CCTR
Calibrating Trajectory Prediction for Uncertainty-Aware Motion Planning in Autonomous Driving

Cao, Chengtai; Chen, Xinhong; Wang, Jianping; Song, Qun; Tan, Rui; Li, Yung Hui

DOI
10.1609/aaai.v38i19.30085

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the AAAI Conference on Artificial Intelligence

Citation (APA)
Cao, C., Chen, X., Wang, J., Song, Q., Tan, R., & Li, Y. H. (2024). CCTR: Calibrating Trajectory Prediction for Uncertainty-Aware Motion Planning in Autonomous Driving. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(19), 20949-20957. https://doi.org/10.1609/aaai.v38i19.30085

CCTR: Calibrating Trajectory Prediction for Uncertainty-Aware Motion Planning in Autonomous Driving

Chengtai Cao[1], Xinhong Chen[1], Jianping Wang[1], Qun Song[2], Rui Tan[3], Yung-Hui Li[4]
[1]City University of Hong Kong
[2]Delft University of Technology
[3]Nanyang Technological University
[4]Hon Hai Research Institute
{chengtcao2-c, xinhchen2-c}@my.cityu.edu.hk, jianwang@cityu.edu.hk

Abstract

Autonomous driving systems rely on precise trajectory prediction for safe and efficient motion planning. Despite considerable efforts to enhance prediction accuracy, inherent uncertainties persist due to data noise and incomplete observations. Many strategies entail formalizing prediction outcomes into distributions and utilizing variance to represent uncertainty. However, our experimental investigation reveals that existing trajectory prediction models yield unreliable uncertainty estimates, necessitating additional customized calibration processes. On the other hand, directly applying current calibration techniques to prediction outputs may yield sub-optimal results due to using a universal scaler for all predictions and neglecting informative data cues. In this paper, we propose Customized Calibration Temperature with Regularizer (CCTR), a generic framework that calibrates the output distribution. Specifically, CCTR 1) employs a calibration-based regularizer to align output variance with the discrepancy between prediction and ground truth and 2) generates a tailor-made temperature scaler for each prediction using a post-processing network guided by context and historical information. Extensive evaluation involving multiple prediction and planning methods demonstrates the superiority of CCTR over existing calibration algorithms and uncertainty-aware methods, with significant improvements of 11%-22% in calibration quality and 17%-46% in motion planning.

Introduction

Over the past decade, autonomous driving has achieved remarkable advancements, driven by substantial progress in key supporting technologies. Notably, trajectory prediction and motion planning have emerged as two pivotal elements within the autonomous driving software pipeline. The prediction module estimates the future locations of surrounding entities based on observed data, while the planning module uses these prediction outputs to derive a collision-free motion path. The sequential interdependence of these two modules raises a pressing concern that the prediction inaccuracy may compromise planning safety and even cause serious accidents. Therefore, addressing the uncertainty between these two modules is vital in ensuring overall driving safety.

Current works mainly deal with the uncertainty associated with trajectory prediction in two manners. On the one hand, end-to-end autonomous driving frameworks have been proposed to circumvent such uncertainty issues (Zeng et al. 2019, 2020; Sadat et al. 2020; Hu et al. 2021). For instance, Zeng et al. (2019) introduce an end-to-end interpretable neural motion planner that learns a cost volume representation to model uncertainty in scene forecasting. Regions with higher cost volume values are more likely to contain other agents or obstacles. However, the end-to-end frameworks introduce complexities in disentangling intertwined uncertainties from various input sources and incorporating individual state-of-the-art designs. On the other hand, some approaches output predictions in the form of a distribution such that uncertainty can be represented as the variance of the distribution. Prior works have employed Gaussian distributions (Alahi et al. 2016; Girgis et al. 2022; Nakamura and Bansal 2022) and Laplace distributions (Zhou et al. 2022, 2023) for such uncertainty modeling. Nevertheless, the absence of an explicit ground truth for variance and the lack of a dedicated loss term for uncertainty introduce potential unreliability in the predicted uncertainty (Feng et al. 2019). Thus, an additional calibration procedure is imperative to adjust the output variance appropriately.

Directly applying existing calibration techniques from other domains may yield sub-optimal results. For instance, when directly utilizing temperature-based calibration methods (Guo et al. 2017; Zhang, Kailkhura, and Han 2020; Feng et al. 2019), a global temperature scaler will be employed for all predictions, regardless of instructive contextual information. While a recent model introduces a variance scaling algorithm designed for real-time safety maintenance and updates (Nakamura and Bansal 2022), its resource-intensive confidence estimation and high-dimensional computations violate the prerequisites for calibration methods outlined in (Zhang, Kailkhura, and Han 2020). Consequently, its applicability in resource-constrained vehicles is constrained.

In this work, we introduce a plug-and-play framework – <u>C</u>ustomized <u>C</u>alibration <u>T</u>emperature with <u>R</u>egularizer (**CCTR**) – that effectively calibrates prediction uncertainty to improve motion planning performance. Specifically, CCTR comprises two modules: 1) a calibration-based regularizer that enforces variance to match the actual divergence between prediction and ground truth, ensuring the calibrated variances accurately capture uncertainty, and 2) a post-processing procedure that customizes temperature ratios to individually scale predicted variances, informed by context and historical information. For efficient post-processing network design, we empirically pinpoint key factors affecting calibration quality. Notably, CCTR does not impose any assumptions or constraints on prediction and planning modules. Therefore, it can be seamlessly integrated with any existing prediction and planning algorithms. Our contributions are summarized as follows:

- Our empirical analyses confirm that current trajectory prediction models are uncalibrated and require a customized calibration strategy. We further identify crucial factors that affect calibration, providing insights for elevating trustworthiness in autonomous driving.
- To effectively handle the uncertainty in trajectory predictions, we develop a general framework CCTR featuring a calibration-based regularizer and a post-processing procedure. CCTR adaptively scales predicted variances to provide more accurate and reliable location estimation of nearby agents for better motion planning.
- We extensively evaluate CCTR using five prediction models and two planning methods, showcasing its preeminence over several state-of-the-art baselines. Specifically, the results reveal that CCTR manifests an 11%-22% improvement in calibration quality and a 17%-46% boost in planning accuracy.

