Adaptive Uncertainty Quantification for Trajectory Prediction Under Distributional Shift

Huiqun Huang[1*], **Sihong He**[1], **Fei Miao**[1]
[1]University of Connecticut
{huiqun.huang, sihong.he, fei.miao}@uconn.edu

Abstract

Trajectory prediction models that can infer both finite future trajectories and their associated uncertainties of the target vehicles in an online setting (e.g., real-world application scenarios) is crucial for ensuring the safe and robust navigation and path planning of autonomous vehicle motion. However, the majority of existing trajectory prediction models have neither considered reducing the uncertainty as one objective during the training stage nor provided reliable uncertainty quantification during inference stage under potential distribution shift. Therefore, in this paper, we propose the **C**onformal **U**ncertainty **Q**uantification under **D**istribution **S**hift framework, `CUQDS`, to quantify the uncertainty of the predicted trajectories of existing trajectory prediction models under potential data distribution shift, while considering improving the prediction accuracy of the models and reducing the estimated uncertainty during the training stage. Specifically, `CUQDS` includes 1) a learning-based Gaussian process regression module that models the output distribution of the base model (any existing trajectory prediction or time series forecasting neural networks) and reduces the estimated uncertainty by additional loss term, and 2) a statistical-based Conformal P control module to calibrate the estimated uncertainty from the Gaussian process regression module in an online setting under potential distribution shift between training and testing data. Experimental results on various state-of-the-art methods using benchmark motion forecasting datasets demonstrate the effectiveness of our proposed design.

1 Introduction

Accurately and efficiently predicting the future trajectories of the target vehicles is pivotal for ensuring the safety of path planning and navigation of autonomous systems in dynamic environments [Hu *et al.*, 2023; Kedia *et al.*, 2023]. However, trajectory uncertainly due to the changing external surrounding environments (e.g., road networks, neighborhood vehicles, passengers, and so on) or intrinsic intention changes of drivers, can lead to both trajectory *distribution shift* that change over time and *overconfidence* in the output of trajectory prediction models. It remains challenging to predict the distribution of the future trajectories of the target vehicles (rather than focusing solely on point estimates), and to quantify the uncertainty of the output trajectories under potential distribution shift between training ans testing data.

Various uncertainty quantification methods have been proposed for trajectory prediction. One popular approach is to estimate the distribution (e.g., Laplace distribution or Gaussian distribution) of future trajectories of target vehicles by direct modeling methods [Zhou *et al.*, 2022; Mao *et al.*, 2023; Zhu *et al.*, 2023; Salzmann *et al.*, 2020; Chen *et al.*, 2023; Tang *et al.*, 2021]. However, this method often overlooks the impact of *model limitation* (the restricted ability of models to represent the real-time trajectories data) and the consequence overconfident uncertainty estimation in the inference stage. Another popular approach is to calibrate the preliminary estimated uncertainty in the inference stage by the statistical-based methods. Specifically, methods such as split conformal prediction (CP) [Shafer and Vovk, 2008; Lindemann *et al.*, 2023; Lekeufack *et al.*, 2023] provide confidence intervals that guarantee to contain the ground truth target data with a predefined probability. However, they are not applicable to the situations when there is distribution shift between the training and testing data.

In this paper, we propose `CUQDS`, i.e., **C**onformal **U**ncertainty **Q**uantification under **D**istribution **S**hift framework, to quantify the output uncertainty of existing trajectory prediction models under distribution shift, while improving the prediction accuracy of the models and reducing the estimated uncertainty during the training stage. The proposed `CUQDS` framework integrates a learning-based Gaussian process regression [Rasmussen *et al.*, 2006] module with a statistics-based conformal P control module [Angelopoulos *et al.*, 2023]. Specifically, in the training stage, `CUQDS` adopts the Gaussian process regression module to estimate the output distribution of base model. The variance of the output distribution quantifies the uncertainty of the predicted trajectories. We further use the output distribution to construct uncertainty interval that guarantees to cover the true target trajectory with

[*]Contact Author

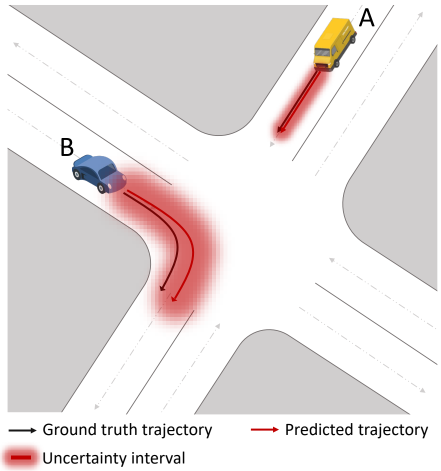

Figure 1: Illustration of the importance of a good uncertainty interval estimation for the predicted trajectories of the target vehicles.

a predefined probability in the long-run. As shown in Fig. 1, a good uncertainty interval should tend to be narrow when the predicted trajectory closely aligns with the ground truth trajectory (e.g., vehicle A), whereas it should be widen when the output trajectory deviates (e.g., vehicle B). To achieve this, we introduce additional loss term to reduce the estimated uncertainty in the training stage, ensuring narrow uncertainty interval while covering the true target trajectory with a predefined probability. In the testing or inference stage, `CUQDS` calibrates the estimated output uncertainty from the Gaussian process regression module upon potential distribution shift by a statistics-based conformal P control module. Different from standard conformal prediction method and its variants who use fixed conformal quantile in the inference stage and violate the distribution shift assumption, we first initialize the conformal quantile using the validation data and update it after every prediction step during the inference stage.

The key contributions of this work are summarized as follows:

1. We propose `CUQDS` framework that integrates both the learning-based and statistical-based modules to provide uncertainty quantification for time series trajectory prediction under distribution shift. The proposed learning-based module is trained alongside the base model to quantify its output uncertainty. In the training stage, the main objectives are to enhance prediction accuracy of base model and reduce the estimated uncertainty by incorporating an additional loss term.

