Query2Uncertainty: Robust Uncertainty Quantification and Calibration for 3D Object Detection under Distribution Shift

Supplementary Material

A. Appendix Section

Appendix Contents

A.1. Uncertainty Evaluation Framework

Overview. Our uncertainty evaluation framework builds upon the official `nuScenes-devkit` [1] and extends existing data structures and evaluation routines, which makes our implementation easy to use and integrate into existing 3D object detection pipelines. In particular, we extend the existing 3D box object class with additional regression uncertainty measures and introduce metrics that ingest classification confidences and 3D box covariance estimates while reusing the nuScenes ground-truth matching and metric reporting.

Interface. As a basis for our implementation, we extend the class `NuScenesBox` with `NuScenesBoxUQ` (see Listing 1) to include variances for centroid and size, and additionally an angular concentration for the rotation.

```
class NuScenesBoxUQ(NuScenesBox):
  def __init__(
    self,
    center: List[float],
    center_cov: List[float],            <- Added
    bbox_list: List[List[float]],
    size: List[float],
    size_cov: List[float],              <- Added
    orientation: Quaternion,
    orientation_kappa: float = np.nan, <- Added
    label: int = np.nan,
    score: float = np.nan,
    velocity: Tuple = (np.nan, np.nan),
    name: Optional[str] = None,
    token: Optional[str] = None
  ):
```

Listing 1. Interface for NuScenes box structure with additional uncertainty fields for centroid, size and orientation.

Further, we extend the class `NuScenesMetric` with `NuScenesMetricUQ` (see Listing 2) to include calibration and evaluation methods for classification and regression uncertainties. It logs per-class uncertainty classification and regression metrics alongside the standard nuScenes mAP/NDS scores.

```
class NuScenesMetricUQ(NuScenesMetric):
  def __init__(
    self,
    *args,
    cls_calib_method='identity',
    reg_calib_method='identity',
    train_calibration=None,
    calib_dir=None,
    **kwargs
  ):

results = {...}  # List[NuScenesBoxUQ]

evaluator = NuScenesMetricUQ(...)

metrics = evaluator.nus_evaluate(
  results=results
)
```

Listing 2. Interface for NuScenes metric extension with calibration hooks and example usage.

The framework is directly compatible with `mmdetection3d` [2] by using the redefined box structure and passing them to our evaluation routine. We hope that this easy-to-use extension will facilitate future research on uncertainty calibration and evaluation for 3D object detection.

Additional Metrics. We implement in our framework additional metrics for measuring the quality of uncertainties, that we did not present in the main paper due to space constraints.

LaACE (Classification). The Location-Aware Adaptive Calibration Error (LaACE) mirrors LaECE but removes the discrete binning by directly comparing every detection's confidence with its location-aware quality term [7]. For each detection we compute $lq_j = 1 - \frac{\min(d_j, \tau)}{\tau}$ using the euclidean center-distance d_j to the matched ground-truth (using the nuScenes TP threshold of $\tau = 2$m) and set $lq_j = 0$ for unmatched predictions. Let s_j denote the detector's confidence for detection j. It then reads

$$\text{LaACE} = \frac{1}{n} \sum_{j=1}^{n} |s_j - lq_j| \tag{1}$$

which yields a location-aware calibration error without the sampling variance of binning while still penalizing confident yet poorly localized detections. In our experiments, LaACE behaves similarly to LaECE but generally yields higher errors (see Table 1), since LaECE is averaged over bins, whereas LaACE is a pointwise metric.

Table 1. **In-Distribution - Classification Calibration** LaACE and LaECE (in %) are reported as mean over detections thresholds from $[0.05, 0.60]$ with 0.05 steps. Class-wise calibration: Cls.; Global calibration: Glb.. +/-: better/worse than the naïve post-hoc method.

		PETR				SECOND			
Calib. Type	Calib. Method	Calibration Error LaACE↓	Calibration Error LaECE↓	Accuracy mAP↑	Accuracy NDS↑	Calibration Error LaACE↓	Calibration Error LaECE↓	Accuracy mAP↑	Accuracy NDS↑
None	Uncal.	32.029	27.211	38.25	45.05	27.136	17.378	54.53	63.57
Sample -based	MCD [4]	33.988	30.110	36.90	44.23	28.792	19.523	51.80	62.30
	DE [8]	35.338	32.754	33.20	43.39	29.563	19.635	50.83	62.36
Train -time	CalDETR [10]	34.304	30.089	37.12	44.28	27.874	19.906	53.57	63.17
	TCD [9]	28.837	24.533	38.27	45.05	28.696	18.869	54.37	63.81
Post -hoc	TS [5] Glb.	31.156	26.115	38.26	45.04	23.175	16.194	54.53	63.58
	TS [5] Cls.	29.916	24.660	38.26	45.04	21.985	13.968	54.53	63.58
	PS [12] Glb.	30.770	25.686	38.26	45.04	22.932	15.810	54.53	63.58
	PS [12] Cls.	28.504	22.782	38.26	45.04	18.283	9.076	54.53	63.58
	IR [14] Glb.	30.534	25.753	38.26	45.04	22.591	15.642	54.53	63.58
	IR [14] Cls.	28.487	22.869	38.26	45.04	18.140	9.078	54.53	63.58
Density -aware (Ours)	DA-TS Glb.	30.145+	24.831+	38.36	45.03	23.046+	15.185+	54.36	63.65
	DA-TS Cls.	29.546+	24.185+	38.36	45.03	21.796+	13.237+	54.36	63.65
	DA-PS Glb.	30.086+	24.960+	38.36	45.03	22.668+	15.043+	54.36	63.65
	DA-PS Cls.	**28.104**+	**22.393**+	38.36	45.03	18.107+	**8.812**+	54.36	63.65
	DA-IR Glb.	29.226+	24.015+	38.36	45.03	18.872+	10.982+	54.36	63.65
	DA-IR Cls.	28.160+	22.550+	38.36	45.03	**17.607**+	8.877+	54.36	63.65