Motivation & Preliminaries

In this section, we first delve into the concept of quantile calibration and present our investigation of calibration quality with several state-of-the-art trajectory predictors, demonstrating they are intrinsically uncalibrated and require an adaptive calibration scheme. Subsequently, we detail calibration measurements and conduct empirical analyses to recognize specific factors that influence calibration.

Definition of Calibration

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output spaces, respectively. For a regression task, given an input instance $\boldsymbol{x}_i \in \mathcal{X}$, the probabilistic model $f : \mathcal{X} \to \mathcal{Y}$ generates a target prediction and its associated uncertainty, characterized by the mean $\hat{\boldsymbol{\mu}}_i$ and variance $\hat{\boldsymbol{\sigma}}_i$ of a predetermined distribution, respectively. In the context of trajectory prediction, each output corresponds to a prediction vector $\hat{\boldsymbol{\mu}}_i = [\hat{\mu}_i^1, \hat{\mu}_i^2]$ and an uncertainty vector $\hat{\boldsymbol{\sigma}}_i = [\hat{\sigma}_i^1, \hat{\sigma}_i^2]$, with horizontal and vertical coordinates indicated by $(\cdot)^1$ and $(\cdot)^2$, respectively. Model f is deemed well calibrated if the estimated uncertainty $\hat{\boldsymbol{\sigma}}_i$ exhibits a positive correlation with the likelihood of the predicted mean $\hat{\boldsymbol{\mu}}_i$ being incorrect. Following the prior works (Guo et al. 2017), we define *quantile calibration* based on ground truth $\boldsymbol{y}_i = [y_i^1, y_i^2]$.

Definition 1 *A trajectory predictor is perfectly quantile calibrated iff for all confidence levels* $p \in [0, 1]$*:*

$$C(p) = \frac{\sum_{i=1}^{M} \mathbb{I}\{\boldsymbol{y}_i \le Q^{-1}_{(\hat{\boldsymbol{\mu}}_i,\hat{\boldsymbol{\sigma}}_i)}(p)\}}{M} \to p \; (M \to \infty). \quad (1)$$

In the above definition, $\boldsymbol{y}_i \le Q^{-1}_{(\hat{\boldsymbol{\mu}}_i,\hat{\boldsymbol{\sigma}}_i)}(p)$ means $y_i^1 \le Q^{-1}_{(\hat{\mu}_i^1,\hat{\sigma}_i^1)}(p)$ and $y_i^2 \le Q^{-1}_{(\hat{\mu}_i^2,\hat{\sigma}_i^2)}(p)$; $\mathbb{I}\{\cdot\}$ is the indicator function; M is the number of samples; the *quantile function* $Q^{-1}_{(\hat{\mu}_i^1,\hat{\sigma}_i^1)}(p)$ of the distribution is given by:

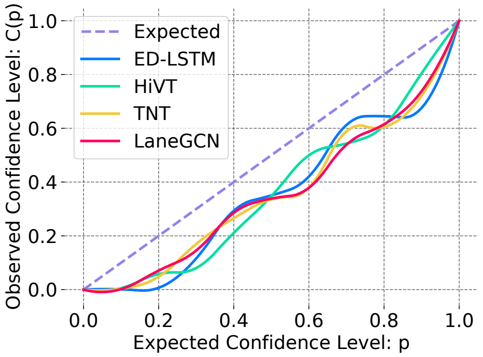

Figure 1: Calibration curves of predictors. The dashed diagonal line represents perfect calibration, while solid curves correspond to the calibration performance of the predictors.

$$Q^{-1}_{(\hat{\mu}_i^1,\hat{\sigma}_i^1)}(p) = \inf\{z : p \le Q_{(\hat{\mu}_i^1,\hat{\sigma}_i^1)}(z)\}, \quad (2)$$

where $Q_{(\hat{\mu}_i^1,\hat{\sigma}_i^1)}(z)$ is the cumulative distribution function (CDF) of the output distribution, determined by its mean $\hat{\mu}_i^1$ and variance $\hat{\sigma}_i^1$. Intuitively, perfect calibration implies that the ground truth $\boldsymbol{y}_i$ falls within a p confidence interval approximately p percent of the time.

Calibration Observations

We demonstrate the necessity of calibration by highlighting the gaps between the current trajectory prediction models and the perfectly calibrated reference. On the Argoverse dataset (Chang et al. 2019), we learn several predictors – ED-LSTM (Chang et al. 2019), HiVT (Zhou et al. 2022), TNT (Zhao et al. 2021), and LaneGCN (Liang et al. 2020) on the training set. Optimization of models involves minimizing the negative log-likelihood (NLL) loss function:

$$\mathcal{L}_{\text{NLL}} = -\log P(\boldsymbol{y}_i \mid \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i). \quad (3)$$

Upon convergence, we calculate $C(p)$ via Eq. (1) on 10,000 scenarios sampled from the validation set.

