Temporal Multimodal Probabilistic Transformers for Safety Monitoring in Autonomous Driving Systems

Yehia Ahmed YEHIAHESHAM.AHMED@GMAIL.COM
Technical University of Munich, Germany

Felippe Roza FELIPPE.SCHMOELLER.DA.ROZA@IKS.FRAUNHOFER.DE
Fraunhofer IKS, Munich, Germany

Núria Mata NURIA.MATA@IKS.FRAUNHOFER.DE
Fraunhofer IKS, Munich, Germany

Editor: Khuong An Nguyen, Zhiyuan Luo, Harris Papadopoulos, Tuwe Löfström, Lars Carlsson and Henrik Boström

Abstract

Ensuring reliable safety monitoring in autonomous driving systems (ADS) under uncertainty is essential for deployment in real-world scenarios. We propose the Temporal Multimodal Probabilistic Transformer (TMPT), a novel deep learning framework that integrates uncertainty quantification (UQ) into lane-keeping safety monitoring. TMPT forecasts lane deviation metrics along with calibrated aleatoric and epistemic uncertainties by processing sequences of multimodal sensor and control data. Our framework combines Transformer-based temporal fusion with deep ensembles and post-hoc calibration to improve predictive accuracy and uncertainty estimation. We evaluate 24 model variants in the CARLA simulator, analyzing the impact of architecture, calibration, and ensembling on both prediction and uncertainty. Calibrated models achieve near-perfect uncertainty reliability (ENCE $<$ 0.03), while uncalibrated models show sharper predictions but overconfident errors. Ensemble methods further improve robustness but incur significant computational cost. Our findings show that aligning model selection with application context—balancing precision, calibration, and efficiency—is critical for safe and practical ADS deployment.

Keywords: Autonomous driving, Safety monitoring, Transformers, Probabilistic deep learning, Uncertainty quantification, Calibration

1. Introduction

Autonomous Driving Systems (ADS) are rapidly advancing, promising safer and more efficient transportation. However, their deployment in complex, real-world environments remains hindered by the challenge of ensuring reliable safety monitoring under uncertainty Yurtsever et al. (2020). Lane-keeping—the ability to maintain position within lane boundaries—is a fundamental safety requirement. Failures in lane-keeping are a leading cause of accidents and financial losses Grewal et al. (2025).

Deep Neural Networks (DNNs) have achieved impressive results in perception and control, but their deterministic outputs often lack reliable confidence measures He and Jiang (2023). Overconfident predictions, especially in edge cases or out-of-distribution (OOD) scenarios, can lead to catastrophic failures Willers et al. (2020). Uncertainty Quantification (UQ) has thus emerged as a critical requirement for safety-critical ADS, enabling risk-aware decision-making and proactive interventions.

This paper presents a unified probabilistic deep learning framework for lane-keeping safety monitoring. We propose the Temporal Multimodal Probabilistic Transformer (TMPT), which fuses multimodal sensor, visual, and control data to forecast lane deviation metrics and their associated uncertainties. Both aleatoric (data-related) and epistemic (model-related) uncertainties are estimated, and model calibration is performed to ensure reliable confidence outputs. We demonstrate, through extensive simulation-based experiments, that calibrated models achieve dramatic reductions in Expected Normalized Calibration Error (ENCE), providing trustworthy uncertainty estimates essential for real-world deployment.

2. Related Work

Developing safety in-service monitoring techniques has been a key focus of recent research efforts Henriksson et al. (2019); Hussain et al. (2022). Approaches such as SelfOracle Stocco et al. (2019), DeepRoad Zhang et al. (2018), and DeepGuard Hussain et al. (2022) analyze real-world driving data to identify anomalous behavior in ADS. However, these techniques are predominantly black-box, relying on input/output traces and lacking introspection into deep neural network (DNN) internals Humbatova et al. (2020). Although white-box methods like ThirdEye Stocco et al. (2022) have been proposed, they are often computationally expensive and less suited for real-time, resource-constrained environments.

Evidential deep learning Sensoy et al. (2018) and early internal consistency checks Weiss and Tonella (2021) attempt to balance interpretability with performance, but often assume static distributions or suffer from instability under domain shift. Our research builds upon and extends these efforts by constructing a white-box safety metrics predictor leveraging state-of-the-art UQ techniques in deep learning (DL) He and Jiang (2023). This work specifically focuses on real-time, uncertainty-aware forecasting of lane-keeping safety metrics in ADS, addressing both aleatoric and epistemic uncertainty using deep ensembles and Transformer-based architectures.

2.1. Safety in Autonomous Driving

Traditional ADS safety monitoring relies on deterministic models with threshold-based alarms Sharifi et al. (2024), which lack robustness in dynamic or unseen conditions. Recent studies advocate for probabilistic safety forecasting Grewal et al. (2025), enabling earlier and more reliable failure detection. Our work aligns with these objectives but enhances them through a transformer-ensemble framework that supports multimodal and temporal data fusion.

2.2. Uncertainty Quantification in Deep Learning

UQ has gained traction for quantifying DNN confidence in safety-critical applications He and Jiang (2023); Weiss and Tonella (2021). Approaches include Bayesian Neural Networks (BNNs) Blundell et al. (2015), MC Dropout Gal and Ghahramani (2016), and deep ensembles Lakshminarayanan et al. (2017). Deep ensembles are particularly attractive due to their performance and ease of implementation. Conformal prediction Angelopoulos et al. (2021) provides theoretical guarantees but assumes independent and identically distributed (IID) data and can be expensive to compute for multistep time series. Our method builds on this by incorporating Transformer-based ensembles to simultaneously estimate aleatoric and epistemic uncertainties Chan et al. (2024), a combination not widely explored in ADS safety literature.

2.3. Probabilistic Forecasting and Transformers

Transformers have been widely adopted for time series and multimodal prediction Vaswani et al. (2023); Zhou et al. (2021). Prior work, such as that by Yao et al. Yao et al. (2022), proposed Transformer-based trajectory forecasting but did not quantify uncertainty via ensemble methods or probabilistic modeling. In contrast, our framework predicts parameters of a Gaussian distribution (μ, σ) for lane deviation metrics and aggregates multiple model outputs to estimate epistemic uncertainty. It is specifically tailored to the lane-keeping safety context in CARLA simulations, and includes real-time calibration techniques.

Furthermore, unlike works that focus on confidence scoring like Grewal et al. (2025), we emphasize probabilistic forecasting of safety metrics—predicting full distributions rather than scalar confidence values. Our system directly estimates both uncertainty types in a modular, multimodal framework, and employs post-hoc calibration for improved reliability.