Uncertainty Realism (Regression). For the centroid regression uncertainty, we additionally evaluate the Uncertainty Realism Criterion [13] with a Mahalanobis distance-based statistical test that operates directly on the posterior predictive Gaussians. Each matched detection i is characterized by its mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$, derived from the uncertainty mechanism, along with the ground-truth regression target $\mathbf{y}_{i,\text{gt}}$. The squared Mahalanobis distance of the ground truth under the predictive Gaussian reads

$$M^2_{\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i}(\mathbf{y}_{i,\text{gt}}) = (\mathbf{y}_{i,\text{gt}} - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\mathbf{y}_{i,\text{gt}} - \boldsymbol{\mu}_i), \quad (2)$$

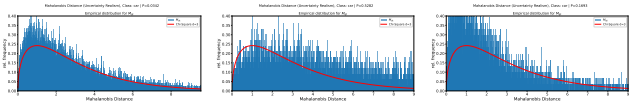
and we collect the sample distances over the test set as $\mathcal{M}_{\text{gt}} = \{M^2_{\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i}(\mathbf{y}_{i,\text{gt}})\}_{i=1}^N$.

It reads $\chi^2(d)$ in distribution when the predictive mean and covariance are realistic, where d is the dimensionality of the regression subspace (e.g., $d = 3$ for centroid).

Following the Uncertainty Realism Criterion [13], the squared Mahalanobis distances for the entire test set $\mathcal{M}_{\text{gt}}$ should follow a $\chi^2(d)$-distribution. We assess this hypothesis with the one-sample Kolmogorov–Smirnov test between $\mathcal{M}_{\text{gt}}$ and $\chi^2(d)$, but instead of using the highly sensitive p-value we record the KS statistic, which proved much more stable in practice. Let $\hat{F}_{\mathcal{M}}$ denote the empirical CDF of $\mathcal{M}_{\text{gt}}$ and $F_{\chi^2(d)}$ the theoretical CDF of the $\chi^2(d)$ law. Our reported score KS_{xyz} corresponds to the maximum CDF deviation

$$\text{KS}_{xyz} = D_{\text{KS}} = \sup_x \left| \hat{F}_{\mathcal{M}}(x) - F_{\chi^2(d)}(x) \right|, \quad (3)$$

where lower values indicate that the predicted covariance matrices generate Mahalanobis distances closer to the desired $\chi^2(d)$ distribution. We multiply KS_{xyz} by 100 for better readability, resulting in values between 0 and 100, where 0 indicates perfect uncertainty realism and a value close to 100 indicates unrealistic uncertainties. We report KS_{xyz} alongside with MCA_{xyz} in Table 2 and since both metrics behave similarly across all methods, we conclude that MCA is a reliable proxy for regression Uncertainty Realism in our experiments. A visual example of the empirical and theoretical CDFs for SECOND is provided in Figure 1. In our main experiments, we focus on MCA_{xyz} as it is easier to interpret and more commonly used in the literature.



(a) Well-Calibrated (b) Underconfident (c) Overconfident

Figure 1. **Uncertainty Realism Visualization.** Empirical squared Mahalanobis Distance distributions (blue) and an ideal distribution (red) for the centroid covariance for a calibrated (a), overconfident (b) and underconfident case (c).

A.2. Calibrator Parameter Count

We summarize the number of learnable parameters for each classification and regression calibrator in Table 3 using SECOND at detection threshold 0.30 as an example. For

Table 2. **In-Distribution - Regression Calibration** Metrics are averaged over all corruptions, severity levels and detection thresholds from 0.05 to 0.60 with 0.05 increments. Class-wise calibration: Cls.; Global calibration: Glb.