2. We introduce the statistical-based conformal P control module to calibrate the estimated output uncertainty from the learning-based module under potential distribution shift during the inference stage. To alleviate the impacts of data distribution shift on uncertainty estimation, the conformal quantile is first initialized using the validation data and keep updating after each prediction step in the inference stage.

3. We validate the effectiveness of our proposed framework on the Argoverse 1 motion forecasting dataset [Chang *et al.*, 2019] and five state-of-the-art baselines. Compared to base models without our `CUQDS`, the experiment results show that our approach improves prediction accuracy by an average of 7.07% and reduce uncertainty of predicted trajectory by an average of 25.41% through incorporation with our `CUQDS`.

2 Related Work

Modeling the distribution of the trajectory of the target vehicles, instead of solely focusing on the point estimation of the future trajectory, proves to be an efficient approach [Chai *et al.*, 2019; Deo and Trivedi, 2018; Phan-Minh *et al.*, 2020; Zeng *et al.*, 2021] to avoid missing potential behavior in trajectory prediction methods. Early works sample multiple potential future trajectories [Liang *et al.*, 2020; Ngiam *et al.*, 2021; Gupta *et al.*, 2018; Rhinehart *et al.*, 2018; Rhinehart *et al.*, 2019; Tang and Salakhutdinov, 2019] to approximate the predicted trajectory distribution. However, these approaches still rely on limited point estimation and suffer from overconfidence upon trajectory prediction in real-world settings. To mitigate these issues, recent works directly estimate the distribution of future trajectories by fitting the (mixture) distribution models based on the embedded features of input trajectories [Zhou *et al.*, 2022; Varadarajan *et al.*, 2022; Cui *et al.*, 2019; Shi *et al.*, 2023]. However, these works mostly train the (mixture) distribution models by the MLP-based predictor in the last layer of the model. Such designs fail to estimate an accurate enough distribution of future trajectories under data distribution shift.

Uncertainty quantification in trajectory prediction is challenging and usually solved by two categories of methods: direct modeling and statistical-based methods. Specifically, direct modeling assumes the target data follows specific distribution, designs the learning-based neural networks for distribution estimation, and introduces the corresponding loss function to model the uncertainty directly. To achieve this, existing works usually assume the data follows Gaussian distribution [Mao *et al.*, 2023] or Laplace distribution [Gu *et al.*, 2024] and model the corresponding distribution by learning-based designs. However, estimating rigorous uncertainty by direct modeling can be challenging, as the model may easily overfit the training dataset such that invalid in the inference stage under data distribution shift. **Conformal prediction** [Shafer and Vovk, 2008; Angelopoulos *et al.*, 2023] stands out as a method in statistical-based inference that has proven effective in constructing predictive sets, ensuring a certain probability of covering the true target values [Ivanovic *et al.*, 2022; Xu *et al.*, 2014]. However, the assumption of identical distribution across all datasets invalidates the standard conformal prediction under distribution shift. Consequently, this necessitates modifications [Xu and Xie, 2023] in conformal prediction to enable timely calibration of model outputs when confronted with distribution shift.

In this study, we propose the `CUQDS` framework to provide output distribution for the predicted trajectories of base model under distribution shift, while improving the prediction accuracy of base model and reducing the estimated uncertainty by introducing additional loss term. In particular, `CUQDS` adopts

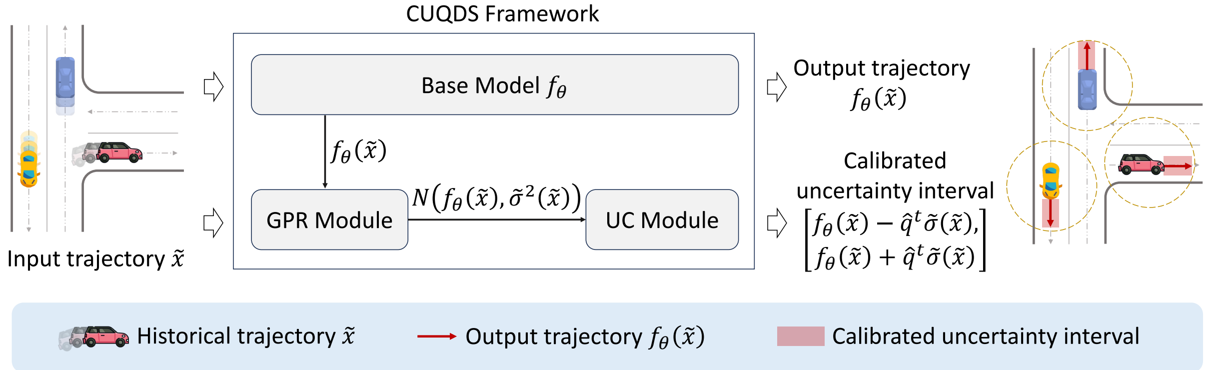

Figure 2: Our CUQDS models the conditional output distribution $\tilde{Y}|\tilde{x}, \mathcal{D} \sim \mathcal{N}\left(f_\theta\left(\tilde{x}\right), \tilde{\sigma}^2\left(\tilde{x}\right)\right)$ of base model f_θ by the Gaussian process regression (GPR) module, and provides the correspond calibrated uncertainty interval $\left[f_\theta\left(\tilde{x}\right) - \hat{q}^t\tilde{\sigma}\left(\tilde{x}\right), f_\theta\left(\tilde{x}\right) + \hat{q}^t\tilde{\sigma}\left(\tilde{x}\right)\right]$ for the predicted trajectories by the uncertainty calibration (UC) module.

a Gaussian process regression module to estimate the output uncertainty of base model, then utilizes a statistical-based conformal P control module to calibrate this output uncertainty by taking into account the model performance on recent trajectory data. Moreover, different from standard conformal prediction methods who use the fixed conformal quantile in the inference stage, we first initialize the conformal quantile using the validation data and update it after every prediction in the inference stage to alleviate the data distribution shift problem.