The apparent gaps between the dashed line and the solid curves in Figure 1 indicate that the prediction approaches lack proper calibration. Furthermore, all the solid curves reside beneath the dashed line, signifying over-confidence in the predictions, i.e., the predicted variance is smaller than the expected uncertainty. Accordingly, planning with over-confident estimation may generate collision-prone paths. As such, learning a scaler that enlarges the variance, akin to the existing temperature scaling methods (Guo et al. 2017; Levi et al. 2022; Zhang, Kailkhura, and Han 2020), seems to be a promising calibration remedy. However, our in-depth scrutiny of individual predictions reveals limitations in these approaches: even though most predictions display over-confidence, approximately 14.37% exhibit under-confidence (i.e., prediction variance is larger than expected). This under-confidence unduly restricts motion planning, resulting in impractical routes or even filtering feasible paths.

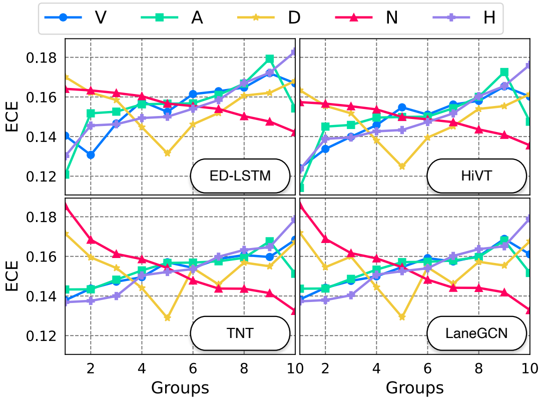

Figure 2: Effect of velocity (V), acceleration (A), distance to the nearest actor (D), number of cars around (N), and forecast horizon (H) on calibration, as measured by expected calibration error (ECE). The non-parallel curves suggest a relationship between ECE and the identified factors.

Hence, an imperative emerges for a customized temperature scaling scheme that can both increase and reduce variance, going beyond the limitations of a uniform temperature ratio.

Calibration Performance Metrics

To quantify the calibration performance, following the previous works (Cui, Hu, and Zhu 2020; Levi et al. 2022), we compute Expected Calibration Error (ECE), Maximum Calibration Error (MCE), and Normalized Calibration Error (NCE) based on a sampled confidence level set $S = \{p_1, \ldots, p_s, \ldots, p_{|S|}\}$ with a size of $|S|$ as follows:

$$\mathrm{ECE} = \frac{1}{|S|}\sum_{s=1}^{|S|}(|C(p_s) - p_s|), \tag{4}$$

$$\mathrm{MCE} = \max_{p_s \in S}(|C(p_s) - p_s|), \tag{5}$$

$$\mathrm{NCE} = \frac{1}{M}\sum_{i=1}^{M}\frac{\|(\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_i) \odot (\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_i) - \hat{\boldsymbol{\sigma}}_i\|_2}{\|\hat{\boldsymbol{\sigma}}_i\|_2}, \tag{6}$$

where $C(p_s)$ is calculated using Eq. (1), $\odot$ represents the element-wise product, and $\|\cdot\|_2$ is the ℓ_2 norm. Intuitively, as depicted in Figure 1, ECE measures the area between the dashed line corresponding to perfect calibration and a solid curve obtained for a predictor; MCE corresponds to the maximum deviation between the dashed line and the solid curve; NCE assesses the disagreement between the square error of the prediction $\hat{\boldsymbol{\mu}}_i$ and the estimated uncertainty $\hat{\boldsymbol{\sigma}}_i$.

Factors Affecting Calibration Performance

We aim to develop an effective and efficient method for calibrating trajectory predictions to enhance their reliability. Given the intricacies of autonomous driving, it is crucial to identify specific factors that affect the calibration of predictors. Through experiments on the same 10,000 scenarios, we pinpoint five decisive factors: velocity (V), acceleration (A), distance to the nearest actor (D), number of cars around (N), and forecast horizon (H). To understand the impact of these factors, we divide the range of each factor observed among the 10,000 scenarios into ten groups evenly and compute the ECE for the scenarios falling into each group. Figure 2 shows the results of all predictors, with the group index increasing with the average value of the concerned factor. Each curve is not parallel to the horizontal axis, indicating the latent relationship between calibration measurement ECE and the identified factors. A general increasing trend in ECE is evident for velocity, acceleration, and forecast horizon. This aligns with our expectations, as higher values in these factors introduce more complexity and uncertainty in the prediction task. In the case of the distance to the nearest actor, we observe a U-shaped relationship, suggesting that extreme closeness (potentially leading to stopping scenarios) or long distance (providing high driving freedom) both pose challenges to prediction, resulting in poorer calibration. As for the number of cars around, ECE exhibits a clear decreasing trend. This implies that a reduced number of nearby vehicles leads to increased driving options and more uncertainty.

Methodology

This section elaborates on CCTR, which consists of two main components: 1) a calibration loss-based regularizer that can be incorporated into any prediction model training to guide the variance to align with the discrepancy between prediction and ground truth, and 2) a subsequent post-processing procedure, informed by the calibration insights from the previous section, that learns tailored temperature coefficients for prediction outputs. Figure 3 overviews the design of CCTR, where numbered text boxes indicate the sequence of operations within CCTR.