		PETR				SECOND			
Calib.	Calib.	$MCA_{xyz}\downarrow$	$KS_{xyz}\downarrow$	mAP↑	NDS↑	$MCA_{xyz}\downarrow$	$KS_{xyz}\downarrow$	mAP↑	NDS↑
Sample -based	MCD [4]	35.878	92.003	36.90	44.23	31.620	87.506	51.80	62.30
	DE [8]	28.871	84.636	33.20	43.39	23.089	76.020	50.83	62.36
	CalDETR [10]	31.834	93.237	37.12	44.28	34.234	95.525	53.57	63.17
Train -time	ES [6]	6.930	19.738	38.25	45.05	3.400	11.461	54.53	63.57
	KL [15]	4.384	13.033	38.25	45.05	4.692	13.205	54.53	63.57
Post -hoc	Depth Glb.	8.254	25.880	38.26	45.04	12.494	36.744	54.53	63.58
	Depth Cls.	2.407	10.881	38.26	45.04	4.749	19.102	54.53	63.58
	TS [3] Glb.	4.028	11.977	38.26	45.04	3.443	12.711	54.53	63.58
	TS [3] Cls.	1.692	7.135	38.26	45.04	1.533	7.281	54.53	63.58
Density -aware	DA-TS Glb.	4.059_{-}	12.087_{-}	38.36	45.03	2.875_{+}	10.753_{+}	54.36	63.65
	DA-TS Cls.	$\mathbf{1.538}_{+}$	$\mathbf{6.671}_{+}$	38.36	45.03	$\mathbf{1.518}_{+}$	$\mathbf{7.053}_{+}$	54.36	63.65

nuScenes with 10 object classes, global calibrators (Glb.) have ten times fewer parameters than class-wise calibrators (Cls.). IR [14] exhibits the highest complexity among all methods, as it employs a piecewise linear function for classification calibration with multiple segment parameters. Note that the number of parameters for IR varies depending on the input distribution of the calibration set used for fitting, while other methods have a fixed parameter count.

Table 3. **Parameter Count of Post-hoc Calibration Methods.** Parameter counts using SECOND for classification and regression calibration at detection threshold 0.30 and using ten object classes.

Task	Method	Glb.	Cls.
Classi -fication	TS [5]	1	10
	PS [12]	2	20
	IR [14]	330	1364
	DA-TS	2	20
	DA-PS	4	40
	DA-IR	377	1362
Regress -ion	Depth	14	140
	TS [3]	7	70
	DA-TS	21	210

A.3. Sample-based Approaches

Monte-Carlo Dropout (MCD) [4] and Deep Ensembles (DE) [8] performed poorly in our experiments, since the clustering of multiple stochastic forward passes into final detections is challenging in object-dense scenes. We follow related work [11] and cluster multiple 3D boxes across stochastic passes with DBSCAN before averaging the box parameters and deriving uncertainties from the sample statistics. We tested a 3D IoU and the L1-Norm to compute the distance between boxes of the same class and provide ablations on DBSCAN's maximum distance threshold (ϵ) for PETR in Table 4. For MCD and DE, we use by default $n = 6$ stochastic forward passes, and additionally provide results for $n = 12$ using the best DBSCAN configuration. For both, the accuracy (mAP, NDS) and uncertainty quality (D-ECE, MCA_{xyz}), the sample-based methods are underperforming compared to the single-forward pass approaches presented in the main paper. Further, a higher number of stochastic passes does not improve the results, but increases the computational overhead. Hence, we keep $n = 6$ for all main experiments in the paper. Compared to previous works [11], we apply MCD and DE to nuScenes, which is a more complex and object-dense dataset than KITTI, which contains only classes `car`, `pedestrian`, and `cyclist`. Hence, both sample-based UQ baselines struggle to provide reliable uncertainty estimates for 3D object detection in object-dense scenes and produce mainly overconfident uncertainties, as shown in Figures 4 and 5. This further underlines the need for dedicated uncertainty quantification methods in 3D object detection, as proposed in our work.

A.4. Train-time Calibration Approaches

CalDETR. Following the methodology of Cal-DETR [10], we adapt the training-time calibration framework for our DETR-based architecture. We implement *Uncertainty-Guided Logit Modulation* by estimating uncertainty via the variance of classification logits across the transformer decoder layers. This variance modulates the final confidence scores during the forward pass, explicitly penalizing predictions that exhibit high disagreement across layers. Furthermore, we incorporate *Logit Mixing* as a regularizer, which interpolates the logits of positive queries with a batch-wise prototypical mean logit to enforce a smoother, well-calibrated embedding space. For the Logit Mixing loss, we generate soft training targets where the ground-truth class receives a probability of α, and the remaining probability mass $(1 - \alpha)$ is distributed among the other classes in the mix. The loss

Table 4. **Sample-base Methods - PETR - In-Distribution.** Accuracy (mAP and NDS) measured without thresholding. D-ECE and MCA_{xyz} measured at threshold 0.05. Bold indicates the configuration used for main experiments.