This approach also extends the work by Sharifi et al. Sharifi et al. (2024), who optimize for inference latency and prediction accuracy. Our contribution prioritizes uncertainty reliability under varied simulation conditions through ensemble modeling and variance scaling.

While Brando et al. Brando et al. (2023) focus on standardizing uncertainty sources conceptually, our work operationalizes those ideas into a functioning real-time architecture validated in simulation.

Key distinctions of our work:

- Estimating aleatoric and epistemic uncertainties via Transformer-based ensembles.
- Focus on probabilistic forecasting over deterministic predictions or confidence scores.
- Application to lane-keeping in ADS, using real-time CARLA simulations and ROS2.
- Integration of temporal and multimodal sensor data into a unified uncertainty pipeline.

3. Methodology

This section outlines the proposed approach for addressing the challenges of probabilistic forecasting and UQ for safety-critical metrics in ADS. The methodology integrates multimodal and temporal data to provide reliable predictions and uncertainty estimates, leveraging advanced machine learning architectures. This chapter details the proposed framework's design, implementation, and training procedures, emphasizing its adaptability to real-time applications and its ability to enhance safety monitoring through robust uncertainty estimation.

3.1. Problem Formulation

Given a sequence of multimodal sensor and control data over a lookback window L, the task is to forecast Lane Deviation Safety Metric (LDSM), which defines deviation distance ($dist$)

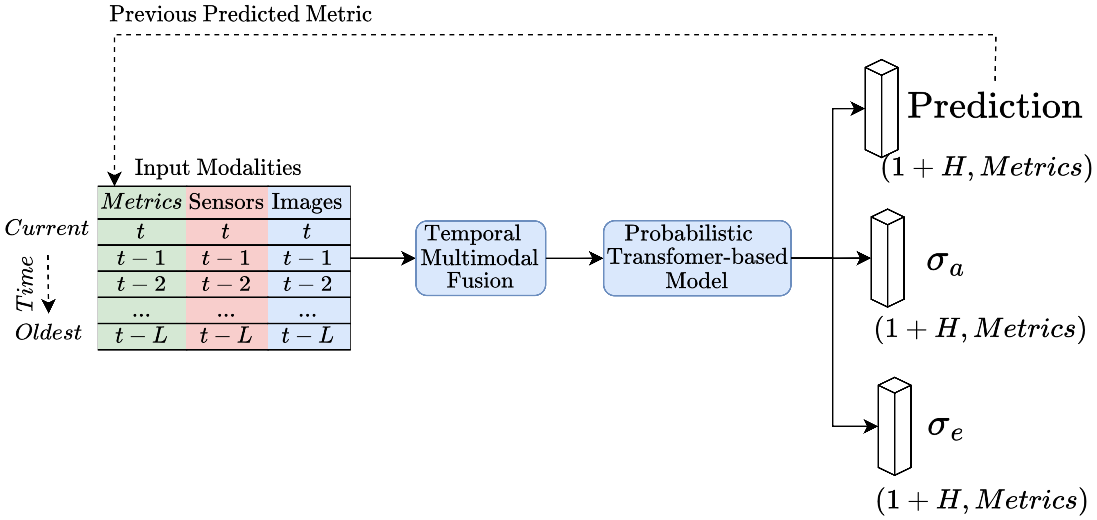
and angle (θ), over a future horizon H, along with their associated uncertainties. Formally, the model predicts for each timestep $h \in [t, t+H]$:

$$\mu_{dist}(h), \mu_{\theta}(h), \sigma_{a,dist}(h), \sigma_{a,\theta}(h) \tag{1}$$

where μ denotes the mean prediction and σ_a the aleatoric uncertainty.

3.2. Overview of the Probabilistic Framework

The proposed framework, shown in Figure 1, is designed to provide probabilistic, uncertainty-aware predictions for safety-critical metrics in ADS. It processes a temporal window of multimodal inputs—including sensor data (IMU, odometry), vehicle control commands, and visual information (camera images, segmentation maps), to forecast lane deviation metrics and their associated uncertainties over a prediction horizon. The architecture is modular, supporting both temporal and modality, specific information fusion, and includes a feedback loop for real-time applications. Modality-specific embedding layers prepare and align the inputs to a fusion dimension ($Fusion_D$), as highlighted in Figure 2. A Transformer-based architecture fuses and temporally encodes these embeddings into a shared representation. This embedding then feeds K separate Neural Networks (NNs) (as shown in Figure 3) to predict the K parameters of a chosen probability distribution (e.g., mean μ_{NN} and standard deviation σ_{NN} for a Gaussian), enabling the derivation of the most likely prediction and aleatoric uncertainty. This design choice, motivated by implementation and training (detailed in Section 3.5), allows for focused training of prediction and uncertainty estimation. A Gaussian application is detailed in Section 3.3.



$Metrics$	Sensors	Images
t	t	t
$t-1$	$t-1$	$t-1$
$t-2$	$t-2$	$t-2$
...	...	...
$t-L$	$t-L$	$t-L$

Figure 1: Generic Architecture designed to incorporate both temporal and modality information into its predictions, aleatoric uncertainty (σ_a), and epistemic uncertainty (σ_e). Furthermore, it allows for a feedback loop where these outputs can be used as inputs for the model in subsequent timesteps.