3 Methodology

3.1 Problem Formulation

Suppose we have a training dataset $\mathcal{D}_1 = \{(x_i, y_i)\}_{i=1}^{N_1} = \{(x^t, y^t)\}_{t=L-1}^{T_1}$, a validation dataset $\mathcal{D}_2 = \{(x_i, y_i)\}_{i=1}^{N_2} = \{(x^t, y^t)\}_{t=L-1}^{T_2}$, and a testing dataset $\tilde{\mathcal{D}} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{N_3} = \{(\tilde{x}^t, \tilde{y}^t)\}_{t=L-1}^{T_3}$. N_1, N_2, and N_3 are the number of data samples, and T_1, T_2, and T_3 are the time periods of each dataset. $T_1 \bigcap T_2 \bigcap T_3 = \varnothing$. As shown above, we use two ways to represent each dataset, where (x_i, y_i) (or $(\tilde{x}_i, \tilde{y}_i)$) denotes the random input-output trajectory pair within each dataset indexed by i, and (x^t, y^t) (or $(\tilde{x}^t, \tilde{y}^t)$) is time series data and represents the input-output trajectory pair of the current time step t. Both x_i and x^t denote the input historical trajectory during the past L historical time steps and are sampled from domain $\mathcal{X} \in \mathbb{R}^{L\times D}$, and y_i and y^t denote the corresponding target trajectory during the J following time steps from domain $\mathcal{Y} \in \mathbb{R}^{J\times D}$. D is the dimension of target features. We will omit the index i or t when there is no conflict.

We assume that we have a time series trajectory prediction model f with parameters θ. We call this model f_θ a base model and it can be implemented as different structures of neural network [Zhou *et al.*, 2022; Liu *et al.*, 2021; Zhou *et al.*, 2023; Liang *et al.*, 2020; Zhong *et al.*, 2022]. The model f_θ is trained using the dataset $\mathcal{D}$, where the target trajectory, denoted as Y, is conditioned on input x and the dataset $\mathcal{D}$. We model the conditional distribution of Y given x and $\mathcal{D}$ as a Gaussian distribution, where $Y|x, \mathcal{D} \sim \mathcal{N}\left(\mu(x), \sigma^2(x)\right)$. In this formulation, $\mu(x)$ and $\sigma^2(x)$ are functions that map the input x to the mean and variance of the Gaussian distribution, respectively. We treat the predicted trajectory of the base model f_θ as the mean $\mu(x)$ directly. Our main task in this study is to estimate the variance $\sigma^2(x)$ to denote the output uncertainty.

We propose the Conformal Uncertainty Quantification framework, CUQDS, as illustrated in Fig. 2, to quantify the uncertainty of the predicted trajectory of base model under potential distribution shift. The main objectives of this framework are to improve prediction accuracy of base model and reduce the output uncertainty. The framework introduces two modules, *1)* In particular, in the training and validation stages (Alg. 1), we propose a learning-based Gaussian process regression module to approximate the conditional output distribution $(Y|x, \mathcal{D})$ of base model by estimating $(Y|x, \theta \sim \mathcal{N}\left(\hat{\mu}\left(x\right), \hat{\sigma}^2\left(x\right)\right))$ based on f_θ. The details are introduced in Sec. 3.2. *2)* In the testing or inference stage (Algorithm. 2), we propose a statistical-based conformal P control module to calibrate the output distribution $\tilde{Y}|\tilde{x}, \theta \sim \mathcal{N}\left(\tilde{\mu}\left(\tilde{x}\right), \tilde{\sigma}^2\left(\tilde{x}\right)\right)$ by considering the potential distribution shift between the training and testing datasets and the performance limitation of f_θ on current time series trajectory input. The conformal quantile of the module is first initialized using the validation data and keep updating after each prediction during the inference stage. We build the calibrated uncertainty interval and ensure it covers the true target trajectory with a predefined probability in long-run. The details are introduced in Sec. 3.3.

To summarize, during training and validation stage, our goal is to find the parameters $[\theta, \omega]$ such that minimizing the loss function $\mathcal{L}$ on training data:

$$[\theta, \omega] = \underset{\theta,\omega}{\arg\min}\, \mathcal{L}\left(\theta, \omega|\mathcal{D}\right). \tag{1}$$

The loss $\mathcal{L}$ is a weighted combination of base model' loss $\mathcal{L}_1(\theta)$ and Gaussian processing regression model's loss $\mathcal{L}_2(\omega, \theta)$.

$$\mathcal{L} = w_1\mathcal{L}_1(\theta) + w_2\mathcal{L}_2(\omega, \theta), \tag{2}$$

where $w_1 \in \mathbb{R}$ and $w_2 \in \mathbb{R}$ are the weights adjusting the influence of two loss terms, respectively. During testing or inference stage, we will calibrate the covariance and estimate the uncertainty interval of the predicted trajectory.

3.2 Gaussian Process Regression Module

Existing literature of trajectory prediction models [Zhou *et al.*, 2022; Zhou *et al.*, 2023; Liang *et al.*, 2020; Zhong *et al.*, 2022] focuses on providing point estimates of future trajectories. However, the uncertainty of the future trajectories due to the changing environment or the intrinsic intention changes of drivers can lead to significant distribution shift and overconfident trajectory prediction. Such distribution shift and overconfident prediction can greatly impact the subsequent decision-making processes, such as robust path planning [Hu *et al.*, 2023; Kedia *et al.*, 2023]. Hence, we propose to consider output uncertainties from both the prediction capability limitation of the model f_θ upon current trajectory inputs and the noise inherent in the trajectory data.

Algorithm 1 Training & validation stage

Input: a base model $f(\cdot)$ with an initialized parameter θ and a covariance function $k(\cdot)$ with an initialized parameter ω, error rate $\alpha \in [0,1]$, score function $s(\cdot)$, total number of training epochs *epo*.

Data: training dataset $D_1 = \{x_i, y_i\}_{i=1}^{N_1}$, validation dataset $D_2 = \{x^t, y^t\}_{t=1}^{T_2}$.

