a method that does not rely on an explicit classifier to provide guidance for the diffusion model. In this guidance approach, an unconditional noise estimator $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ and a conditional noise estimator $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}, c, t)$ are jointly trained. Both are implemented through one neural network. Thus, the class identifier c of the unconditional model is set to zero for the generation process such that $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}, c = 0, t)$. During sampling, the noise estimate $\tilde{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, c, t)$ of the guided DDPM is determined by

$$\tilde{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, c, t) = (1+w)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, c, t) - w\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t), \quad (7)$$

where $w \in \mathbb{R}$ is the guidance scale. The guidance scale is used in conditional diffusion models to balance diversity and sample fidelity. It controls how much influence the condition has over the generation process: it decreases the unconditional likelihood with a negative score term while simultaneously increasing the conditional likelihood of a sample[18]. A higher guidance scale w can lead to samples that closely match the conditioning information, resulting in higher fidelity but potentially lower diversity. Vice versa, a lower guidance scale can result in more diverse samples but with less fidelity to the conditioning information.

4. Method

This section introduces the architecture of cVMD. As illustrated in Fig. 1, the network consists of three main components: the vehicle motion diffusion module, the context conditioning module and the uncertainty quantification unit (UQ). The module for context conditioning captures and categorizes the scenario context, while the vehicle motion diffusion module performs the trajectory prediction. The UQ embedded in cVMD estimates the model uncertainty. This uncertainty is also used in the uncertainty-adaptive trajectory prediction. The subsequent subsections provide a detailed explanation of each component and how they are integrated within cVMD. Initially, the considered problem formulation is presented.

4.1. Problem Formulation

Let dataset $\mathcal{D} = \{(\boldsymbol{\xi}^{(m)}, \mathbf{Y}^{(m)}, \boldsymbol{s}^{(m)})\}_{m=1}^{M}$ be composed of M distinct data samples. Each data sample holds the traffic scenario observations $\boldsymbol{\xi}^{(m)} \in \mathbb{R}^{N\times F\times T_{\text{obs}}}$, the future trajectory $\mathbf{Y}^{(m)} \in \mathbb{R}^{2\times T_{\text{pred}}}$ of a selected target vehicle and the maneuver class $\mathbf{s}^{(m)} \in \mathbb{R}^3$. The maneuver class $\mathbf{s}^{(m)}$ is an one-hot encoded vector, indicating if the future trajectory $\mathbf{Y}^{(m)}$ is a lane change left (lcl), lane change right (lcr) or keep lane (kl) maneuver. Based on the motion observation $\boldsymbol{\xi}^{(m)}$ of $N = 9$ vehicles within an interactive traffic scenario for the time span $T_{\text{obs}} = 3\,\text{s}$, the task is to predict the trajectory $\mathbf{Y}^{(m)}$ of the target vehicle $i \in \{1, \dots, N\}$. During the observation period, a total of $F = 4$ vehicle features are taken into account such that the motion information of the j-th participating vehicle is $\boldsymbol{\xi}_j^{(m)} = [\boldsymbol{x}_j, \boldsymbol{y}_j, \boldsymbol{v}_{j,\text{x}}, \boldsymbol{v}_{j,\text{y}}]^{\text{T}}$, containing the past longitudinal and lateral positions $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ and velocities $(\boldsymbol{v}_{j,\text{x}}, \boldsymbol{v}_{j,\text{y}})$ up to the current time step t_0. Based on the observed traffic scenario $\boldsymbol{\xi}^{(m)}$, the network is tasked with predicting the trajectory of the selected targeted vehicle $\mathbf{Y}^{(m)} = [\boldsymbol{x}_{\text{pred}}, \boldsymbol{y}_{\text{pred}}]^{\text{T}}$, where $\boldsymbol{x}_{\text{pred}}, \boldsymbol{y}_{\text{pred}} \in \mathbb{R}^{T_{\text{pred}}}$. The prediction horizon for the trajectory is set to $T_{\text{pred}} = 5\,\text{s}$.

4.2. Context Conditioning

The context conditioning module is used to determine and categorize the context of an observed traffic scenario $\boldsymbol{\xi}^{(m)} \in \mathbb{R}^{N\times F\times T_{\text{obs}}}$. The underlying goal is to discretize the space of possible scenario constellations. Although there are an infinite number of possible traffic scenario constellations, this work assumes that they can be decomposed into a discrete set of scenario representatives $q \in \{1, \dots, Q\}$. The rationale behind this is that comparable traffic scenarios lead to similar motion patterns for selected traffic participants. While similar traffic scenarios may differ in terms of the exact positioning and movement of the vehicles, they do provide a degree of context similarity. High contextual similarity between a new traffic scenario and a previously categorized scenario enhances certainty regarding the future behavior of the participants. To put it differently, the context conditioning module performs a clustering process, whereby each scenario $\boldsymbol{\xi}^{(m)}$ is assigned to a distinct cluster q corresponding to its specific scenario context. In this work, a VQ-VAE [12] is applied to interpret, categorize and cluster traffic scenarios with high contextual similarity. Given a traffic scenario observation $\boldsymbol{\xi}^{(m)}$, the VQ-VAE assigns a context category q to this observation. VQ-VAE combines the concepts of Variational Autoencoders (VAEs) [22] and Vector Quantization (VQ). The architecture consists of an encoder E that maps input data $\boldsymbol{\xi}^{(m)}$ to a latent representation $\hat{\boldsymbol{z}}^{(m)} \in \mathbb{R}^{R_q}$ with the vector dimension R_q and a decoder D that reconstructs the input data from the latent representation. In VQ-VAE, the