3.3. TMPT Architecture

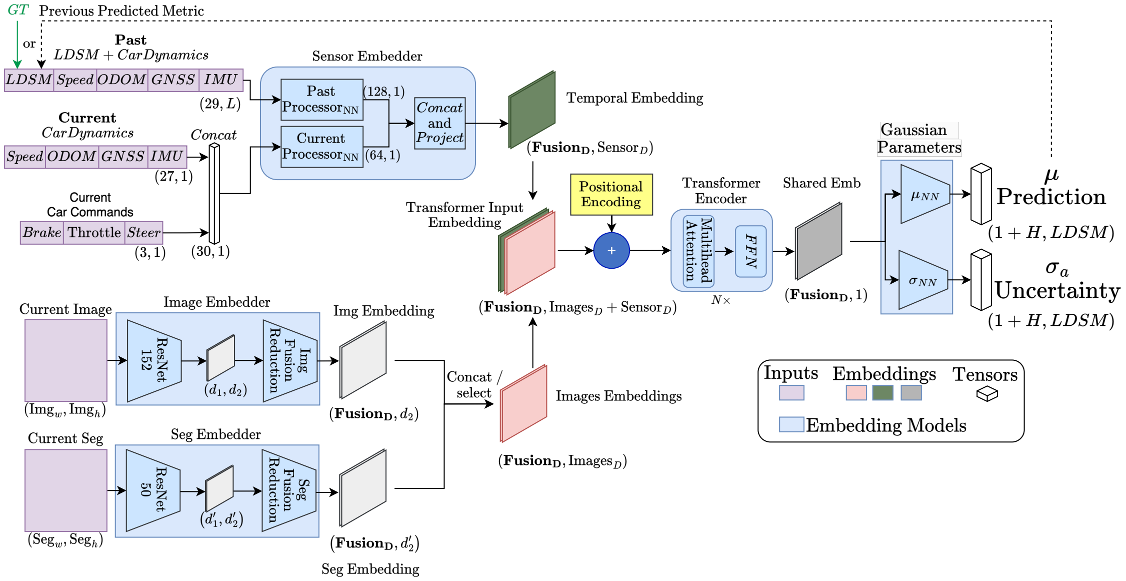


Figure 2: TMPT is a detailed architecture of the method applied to LDSM. Intricate temporal multimodal fusion of inputs is performed, and the outputs are modeled as a Gaussian distribution. The prediction corresponds to the Gaussian mean (μ), and the aleatoric uncertainty is represented by the standard deviation (σ_a).

The Temporal Multimodal Probabilistic Transformer (TMPT) method is applied to the LDSM, a safety metric defined by the combination of lateral distance (dev_{dist}) and angle (dev_θ) deviations. The distribution of the LDSM is modeled as Gaussian, with its mean (μ) and standard deviation (σ) being predicted by two dedicated NN (μ_{NN} and σ_{NN}). Throughout the following sections, μ represents the mean prediction, while σ_a quantifies aleatoric uncertainty.

Figure 2 presents the TMPT architecture which consists of the following:

- **Embedding Networks:** Separate NN process each modality:
  - **Camera Image**: Pre-trained convolutional neural network (CNN) extracts features, reduced to $Fusion_D$ (e.g., 512) by **Img Fusion Reduction**.
  - **Semantic Image**: Pre-trained CNN extracts features, reduced to $Fusion_D$ by **Seg Fusion Reduction**.
  - **Image Embedding Fusion/Selection**: Concatenates or selects from camera and segmentation embeddings ($Images_D$ dimension).
  - **Vehicle Dynamics and Car Controls**: Past Processor$_{NN}$ (temporal data of length L) and Current Processor$_{NN}$ (single timestep) are concatenated and projected to $(Fusion_D, Sensor_D)$ by **Concat and Project**.

- **Transformer Encoder:** A single-stage transformer (e.g., 6 or 12 layers, 8 heads, FFN size 2048, $Fusion_D$ 512) fuses all embeddings, capturing temporal and cross-modal dependencies.

- **Output Heads:** Fully connected (FC) layers predict the mean (μ_{NN}) and standard deviation (σ_{NN}) for Gaussian modeling for each safety metric and timestep ($(1+H) \times LDSM$).

The 24 models evaluated in this work, summarized in Table 1, share this architecture template but vary in configuration. CNN-based models (M1–M6) omit the Transformer and differ by loss function. Transformer based models (M7–M15) differ in depth, batch size, and use of positional encoding. M16–M22 apply post-hoc calibration using variance scaling, while M23–M24 ensemble calibrated models (M17, M21, M22) using either mean or uncertainty-weighted aggregation. The visual fusion strategy also varies: for example, M16 uses only RGB embeddings, while M20 concatenates both RGB and segmentation features before projection. These variations enable analysis of how architecture, fusion, calibration, and ensembling affect performance and uncertainty reliability. The models are grouped as follows:

- **M1–M6:** CNN baselines with different loss functions (e.g., NLL, L1, hybrid).

- **M7–M15:** Transformer-based models with differing depths and embedding strategies.

- **M17–M22:** Calibrated models using post-hoc variance scaling.

- **M23–M24:** Ensembles aggregating predictions from M17, M21, and M22.

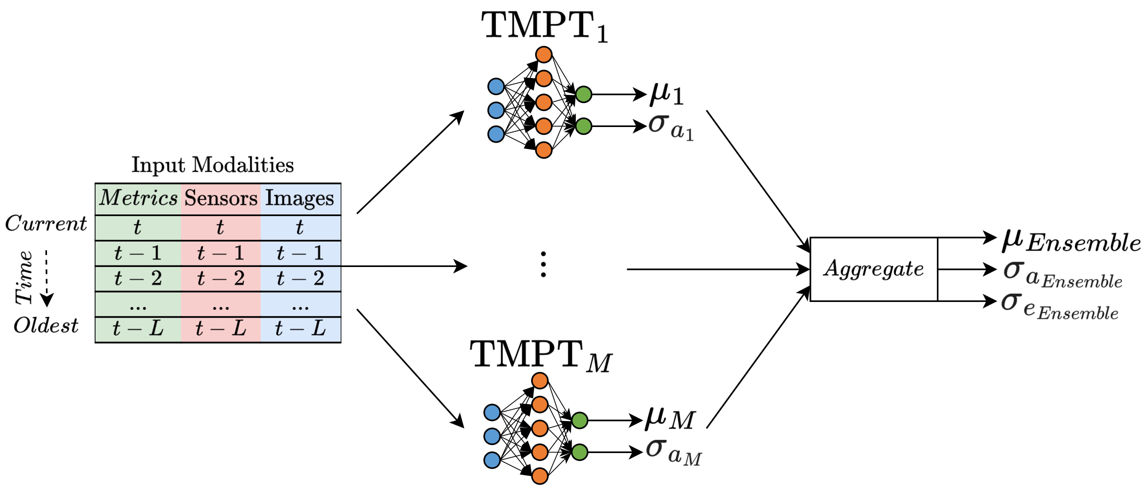
3.4. Uncertainty Estimation and Calibration

Aleatoric uncertainty represents irreducible data noise, is captured by training the model to predict both the mean and standard deviation for each output using a negative log-likelihood loss. This is implemented as separate neural network heads for each distribution parameter, leveraging a shared temporal embedding. Thus, aleatoric uncertainty is estimated by the predicted standard deviation of the Gaussian output.