Output: well-trained base model $f_\theta(\cdot)$, updated statistic $\hat{q}^1$, score set $S = \{s(x^t, y^t)\}_{t=L-1}^{T_2}$ and error set $E = \{e^t\}_{t=L-1}^{T_2}$ estimated from the validation data.

```
Initialization: q̂^0 = 1.
for epoch in epo do
    Training CUQDS_{θ,ω} with loss L and training dataset D_1.
    Update {θ, ω} ← argmin_{θ,ω} L(θ, ω|D)
    Initialize score set S = {}, error set E = {}.
    for t = 1 : T_2 do
        f_θ(x^t), σ̂(x^t) ← CUQDS_{θ,ω}(x^t)
        Compute score s^t = s(x^t, y^t) by Eq. 8.
        Compute conformal prediction set C^t by Eq. 9.
        Compute e^t = 1_{ỹ^t ∉ C^t}.
        S ← s^t, E ← e^t.
        η = β max(S), then q̂^t = q̂^{t-1} + η(Ē − α).
Update q̂^1 ← q̂^{T_2}.
```

Gaussian Process

Gaussian process regression often acts as the surrogate model to estimate the output distribution of existing models [Erlygin *et al.*, 2023]. It introduces additional loss term to guild the learning process of existing models and reduce the output uncertainty. Gaussian process regression assumes any combinations of data samples follows different join distribution and thus good at capturing the nonlinear relationship among time series data samples. In this study, we introduce to utilize the Gaussian process regression method to estimate the preliminary output distribution of existing trajectory prediction models.

We view the base model f_θ as a black box, whose design can be an existing neural network structure in the literature (examples include [Zhou *et al.*, 2022; Liu *et al.*, 2021; Zhou *et al.*, 2023; Liang *et al.*, 2020; Zhong *et al.*, 2022]). We aim to estimate the output distribution $(Y|x, \theta \sim \mathcal{N}\left(\hat{\mu}\left(x\right), \hat{\sigma}^2\left(x\right)\right))$ corresponding to the time series trajectory inputs x by a learning-based Gaussian process regression module [Rasmussen *et al.*, 2006]. We treat the output of base model $f_\theta\left(\cdot\right)$ on x as the mean value $\hat{\mu}\left(x\right)$ of the estimated output distribution and the correspond variance $\hat{\sigma}^2\left(x\right)$ as the uncertainty of $f_\theta\left(x\right)$. In real-world settings of autonomous vehicles, the collected historical trajectories of the target vehicles inevitably contain noisy data due to the processing limitations during perception and object tracking steps. Such noise usually hardly captured by the trajectory prediction models such that greatly impact the output trajectory and the correspond uncertainty. To mitigate this problem, we model such noisy impact as $\epsilon \in \mathbb{R}^D$ and assume it follows Gaussian distribution $\mathcal{N}\left(0, \sigma_\epsilon^2\right)$ with zero mean for simplicity and variance of $\sigma_\epsilon^2 \in \mathbb{R}^D$. σ_ϵ is set as trainable vector. Now the trajectory prediction process of base model is denoted as $Y|_x = \hat{\mu}(x) + \epsilon$, where (x, y) is any data samples from $\mathcal{D}$. Then we have:

$$Y|_x \sim N\left(\hat{\mu}\left(x\right), \frac{\sum_{i=1}^{N_1} k\left(x, x_i\right)}{N_1} + \sigma_\epsilon^2 I\right), \tag{3}$$

where $k(\cdot,\cdot)$ is the kernel function or the covariance function. For example, $k\left(x, x'\right)$ quantifies the similarity between trajectory input data x and x'.

Learnable Kernel Function
In this paper, we define the covariance function as the radial basis function

$$k\left(x, x'\right) = l_1^2 exp\left(-\frac{\left(x - x'\right)^2}{2l_2^2}\right), \tag{4}$$

where $l_1 \in \mathbb{R}$ and $l_2 \in \mathbb{R}$ are the parameters to be trained. The closer x and x' are, the higher the value of $k(x, x')$, reaching its maximum value of l_1^2.

Eigenvector Inducing Variables
As shown in Eq. 3, the covariance $\hat{\sigma}^2(x)$ in standard Gaussian process regression method is typically optimized based on the whole training data. However, this can be computationally expensive when the number of training samples (N_1) is large, given that the entire Gaussian process regression model requires $O(N_1^3)$ computational complexity and $O(N_1^2)$ memory complexity. To alleviate such complexity while maintaining the effectiveness of the Gaussian process regression module, we propose a modification to approximate the covariance $\hat{\sigma}^2(x)$ in Eq. 3 by extracting $M \in \mathbb{R}$, $M \ll N_1$, inducing variables $\{v_i\}_{i=1}^M, v_i \in \mathbb{R}^{L\times D}$ by Principal Component Analysis (PCA). These M inducing variables summarize the key information of training dataset. More specifically, we first standardize each input trajectory data sample $x_i \in \mathbb{R}^{L\times D}$ in training dataset $\mathcal{D}_1$ by its own mean and standardisation along time dimension. Then we build the $N_1 \times N_1$ covariance matrix by $cov(x, x') = \boldsymbol{E}[(x - \boldsymbol{E}(x))(x' - \boldsymbol{E}(x'))]$. The further steps follow the standard PCA processes.