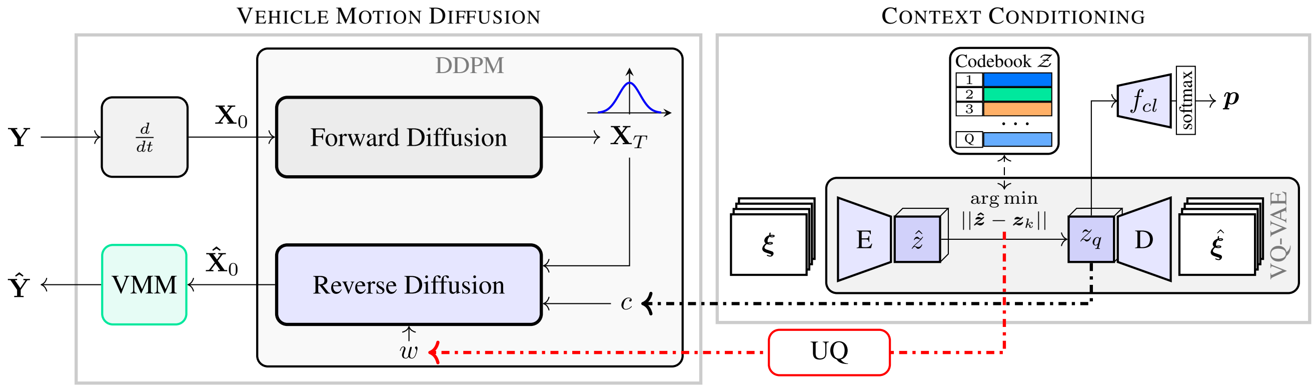

Figure 1. Architecture of cVMD, composed of three primary components: vehicle motion diffusion module, context conditioning module, and uncertainty quantification unit (UQ). The context conditioning module, realized as VQ-VAE, discretizes the traffic scenario $\boldsymbol{\xi}$. The index q of the discretized scenario context is then passed to the diffusion model as condition c. UQ determines the prediction uncertainty, which is used to adaptively modify the guidance scale w of the vehicle motion diffusion module used for the trajectory prediction.

latent space is discretized, $\boldsymbol{z}_q^{(m)} = f_q(\hat{\boldsymbol{z}}^{(m)})$, prior to being fed into the decoder, rather than directly reconstructing the original input data. During quantization, the latent representation is mapped to a single codebook vector $\hat{\boldsymbol{z}}^{(m)} \rightarrow \boldsymbol{z}_q^{(m)}$ from a finite set of codebook vectors $\mathcal{Z} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_Q\}$, using the Euclidean distance

$$\boldsymbol{z}_q^{(m)} = f_q(\hat{\boldsymbol{z}}^{(m)}) = \arg\min_{\boldsymbol{z}_k \in \mathcal{Z}} ||\hat{\boldsymbol{z}}^{(m)} - \boldsymbol{z}_k||_2^2, \tag{8}$$

with $\boldsymbol{z}_k \in \mathbb{R}^{R_q}$. Subsequently, the decoder tries to reconstruct the input data based on the determined codebook entry $\hat{\boldsymbol{\xi}}^{(m)} = D(\boldsymbol{z}_q^{(m)})$. During training, the model parameters and codebook vectors $\mathcal{Z}$ are optimized via

$$\begin{aligned}\mathcal{L}_{\text{vq}} = ||\boldsymbol{\xi}^{(m)} - \hat{\boldsymbol{\xi}}^{(m)}||^2 + ||\text{sg}[E(\boldsymbol{\xi}^{(m)})] - \boldsymbol{z}_q^{(m)}||_2^2 \\ + ||\text{sg}[\boldsymbol{z}_q^{(m)}] - E(\boldsymbol{\xi}^{(m)})||_2^2,\end{aligned} \tag{9}$$

where sg[$\cdot$] denotes the stop-gradient operation. In this work, the loss function of the VQ-VAE (Eq. 9) is extended to include a classification task in the latent space. To highlight the details in the constellations of a traffic scenario that lead to different following maneuvers (lcl, lcr, kl), a linear classifier $f_{cl}(\boldsymbol{z}_q^{(m)})$ is added. The classifier assigns a maneuver class to each selected codebook entry $\boldsymbol{z}_q^{(m)}$. To penalize false classification a cross-entropy loss

$$\mathcal{L}_{\text{cl}} = -\sum_{i=1}^{S} s_i^{(m)} \log(p_i^{(m)}), \tag{10}$$

$$\boldsymbol{p}^{(m)} = \text{softmax}(f_{cl}(\boldsymbol{z}_q)), \tag{11}$$

is applied, where $\boldsymbol{s}^{(m)} \in \mathbb{R}^S$ is the ground truth label for the $S = 3$ different maneuver classes (lcl, kl, lcr) and the predicted class $\boldsymbol{p}^{(m)} \in \mathbb{R}^S$. With the introduction of the loss $\mathcal{L}_{\text{cl}}$, the latent space is additionally forced to form meaningful codebook entries. The complete objective for training the context conditioning architecture reads

$$\mathcal{L}_{\text{cc}} = \mathcal{L}_{\text{vq}} + \lambda \mathcal{L}_{\text{cl}}, \tag{12}$$

where λ is an adaptive weight.