Epistemic uncertainty represents reducible uncertainty stemming from limited or lack of knowledge or model parameters. Epistemic uncertainty is captured using deep ensembles: multiple TMPT models trained with different initializations and data splits, as shown in Figure 3. The ensemble mean and variance provide the final prediction and epistemic uncertainty, respectively. For M ensemble members with predictions μ_i, aleatoric uncertainties $\sigma_{a,i}$, and w_i representing weighting proportional to the inverse uncertainty of each model. The ensemble's predictions are aggregated, as follows:

- Mean prediction: The ensemble mean provides the final forecast.

$$\mu_{\text{Ensemble}} = \sum_{i=1}^{M} w_i \mu_i \tag{2}$$



	Metrics	Sensors	Images
Current	t	t	t
	$t-1$	$t-1$	$t-1$
	$t-2$	$t-2$	$t-2$
	...	...	...
Oldest	$t-L$	$t-L$	$t-L$

Figure 3: The outputs of individual TMPT models are combined to estimate the weighted average prediction (μ_{Ensemble}) and the epistemic uncertainty of the ensemble ($\sigma_{e_{\text{Ensemble}}}$). The aleatoric uncertainties of the M individual TMPT models ($\sigma_{a_1}, \ldots, \sigma_{a_M}$) are utilized to weight their respective predictions, resulting in enhanced overall prediction performance. Moreover, ($\sigma_{a_1}, \ldots, \sigma_{a_M}$) are aggregated to provide the average aleatoric uncertainty of the ensemble ($\sigma_{a_{\text{Ensemble}}}$).

- Epistemic uncertainty: The standard deviation of ensemble predictions quantifies epistemic uncertainty.

$$\sigma_{e_{\text{Ensemble}}} = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(\mu_i - \mu_{Ensemble})^2} \tag{3}$$

- Aleatoric aggregation: Individual models' aleatoric uncertainties are combined using a weighted average, where weights are derived from each model's predicted uncertainty.

$$\sigma_{a_{\text{Ensemble}}} = \sum_{i=1}^{M} w_i \sigma_{a_i} \tag{4}$$

- Total uncertainty: The overall predictive uncertainty is computed as the square root of the sum of squared aleatoric and epistemic uncertainties.

$$\sigma_{total} = \sqrt{\sigma^2_{a_{\text{Ensemble}}} + \sigma^2_{e_{\text{Ensemble}}}} \tag{5}$$

Calibration is performed using variance scaling Guo et al. (2017), fitting a scalar to align predicted uncertainties with observed errors on a validation set.

3.5. Training Procedure

Training proceeds in four stages:

1. **Regression Training:** Optimize prediction heads for mean accuracy.
2. **Uncertainty Training:** Freeze prediction heads, train uncertainty heads using negative log-likelihood loss.
3. **Calibration:** Fit variance scaler on validation data.
4. **Ensembling:** Aggregate predictions and uncertainties from multiple models.

4. Experimental Setup

4.1. Simulation Environment and Dataset

Experiments are conducted using the CARLA simulator (v0.9.15) with ROS2 middleware integration for real-time data streaming and control. Diverse scenarios are generated by varying weather, road layouts, vehicle speeds, and spawn locations. The dataset comprises 1,915 simulations across six towns and 22 weather conditions, with multimodal data (images, segmentation, IMU, odometry, control commands) collected at 50 miliseconds (ms) intervals Dosovitskiy et al. (2017). 80% of the simulations are used for training, 10% for validation, and 10% for testing. Each sample consists of:

- **Visual data:** RGB front camera and semantic segmentation.
- **Sensor data:** Inertial Measurement Unit (IMU), odometry.
- **Controls:** Steering, throttle, brake inputs.
- **Labels:** Ground-truth lateral deviation and heading angle from lane centerline.

4.2. Safety Metrics and Thresholds

Lane deviation is quantified by:

- **Lateral distance** (dev_{dist}): Euclidean distance from lane centerline.
- **Angular deviation** (dev_{θ}): Angle between vehicle heading and lane direction.

Safety thresholds are set at ±0.5 m for distance and ±5° for angle, following Euro NCAP and industry standards.

4.3. Evaluation Metrics

Two sets of metrics are used to asses the performance:

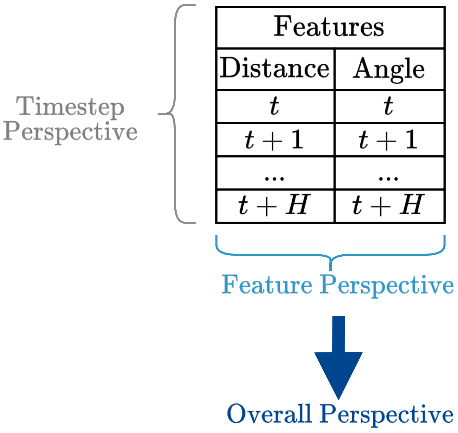
- **Regression:** Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R^2. Regression metrics are deatiled in depth in Appendix A.1.
- **Uncertainty Calibration:** ENCE Levi et al. (2020), Uncertainty Calibration Error (UCE), Quantile Calibration Error (QCE), Quantile Loss (QL). Uncertainty metrics are explained with more details in Appendix A.2.

5. Results and Discussion

This section presents a detailed analysis of 24 distinct architectures for lane deviation prediction, rigorously evaluated across regression performance, uncertainty calibration, temporal stability, and computational efficiency. These models range from simple CNN to sophisticated transformer-based ensembles. Table 1 summarizes the architectural configurations, fine-tuning relationships, and calibration or ensemble associations for each model. The results provide a comprehensive overview of their comparative performance, revealing that no single architecture is universally optimal across all metrics. Instead, the study highlights the necessity of context-driven model selection, carefully balancing the trade-off between predictive accuracy and reliable uncertainty estimates. The analysis also incorporates insights into the benefits of fine-tuning regression models for uncertainty prediction and the impact of calibration strategies.

We present the results grouped into five model families to highlight the impact of architecture, calibration, and ensembling: **CNN (UnCalib)** includes M1–M6; **CNN (Calib)** covers M17 and M18; **TMPT (UnCalib)** spans M7–M15; **TMPT (Calib)** includes M19–M22; and **Ensembles** comprise M23 (averaging) and M24 (uncertainty-weighted).

While multiple metrics are evaluated, it is worth noting that ENCE is the recommended metric for assessing uncertainty in regression problems Levi et al. (2020). Similar to regression losses like RMSE and MAE, a lower ENCE indicates better performance, specifically in the calibration of the model's uncertainty. The models are evaluated in the following sections from various perspectives determined by the output dimensions. Specifically, we assess predictions and their uncertainty for each timestep and the two LDSM features (distance and angle), as shown in Figure 4.



Features	
Distance	Angle
t	t
$t+1$	$t+1$
...	...
$t+H$	$t+H$

Figure 4: Evaluation perspectives based on output dimensionality. For LDSM (distance and angle) predicted over $1 + H$ timesteps, both prediction and uncertainty outputs have shape $(1 + H, 2)$.