Calibration Regularizer

When trained with NLL loss defined in Eq. (3), the trajectory predictor learns to output variances in an unsupervised manner since there is no explicit ground truth for variances. Consequently, this loss formulation does not inherently ensure well-calibrated uncertainty. The NCE metric shown in Eq. (6) suggests that, for each sample $\boldsymbol{x}_i$, the predicted variance $\hat{\boldsymbol{\sigma}}_i$ should match the actual difference between prediction and ground truth. In this regard, we introduce a calibration-oriented regularizer $\mathcal{L}_{\mathrm{CAL}}$ to guide the variance:

$$\mathcal{L}_{\mathrm{CAL}} = \|(\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_i) \odot (\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_i) - \hat{\boldsymbol{\sigma}}_i\|_2. \tag{7}$$

Accordingly, the loss function $\mathcal{L}_{\mathrm{CCTR}}$ for trajectory prediction models in our CCTR is:

$$\mathcal{L}_{\mathrm{CCTR}} = \mathcal{L}_{\mathrm{NLL}} + \lambda \cdot \mathcal{L}_{\mathrm{CAL}}. \tag{8}$$

The hyper-parameter λ controls the effect of calibration loss $\mathcal{L}_{\mathrm{CAL}}$, forcing the predicted variances to accurately capture the deviation between prediction and ground truth for improved calibration performance.

Post-Processing Procedure

Following (Guo et al. 2017), we develop a post-processing algorithm to calibrate the obtained distributions from the predictor. Building on our findings in the previous section,

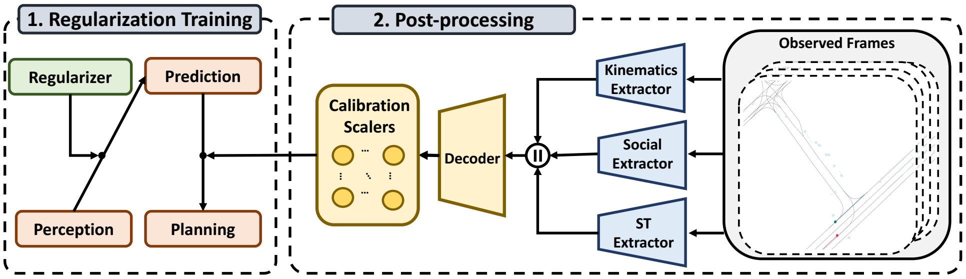

Figure 3: Framework of our CCTR, where ST is the abbreviation of spatiotemporal and || is the concatenation operation. The numbered text boxes illustrate the sequence of operations in CCTR.

we propose to consider the instructive context information within each driving scene and derive a customized temperature scaler to calibrate the uncertainty in each prediction.

We start by generating Bird's Eye View (BEV) images of observed frames based on map information and track records, compressing spatiotemporal knowledge and lane constraints into a 4-dimensional input matrix $\mathbb{M}_{\text{in}} \in \mathbb{R}^{F_i \times \mathbb{W} \times \mathbb{H} \times \mathbb{C}}$. Here, F_i, $\mathbb{W}$, $\mathbb{H}$, and $\mathbb{C}$ represent the number of observed frames, the width and height of the image, and the number of channels in the image. This input matrix facilitates the extraction of kinematic and social features $\mathbb{M}_{\text{ks}}$, as well as spatiotemporal information $\mathbb{M}_{\text{st}}$.

Kinematics & Social Features: We extract kinematic and social factors crucial to calibration for each agent in observed frames, including velocity, acceleration, distance to the nearest actor (both in front and behind), and the count of nearby cars. This results in $\mathbb{M}_{\text{ks}} \in \mathbb{R}^{L \times 5F_i}$, where L is the number of agents.

Spatial & Temporal Information: To incorporate surrounding knowledge and lane restrictions for calibration, we employ a spatiotemporal (ST) extractor that learns a comprehensive and concise representation of environmental information. We combine convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for spatial and sequential information extraction, respectively (Donahue et al. 2015; Zhang et al. 2019). A CNN is first applied to observed frames $\mathbb{M}_{\text{in}}$, obtaining $\mathbb{M}_{\text{sp}} \in \mathbb{R}^{F_i \times d_1}$ which embeds spatial knowledge of each BEV image into a d_1-dimensional vector. Then, an RNN with the gated recurrent unit (GRU) (Cho et al. 2014) is utilized for the representations of observed frames. Each row $\boldsymbol{z}_t \in \mathbb{R}^{d_1} (t \in \{1, \ldots, F_i\})$ in $\mathbb{M}_{\text{sp}}$ serves as the input of GRU. After the update of GRU, the final d_2-dimensional hidden state $\boldsymbol{h}_{F_i} \in \mathbb{R}^{d_2}$ represents the spatiotemporal information representation. As all agents in the same scenario share a single spatiotemporal embedding, we replicate it for L times to form $\mathbb{M}_{\text{st}} \in \mathbb{R}^{L \times d_2}$.

Decoder: We concatenate the matrices $\mathbb{M}_{\text{ks}}$ and $\mathbb{M}_{\text{st}}$ to yield the final representation that encodes instructive kinematics, social, and spatiotemporal information. This representation is then fed to a linear layer with Softplus activation as a decoder to output tailored temperature ratios:

$$\mathbb{M}_{\text{tm}} = \text{Decoder}(\mathbb{M}_{\text{ks}} \mid\mid \mathbb{M}_{\text{st}}), \tag{9}$$

where || denotes concatenation, $\mathbb{M}_{\text{tm}} \in \mathbb{R}^{L \times 2F_o}$ is the output matrix, and F_o is the prediction step size.

Training: Our calibration model is trained on the validation set by minimizing the NLL with rescaled variance $\mathcal{L}_{\text{RNLL}}$:

$$\mathcal{L}_{\text{RNLL}} = -\log P(\boldsymbol{y}_i \mid \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i \odot \boldsymbol{\gamma}_i), \tag{10}$$

where $\boldsymbol{\gamma}_i$ is the corresponding temperature ratios from $\mathbb{M}_{\text{tm}}$.