Once the training of the VQ-VAE has converged, each codebook entry $\boldsymbol{z}_q \in \mathcal{Z}$ is an embedding for a specific traffic scenario context. Thereby, the vast space of scenario constellations is divided into Q distinct scenario clusters, with each cluster q representing a similar scenario context.

4.3. Vehicle Motion Diffusion (VMD)

The aim of the VMD module is to model and predict feasible patterns of vehicle motion. The trajectory prediction task is performed by classifier-free guided DDPM. Unlike the context conditioning module, this module does not have access to the observed scenario context $\boldsymbol{\xi}^{(m)}$. Instead, it only receives the scenario context index $c^{(m)} = q$ as an input condition for the DDPM. When presented with condition $c^{(m)}$, the VMD module generates a context-adaptive future trajectory prediction $\hat{\mathbf{Y}}^{(m)} = [\hat{\boldsymbol{x}}_{\text{pred}}, \hat{\boldsymbol{y}}_{\text{pred}}]^{\text{T}}$ for the selected target vehicle. However, the DDPM does not directly predict the trajectory coordinates $(\hat{\boldsymbol{x}}_{\text{pred}}, \hat{\boldsymbol{y}}_{\text{pred}})$ as is common in most approaches. Instead, it learns to predict a sequence of motion parameters $\hat{\mathbf{X}}_0^{(m)}$ for a Vehicle Motion Model (VMM). The VMM transforms the motion parameters $\hat{\mathbf{X}}_0^{(m)}$ into trajectory $\hat{\mathbf{Y}}^{(m)}$.

4.3.1 DDPM

A classifier-free guided DDPM is utilized to forecast the sequence of motion parameters $\hat{\mathbf{X}}_0^{(m)}$. The DDPM is implemented as described in the preliminaries. The forward diffusion process has no learnable parameters, whereas the reverse diffusion process is approximated utilizing U-Net [39] architecture. During the training phase, DDPM learns which trajectories can be followed, or are likely to be followed, for a scenario context $c^{(m)} = q$. The condition $c^{(m)}$ is a guidance modality to generate scenario-dependent trajectory predictions. Due to the discretization of the scenario context, however, DDPM is not trained with a single most likely trajectory for each scenario q, but rather with a set of possible trajectories. Depending on the total number of scenario representatives Q, the possible trajectories for q can vary greatly (for a low value of Q) or hardly at all (for a high value of Q). The rationale behind this is, that comparable traffic scenarios lead to similar motion patterns for a given target vehicle. However, the exact execution of the motion pattern can vary. Even in identical scenarios, different drivers will respond with distinct maneuvers or trajectories. The DDPM is able to represent the inherent uncertainty in predicting trajectories, which is challenging to achieve using discriminative ML architectures. Although this approach may result in lower performance in traditional metrics, such as average error between the predicted the trajectory and ground truth, it has the advantage of capturing the inherent stochasticity.

During the application phase, a trajectory prediction is generated based on the given condition c using only the reverse diffusion path. The forward diffusion path is not necessary and therefore discarded. In this phase, however, the network additionally considers the guidance scale w. This hyperparameter determines how much influence the context condition has on the trajectory prediction. In general, the guidance scale w is a fixed value. In this work, however, the parameter is introduced as variable and is parameterized based on an estimate of the model's uncertainty of the trajectory prediction. The more confident the model is that it has seen a similar scenario before, the higher its prediction confidence and the higher its guide scale w. A detailed explanation of the realization is explained in Sec. 4.4.

4.3.2 Vehicle Motion Model (VMM)

VMMs are mathematical representations of the motion kinematics of a vehicle. These models aim to capture the relationship between the vehicle's inputs and its resulting motion by considering underlying non-holonomic constraints. In this work, a VMM with variable yaw rate $\dot{\psi}_t$ and longitudinal acceleration $a_{x,t}$ is used to represent the kinematics of vehicles. As described in [7], the position (x_t, y_t), velocity v_t and heading ψ_t of a vehicle at time step $t + \tau$ using this specific VMM are determined computing

$$x_{t+\tau} = x_t + v_t c(\psi_t)\tau + (a_{x,t} c(\psi_t) - \dot{\psi}_t v_t s(\psi_t)) \frac{\tau^2}{2} \tag{13}$$

$$y_{t+\tau} = y_t + v_t s(\psi_t) v + (a_{x,t} s(\psi_t) + \dot{\psi}_t v_t c(\psi_t)) \frac{\tau^2}{2} \tag{14}$$

$$v_{t+\tau} = v_t + a_{x,t}\tau \tag{15}$$

$$\psi_{t+\tau} = \psi_t + \dot{\psi}_t \tau \tag{16}$$

where $c(\psi_t) = \cos(\psi_t)$, $s(\psi_t) = \sin(\psi_t)$ and time increment τ. For the used VMM the motion parameters are defined as $\mathbf{X}_0^{(m)} = [\dot{\boldsymbol{\psi}}, \boldsymbol{a}_x]$, with $\dot{\boldsymbol{\psi}}, \boldsymbol{a}_x \in \mathbb{R}^{T_{\text{pred}}}$.