Inference under Distribution Shift

Through the Gaussian process regression with a learnable kernel function, we are able to estimate the target trajectory's distribution that in the training/validation dataset. However, under potential distribution shift between training and testing data in real-world settings, solely keep using the information learn from the training/validation datasets will lead to high generalization error. To take into account the potential distribution shift between the dataset $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2\}$ and the testing dataset $\tilde{\mathcal{D}}$, we instead estimate the conditional output distribution of base model in the testing stage by $\tilde{Y}|\tilde{x}, \mathcal{D} \sim \mathcal{N}(\tilde{\mu}(\tilde{x}), \tilde{\sigma}(\tilde{x}))$, where $(\tilde{x}, \tilde{y}) \in \tilde{\mathcal{D}}$. Then we compute the variance of the conditional output distribution of base model in the testing stage by

$$\tilde{\sigma}^2(\tilde{x}) = k(\tilde{x}, \tilde{x}) - K_{\tilde{x}M}[K_{MM} + \sigma_\epsilon^2 I]^{-1}(K_{\tilde{x}M})^\intercal, \quad (5)$$

where $K_{\tilde{x}M} = (k(\tilde{x}, v_i), \ldots, k(\tilde{x}, v_M))^\intercal$ and $[K_{MM}]_{ij} = k(v_i, v_j)$. We then construct the uncertainty interval as $[f_\theta(\tilde{x}) - \tilde{\sigma}(\tilde{x}), f_\theta(\tilde{x}) + \tilde{\sigma}(\tilde{x})]$, indicating there is a high probability that it will cover the true target trajectories.

To find the optimal parameters for the covariance function and the noise ϵ, we introduce a new loss term $\mathcal{L}_2$:

$$\mathcal{L}_2(\omega, \theta) = \frac{1}{N_1}\sum\nolimits_{i=1}^{N_1}(-\frac{1}{2}e_i^\top \left[\bar{K}_{x_iM} + \sigma_\epsilon^2 I\right]^{-1} e_i \quad (6)$$

$$-\frac{1}{2}log\left|\bar{K}_{x_iM} + \sigma_\epsilon^2 I\right|) + \frac{1}{2N_1}log 2\pi), \quad (7)$$

where $\omega = [l_1, l_2, \sigma_\epsilon]$ contains all trainable parameters in the Gaussian process regression module, $\bar{K}_{x_iM} = \frac{1}{M}\sum_{j=1}^{M} k(x_i, v_j)$, and $e_i = y_i - f_\theta(x_i)$.

3.3 Uncertainty Quantification under Distribution Shift through Calibration

The learning-based Gaussian process regression module is prone to overfit the training data and provide overconfident uncertainty estimation under distribution shift. Moreover, such uncertainty is more concerned about the aleotoric uncertainty, referring to the probabilistic nature of noise in data [Kendall and Gal, 2017], but fails to consider the performance limitation of base model on different trajectory data, namely the epistemic uncertainty. To solve these problems, we propose a conformal P control module to calibrate the output uncertainty from the Gaussian process regression module by considering the performance of base model under potential distribution shift. The idea of this method is based on both the conformal prediction [Shafer and Vovk, 2008] and the P control in [Angelopoulos *et al.*, 2023]. However, different from standard conformal prediction, we assume the training data and testing data follow different distributions. We initialize the conformal quantile using the validation data and keep updating it after every prediction step in the inference stage to provide trustworthy uncertainty quantification.

Conformal Prediction for Long-run Converage

We aim to design a statistical-based method to calibrate the preliminary output uncertainty from the Gaussian process regression module in the inference stage. Our goal is to achieve a long-run average coverage rate in time, ensuring that the calibrated uncertainty interval covers the true target trajectories with a probability of $1 - \alpha$ [Angelopoulos *et al.*, 2023], namely $\sum_{t=1}^{T} \mathtt{err}^t / T \to \alpha$ as $T \to \infty$. $\mathtt{err}^t = \mathbb{1}(y^t \notin \mathcal{C}^t)$ and $\alpha \in (0, 1)$ is a predefined target error rate threshold. In particular, we seek to design a conformal score function $s(x, y)$ (smaller scores encode better agreement between x and y), construct a conformal prediction set $\mathcal{C}^t(y|s(x^t, y) \leq q^t)$, estimate the conformal quantile $\hat{q}^t$ from the updated prediction set, and then calibrate the estimated uncertainty interval from the Gaussian process regression module by the updated conformal quantile. We define the score function as

$$s(x, y) = \frac{|y - f_\theta(x)|}{\sigma(x)}, \quad (8)$$

where y is the ground truth, $f_\theta(x)$ and $\sigma(x)$ are respectively the mean and std of the output distribution of the base model. Then, the conformal prediction set at time t equals to an uncertainty interval $\{y | y \in [f_\theta(x^t) - q^t \sigma(x^t), f_\theta(x^t) + q^t \sigma(x^t)]\}$. Now the key of uncertainty calibration under potential distribution shift is to update the conformal quantile q^t by considering model performance on recent inputs.

Uncertainty Calibration in CUQDS Framework

To update the conformal quantile q^t under distribution shift, we first initialize the conformal quantile using the validation data as shown in Alg.1. In particular, during each validation iteration, we first initialize the score set S which records the conformal scores and the error set E which records the coverage errors of uncertainty intervals as empty. For each data sample (x^t, y^t) in validation data D_2, we first compute the predicted trajectory $f_\theta(x^t)$ from the base model and the standard deviation $\hat{\sigma}(x^t)$ from the Gaussian process regression module for the trajectory input x^t. The conformal score for this prediction is calculated using the score function $s(x^t, y^t)$, which evaluates the model's performance for current data sample. Then, a conformal prediction set $\mathcal{C}^t$ is computed to determine whether the true label y^t falls within the estimated uncertainty interval. The error e^t is set to 1 if y^t is not within $\mathcal{C}^t$, indicating a prediction error, and 0 otherwise. The estimated conformal score $s(x^t, y^t)$ is added to the score set S, and the error e^t is added to the error set E. The conformal quantile $\hat{q}^t$ is then updated based on both the existing conformal scores and coverage errors. The updating rule involves a learning rate $\eta = \beta \max(S)$ and adjusts $\hat{q}^t$ according to the formula

$$\hat{q}^t = \hat{q}^{t-1} + \eta(\bar{E} - \alpha),$$

where $\bar{E}$ is the average coverage error in the coverage error set and α is the predefined error rate. This step allows the model to adapt its uncertainty estimation based on the observed performance during validation.