Method	Measure	ϵ	mAP↑	NDS↑	D-ECE↓	MCA_{xyz}↓
MCD $n=6$ [4]	3D IoU	0.50	35.53	43.94	9.67	36.30
		0.55	36.66	44.21	7.93	34.58
		0.60	**36.57**	**44.32**	**9.70**	**34.91**
		0.65	36.65	44.29	9.76	34.44
		0.70	35.53	43.94	9.67	36.30
	L1	0.60	29.86	44.19	9.29	36.88
		0.70	31.86	44.27	9.67	36.60
		0.75	33.07	44.31	8.45	35.72
		0.80	33.59	44.30	8.46	35.38
		0.90	34.43	44.16	8.30	34.73
MCD $n=12$ [4]	3D IoU	1.00	31.94	44.34	8.88	34.19
DE $n=6$ [8]	3D IoU	0.50	30.96	42.46	11.74	29.30
		0.55	32.45	42.92	11.67	27.80
		0.60	33.20	43.05	11.68	26.42
		0.65	33.33	42.82	11.46	25.053
		0.70	32.79	42.28	11.66	23.593
	L1	0.70	22.74	36.60	10.01	30.60
		0.75	23.89	40.40	10.35	30.11
		0.80	24.85	40.65	10.22	29.59
		0.90	25.51	42.47	10.98	28.78
		1.00	**27.53**	**43.39**	**10.41**	**27.48**
		1.05	27.88	43.31	11.10	27.11
DE $n=12$ [8]	L1	1.00	23.52	42.38	11.22	24.20

is trained jointly with the standard 3D detection objectives (Classification Focal Loss, L_1). We vary α as hyperparameter for PETR and SECOND and summarize the results in Table 5. Since $\alpha = 0.8$ gave us most stable results across both architectures, we use this value for all main experiments. We did not evaluate $\alpha < 0.5$ thoroughly, since the model focuses more on calibration and the NDS score drops significantly.

Table 5. **CalDETR α Ablation - In-Distribution.** D-ECE reported across detection thresholds from 0.05 to 0.60 with 0.05 increments. Bold indicates the configuration used for main experiments.

α	SECOND		PETR	
	D-ECE↓	NDS↑	D-ECE↓	NDS↑
0.5	15.541	63.21	9.391	43.51
0.8	15.450	63.17	9.264	44.28
0.9	18.491	63.18	10.567	43.84

TCD. We integrate the Train-time Calibration for Detection (TCD) loss [9] as an auxiliary differentiable objective within the transformer decoder layers to align predicted confidence with localization accuracy. The loss formulation consists of two distinct components: a global classification alignment term and a local detection quality term. For the classification component (d_{cls}), we compute the absolute difference between the mean predicted confidence (sigmoid probabilities) and the mean ground-truth accuracy (one-hot encoded labels) across the mini-batch. This enforces that the average confidence reflects the empirical precision of the model for each class. For the detection component (d_{det}), we align the confidence of positive object queries with their structural fidelity. We calculate the L_1 distance between the maximum class probability of a query and its corresponding Intersection-over-Union (IoU) with the ground truth box.

The TCD loss is optimized jointly with the Focal Loss used for classification and the L_1 losses used for box regression. The total TCD loss is defined as

$$\mathcal{L}_{TCD} = 0.5 \cdot \left(\mathcal{L}_{cls}^{cal} + \mathcal{L}_{det}^{cal}\right). \tag{4}$$

This auxiliary loss is applied at every decoder layer together with the standard detection losses.

A.5. Density-Aware Calibration

The gain parameter γ controls how strongly the standardized density deviation modulates the per-query calibrators; larger values amplify the density-aware corrections (making the method more sensitive to distribution shift), while smaller values keep the adjustment closer to the base Post-hoc method behavior. We therefore ablate γ and summarize their effect on MCA_{xyz} (see Table 6 and Table 7) and on D-ECE (see Table 8 and Table 9). We tune γ on the in-distribution validation set and fixed this value for the distribution-shift experiments. A larger γ allows the calibrator to react more aggressively to density drops, which is beneficial under strong shifts, but risks over-correcting on ID data. For DA-TS (regression), we choose $\gamma = 0.15$ for SECOND and $\gamma = 0.3$ for PETR, as these values yield the best trade-off between ID and distribution-shift performance. For classification, we chose $\gamma = 0.2$ for both architectures and for both, DA-TS and DA-PS, as it provides a good trade-off between ID and distribution-shift performance. The results indicate that a better tuning of γ per architecture and calibrator could further improve the results under domain shift.

A.6. Distribution Shift Results

For SECOND, we present for each corruption the centroid regression quality metrics (MCA_{xyz}, KS_{xyz}) in Table 10 and the remaining parameters (MCA_{lwh}, MCA_{ϕ}) in Table 11. For PETR, we present corresponding results in Tables 13 and 14. The reported values are averaged over three severity levels and detection thresholds from 0.05 to 0.60 in steps of 0.05.