5.1. Key Insights and Trade-offs

Prediction vs. Calibration: The TMPT (UnCalib) model with positional encoding (M16) achieved the best raw accuracy (e.g., RMSE = 7.34, $R^2 = 0.93$) but suffered from poor calibration (ENCE = 71.18). CNN (Calib) like M17 offered slightly worse precision but drastically better ENCE (0.50). TMPT (Calib) model M20 achieved near-perfect calibration (ENCE = 0.02) at the cost of higher prediction error. Ensembles (M24) offered a balance but required high compute.

Implication: Calibrated models (e.g., M17, M20) are preferable in safety-critical settings due to their reliable uncertainty estimates. Uncalibrated models like M16 should be used cautiously as their confidence may be misleading.

Feature-wise Findings: Angular deviation is significantly harder to predict and calibrate than lateral distance. Across all model groups, angle predictions consistently have higher MAE, RMSE, and ENCE. Even ensemble models (e.g., M24) showed stronger calibration for distance than angle. This reflects the higher volatility and nonlinearity in heading deviation.

Temporal Trends: As the prediction horizon H increases, all models exhibit rising error and uncertainty. TMPT models are more resilient than CNNs over long horizons, likely due to their stronger temporal modeling. For example, M16 maintained acceptable MAE and calibration over up to 16 steps, while CNN-based models saw early degradation. Ensembles helped mitigate this effect by averaging across drifted predictions.

5.2. Overall Predictive and Uncertainty Performance

The experimental results demonstrate significant variations in model performance, with distinct tradeoffs between prediction performance and uncertainty calibration. As shown in Table 2 summarizes the performance of key models. The calibrated CNN (M17) achieves the lowest MAE (1.58) and NLL (1.47), rivaling more complex transformer models. The transformer with positional encoding (M16) delivers the best RMSE (7.34) and R^2 (0.93) but suffers from poor calibration (ENCE=71.18). In contrast, the calibrated transformer (M20) achieves near-perfect calibration (ENCE=0.02), albeit with higher MAE.

Calibration reduces ENCE by up to 99.9% across models (see Table 3), with transformer-based and ensemble models achieving near-perfect calibration. Figures 6 and 5 visualizes the trade-off between regression accuracy and uncertainty quality.

5.3. Feature-Specific and Temporal Analysis

Feature-specific results (Table 4) show that angular deviation is more challenging than distance, with higher errors and calibration difficulty. The calibrated transformer (M20) achieves ENCE values of 0.0057 (angle) and 0.0351 (distance), outperforming all other models in uncertainty reliability.

Temporal analysis reveals that transformer-based models maintain stable error growth over longer horizons, supporting their suitability for proactive safety monitoring.

Table 1: Complete model architectures and configurations.

Model ID	**Architectural Configuration**
M1	Base CNN with NLL loss
M2	CNN with L_1 normalization
M3	CNN with L_2 normalization
M4	Hybrid loss CNN
M5	Fine-tuned L_2-CNN from base model M3
M6	Attention-based ResNet-152
M7	6-layer Transformer (550 batch)
M8	6-layer Transformer (2K batch)
M9	12-layer Transformer (550 batch)
M10	12-layer Transformer (2K batch)
M11	12-layer Transformer (5L-10H)
M12	Fine-tuned 12-layer Transformer from based model M11
M13	6-layer Transformer (5L-10H)
M14	Fine-tuned 6-layer Transformer from based model M13
M15	12-layer Transformer (60L-60H)
M16	6-layer Transformer + Pos Encoding
M17	Calibrated M5 (*netcal* var scaling)
M18	Calibrated M14 (unscaled model var scaling)
M19	Calibrated M14 (*netcal* var scaling)
M20	Calibrated M12 (*netcal* var scaling)
M21	Calibrated M10 (*netcal* var scaling)
M22	Calibrated M8 (*netcal* var scaling)
M23	Averaged Ensemble(M17+M21+M22)
M24	Uncertainty Weighted Ensemble (M17+M21+M22)

Table 2: Comprehensive Overall Performance Comparison

Model	↓ **MAE**	↓ **RMSE**	↑ **R**^{2}	↓ **NLL**	↓ **ENCE**	↑ **Cv**
M16	1.70	**7.34**	**0.93**	8.65e9	71.18	1.38
M17	**1.58**	7.70	**0.93**	**1.47**	0.50	**18.14**
M20	10.12	22.81	0.03	3.59	**0.02**	0.86
M24	5.03	18.57	0.22	2.30	0.46	1.48

Table 3: Calibration and ensemble performance across models with different architectures and lookback-horizon configurations (L-H). Each row presents pre- and post-calibration uncertainty metrics: ENCE, UCE, QCE, and QL. Models use either aleatoric uncertainty only (σ_a) or both aleatoric and epistemic uncertainties (σ_a & σ_e). Ensemble configurations (M23 and M24) are also evaluated for both uncertainty types, including average and uncertainty-weighted ensemble approaches. Improvements are reflected as raw values with corresponding percentage changes.

Arch	**Model Pair**	**ENCE ↓**	**UCE ↓**	**QCE ↓**	**QL ↓**
CNN (Basline) $5L-5H$	M5 → M17	399.07 → 0.50 (↓99.87%)	**59.46** → 64764.39 (↑+108,900%)	0.461 → 0.370 (↓19.70%)	**0.79** → 3.06 (↑+288.6%)
TMPT 6-Layers $5L-5H$	M8 → M22	288.25 → 2.14 (↓99.26%)	1181.88 → 462.23 (↓60.89%)	**0.278** → 0.187 (↓32.79%)	6.07 → 4.04 (↓33.46%)
TMPT 12-Layers $5L-5H$	M10 → M21	**142.51** → 0.10 (↓99.93%)	501.94 → 55.70 (↓88.91%)	0.494 → 0.217 (↓56.08%)	5.14 → 3.03 (↓40.93%)
TMPT 6-Layers $5L-10H$	M14 → M19	154.85 → 0.36 (↓99.76%)	61.21 → 27733.53 (↑+45,278%)	0.419 → **0.150** (↓64.26%)	0.90 → 7.68 (↑+749.5%)
TMPT 12-Layers $5L-10H$	M12 → M20	172.58 → **0.02** (↓99.99%)	520.27 → **5.84** (↓98.88%)	0.477 → 0.237 (↓50.27%)	5.02 → 3.47 (↓30.85%)
Avg. Ens. uses σ_a	M23: M17,M21,M22	0.6728	1456.0667	0.3410	3.4288
Avg. Ens. uses σ_a & σ_e	M23: M17,M21,M22	0.7394	1611.2582	0.3575	3.3705
Uncert. Wtd. Ens. uses σ_a	M24: M17,M21,M22	0.4583	160.9781	0.2688	2.0145
Uncert. Wtd. Ens. uses σ_a & σ_e	M24: M17,M21,M22	0.5862	8.0748	0.3129	**2.0058**