During the testing stage, the conformal P control module inherits the conformal quantile, the score set S, and the error set E from the final validation iteration. The following conformal quantile updating steps are similar as in the validation stage as shown in Alg.2. For each time step t in the testing stage, we get the predicted trajectory $f_\theta(\tilde{x}^t)$ from the well trained base model and variance $\hat{\sigma}^2(\tilde{x}^t)$ from the well trained Gaussian process regression module. We then calibrate the output uncertainty and construct the uncertainty interval by using the current conformal quantile $\hat{q}^t$,

$$\mathcal{C}^t = \left[f_\theta(\tilde{x}^t) - \hat{q}^t\tilde{\sigma}(\tilde{x}^t), f_\theta(\tilde{x}^t) + \hat{q}^t\tilde{\sigma}(\tilde{x}^t)\right]. \quad (9)$$

Algorithm 2 Testing stage

Input: the well-trained framework $\texttt{CUQDS}_{\theta,\omega}$, the conformal quantile $\hat{q}^1$, the error set E, the conformal score set S, the error rate α from the training stage, score function $s(\cdot)$.

Output: predicted trajectory and the correspond calibrated uncertainty interval.

Data: testing data $\tilde{D} = \{(\tilde{x}^t, \tilde{y}^t)\}_{t=L-1}^{T_3}$

for $t = 1 : T_3$ **do**
- $f_\theta(\tilde{x}^t), \hat{\sigma}(\tilde{x}^t) \leftarrow \texttt{CUQDS}_{\theta,\omega}(\tilde{x}^t)$
- Calibrate std $\tilde{\sigma}(\tilde{x}^t)$ by $\hat{q}^{t-1}\tilde{\sigma}(\tilde{x}^t)$.
- Compute score $s(\tilde{x}^t, \tilde{y}^t)$ by Eq. 8.
- Compute conformal prediction set $\mathcal{C}^t$ by Eq. 9.
- Compute $e^t = \mathbb{1}_{\tilde{y}^t \notin \mathcal{C}^t}$.
- $S \leftarrow s^t, E \leftarrow e^t$.
- $\eta = \beta \max(S)$, then $\hat{q}^t = \hat{q}^{t-1} + \eta(\bar{E} - \alpha)$.

4 Experiment

4.1 Experimental Setups

Dataset & Key Setups: We use the Argoverse 1 motion forecasting dataset [Chang *et al.*, 2019] to verify the efficacy of our approach. This dataset collects trajectory data from Miami and Pittsburgh, with a sample rate of 10 Hz. Given that the ground truth future trajectories are not provided in this official test sequences but are essential to our CUQDS in the testing stage to update the conformal quantile $\hat{q}^t$, we repartition the sequences. In particular, we split the 205,942 official training sequences into 166,470 training data and 39,472 validation data, and use the official 39,472 validation sequences as testing data in this study.

In all experiments of this study, α is set as 0.1. We implement existing base models by using their default settings, unless otherwise specified. The host machine is a server with IntelCore i9-10900X processors and four NVIDIA Quadro RTX 6000 GPUs.

Prediction Accuracy Evaluation Metrics: Following the standard evaluation protocol, we utilize metrics including minimum Average Displacement Error ($\texttt{minADE}_k$), minimum Final Displacement Error ($\texttt{minFDE}_k$), and Miss Rate ($\texttt{MR}_k$) to evaluate the prediction accuracy of the model. As common, k is selected as 1 and 6.

Uncertainty Evaluation Metrics: To further verify the efficacy of our CUQDS in reducing the predicted uncertainty to provide narrow while accurate uncertainty interval, we adopt the Negative Log-Likelihood (NLL) [Feng *et al.*, 2021] to assess the level of uncertainty in the predicted distribution. For this metric, lower values connote a higher degree of precision in uncertainty estimation and narrower uncertainty interval.

4.2 Baselines

In this study, we employ the following models as baselines for comparison.

1. HiVT [Zhou *et al.*, 2022]: HiVT is a transformer-based trajectory prediction model which models the output distribution as a mixture Laplace distribution. We set hidden units as 64 and radius as 50.
2. LaneGCN [Liang *et al.*, 2020]: LaneGCN is a graph convolutional network based model that predicts the k possible future trajectories and their confidence scores by a MLP-based prediction header.
3. LBA [Zhong *et al.*, 2022]: LBA improves trajectory forecasting accuracy of existing models by fusing information from observed trajectories, HD maps, and local behavior data.
4. LBF [Zhong *et al.*, 2022]: LBF improves trajectory forecasting accuracy of existing models by employing a local-behavior-free prediction framework which infers the impact of missing data when the local behavior historical data is insufficient or unavailable.
5. SPCI [Xu and Xie, 2023]: SPCI proposes a learning-based estimator to predict the conditional quantile directly and bases on split conformal prediction to provide conformal prediction interval for existing time series forecasting models. We implement additional module whose structure is the same as the estimator to predict the variance of the predicted trajectory and add the KLD [Meyer and Thakurdesai, 2020] loss term to reduce the estimated uncertainty.

4.3 Main Results

We adopt HiVT, LaneGCN, LBA, and LBF as base models and apply our CUQDS on them to verify the efficacy of CUQDS. To compare our CUQDS with state-of-the-art methods, we also apply SPCI into the above four base models. As presented in Table. 1 and Table. 2, our CUQDS improves the prediction accuracy 7.07% on average. Compared with incorporating SPCI in the base models, our CUQDS reduces the output uncertainty which quantified by NLL by 25.41% on average, and improves the average coverage rate of the estimated uncertainty interval by 21.02%. The results indicate that our CUQDS is capable of providing the trust worthy uncertainty quantification for the output trajectory of base model under potential distribution shift. Our estimated uncertainty intervals are narrow and achieve high coverage rate of covering the true target trajectories.