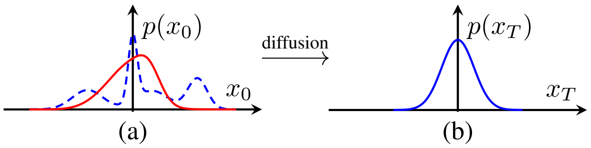

Figure 2. Diffusion processes transform (a) data $p(x_0)$ into to Gaussian noise (b) $p(x_T)$ (——). While the distribution of positional data (----) is highly depending on the map data, the motion parameter distribution (——) is more likely to resemble a Gaussian distribution.

The ground truth motion parameters $\mathbf{X}_0^{(m)} = [\dot{\boldsymbol{\psi}}, \boldsymbol{a}_x]$ are calculated by the numerical derivations $\boldsymbol{a}_x = \frac{d^2 \boldsymbol{x}_{\text{pred}}}{dt^2}$ and $\dot{\boldsymbol{\psi}} = \arctan\left(\frac{d\boldsymbol{y}_{\text{pred}}}{dt}, \frac{d\boldsymbol{x}_{\text{pred}}}{dt}\right)$. Thus, during training, the DDPM learns to predict the sequence of motion parameters $\hat{\mathbf{X}}_0^{(m)} = [\hat{\dot{\boldsymbol{\psi}}}, \hat{\boldsymbol{a}}_x]$. Note that these motion parameters have known physical limits ($\dot{\boldsymbol{\psi}}_{\text{max}} = \pm 71.26\,\text{deg/s}$[23], $\boldsymbol{a}_{x,\text{max}} = \pm 9\,\text{m/s}^2$[49]), that are taken into account during trajectory prediction. Bounding each prediction to these physical limits ensures that the values do not exceed defined limits and that the predicted trajectory can be executed by a vehicle. The equations Eq. (13)-(16) incorporate the non-holonomic constraints of vehicles, relate the vehicle's motion parameters to its motion, and allow reliable prediction of vehicle trajectories. The use of motion parameters as predicted quantities introduces an additional benefit to the learning process of the DDPM. As explained initially, the idea of diffusion models is to transform the data distribution gradually into a Gaussian distribution, and then learn how to reverse this process. While the distributions of acceleration and yaw rate values are likely to be very approximate to a Gaussian distribution, this is less likely to be the case for trajectory position data. This is due to the fact that the distribution of trajectory position data is highly dependent on the dataset used and the conditions of the infrastructure. If the original data distribution is already Gaussian, it simplifies both the forward and backward process. Hence, also the learning task is simpler since finding a reverse mapping from complex data distributions back to a Gaussian distribution is often challenging. The intuition behind this is illustrated in Fig. 2.

4.4. Uncertainty Quantification

Quantifying the uncertainty in a model's predictions and determining the level of confidence in those predictions is extremely important in safety-critical applications. Operating with a false sense of certainty can be hazardous and wrong predictions can have severe consequences. Therefore, the identification of model limitations and the assessment of model uncertainty is a significant aspect of this work. Previous studies [6, 21] have already shown that it is possible to obtain a measure of model uncertainty in the latent representation within encoder-decoder architectures. Based

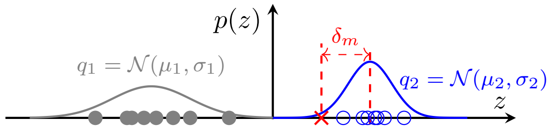

Figure 3. Example of univariate MLE based on a set of samples $\mathcal{H}_1$(●) and $\mathcal{H}_2$(○). A new data sample (×) is assigned to q_2 with the quantified uncertainty δ_m.

on these findings, in this work the prediction uncertainty of the model δ_m is estimated using Maximum Likelihood Estimation (MLE) in the latent space of the VQ-VAE and inferential statistics. The resulting model uncertainty is incorporated into the prediction of the vehicle's trajectory. As previously explained, the idea of VQ-VAEs is to map similar scenarios in close proximity within the latent space. The associated codebook entry $\boldsymbol{z}_q^{(m)}$ for each sample embedding $\hat{\boldsymbol{z}}^{(m)}$ can be interpreted as representative of the respective scenario context. Thus, the distance of a new data point from the codebook entry in the latent space indicates the similarity to the respective scenario context representative. Once the training procedure of the VQ-VAE has converged, each data sample within $\mathcal{D}_{\text{train}}$ is ultimately assigned to the closest codebook entry $\boldsymbol{z}_q^{(m)}$ according to Eq. 8. This generates a set of assigned samples $\mathcal{H}_q = \{\hat{\boldsymbol{z}}^{(1)}, \hat{\boldsymbol{z}}^{(2)}, \dots, \hat{\boldsymbol{z}}^{(h_q)}\}$ for each codebook entry $q = 1, \dots, Q$, where h_q is the total number of samples assigned to q. Each set $\mathcal{H}_q$ is used to approximate the true class conditional distribution in the latent space using a tractable distribution from within a variational family $\mathcal{Q}$. In this work, all class conditional distributions $q_q \in \mathcal{Q}$ are assumed to follow multivariate Gaussian distributions $q_q(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ with mean $\boldsymbol{\mu}_q$ and covariance $\boldsymbol{\Sigma}_q$. The distribution's parameters are defined as