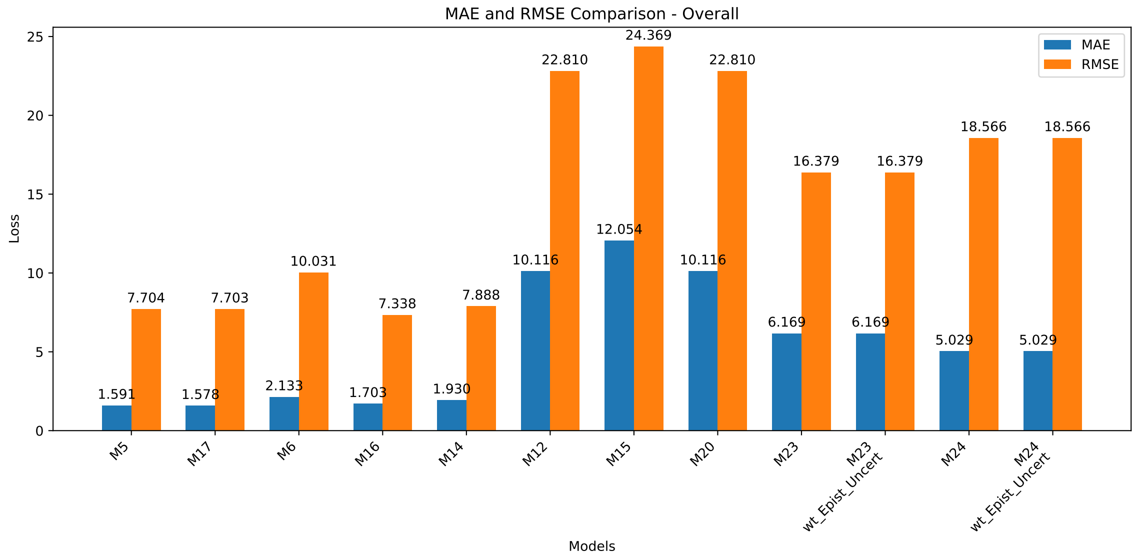


Figure 5: Comparison of loss values, a lower value indicates better regression performance.

Table 4: Feature-specific performance comparison across all models.

Model	**Feature**	↓ **MAE**	↓ **RMSE**	↓ **ENCE**	↑ **Cv**^{2}
M5	Angle	3.0544	10.8900	295.6379	**19.5938**
	Dist	**0.1272**	0.3515	548.9150	4.6954
M15	Angle	22.9111	34.3490	344.1216	1.0964
	Dist	1.1973	2.8074	37.8949	1.1177
M12	Angle	19.2532	32.1691	213.4511	0.1466
	Dist	0.9779	2.3894	132.2334	0.1284
M20	Angle	19.2532	32.1691	**0.0057**	0.0324
	Dist	0.9779	2.3894	**0.0351**	0.0440
M23	Angle	11.7633	23.1140	0.5881	1.3664
	Dist	0.5743	1.5030	0.7784	4.0375
M4	Angle	11.5475	32.1558	393.4157	3.4032
	Dist	0.2668	1.6198	454.1697	5.6277
M24	Angle	9.3250	26.1694	0.6846	0.9592
	Dist	0.7338	2.1324	0.3079	1.4895
M6	Angle	4.1254	14.1598	225.1769	2.5940
	Dist	0.1412	0.8660	22918.1289	**9.9127**
M14	Angle	3.6339	11.1474	195.3533	3.6133
	Dist	0.2253	0.4122	99.6171	7.1671
M16	Angle	3.2082	**10.3715**	83.4567	1.6427
	Dist	0.1983	0.3635	6.7807	1.0383
M17	Angle	**3.0288**	10.8882	0.3535	**18.7989**
	Dist	**0.1282**	**0.3506**	0.7863	5.0375

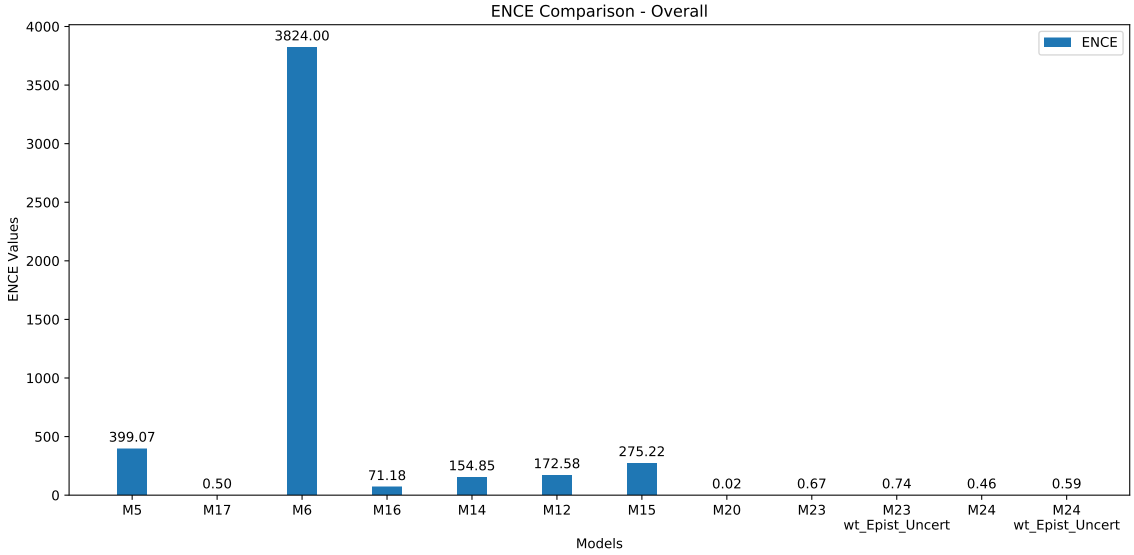


Figure 6: ENCE Comparison. A lower ENCE indicates better uncertainty estimation and calibration.

5.4. Computational Efficiency

Understanding model size and resource requirements is critical for selecting architectures suited to specific deployment contexts (e.g., embedded devices, real-time inference, or cloud processing). Table 5 summarizes parameter counts across representative models.

Table 5: Model parameter count and architectural efficiency.