Discussion

CCTR satisfies all three desiderata for calibration methods stated in (Zhang, Kailkhura, and Han 2020):

- **Accuracy Preservation**: CCTR achieves comparable or even improved prediction performance with an appropriate hyperparameter λ, compared to the original predictor implementations.
- **Data Efficiency**: CCTR is a parametric model with only 1.5 Mb size, which theoretically requires less data to converge (Zhang, Kailkhura, and Han 2020). Our evaluation with varying validation data sizes further supports this advantage.
- **Expressiveness**: CCTR exhibits much better performance than calibration baselines when exposed to more data, owing to its capacity to effectively leverage available data to produce contextually-informed distribution-specific temperature scalers.

Furthermore, CCTR can be utilized with diverse spatiotemporal extractors, e.g., 3D CNN (Ji et al. 2012), and be integrated with any prediction and planning approaches.

Experiments

This section presents the experimental evaluation of CCTR against several prevailing calibration methods and uncertainty-aware baselines to demonstrate its superiority in uncertainty estimation and downstream planning tasks. Additionally, we investigate the contribution of each component within CCTR and empirically validate how CTTR fulfills the three desiderata for calibration methods.

Experimental Settings

Dataset: We conduct our experiments on the Argoverse dataset (Chang et al. 2019), which provides agent trajectories with high-definition map data. The dataset consists of 205,942 training and 39,472 validation scenarios. As the

Model	Method	ECE	MCE	NCE	ℓ_2 error (RRT*)	ℓ_2 error (NMP)	ADE	FDE
ED-LSTM	Original	0.1567	0.2240	0.1326	8.6258	3.4711	**1.74**	3.99
	TS	0.1540	0.2184	0.1291	6.7297	2.6980	\	\
	IR	0.1535	0.2180	0.1289	6.6031	2.4724	\	\
	ETS	0.1493	0.2080	0.1204	5.5453	2.3090	\	\
	CCTR	**0.1335**	**0.2005**	**0.1030**	**4.6017**	**2.1459**	**1.74**	**3.95**
HiVT	Original	0.1488	0.2196	0.1286	7.8968	3.1290	**0.69**	1.04
	TS	0.1425	0.2142	0.1197	6.3471	2.5821	\	\
	IR	0.1402	0.2115	0.1188	6.1556	2.4166	\	\
	ETS	0.1406	0.2087	0.1178	5.8601	2.2802	\	\
	CCTR	**0.1295**	**0.1986**	**0.0984**	**4.3310**	**2.0102**	**0.69**	**1.03**
TNT	Original	0.1536	0.2205	0.1318	8.2737	3.3691	0.93	**1.69**
	TS	0.1512	0.2195	0.1284	6.4402	2.5418	\	\
	IR	0.1518	0.2182	0.1284	6.1299	2.4328	\	\
	ETS	0.1510	0.2131	0.1232	5.4753	2.3870	\	\
	CCTR	**0.1313**	**0.1998**	**0.1025**	**4.5503**	**2.2002**	**0.92**	**1.69**
LaneGCN	Original	0.1542	0.2213	0.1307	8.0139	3.3217	0.72	**1.10**
	TS	0.1528	0.2183	0.1285	6.7524	2.6032	\	\
	IR	0.1496	0.2114	0.1271	6.6016	2.5814	\	\
	ETS	0.1513	0.2165	0.1272	6.1427	2.3995	\	\
	CCTR	**0.1335**	**0.2017**	**0.1028**	**4.5211**	**2.1259**	**0.71**	**1.10**
AutoBots	Original	0.1550	0.2205	0.1301	7.6814	3.3747	0.73	1.10
	TS	0.1531	0.2162	0.1279	6.5682	2.7313	\	\
	IR	0.1494	0.2108	0.1264	6.4779	2.6652	\	\
	ETS	0.1461	0.2071	0.1255	5.9251	2.3890	\	\
	CCTR	**0.1328**	**0.2003**	**0.1020**	**4.3785**	**2.2348**	**0.72**	**1.09**

Table 1: Calibration (third column), planning (fourth column), and prediction (last column) performance comparisons among calibration algorithms. The bold values denote the best performance and \ means it does not affect the prediction results.

ground truth is unavailable for the testing example, we separate 10,000 training samples for testing purposes. The goal is to output the future movements of agents for the next three-second given the first two-second trajectories.

Calibration Baselines: To show the CCTR's superiority in uncertainty calibration, we compare it with the following prevalent calibration methods:

- *Temperature Scaling (TS)* (Levi et al. 2022), which simply utilizes a global temperature to scale the variance.
- *Isotonic Regression (IR)* (Kuleshov, Fenner, and Ermon 2018), which trains an auxiliary model based on isotonic regression to fit $C(p)$ defined in Eq. (1).
- *Ensemble Temperature Scaling (ETS)* (Zhang, Kailkhura, and Han 2020), which learns a mixture of uncalibrated, TS-calibrated, and uniform probabilistic outputs.

Prediction & Planning Methods: To indicate CCTR's adaptability, we integrate it with five state-of-the-art trajectory prediction models: ED-LSTM (Chang et al. 2019), HiVT (Zhou et al. 2022), TNT (Zhao et al. 2021), LaneGCN (Liang et al. 2020), and AutoBots (Girgis et al. 2022), along with two planning algorithms: sample-based RRT* (Kuffner and LaValle 2000) and imitation learning-based NMP (Zeng et al. 2019; Hu et al. 2021).

Metrics: For calibration assessment, we use three widely used measurements: ECE, MCE, and NCE defined in Eq. (4), Eq. (5), and Eq. (6), respectively. For motion planning evaluation, we employ the ℓ_2 distance between corresponding waypoints of the planned trajectory and the ground truth of human driving. Regarding prediction accuracy, we follow the benchmark and utilize Average Displacement Error (ADE) and Final Displacement Error (FDE).