A.7. Qualitative Results

We provide qualitative examples, using PETR, of our density-aware calibration methods for an In-Distribution setting in Figures 2 and 3. Additional qualitative results for DE [8] and MCD [4] are provided in Figures 4 and 5, respectively. For distribution shift settings, we provide qualitative examples in Figures 6-9.

Table 6. **Ablation on γ - PETR - Regression.**

Method	γ	MCA$_{xyz}$ ↓ In-Distribution	MCA$_{xyz}$ ↓ Distribution-Shift
DA-TS Cls.	0.10	1.609	3.709
	0.20	1.544	3.571
	0.30	1.538	3.546
	0.40	1.550	3.537
TS [3] Cls.	-	1.692	3.780

Table 7. **Ablation on γ - SECOND - Regression.**

Method	γ	MCA$_{xyz}$ ↓ In-Distribution	MCA$_{xyz}$ ↓ Distribution-Shift
DA-TS Cls.	0.05	1.526	4.584
	0.10	1.538	4.241
	0.15	1.518	4.037
	0.20	1.589	4.031
TS [3] Cls.	-	1.533	5.264

Table 8. **Ablation on γ - PETR - Classification.**

Method	γ	D-ECE ↓ In-Distribution	D-ECE ↓ Distribution-Shift
DA-TS Cls.	0.10	5.922	9.934
	0.20	5.772	9.655
	0.30	5.720	9.662
	0.40	5.742	9.719
DA-PS Cls.	0.10	4.202	7.978
	0.20	3.899	7.770
	0.30	3.968	7.764
	0.40	4.115	7.801

Table 9. **Ablation on γ - SECOND - Classification.**

Method	γ	D-ECE ↓ In-Distribution	D-ECE ↓ Distribution-Shift
DA-TS Cls.	0.10	7.793	13.461
	0.20	7.351	13.018
	0.30	7.189	12.901
	0.40	7.035	12.746
DA-PS Cls.	0.10	2.467	7.150
	0.20	2.224	6.777
	0.30	2.342	6.731
	0.40	2.623	6.819

Table 10. **SECOND - Regression Calibration under Distribution Shift.** Class-wise calibration (Cls.) and Global calibration (Glb.) reported for each corruption as calibration error MCA$_{xyz}$ and KS$_{xyz}$ (lower is better). Results are averaged over all detection thresholds and corruption severities. Ours: DA-TS.

Corruption	MCA$_{xyz}$ ↓ (Calib: Cls.)				MCA$_{xyz}$ ↓ (Calib: Glb.)				KS$_{xyz}$ ↓ (Calib: Cls.)				KS$_{xyz}$ ↓ (Calib: Glb.)			
	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS
Snow	5.936	7.783	4.513	**2.237**	5.936	14.997	4.966	**2.959**	15.426	26.635	11.788	**6.881**	15.426	44.587	14.204	**9.642**
Fog	6.389	7.905	5.164	**3.216**	6.389	14.684	5.181	**3.456**	16.297	27.358	13.799	**9.018**	16.297	43.745	14.850	**10.892**
Motionblur	4.824	7.875	3.528	**2.167**	4.824	14.959	3.871	**2.744**	12.417	26.126	9.568	**7.673**	12.417	43.467	11.826	**9.732**
Beams Red.	6.820	12.638	5.952	**5.262**	6.820	19.120	6.066	**5.470**	13.701	39.886	13.198	**15.652**	**13.701**	55.776	14.371	16.020
Points Red.	6.034	7.066	4.792	**2.388**	6.034	14.446	5.045	**3.091**	15.753	25.526	12.936	**7.266**	15.753	43.371	13.720	**9.472**
Spatial Mis.	8.172	16.117	**7.632**	8.949	**8.172**	22.092	8.987	9.908	25.491	45.918	**24.043**	28.408	**25.491**	59.510	27.643	31.072
Mean	6.363	9.897	5.264	**4.037**	6.363	16.716	5.686	**4.605**	16.514	31.908	14.222	**12.483**	16.514	48.410	16.103	**14.472**

Table 11. **SECOND - Regression Calibration under Distribution Shift.** Class-wise calibration (Cls.) and Global calibration (Glb.) reported for each corruption as calibration error MCA$_{lwh}$ and MCA$_{\phi}$ (lower is better). Results are averaged over all detection thresholds and corruption severities. Ours: DA-TS.