Model ID	**Architecture**	**Calibration**	**Params (M)**
M5	CNN (Simple CNN + FC)	Uncalibrated	302.6
M17	CNN (Simple CNN + FC)	Calibrated	302.6
M16	TMPT (ResNet152, 6 layers)	Uncalibrated	93.9
M20	TMPT (ResNet152, 12 layers)	Calibrated	119.0
M23/M24	Ensemble (M16, M20, M17)	Calibrated	515.5

Despite its simplicity, the CNN-based predictor (M5/M17) contains over 300M parameters due to flattening high-resolution features into large FC layers. As CNN layers convert the intial image dimensionality from 3 (RGB) to 32, 64, 128, then 256. TMPT models (e.g., M16, M20) with frozen ResNet152 backbones and transformer encoders are more parameter-efficient while supporting temporal attention and multimodal fusion.

Ensemble models like M23 and M24, combining CNN and TMPT variants, exceed 500M parameters and are impractical for real-time edge deployment. However, they provide superior calibration and robustness in uncertainty estimation, especially under distributional shifts. While we did not profile GPU inference time due to hardware constraints, model testing on GPU or cloud platforms (Google Colab, AWS EC2, Paperspace) is feasible for benchmarking latency and memory footprint. Optimization for real-time deployment, including quantization and pruning, is left as future work.

6. Conclusion and Future Work

This work introduced Temporal Multimodal Probabilistic Transformers (TMPT), a framework for uncertainty-aware forecasting of safety metrics in ADS. By integrating both aleatoric and epistemic uncertainty, along with post-hoc calibration, TMPT enables models that are not only accurate but reliably calibrated.

Through extensive simulation in CARLA across 24 model variants, we found clear tradeoffs between prediction accuracy, calibration, and computational efficiency. CNN models (e.g., M5, M17) yield strong accuracy but unreliable uncertainty unless calibrated. TMPT models (e.g., M16, M20) provide a better balance between sharpness and calibration, with M20 achieving near-perfect ENCE. Ensemble models (e.g., M23, M24) improve robustness further, though at higher computational cost. Overall, model selection should reflect deployment constraints: TMPT (Calib) or Ensemble models are ideal for safety-critical scenarios, while calibrated CNNs offer efficient alternatives for limited-resource settings.

Future Work. Several directions remain. First, real-world deployment is needed to evaluate generalization under distribution shift. This includes testing on unseen weather conditions and performing structured OOD experiments to assess uncertainty behavior. Second, profiling inference speed, memory usage, and latency on GPUs will support practical benchmarking. Third, expanding to multi-agent safety forecasting, longer horizons, and integrating modalities like LiDAR or radar can improve realism and robustness. Lastly, more advanced aggregation schemes will enhance predictive reliability in ensemble models.

Acknowledgments

This work was funded by a project under Fraunhofer-Institut für Kognitive Systeme.

References

Anastasios N. Angelopoulos, Stephen Bates, Michael I. Jordan, and Jitendra Malik. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2021. URL https://arxiv.org/abs/2107.07511.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015. URL https://arxiv.org/abs/1505.05424.

Axel Brando, Isabel Serra, Enrico Mezzetti, Francisco Javier Cazorla Almeida, and Jaume Abella Ferrer. Standardizing the probabilistic sources of uncertainty for the sake of safety deep learning. In *Proceedings of the Workshop on Artificial Intelligence Safety 2023 (SafeAI 2023) co-located with the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023): Washington DC, USA, February 13-14, 2023.*, volume 3381.

CEUR Workshop Proceedings, 2023. URL https://upcommons.upc.edu/handle/2117/388215.

Matthew A. Chan, Maria J. Molina, and Christopher A. Metzler. Estimating epistemic and aleatoric uncertainty with a single model, 2024. URL https://arxiv.org/abs/2402.03478.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016. URL https://arxiv.org/abs/1506.02142.

Ruben Grewal, Paolo Tonella, and Andrea Stocco. Predicting safety misbehaviours in autonomous driving systems using uncertainty quantification, 2025. URL https://arxiv.org/abs/2404.18573.

Chuan Guo, Geoff Pleiss, Yu Sun Raghavan, John Hoffman, Laurens van der Lee, and Kilian Q. Weinberger. netcal: Calibration of Neural Networks. URL https://pypi.org/project/netcal/.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Wenchong He and Z Jiang. A survey on uncertainty quantification methods for deep neural networks: An uncertainty source perspective. *perspective*, 1:88, 2023.

Jens Henriksson, Christian Berger, Markus Borg, Lars Tornberg, Cristofer Englund, Sankar Raman Sathyamoorthy, and Stig Ursing. Towards structured evaluation of deep neural network supervisors. In *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 27–34. IEEE, 2019. URL https://ieeexplore.ieee.org/iel7/8716376/8718207/08718225.pdf.

Nargiz Humbatova, Gunel Jahangirova, Gabriele Bavota, Vincenzo Riccio, Andrea Stocco, and Paolo Tonella. Taxonomy of real faults in deep learning systems. In *Proceedings of the ACM/IEEE 42nd international conference on software engineering*, pages 1110–1121, 2020. URL https://doi.org/10.48550/arXiv.1910.11015.

Manzoor Hussain, Nazakat Ali, and Jang-Eui Hong. Deepguard: A framework for safeguarding autonomous driving systems from inconsistent behaviour. *Automated Software Engineering*, 29(1):1, 2022. URL https://doi.org/10.1007/s10515-021-00310-0.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017. URL https://arxiv.org/abs/1612.01474.

Dan Levi, Liran Gispan, Niv Giladi, and Ethan Fetaya. Evaluating and calibrating uncertainty prediction in regression tasks, 2020. URL https://arxiv.org/abs/1905.11659.

Murphy Y. Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty, 2018. URL https://arxiv.org/abs/1806.01768.

Sepehr Sharifi, Andrea Stocco, and Lionel C. Briand. System safety monitoring of learned components using temporal metric forecasting, 2024. URL https://arxiv.org/abs/2405.13254.

Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. Misbehaviour prediction for autonomous driving systems, 2019. URL https://arxiv.org/abs/1910.04443.

Andrea Stocco, Paulo J Nunes, Marcelo d'Amorim, and Paolo Tonella. Thirdeye: Attention maps for safe autonomous driving systems. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–12, 2022. URL https://dl.acm.org/doi/pdf/10.1145/3551349.3556968.

PyTorch Team. torcheval.metrics.r2score. URL https://pytorch.org/torcheval/main/generated/torcheval.metrics.R2Score.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Michael Weiss and Paolo Tonella. Fail-safe execution of deep learning based systems through uncertainty monitoring. In *2021 14th IEEE conference on software testing, verification and validation (ICST)*, pages 24–35. IEEE, 2021. URL https://arxiv.org/abs/2102.00902.

Oliver Willers, Sebastian Sudholt, Shervin Raafatnia, and Stephanie Abrecht. Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks, 2020. URL https://arxiv.org/abs/2001.08001.