Compared with the transformer-based model HiVT, applying our CUQDS in HiVT provides the output uncertainty of HiVT by considering the distribution difference between the training and testing data. Compared with the base models of LaneGCN, LBA, and LBF who provide a confidence score for each output trajectory, our CUQDS provides the output distribution of base model instead of point estimates and calibrates the correspond uncertainty by taking into account the model performance on recent inputs. SPCI try to predict the conformal quantile by the learning-based estimator to provide the

Table 1: Prediction results and performance comparison on testing dataset when with and without our CUQDS.

Scheme	minADE$_6$ ↓	minFDE$_6$ ↓	MR$_6$ ↓	minADE$_1$ ↓	minFDE$_1$ ↓	MR$_1$ ↓
HiVT	0.692	1.043	0.106	1.291	2.895	0.499
HiVT+SPCI	0.694	1.047	0.107	1.302	2.921	0.501
HiVT+CUQDS	**0.682**	**1.034**	**0.097**	**1.218**	**2.696**	**0.455**
LaneGCN	0.719	1.094	0.104	1.375	3.023	0.505
LaneGCN+SPCI	0.739	1.169	0.114	1.585	3.573	0.568
LaneGCN+CUQDS	**0.704**	**1.033**	**0.097**	**1.255**	**2.744**	**0.482**
LBA	0.717	1.094	0.103	1.395	3.100	0.503
LBA+SPCI	0.719	1.098	0.104	1.402	3.100	0.523
LBA+CUQDS	**0.705**	**1.044**	**0.092**	**1.251**	**2.801**	**0.483**
LBF	0.720	1.098	0.104	1.530	3.467	0.549
LBF+SPCI	0.721	1.099	0.105	1.533	3.469	0.550
LBF+CUQDS	**0.714**	**1.091**	**0.097**	**1.455**	**3.188**	**0.522**

Table 2: Compare the NLL distribution distance between ground truth and predicted distribution, and the average cover rate (CR) of the estimated uncertainties interval under settings of 1) incorporate SPCI into base models, 2) incorporate our CUQDS into base models.

Scheme	+SPCI (NLL)	+CUQDS (NLL)	+SPCI (CR)	+CUQDS (CR)
HIVT	21.354	16.354	0.705	0.832
LaneGCN	25.532	18.896	0.603	0.746
LBA	23.634	17.453	0.602	0.721
LBF	23.723	17.535	0.614	0.716

Table 3: Compare the coverage rates of the uncertainty interval with and without the uncertainty calibration (UC) module.

Scheme	Without UC	With UC
HiVT+CUQDS	0.603	0.832
LaneGCN+CUQDS	0.497	0.746
LBA+CUQDS	0.570	0.721
LBF+CUQDS	0.569	0.716

prediction interval. However, such estimator is prone to overfit the training data and invalid under distribution shift.

4.4 Ablation Study

• Conformal Prediction VS conformal P control module: We conduct ablation study on replacing the P control uncertainty calibration module with the standard split conformal prediction [Shafer and Vovk, 2008]. More specificity, the standard split conformal prediction (CP) estimates the conformal quantile $\hat{q}$ as the $\frac{(1-\alpha)(N_2+1)}{N_2}$ smallest element in the conformal score set $\mathcal{S} = \{s(x_i, y_i)\}_{i=1}^{N_2}$ which uses validation dataset. This conformal quantile $\hat{q}$ is only calculated once and fixed in the testing stage to calibrate the uncertainty.

We implement the standard split conformal prediction setting in base model LaneGCN. The average coverage rates of the uncertainty intervals estimated by our P control module and the split conformal prediction are 0.746 and 0.402, respectively. Our P control module outperforms the standard split conformal prediction. This helps to prove the effectiveness of our uncertainty calibration module in adapting to potential distribution shift.

Table 4: Results of our CUQDS and replacing the Gaussian process regression module with the self-attention based design for uncertainty estimation.

Scheme	minADE$_6$ ↓	minFDE$_6$ ↓	MR$_6$ ↓	minADE$_1$ ↓	minFDE$_1$ ↓	MR$_1$ ↓
HiVT+CUQDS	0.682	1.034	0.097	1.218	2.696	0.455
HiVT+DM	0.694	1.047	0.107	1.302	2.921	0.501

• With VS without the P control uncertainty calibration module: We further verify the efficacy of the P control uncertainty calibration module by removing the whole calibration process in the testing stage. As shown in Table. 3, by calibrating the output uncertainty of the base model by the P control uncertainty calibration module, the coverage rate of the calibrated uncertainty interval improves 35.10% on average comparing with without calibration.

• Gaussian process regression module VS direct modeling: To validate the effectiveness of our Gaussian process regression module, we replace the Gaussian process regression module with the self-attention based design [Mao *et al.*, 2023] to predict the variance of the output distribution and add the KLD loss term to reduce the estimated uncertainty during the training stage. We apply this setting on the base model HiVT. As shown in Table. 4, both of the methods achieves better prediction accuracy than without considering uncertainty in the base model. Our CUQDS slightly exceed the direct modeling method in all prediction accuracy evaluation metrics. The results prove that our CUQDS is sufficient in providing output uncertainty estimation for the trajectory prediction base model and improving the prediction accuracy.

5 Conclusion & Discussion

In this study, we present the framework CUQDS to estimate the output distribution for any existing trajectory prediction models under distribution shift, while improving the prediction accuracy and reduce the output uncertainty by introducing additional loss term. CUQDS adopts the Gaussian process regression module to model the output distribution of the base model. In addition, CUQDS utilizes a statistical-based P control module to calibrate the estimated uncertainty by considering the performance limitation of base model upon recent inputs. The experiment results demonstrate the efficacy of CUQDS in improving the prediction accuracy and reducing the prediction uncertainty. Our findings highlight the importance of quantifying the uncertainty of output trajectory in trajectory prediction under distribution shift. In the future work, we plan to extend our work to more state-of-the-art models, provide uncertainty of output trajectory in each time step, and further compare the performance difference of estimating uncertainty and output distribution by different designs.

References

[Angelopoulos *et al.*, 2023] Anastasios N Angelopoulos, Emmanuel J Candes, and Ryan J Tibshirani. Conformal pid control for time series prediction. *arXiv preprint arXiv:2307.16895*, 2023.