Corruption	MCA$_{lwh}$ ↓ (Calib: Cls.)				MCA$_{lwh}$ ↓ (Calib: Glb.)				MCA$_{\phi}$ ↓ (Calib: Cls.)				MCA$_{\phi}$ ↓ (Calib: Glb.)			
	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS
Snow	5.722	6.881	5.049	**3.719**	5.722	16.503	6.135	**5.161**	17.530	8.127	9.364	**8.709**	17.530	16.615	16.410	**10.923**
Fog	6.645	5.340	5.012	**3.516**	6.645	15.506	7.685	**5.857**	15.661	7.822	7.315	**7.194**	15.661	16.738	14.727	**8.183**
Motionblur	5.574	6.358	4.354	**4.195**	5.574	16.753	5.859	**6.092**	14.453	7.996	7.578	**7.242**	14.453	17.106	13.675	**7.935**
Beams Red.	7.969	9.896	6.848	**6.418**	7.969	18.620	8.581	**8.397**	15.924	10.773	**9.412**	9.461	15.924	18.069	15.163	**10.344**
Points Red.	6.378	3.675	4.850	**2.940**	6.378	14.458	7.197	**5.308**	15.944	9.076	7.973	**7.589**	15.944	17.210	15.010	**9.132**
Spatial Mis.	5.703	4.519	4.462	**2.859**	5.703	15.346	6.695	**4.365**	13.803	8.891	8.573	**8.437**	13.803	15.399	12.107	**8.725**
Mean	6.332	6.111	5.096	**3.941**	6.332	16.198	7.025	**5.863**	16.328	8.781	8.379	**8.105**	16.328	16.182	14.338	**9.207**

Table 12. **SECOND - Classification Calibration under Distribution Shift.** All results are reported for class-wise calibration (Cls.) and are averaged over all detection thresholds and all three corruption severities.

Corruption	D-ECE↓ (Calib: Cls.)						LaECE↓ (Calib: Cls.)					
	Baseline			Density-aware (Ours)			Baseline			Density-aware (Ours)		
	TS[5]	PS[12]	IR[14]	DA-TS	DA-PS	DA-IR	TS[5]	PS[12]	IR[14]	DA-TS	DA-PS	DA-IR
Snow	14.493	7.217	7.358	**7.316**	7.403	8.993	16.448	8.987	9.153	13.278	**8.676**	8.822
Fog	13.054	6.492	6.364	12.581	**6.302**	7.036	15.433	8.970	8.952	15.424	7.881	**7.441**
Motionblur	10.073	4.570	4.695	10.172	**4.408**	4.887	14.256	8.725	8.770	14.547	8.110	**7.940**
Beams Red.	15.583	8.612	8.587	15.125	8.022	**6.785**	18.223	11.734	11.655	13.287	7.451	**7.158**
Points Red.	10.606	4.513	4.406	10.374	4.635	**4.319**	14.423	7.658	7.753	17.420	10.783	**8.514**
Spatial Mis.	16.370	10.820	11.042	15.212	9.892	9.348	25.130	20.239	20.278	13.638	**7.243**	7.288
Mean	13.363	7.037	7.076	13.018	**6.777**	6.895	17.319	11.052	11.093	16.364	10.051	**9.273**

Table 13. **PETR - Regression Calibration under Distribution Shift.** Class-wise calibration (Cls.) and Global calibration (Glb.) reported for each corruption as calibration error MCA_{xyz} and KS_{xyz} (lower is better). Results are averaged over all detection thresholds and corruption severities. Ours: DA-TS.

Corruption	MCA_{xyz} ↓ (Calib: Cls.)				MCA_{xyz} ↓ (Calib: Glb.)				KS_{xyz} ↓ (Calib: Cls.)				KS_{xyz} ↓ (Calib: Glb.)			
	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS
Snow	5.967	7.356	4.141	**3.599**	5.967	13.063	5.811	**5.139**	17.996	24.158	13.264	**12.018**	17.996	39.370	17.494	**15.294**
Fog	4.670	6.242	3.448	**3.149**	4.670	11.775	4.539	**4.385**	12.634	21.780	10.711	**9.734**	12.634	36.086	12.117	**11.290**
Motionblur	5.062	8.349	3.618	**3.524**	5.062	13.733	4.906	**4.444**	13.538	26.861	11.115	**10.084**	13.538	41.416	13.015	**12.773**
Brightness	5.572	9.220	4.781	**4.501**	5.572	13.984	5.441	**5.390**	14.565	28.557	14.342	**12.916**	14.565	43.209	14.124	**14.079**
Darkness	4.236	5.665	**2.911**	2.954	4.236	11.155	**4.151**	4.319	11.628	20.615	9.748	**9.118**	11.628	35.430	11.315	**11.607**
Mean	5.101	7.367	3.780	**3.546**	5.101	12.742	4.970	**4.735**	14.072	24.394	11.836	**10.774**	14.072	39.102	13.613	**13.009**

Table 14. **PETR - Regression Calibration under Distribution Shift.** Class-wise calibration (Cls.) and Global calibration (Glb.) reported for each corruption as calibration error MCA_{xyz} and KS_{xyz} (lower is better). Results are averaged over all detection thresholds and corruption severities. Ours: DA-TS.