$$\boldsymbol{\mu}_q = \boldsymbol{z}_q \tag{17}$$

$$\boldsymbol{\Sigma}_q = \mathbb{E}[(\hat{\boldsymbol{z}}^{(h)} - \boldsymbol{\mu}_q)(\hat{\boldsymbol{z}}^{(h)} - \boldsymbol{\mu}_q)^{\text{T}}], \tag{18}$$

where $\hat{\boldsymbol{z}}^{(h)} \in \mathcal{H}_q$. After using MLE to fit the class probability distributions as exemplified in Fig. 3, the likelihood of a new observation $\hat{\boldsymbol{z}}^{(m)} \in \mathcal{D}_{\text{test}}$ under the fitted model can be identified and the model uncertainty δ_m can be estimated. Similar to [21], the model uncertainty δ_m is quantified using the Mahalanobis distance (M-distance)

$$\delta_m(q_q, \hat{\boldsymbol{z}}^{(m)}) = \sqrt{\left(\boldsymbol{\mu}_q - \hat{\boldsymbol{z}}^{(m)}\right)^{\text{T}} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \left(\boldsymbol{\mu}_q - \hat{\boldsymbol{z}}^{(m)}\right)}. \tag{19}$$

Given a sample $\hat{\boldsymbol{z}}^{(m)}$, the M-distance evaluates the distance of the samples to the class conditional distribution $q_q(\boldsymbol{z})$. If the model uncertainty is above a certain threshold t_c, the observation could be classified as an outlier. This means that the current scenario is unlikely to be appropriately represented by any codebook entry and is therefore a potentially unknown scenario. Vice versa, the lower the M-distance, the more confident the model is in its prediction. In the context of the introduced cVMD, model uncertainty plays an important role in the prediction of the vehicle trajectory. On the one hand, a high model uncertainty indicates that the model limits have been exceeded and a reliable trajectory prediction cannot be guaranteed. On the other hand, the uncertainty quantification can be used to adaptively parameterize the guidance scale w of the diffusion model to influence the trajectory generation process.

4.5. Uncertainty-adaptive Guidance Scale

The guidance scale w controls to what extend the trajectory prediction process is conditioned on the provided context condition of the scenario. The higher the value, the more the model amplifies the provided condition to predict the trajectory. However, this does not mean that the value should always be set to maximum, as more guidance means less diversity and quality. A high level of guidance reduces the variety in the trajectory predictions and may create a risk of overemphasizing the condition in the generation process. In this work, the guidance scale is therefore calculated adaptively, based on the model's identified prediction uncertainty δ_m, using

$$w = w_{\text{min}} + \left(1 - \frac{\min(\delta_m, t_c)}{t_c}\right)(w_{\text{max}} - w_{\text{min}}), \tag{20}$$

where $w_{\text{min}}, w_{\text{max}} \in \mathbb{R}$ are the minimal and maximal parameters for w. According to Eq. 20, when the model uncertainty is low, the guidance scale is consequently large. As a result, the model's generation of the trajectory prediction is strongly conditioned on the scenario context. This setting implies that there is a comprehensive understanding of the existing context condition, as similar scenarios have been encountered before. However, due to the non-deterministic nature of diffusion models, each trajectory generation process inherently embodies a degree of stochasticity. This reflects the real-world principle that the same maneuver can be performed in many different ways due to individual driving behaviors. Conversely, if there is a high model uncertainty, the future trajectory is less certain, suggesting that the model has not been previously exposed to a similar situation. The resulting lower guidance scale leads to a trajectory prediction with more variability, implying reduced prediction reliability.

5. Dataset and Experiments

In this work, the performance of the proposed cVMD architecture is experimentally evaluated and compared with state-of-the-art models. In addition, an ablation study for the parameterization of guidance scale w is performed to evaluate the optimal hyperparameter configuration. Since the performance of the overall architecture is sensitive to the discretization quality of the VQ-VAE, also the robustness of the context conditioning is investigated.

Dataset. For the experiments, the publicly available

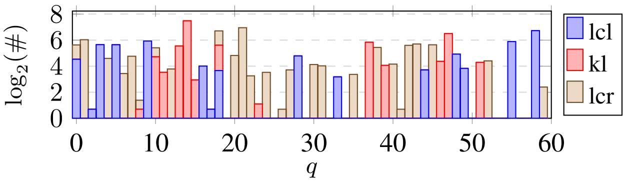

Figure 4. Stacked histogram for the selected context condition q from the codebook. The histogram has a logarithmic scale.

highway dataset highD [24] is used due to its extent of application-oriented scenarios. The highD dataset contains drone recordings of German highways taken at a frequency of 25 Hz. The dataset naturally contains a large imbalance of scenarios, as lane changes occur less frequently than lane keeping. Therefore, it is pre-processed in such a way that the extracted scenarios are distributed in a uniform manner. The resulting data format is consistent with the previously explained problem definition. The extracted scenarios are split into the subsets $\mathcal{D}_{\text{train}}$ (9,841 samples for training) and $\mathcal{D}_{\text{test}}$ (4,217 samples for testing and experiments).