Implementation Details: Our post-processing model is trained for 50 epochs using Adam optimizer (Kingma and Ba 2015). The batch size is set to 128, with an initial learning rate of 5×10^{-4} and halved every ten epochs. The 3-layer CNN uses 3×3 convolution filters, with filter counts of 8, 16, and 32, respectively. Each convolution layer is followed by a linear layer with ReLU activation and a 4×4 max-pooling layer. The hyper-parameter λ, hidden size, number of layers, and the dropout rate of GRU are 0.1, 128, 3, and 0.1, respectively. We find that our model is relatively insensitive to most hyperparameters, except for the GRU's hidden size. We test a range of values, including $\{32, 64, 128, 256, 512\}$, and observe that the performance improves up to a hidden size of 128. Beyond this point, we notice similar or worse performance due to overfitting. As a result, we set the hidden size of GRU to 128. We use 5 random seeds to conduct experiments and we find that all methods are insensitive to the selection of random seeds. Consequently, we only report the mean value and ignore the standard deviation for simplicity. For uncertainty-aware planning, the uncertainty estimation from the trajectory prediction module with or without calibration assumes the oval region with one variance as the possible location of surrounding vehicles.

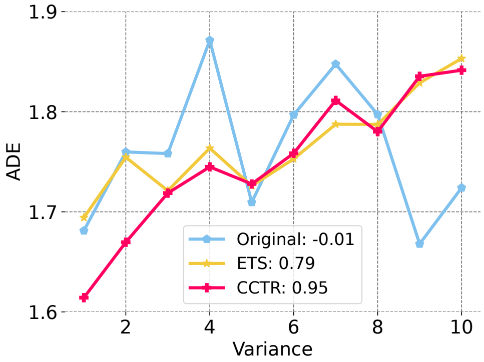

Figure 4: The relationship (with correlation coefficient) between ADE and variance. CCTR demonstrates a clear positive correlation between these two variables.

Method	ℓ_2 error	Training (hr)	Inference (ms)
MCDropout	3.8396	**1.9**	256
DE	4.0127	3.4	149
E2E	2.3012	4.3	217
FF	2.4184	4.1	201
CCTR-EN	**2.1459**	2.1	**137**

Table 2: Planning comparisons with end-to-end planners and uncertainty-aware methods.

Results

Calibration Comparison: The calibration comparisons are presented in the third column of Table 1. CCTR consistently outperforms all calibration baselines by a significant margin. For example, with the ED-LSTM predictor, CCTR reduces the ECE by 15% and 11% compared with the original prediction and the most potent baseline (ETS), respectively. This empirical evidence demonstrates that CCTR, which generates customized temperature ratios based on critical factors in the dynamic environment, is more effective than all baselines in achieving more calibrated predictions.

Moreover, we compare CCTR with baselines on their ability to signal unreliable outputs. To this end, we sort the variances in ascending order, divide them into ten groups, calculate the average ADE for each, and plot ADE against variance using the ED-LSTM model in Figure 4. For clarity, we only present the curve for ETS as it consistently performs the best among the baselines. The original trajectory prediction with a correlation coefficient of -0.01 displays no correlation between variance (uncertainty) and ADE (precision). Calibration by ETS partially mitigates this issue, showing a better correlation with a correlation coefficient of 0.79. In contrast, the curve of CCTR is a roughly increasing polyline with a correlation coefficient of 0.95, indicating a positive correlation between the two variables. In other words, for models calibrated using CCTR, higher variance implies a likely erroneous prediction. Hence, additional mechanisms can be designed based on variance to proactively avert potential danger caused by uncertain predictions.

Planning Performance: The fourth column of Table 1 highlights CCTR's superior performance over all calibration baselines in motion planning. Specifically, using the ED-LSTM predictor and RRT* planners, CCTR decreases the ℓ_2 error by 46% and 17% compared to the original prediction and the most effective baseline (ETS), respectively. This affirms that calibrated predictions are beneficial for subsequent planning. For a comprehensive evaluation, we combine CCTR with ED-LSTM predictor and NMP planner, obtaining CCTR-EN, and compare it with 1) two leading prediction uncertainty-aware planners – FF (Hu et al. 2021) and E2E (Zeng et al. 2019), and 2) two popular uncertainty capture methods without calibration – MCDropout (Gal and Ghahramani 2016) and Deep Ensemble (DE) (Lakshminarayanan, Pritzel, and Blundell 2017). The results in Table 2 demonstrate that: 1) CCTR-EN achieves the minimum ℓ_2 error, showcasing the strength of calibrating uncertainty estimation. Moreover, the ℓ_2 error can be further reduced with improved prediction and planning algorithms (e.g., 2.0102 with HiVT); 2) CCTR-EN converges quickly since it only requires adding a regularization term in prediction model training and learning a small model on the validation set. In contrast, end-to-end models exhibit slow convergence due to intermediate agents and multi-task objectives; 3) For inference time, CCTR-EN introduces a negligible delay as it only adds the forward time of calibration models, while other methods require time-consuming operations, such as multiple predictions of MCDropout.

Method	ECE	MCE	NCE
CCTR-KI	0.1395	0.2071	0.1193
CCTR-SO	0.1402	0.2089	0.1216
CCTR-ST	0.1420	0.2091	0.1133
CCTR-RE	0.1367	0.2013	0.1198
CCTR-PP	0.1468	0.2112	0.1086
CCTR	**0.1335**	**0.2005**	**0.1030**

Table 3: Ablation study for CCTR with ED-LSTM.