Corruption	MCA_{lwh} ↓ (Calib: Cls.)				MCA_{lwh} ↓ (Calib: Glb.)				MCA_{ϕ} ↓ (Calib: Cls.)				MCA_{ϕ} ↓ (Calib: Glb.)			
	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS	KL[15]	Depth	TS [3]	DA-TS
Snow	6.688	5.081	4.399	**4.301**	6.688	17.397	**6.510**	7.408	13.025	14.717	12.151	**10.920**	13.025	23.381	13.518	**10.348**
Fog	5.680	**3.804**	3.856	3.986	5.680	16.092	**5.453**	6.605	12.594	11.079	9.374	**8.907**	12.594	20.097	12.129	**10.729**
Motionblur	6.861	**4.235**	4.397	4.407	6.861	16.818	**6.720**	7.728	12.231	11.473	9.666	**9.057**	12.231	20.155	11.857	**10.399**
Brightness	6.305	6.189	4.829	**4.698**	6.305	17.036	**6.100**	6.395	11.640	12.988	10.819	**9.764**	11.640	21.218	11.820	**9.932**
Darkness	5.631	4.060	**3.645**	3.692	5.631	16.473	**5.466**	6.415	13.009	10.922	9.445	**9.016**	13.009	19.301	12.526	**11.024**
Mean	6.233	4.674	4.226	**4.217**	6.233	16.763	**6.050**	6.910	12.500	12.236	10.291	**9.533**	12.500	20.830	12.370	**10.486**

Table 15. **PETR - Classification Calibration under Distribution Shift.** All results are reported for class-wise calibration (Cls.) and are averaged over all detection thresholds and all three corruption severities.

Corruption	D-ECE↓ (Calib: Cls.)						LaECE↓ (Calib: Cls.)					
	Baseline			Density-aware (Ours)			Baseline			Density-aware (Ours)		
	TS[5]	PS[12]	IR[14]	DA-TS	DA-PS	DA-IR	TS[5]	PS[12]	IR[14]	DA-TS	DA-PS	DA-IR
Snow	10.666	8.178	7.496	10.481	8.215	**7.246**	26.919	24.650	24.456	25.633	23.166	**22.201**
Fog	8.398	6.556	**5.946**	8.060	7.043	6.851	26.287	23.938	23.816	25.031	22.729	**21.820**
Motionblur	9.315	6.778	**6.274**	8.840	6.834	6.744	25.696	23.473	23.089	24.955	22.578	**21.629**
Brightness	11.788	9.551	9.137	11.757	9.784	**8.709**	28.557	26.028	25.676	27.906	24.983	**23.736**
Darkness	11.190	8.886	8.394	9.136	6.973	**6.504**	26.945	24.917	24.630	24.720	22.200	**21.675**
Mean	10.271	7.990	7.449	9.655	7.770	**7.211**	26.881	24.602	24.333	25.649	23.131	**22.212**

Figure 2. **In-Distribution** PETR with KL [15] - Uncalibrated.

Figure 3. **In-Distribution** PETR with KL [15] calibrated with DA-TS for regression and DA-IR for classification.

Figure 4. **In-Distribution** PETR - DE [8].

Figure 5. **In-Distribution** PETR - MCD [4].

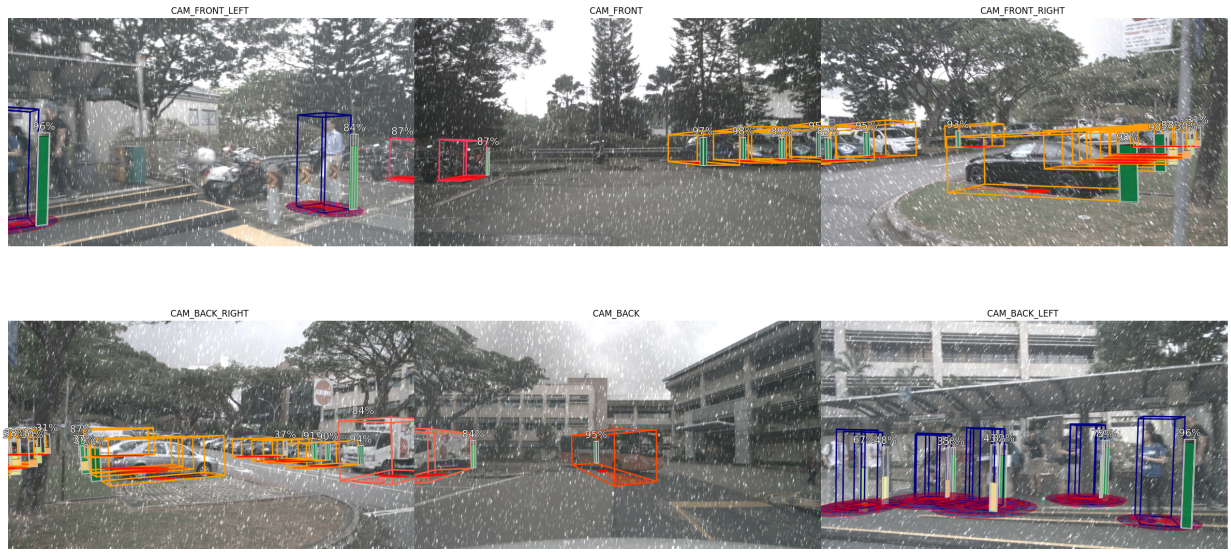


Figure 6. **Distribution Shift - Snow Level 2** PETR with KL [15] - Uncalibrated.