Implementation Details. The training processes of the vehicle motion diffusion module and the context conditioning module are decoupled. First, the context conditioning module is trained with batch size $B_1 = 64$, learning rate $lr_1 = 4.5 \times 10^{-6}$ and $\lambda = 1$ for a total number of epochs $E_1 = 1200$. The VQ-VAE codebook is configured with $Q = 60$ entries, where each codebook entry is of dimension $\boldsymbol{z}_q \in \mathbb{R}^{64}$. Once the training procedure for the VQ-VAE is completed, its parameters are fixed. Secondly, the vehicle motion diffusion module is trained with batch size $B_2 = 64$ and learning rate $lr_2 = 1.0 \times 10^{-4}$ for $E_2 = 50$ epochs. For details of architecture implementation and code see: https://github.com/mb-team-thi/conditioned-vehicle-motion-diffusion.

6. Evaluation

6.1. Codebook entry

The VQ-VAE discretizes the infinite scenario space by assigning each scenario embedding $\hat{\boldsymbol{z}}^{(m)}$ a specific context condition, denoted as q. A visual representation of the distribution of the context indices that are assigned to the samples of $\mathcal{D}_{\text{train}}$ on the basis of Eq. 8 is given in Fig. 4. The color of a bar indicates the maneuver class (lcl, kl, lcr) of the target agent's future trajectory according to the scenario context. If a single bar is composed of different color segments, it means that one categorized scenario context leads to different maneuver classes. Ideally, however, each bar should be represented by a single color. This way, there is no ambiguity as to which maneuver category is going to be performed by the target vehicle. As can be seen, the VQ-VAE training process resulted in the maneuver converging using $Q_t = 49$ of the $Q = 60$ entries. From the stacked histogram, it is noticeable that most traffic scenario contexts q are followed by a typical target agent maneuver. Yet, some scenario types, *e.g.* $q = 18$, have no clear following maneuver. This means that in certain scenario constellations, the target vehicle reacted with different maneuvers. To quantify the level of maneuver diversity for the context conditions, average Shannon entropy $H_{\text{avg}} = \mathbb{E}_{q=1,\dots,Q}[H_{\text{q}}]$ is calculated. The entropy H_{q} for condition q is

Ablation	highD	
w	ADE [m]	FDE [m]@5 s
1	1.90	4.02
3	1.85	3.90
5	1.82	3.82
7	1.88	3.92
13	1.93	4.01
uc	**1.79**	**3.76**

Table 1. Ablation study results showing influence of guidance scale w on the trajectory prediction performance.

$$H_{\text{q}} = -\sum_{i=1}^{S} p_i \log_2 p_i, \tag{21}$$

where p_i is the probability of the maneuver class i being assigned to condition q and $S = 3$. Thus, H_{q} is a measure of the unpredictability of the maneuver class for the context condition q. As there are three potential classes S, Shannon entropy can range from 0 (complete purity) to $\log_2(3) = 1.585$ (complete impurity, instances are evenly distributed among all classes). Computing the Shannon entropy separately for the training and test datasets resulted in a significantly lower entropy for the training dataset $H_{\text{avg}}(\mathcal{D}_{\text{train}}) = 0.01$ in comparison to the test dataset $H_{\text{avg}}(\mathcal{D}_{\text{test}}) = 0.39$. Hence, on average, the distribution of maneuver classes for a context condition q is more impure for $\mathcal{D}_{\text{test}}$ than $\mathcal{D}_{\text{train}}$. While this is an indication that the VQ-VAE did not learn to generalize effectively, the entropy value of 0.39 still indicates that there is a relative majority of one class for each context condition q. However, since the model's ability to correctly predict future trajectories is only as robust as the capacity of the clustering algorithm, future work is aimed at improving clustering performance in terms of clear scenario differentiation and generating more appropriate codebook entries. Nevertheless, the current latent space of the embeddings $\hat{\boldsymbol{z}}^{(m)}$ and the generated clusters can be thoroughly examined using the visualization tool `VQSPEC`[*], based on [11, 27, 28, 44].

6.2. Ablation study

The ablation study evaluates the importance of the hyperparameter w within the predictive diffusion model. Tab. 1 shows the results of the trajectory prediction performance

[*] https://mb-team-thi.github.io/VQSPEC/

Architecture	highD	
	ADE [m]	FDE [m]@5 s
GFTNNv2 [33]	**0.72**	**1.80**
HSTA [48]	2.18	4.56
CS-LSTM [13]	2.88	5.71
MHA-LSTM(+f) [29]	2.58	5.44
Two-channel [30]	2.97	6.30
RA-GAT [14]	3.46	6.93
cVMD ($\boldsymbol{w = uc}$)	1.79	3.76

Table 2. Prediction performance of different state-of-the-art architectures based on the metrics ADE and FDE.

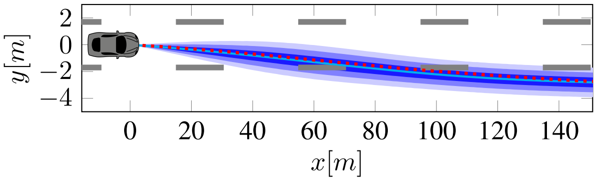

Figure 5. Trajectory prediction $\boldsymbol{\mu}_p$ (—) with confidence intervals $\boldsymbol{\sigma}_p$ (■), $2\boldsymbol{\sigma}_p$ (■), $3\boldsymbol{\sigma}_p$ (■) for a scenario assigned to index $q = 4$ and the ground truth trajectory (⋯).