Ablation Results: CCTR comprises four fundamental components: kinematics feature extractor (KI), social feature extractor (SO), spatiotemporal feature extractor (ST), and regularizer (RE). To assess the impact of each component on calibration performance, we implement four variants of CCTR: CCTR-KI, CCTR-SO, CCTR-ST, and CCTR-RE, where each variant removes the corresponding component. Additionally, we remove the entire post-processing (PP) module, resulting in the variant CCTR-PP, to emphasize its contribution. Table 3 presents the results of all variants with the ED-LSTM predictor. The noticeable performance gap between all variants and CCTR confirms the efficacy of each respective component. The worse result of CCTR-ST suggests that the spatiotemporal information ignored by previous works is crucial for improvement. The inferior results of CCTR-KI and CCTR-SO align with our calibration observations that kinematics and social features are key factors in uncertainty calibration.

Three Merits of CCTR: As mentioned before, CCTR meets all three desiderata for calibration approaches. Now, we provide experimental evidence as follows:

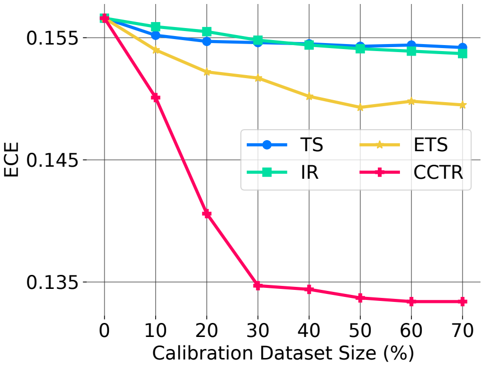

Figure 5: ECE comparisons of various calibration methods with validation data size.

– *Accuracy Preservation*: The last column of Table 1 demonstrates that, although CCTR introduces a new regularization loss, it manages to maintain and even boost the prediction performance of multiple trajectory predictors.
– *Data Efficiency*: We examine the influence of calibration sample size on calibration performance with ED-LSTM in Figure 5. CCTR is data-efficient, requiring only a few calibration samples to achieve decent calibration performance.
– *Expressiveness*: A more expressive method should yield a lower ECE when sufficient calibration samples are available. Figure 5 shows a substantial gap between CCTR and other baselines when more data is available for calibration. Additionally, the temperature ratios obtained by CCTR are more customized, with approximately 13% of generated ratios being less than one (i.e., decreasing the variance).

Related Work

As most current trajectory prediction models are deep neural networks (DNNs) (Huang et al. 2022), we first review the literature on capturing and calibrating uncertainty in traditional DNNs and then elaborate on specific methods in existing prediction and planning (Pred & Plan) approaches.

Uncertainty & Calibration in DNNs

As suggested by Kendall and Gal (2017), DNNs exhibit two major uncertainties: epistemic and aleatoric uncertainty. To address epistemic uncertainty, Bayesian learning offers a mathematically grounded framework, and several Bayesian approximation methods have been designed to estimate uncertainty, such as MCDropout (Gal and Ghahramani 2016). Moreover, with the help of ensemble learning, Deep Ensemble (Lakshminarayanan, Pritzel, and Blundell 2017) integrates multiple models for uncertainty estimation.

On the other hand, for aleatoric uncertainty estimation, Kendall and Gal (2017) propose a unified Bayesian learning-based approach that directly maps input data to aleatoric uncertainty estimations. Our framework aligns with this direction, where the uncertainty is represented as the predicted variance of the output distribution. However, the absence of effective distribution calibration yields unreliable uncertainty estimations and further harms downstream tasks (Guo et al. 2017; Wang et al. 2021). For instance, CNNs exhibit over-confidence (Guo et al. 2017) while Graph Neural Networks display under-confidence (Wang et al. 2021). To mitigate this problem, temperature scaling-based methods are proposed to refine model outputs. Nevertheless, the temperature ratio in these approaches is unified for all outcomes, restricting calibration efficacy. Other methods, like isotonic regression-based algorithms, have been developed to calibrate neural network regressors in a more refined way (Kuleshov, Fenner, and Ermon 2018). However, they neglect the informative context of data and lack generalizability. To address these issues, our CCTR leverages crucial contextual information to learn a *tailored* temperature ratio for each prediction, improving calibration performance.

Uncertainty & Calibration in Pred & Plan

In addition to the above DNN-oriented approaches, specific efforts have been made to incorporate uncertainty into trajectory prediction for motion planning (Liu et al. 2023), which can be categorized into two groups. A line of existing work involves using surrogates to model the uncertainty, such as cost volumes (Zeng et al. 2020), semantic occupancies (Sadat et al. 2020), and freespace (Hu et al. 2021). However, these approaches require integrating the prediction and planning components in an end-to-end framework, sacrificing the benefit of modularity.

Another line of work adheres to the traditional stack where prediction and planning are sequential modules with some mechanisms designed between them to estimate uncertainty. For example, Wu, Huang, and Lv (2022) consider variance in the predicted distribution as uncertainty and recklessly incorporate it into planning. However, uncalibrated distribution variance can be unreliable for downstream tasks. Building on this, Nakamura and Bansal (2022) further propose to scale variances for safe planning. However, their approach is not data-efficient due to resource-intensive confidence estimation and high-dimensional reachable set computations, which limit its practical applicability. To address these limitations, our CCTR directly calibrates the distribution variance of the predicted outputs, aiming to efficiently achieve more precise uncertainty estimation for the subsequent planning task.