Hai-Yan Yao, Wang-Gen Wan, and Xiang Li. End-to-end pedestrian trajectory forecasting with transformer network. *ISPRS International Journal of Geo-Information*, 11(1), 2022. ISSN 2220-9964. doi: 10.3390/ijgi11010044. URL https://www.mdpi.com/2220-9964/11/1/44.

Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8: 58443–58469, 2020. ISSN 2169-3536. doi: 10.1109/access.2020.2983149. URL http://dx.doi.org/10.1109/ACCESS.2020.2983149.

Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In *Proceedings of the 33rd ACM/IEEE international conference on automated software engineering*, pages 132–142, 2018. URL http://doi.acm.org/10.1145/3238147.3238187.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting, 2021. URL https://arxiv.org/abs/2012.07436.

Appendix A. Metrics

Evaluation metrics were employed to comprehensively assess the model's performance in predicting the mean (μ) and uncertainty (σ) of lane deviation distance and angle. This evaluation focused on both prediction accuracy and uncertainty quality. Following recommendations from Levi et al. (2020), the focus is placed on metrics designed explicitly for regression uncertainty evaluation.

A.1. Regression Metrics

For evaluating the quality of regression values, the following metrics are considered:

1. **Mean Absolute Error (MAE)**:
   Measures average difference predicted and ground truth.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\mu_i - y_i|, \tag{6}$$

   where μ_i is the predicted mean value, y_i is the ground truth, and N is the number of samples.

2. **Root Mean Squared Error (RMSE)**:
   RMSE penalizes larger errors more than MAE, making it sensitive to outliers.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mu_i - y_i)^2}. \tag{7}$$

   RMSE penalizes larger errors more than MAE, making it sensitive to outliers.

3. **R-squared Score (R^2)**: Measures how well predicted values approximate the true targets. $R^2 = 1$ indicates a perfect fit, while $R^2 = 0$ means the model performs no better than the mean. It is computed as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\mu_i - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{8}$$

   where μ_i is the prediction, y_i the ground truth, and $\bar{y}$ the sample mean of ground truth. R^2 is computed via *torcheval* Team.

A.2. Uncertainty Metrics

For evaluating the quality of uncertainty values, the following metrics are used. Moreover, the python library *netcal* implementation for these metrics Guo et al..

1. **Negative Log-Likelihood (NLL)**:
   NLL evaluates both prediction accuracy and uncertainty quality by treating predictions as Gaussian distributions:

$$\text{NLL} = -\frac{1}{N}\sum_{i=1}^{N}\log p(y_i|\mu_i, \sigma_i) = -\frac{1}{N}\sum_{i=1}^{N}\left[-\frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \log(\sigma_i) - C\right], \tag{9}$$

   where σ_i is the predicted standard deviation and C is a constant.

2. **MSLL**:
   MSLL measures the quality of probabilistic predictions by considering both the accuracy of the predicted mean and the calibrated uncertainty of the predicted variance, normalized by the variance.

$$\text{MSLL} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{(y_i - \mu_i)^2}{2\sigma_i^2} + \log(\sigma_i)\right]. \tag{10}$$

3. **Expected Normalized Calibration Error (ENCE)**: *[Highly Recommended]*
   Measures the average normalized difference between predicted and actual probabilities across bins. Levi et al. highly recommend using ENCE metric as a primary indicator of the quality of the uncertainty Levi et al. (2020), defined as:

$$\text{ENCE} = \frac{1}{B}\sum_{b=1}^{B}\frac{|\text{RMV}_b - \text{RMSE}_b|}{\text{RMV}_b}, \tag{11}$$

   where RMV_b is the Root Mean Variance and RMSE_b is the RMSE for bin b. The **RMV** and **RMSE** are computed as :

$$\text{RMV}_b = \sqrt{\frac{1}{|B_b|}\sum_{i\in B_j}\sigma_i^2}, \tag{12}$$

$$\text{RMSE}_b = \sqrt{\frac{1}{|B_b|}\sum_{i\in B_j}(\mu_i - y_i)^2}, \tag{13}$$

   where B_b is the set of samples in bin b, grouped by predicted uncertainty levels.

4. **RMV vs. RMSE Reliability Diagram:**
   Following Levi et al. (2020), the authors use a reliability diagram that compares the predicted uncertainty RMV to the observed error RMSE. This approach provides a more reliable assessment of calibration quality than traditional methods.

5. **Coefficient of Variation (CV)**:

$$\text{CV} = \frac{\sigma_\sigma}{\mu_\sigma}, \tag{14}$$

   where σ_σ is the standard deviation of the predicted uncertainties across all samples and μ_σ is mean of the predicted uncertainties across all samples.
   CV measures the dispersion in predicted uncertainties, with higher values indicating more informative uncertainty estimates. Levi et al. recommend using this metric as supplementary indicatory to ENCE of the quality of the uncertainty Levi et al. (2020).

6. **Uncertainty Calibration Error (UCE)**: Measures mismatch between predicted confidence and actual accuracy.

$$\mathrm{UCE} = \sum_{b=1}^{B} \frac{|B_b|}{N} |u_b - e_b|, \tag{15}$$

where u_b is the average predicted uncertainty in bin b, e_b is the average observed error in bin b, and $|B_b|$ is the number of samples in bin b, and N is total number of predictions.

7. **Quantile Loss (QL)**:

$$\mathrm{QL}_q = \frac{1}{N} \sum_{i=1}^{N} \max[q(y_i - \hat{y}_{q,i}),\ (q-1)(y_i - \hat{y}_{q,i})], \tag{16}$$

where q is the quantile level, y_i the ground truth, and $\hat{y}_{q,i}$ the predicted q-th quantile. We use nine quantiles $Q = \{0.1, 0.2, \ldots, 0.9\}$ to capture uncertainty across the distribution, lower q targets the lower tail, $q = 0.5$ the median, and higher q the upper tail. The final QL is the average over all $q \in Q$, an overall quantile prediction quality.

8. **Quantile Calibration Error (QCE)**:

$$\mathrm{QCE} = \frac{1}{|Q|} \sum_{q \in Q} |q - \hat{q}(q)|, \tag{17}$$

where Q is the set of quantile levels and $\hat{q}(q)$ is the empirical frequency at quantile level q. $Q = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ is used to evaluate QCE.

(a) Divide examples into bins based on their predicted uncertainty (σ_i^2) using equal-frequency binning.

(b) Compute RMV and RMSE for each bin.

(c) Plot RMSE as a function of RMV for all bins. For a well-calibrated model, this plot should approximate the identity line.