as the value of parameter w is varied. Similar to [32], the Average Displacement Error (ADE) and the Final Displacement Error (FDE) at $T_{\text{pred}} = 5\,\text{s}$ are used to evaluate performance in the vehicle trajectory prediction task. In contrast to the settings $w = \{1, 3, 5, 7, 10, 13\}$, setting $w = uc$ indicates the uncertainty-adaptive computation of guidance scale w according to Eq. 20. The hyperparameters are set to $t_c = 10$, $w_{\text{min}} = 1$ and $w_{\text{max}} = 7$. In the conducted ablation study, the prediction performance improved progressively when the guidance scale was increased from $w = 1$ to $w = 5$. Increasing the guiding scale beyond $w = 5$ leads to a gradual decrease in performance. Thus, based on the evaluation metrics used, $w = 5$ results in the best performing prediction model when using a fixed guidance scale. However, the overall best performing prediction was achieved when setting the guidance scale uncertainty-adaptive. This highlights the importance and effectiveness of the proposed approach in setting the guidance scale as a function of model uncertainty, thereby managing the fidelity-diversity trade-off in the diffusion model generation process. Information on uncertainty can help assess and influence the level of confidence a model has in its trajectory predictions.

6.3. Prediction performance

To ensure fair benchmarking, all architectures are trained and tested on the same data $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. Tab. 2 compares the prediction accuracy of the proposed cVMD and state-of-the-art approaches based on the metrics ADE and FDE. For cVMD, only one trajectory prediction per scenario was generated and evaluated. Although the proposed cVMD did not outperform the best-performing prediction model, GFTNNv2, it demonstrated superior predictive capabilities to the other leading models in the field. However, this outcome was somewhat anticipated due to the fundamental differences between the proposed model and the existing ones. Unlike the models being compared, the proposed diffusion-based cVMD considers the inherent uncertainties related to the future trajectories of traffic participants. DDPMs rely on stochastic processes to generate future trajectories. While this stochasticity generally allows the model to produce multiple plausible predictions, it can also cause the model to have inferior performance compared to deterministic models. Nevertheless, the inherent stochasticity of DDPMs can be used to its advantage. DDPMs allow for the generation of a set of potential trajectories, derived from the same initial condition. This spectrum of possible trajectories can be used to approximate a statistical confidence interval, representing the range within which the actual trajectory is likely to fall a certain percentage of the time. Fig. 5 illustrates this concept. As an example, eight generated trajectories for a scenario assigned to index $q = 4$ are converted to a mean trajectory prediction $\boldsymbol{\mu}_p$ with a confidence interval constrained by the standard deviation $\boldsymbol{\sigma}_p$. Note that the observed variance within the generated trajectories is related to the parameterization of guidance scale w, linking the model uncertainty δ_m within the latent space to the confidence interval of the trajectory prediction (cf. Eq. 20). Such a confidence interval is an effective way to represent the uncertainty associated with these predictions and provides a measure of the reliability.

7. Conclusion

The proposed cVMD architecture for vehicle trajectory prediction in interactive highway scenarios allows the generation of guaranteed drivable trajectories while taking into account the inherent multimodality of real-world scenarios. Unlike fully data-driven prediction methods, cVMD includes non-holonomic motion constraints and physical limitations into the generative prediction module. Another unique feature of cVMD is its ability to quantify the model's prediction uncertainty. Incorporating model uncertainty into the trajectory prediction process has been shown to improve the network's trajectory prediction performance. When evaluated on the publicly available highD dataset, cVMD demonstrated highly competitive capabilities with established state-of-the-art architectures.

A notable limitation of the diffusion-based cVMD is its extended inference time, which currently prevents it from being used in real-time applications. Minimizing the cVMD's inference time will be the focus of future efforts. Furthermore, experiments have shown that there are limits to the effectiveness of the context conditioning module used, which should be improved with further research.

Acknowledgement. The work was supported by Audi AG.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016. 1, 2

[2] Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. In *Medical Imaging with Deep Learning*, 2018. 1

[3] Mohammadhossein Bahari, Saeed Saadatnejad, Ahmad Rahimi, Mohammad Shaverdikondori, Amir Hossein Shahidzadeh, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Vehicle trajectory prediction works, but not everywhere. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17102–17112, 2022. 1

[4] Lakshman Balasubramanian, Jonas Wurst, Robin Egolf, Michael Botsch, Wolfgang Utschick, and Ke Deng. Scenediffusion: Conditioned latent diffusion models for traffic scene prediction. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3914–3921, 2023. 2

[5] Vibha Bharilya and Neetesh Kumar. Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions. *arXiv e-prints*, 2023. arXiv:2307.07527. 1

[6] Vanessa Böhm, François Lanusse, and Uroš Seljak. Uncertainty quantification with generative models. *arXiv e-prints*, 2019. arXiv:1910.10046. 5

[7] Michael Botsch and Wolfgang Utschick. *Fahrzeugsicherheit und automatisiertes Fahren: Methoden der Signalverarbeitung und des maschinellen Lernens*. Hanser, München, 2020. 5

[8] Defu Cao, Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Spectral temporal graph neural network for trajectory prediction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, page 1839–1845, 2021. 2

[9] Kehua Chen, Xianda Chen, Zihan Yu, Meixin Zhu, and Hai Yang. Equidiff: A conditional equivariant diffusion model for trajectory prediction. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems*, 2023. 2