Conclusion

This paper presents a novel CCTR framework to address the challenge of proper uncertainty calibration in trajectory prediction models, improving their reliability. CCTR offers a solution by introducing a calibration-oriented regularizer to align predicted variances with ground truth divergence and generating tailor-made temperature scalers for each prediction based on context and historical information. Extensive experiments demonstrate the superiority of CCTR over various baselines in uncertainty estimation and downstream planning tasks, leading to better-calibrated predictions and more trustworthy planning. Moreover, the ablation studies show the effectiveness of each component, with in-depth empirical analysis verifying CCTR's desirable properties. Future work can exploit more advanced post-processing modules to further improve calibration quality.

Acknowledgments

The work is supported in part by a project from the Hong Kong Research Grant Council under GRF 11210622 and in part by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023-006).

References

Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; and Savarese, S. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 961–971.

Chang, M.-F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. 2019. Argoverse: 3d Tracking and Forecasting with Rich Maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8748–8757.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1724–1734.

Cui, P.; Hu, W.; and Zhu, J. 2020. Calibrated Reliable Regression Using Maximum Mean Discrepancy. In *Advances in Neural Information Processing Systems*, 17164–17175.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2625–2634.

Feng, D.; Rosenbaum, L.; Glaeser, C.; Timm, F.; and Dietmayer, K. 2019. Can We Trust You? On Calibration of a Probabilistic Object Detector for Autonomous Driving. arXiv:1909.12358.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning*, 1050–1059.

Girgis, R.; Golemo, F.; Codevilla, F.; Weiss, M.; D'Souza, J. A.; Kahou, S. E.; Heide, F.; and Pal, C. 2022. Latent Variable Sequential Set Transformers for Joint Multi-Agent Motion Prediction. In *International Conference on Learning Representations*.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, 1321–1330.

Hu, P.; Huang, A.; Dolan, J.; Held, D.; and Ramanan, D. 2021. Safe Local Motion Planning with Self-Supervised Freespace Forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12732–12741.

Huang, Y.; Du, J.; Yang, Z.; Zhou, Z.; Zhang, L.; and Chen, H. 2022. A Survey on Trajectory-Prediction Methods for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 7(3): 652–674.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 221–231.

Kendall, A.; and Gal, Y. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems*, 5574–5584.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Kuffner, J. J.; and LaValle, S. M. 2000. RRT-Connect: An Efficient Approach to Single-Query Path Planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 995–1001.

Kuleshov, V.; Fenner, N.; and Ermon, S. 2018. Accurate Uncertainties for Deep Learning Using Calibrated Regression. In *International Conference on Machine Learning*, 2796–2804.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Advances in Neural Information Processing Systems*, 6402–6413.

Levi, D.; Gispan, L.; Giladi, N.; and Fetaya, E. 2022. Evaluating and Calibrating Uncertainty Prediction in Regression Tasks. *Sensors*, 22(15): 5540.

Liang, M.; Yang, B.; Hu, R.; Chen, Y.; Liao, R.; Feng, S.; and Urtasun, R. 2020. Learning Lane Graph Representations for Motion Forecasting. In *European Conference on Computer Vision*, 541–556.

Liu, X.; Jiao, R.; Wang, Y.; Han, Y.; Zheng, B.; and Zhu, Q. 2023. Safety-Assured Speculative Planning with Adaptive Prediction. arXiv:2307.11876.

Nakamura, K.; and Bansal, S. 2022. Online Update of Safety Assurances Using Confidence-based Predictions. arXiv:2210.01199.

Sadat, A.; Casas, S.; Ren, M.; Wu, X.; Dhawan, P.; and Urtasun, R. 2020. Perceive, Predict, and Plan: Safe Motion Planning through Interpretable Semantic Representations. In *European Conference on Computer Vision*, 414–430.

Wang, X.; Liu, H.; Shi, C.; and Yang, C. 2021. Be Confident! Towards Trustworthy Graph Neural Networks via Confidence Calibration. In *Advances in Neural Information Processing Systems*, 23768–23779.

Wu, J.; Huang, Z.; and Lv, C. 2022. Uncertainty-Aware Model-Based Reinforcement Learning: Methodology and Application in Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 8(1): 194–203.

Zeng, W.; Luo, W.; Suo, S.; Sadat, A.; Yang, B.; Casas, S.; and Urtasun, R. 2019. End-to-End Interpretable Neural Motion Planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8660–8669.

Zeng, W.; Wang, S.; Liao, R.; Chen, Y.; Yang, B.; and Urtasun, R. 2020. Dsdnet: Deep Structured Self-Driving Network. In *European Conference on Computer Vision*, 156–172.

Zhang, C.; Song, D.; Chen, Y.; Feng, X.; Lumezanu, C.; Cheng, W.; Ni, J.; Zong, B.; Chen, H.; and Chawla, N. V. 2019. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1409–1416.

Zhang, J.; Kailkhura, B.; and Han, T. Y.-J. 2020. Mix-n-Match: Ensemble and Compositional Methods for Uncertainty Calibration in Deep Learning. In *International Conference on Machine Learning*, 11117–11128.

Zhao, H.; Gao, J.; Lan, T.; Sun, C.; Sapp, B.; Varadarajan, B.; Shen, Y.; Shen, Y.; Chai, Y.; Schmid, C.; et al. 2021. Tnt: Target-Driven Trajectory Prediction. In *Conference on Robot Learning*, 895–904.

Zhou, Z.; Wang, J.; Li, Y.-H.; and Huang, Y.-K. 2023. Query-Centric Trajectory Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17863–17873.

Zhou, Z.; Ye, L.; Wang, J.; Wu, K.; and Lu, K. 2022. Hivt: Hierarchical Vector Transformer for Multi-Agent Motion Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8823–8833.