[Chai *et al.*, 2019] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

[Chang *et al.*, 2019] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Pro. of IEEE/CVF*, pages 8748–8757, 2019.

[Chen *et al.*, 2023] Guangyi Chen, Zhenhao Chen, Shunxing Fan, and Kun Zhang. Unsupervised sampling promoting for stochastic human trajectory prediction. In *Proc. of IEEE/CVF*, pages 17874–17884, 2023.

[Cui *et al.*, 2019] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 ICRA*, pages 2090–2096. IEEE, 2019.

[Deo and Trivedi, 2018] Nachiket Deo and Mohan M Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *Proc. of IEEE CVPR workshops*, pages 1468–1476, 2018.

[Erlygin *et al.*, 2023] Leonid Erlygin, Vladimir Zholobov, Valeriia Baklanova, Evgeny Sokolovskiy, and Alexey Zaytsev. Uncertainty estimation for time series forecasting via gaussian process regression surrogates. *SIGKDD 2023*, 2023.

[Feng *et al.*, 2021] Di Feng, Ali Harakeh, Steven L Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):9961–9980, 2021.

[Gu *et al.*, 2024] Xunjiang Gu, Guanyu Song, Igor Gilitschenski, Marco Pavone, and Boris Ivanovic. Producing and leveraging online map uncertainty in trajectory prediction. *Proc. of the IEEE/CVF CVPR*, 2024.

[Gupta *et al.*, 2018] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proc. of IEEE CVPR*, pages 2255–2264, 2018.

[Hu *et al.*, 2023] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proc. of the IEEE/CVF*, pages 17853–17862, 2023.

[Ivanovic *et al.*, 2022] Boris Ivanovic, Yifeng Lin, Shubham Shrivastava, Punarjay Chakravarty, and Marco Pavone. Propagating state uncertainty through trajectory forecasting. In *2022 ICRA*, pages 2351–2358. IEEE, 2022.

[Kedia *et al.*, 2023] Shubham Kedia, Yu Zhou, and Sambhu H Karumanchi. Integrated perception and planning for autonomous vehicle navigation: An optimization-based approach. In *Proc. of the IEEE/CVF*, pages 3205–3214, 2023.

[Kendall and Gal, 2017] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[Lekeufack *et al.*, 2023] Jordan Lekeufack, Anastasios N Angelopoulos, Andrea Bajcsy, Michael I. Jordan, and Jitendra Malik. Conformal decision theory: Safe autonomous decisions from imperfect predictions. *arXiv preprint arXiv:2310.05921*, 2023.

[Liang *et al.*, 2020] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *Computer Vision–ECCV 2020*, pages 541–556. Springer, 2020.

[Lindemann *et al.*, 2023] Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 2023.

[Liu *et al.*, 2021] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proc. of the IEEE/CVF Conference on CVPR*, pages 7577–7586, 2021.

[Mao *et al.*, 2023] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proc. of IEEE/CVF*, pages 5517–5526, 2023.

[Meyer and Thakurdesai, 2020] Gregory P Meyer and Niranjan Thakurdesai. Learning an uncertainty-aware object detector for autonomous driving. In *2020 IEEE/RSJ International Conference on IROS*, pages 10521–10527. IEEE, 2020.

[Ngiam *et al.*, 2021] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021.

[Phan-Minh *et al.*, 2020] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proc. of IEEE/CVF*, pages 14074–14083, 2020.

[Rasmussen *et al.*, 2006] Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006.

[Rhinehart *et al.*, 2018] Nicholas Rhinehart, Kris M Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proc. of ECCV*, pages 772–788, 2018.

[Rhinehart *et al.*, 2019] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proc. of IEEE/CVF*, pages 2821–2830, 2019.

[Salzmann *et al.*, 2020] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020*, pages 683–700. Springer, 2020.

[Shafer and Vovk, 2008] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[Shi *et al.*, 2023] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *arXiv preprint arXiv:2306.17770*, 2023.

[Tang and Salakhutdinov, 2019] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. *NeurIPS*, 32, 2019.

[Tang *et al.*, 2021] Bohan Tang, Yiqi Zhong, Ulrich Neumann, Gang Wang, Siheng Chen, and Ya Zhang. Collaborative uncertainty in multi-agent trajectory forecasting. *NeurIPS*, 34:6328–6340, 2021.

[Varadarajan *et al.*, 2022] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 ICRA*, pages 7814–7821. IEEE, 2022.

[Xu and Xie, 2023] Chen Xu and Yao Xie. Sequential predictive conformal inference for time series. In *International Conference on Machine Learning*, pages 38707–38727. PMLR, 2023.

[Xu *et al.*, 2014] Wenda Xu, Jia Pan, Junqing Wei, and John M Dolan. Motion planning under uncertainty for on-road autonomous driving. In *2014 IEEE ICRA*, pages 2507–2512. IEEE, 2014.

[Zeng *et al.*, 2021] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *2021 IEEE/RSJ International Conference on IROS*, pages 532–539. IEEE, 2021.

[Zhong *et al.*, 2022] Yiqi Zhong, Zhenyang Ni, Siheng Chen, and Ulrich Neumann. Aware of the history: Trajectory forecasting with the local behavior data. In *European Conference on Computer Vision*, pages 393–409. Springer, 2022.

[Zhou *et al.*, 2022] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proc. of IEEE/CVF*, pages 8823–8833, 2022.

[Zhou *et al.*, 2023] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proc. of the IEEE/CVF CVPR*, pages 17863–17873, 2023.

[Zhu *et al.*, 2023] Dekai Zhu, Guangyao Zhai, Yan Di, Fabian Manhardt, Hendrik Berkemeyer, Tuan Tran, Nassir Navab, Federico Tombari, and Benjamin Busam. Ipcc-tp: Utilizing incremental pearson correlation coefficient for joint multi-agent trajectory prediction. In *Proc. of IEEE/CVF*, pages 5507–5516, 2023.