[10] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv e-prints*, 2014. arXiv:1412.3555. 2

[11] Grant Custer. *https://github.com/GrantCuster/umap-explorer*, 2019. 7

[12] Aaron den van Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv e-prints*, 2018. arXiv:1711.00937. 3

[13] Nachiket Deo and Mohan M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1549–15498, 2018. 1, 2, 8

[14] Zhezhang Ding, Ziyang Yao, and Huijing Zhao. RA-GAT: repulsion and attraction graph attention for trajectory prediction. In *24th IEEE International Intelligent Transportation Systems Conference, (ITSC)*, pages 734–741, 2021. 2, 8

[15] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, and et. al. A survey of uncertainty in deep neural networks. In *2022 Artificial Intelligence Review*, pages 1513–1589, 2022. 1

[16] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv e-prints*, 2022. arXiv:2210.08933. 2

[17] Hao Zhou, Dongchun Ren, Huaxia Xia, Mingyu Fan, Xu Yang, and Hai Huang. Ast-gnn: An attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction. *Neurocomputing*, 445:298–308, 2021. 2

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv e-prints*, 2022. arXiv:2207.12598. 2, 3

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020. 2

[20] Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 7(3):652–674, 2022. 1

[21] Philipp Joppich, Sebastian Dorn, Oliver de Candido, Wolfgang Utschick, and Jakob Knollmüller. Classification and uncertainty quantification of corrupted data using semi-supervised autoencoders. *arXiv e-prints*, 2021. arXiv:2105.13393. 5, 6

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv e-prints*, 2022. arXiv:1312.6114. 3

[23] János Kontos, Balázs Kránicz, and Ágnes Vathy-Fogarassy. Prediction for future yaw rate values of vehicles using long short-term memory network. *Sensors*, 23(12), 2023. 5

[24] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125. 2018. 7

[25] Yuandu Lai, Yucheng Shi, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Exploring uncertainty in regression neural networks for construction of prediction intervals. *Neurocomputing*, 481:249–257, 2022. 1

[26] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. *arXiv e-prints*, 2022. arXiv:2205.14217. 2

[27] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers and Geosciences*, 19 (3):303–342, 1993. 7

[28] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. 7

[29] Kaouther Messaoud, Itheri Yahiaoui, Anne Verroust-Blondet, and Fawzi Nashashibi. Attention based vehicle trajectory prediction. *IEEE Transactions on Intelligent Vehicles 2021*, 6:175–185. 1, 8

[30] Xiaoyu Mo, Yang Xing, and Chen Lv. Graph and recurrent neural network-based vehicle trajectory prediction for highway driving. *arXiv e-prints*, 2021. arXiv:2107.03663. 8

[31] Marion Neumeier, Michael Botsch, Andreas Tollkuhn, and Thomas Berberich. Variational autoencoder-based vehicle trajectory prediction with an interpretable latent space. In *2021 IEEE 24th International Intelligent Transportation Systems Conference (ITSC)*, pages 820–827, 2021. 2

[32] Marion Neumeier, Andreas Tollkühn, Michael Botsch, and Wolfgang Utschick. A multidimensional graph fourier transformation neural network for vehicle trajectory prediction. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 687–694, 2022. 2, 8

[33] Marion Neumeier, Sebastian Dorn, Michael Botsch, and Wolfgang Utschick. Prediction and interpretation of vehicle trajectories in the graph spectral domain. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1172–1179, 2023. 8

[34] Marion Neumeier, Andreas Tollkühn, Sebastian Dorn, Michael Botsch, and Wolfgang Utschick. Optimization and interpretability of graph attention networks for small sparse graph structures in automotive applications. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8, 2023. 2

[35] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv e-prints*, 2021. arXiv:2102.09672. 2

[36] Hoang NT and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv e-prints*, 2019. arXiv:1905.09550. 2

[37] Ethan Pronovost, Kai Wang, and Nick Roy. Generating driving scenes with diffusion. *arXiv e-prints*, 2023. arXiv:2305.18452. 2

[38] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv e-prints*, 2022. arXiv:2204.06125. 2

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv e-prints*, 2015. arXiv:1505.04597. 4

[40] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv e-prints*, 2022. arXiv:2205.11487. 2

[41] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023. 2

[42] Ho Suk, Yerin Lee, Taewoo Kim, and Shiho Kim. Chapter ten - addressing uncertainty challenges for autonomous driving in real-world environments. In *Artificial Intelligence and Machine Learning for Open-world Novelty*, pages 317–361. 2024. 1

[43] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[44] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (86):2579–2605, 2008. 7

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[46] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[47] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4697–4708, 2020. 1

[48] Ya Wu, Guang Chen, Zhijun Li, Lijun Zhang, Lu Xiong, Zhengfa Liu, and Alois Knoll. Hsta: A hierarchical spatio-temporal attention model for trajectory prediction. *IEEE Transactions on Vehicular Technology*, 70(11):11295–11307, 2021. 8

[49] Nidzamuddin Md. Yusof, Juffrizal Karjanto, Jacques Terken, Frank Delbressine, Muhammad Zahir Hassan, and Matthias Rauterberg. The exploration of autonomous vehicle driving styles: Preferred longitudinal, lateral, and vertical accelerations. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, page 245–252, 2016. 5

[50] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. *arXiv e-prints*, 2022. arXiv:2210.17366. 2