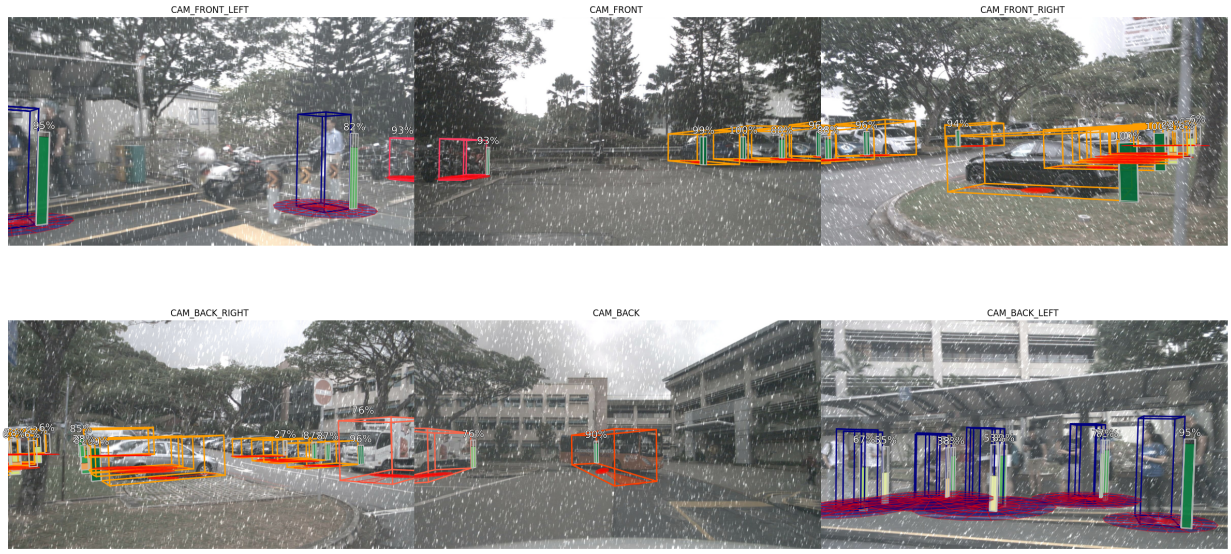


Figure 7. **Distribution Shift - Snow Level 2** PETR with KL [15] calibrated with DA-TS for regression and DA-IR for classification.

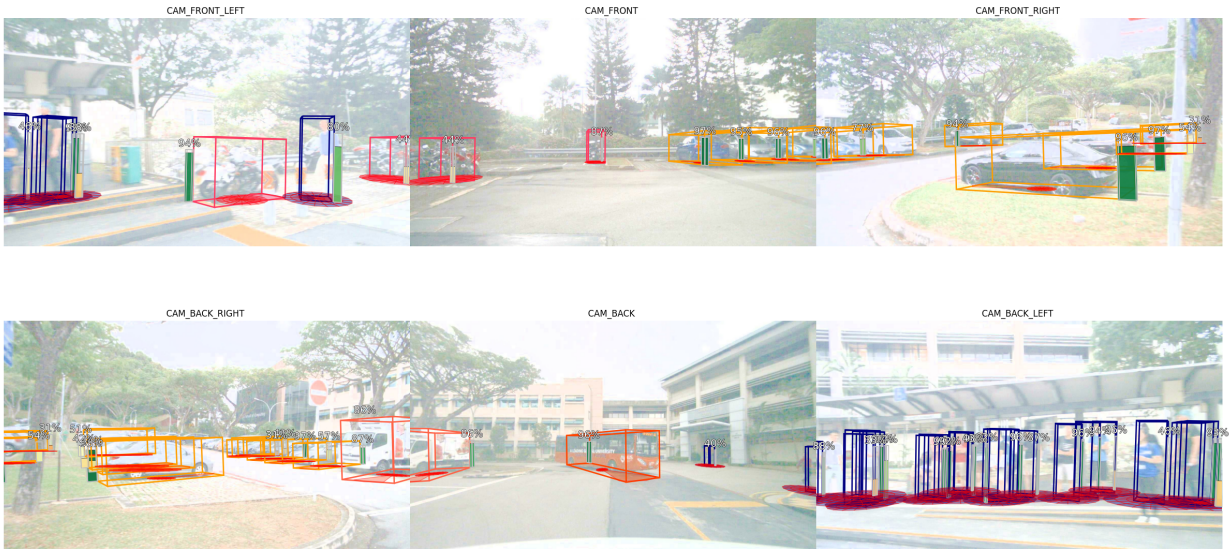


Figure 8. **Distribution Shift - Brightness Level 1** PETR with KL [15] - Uncalibrated.

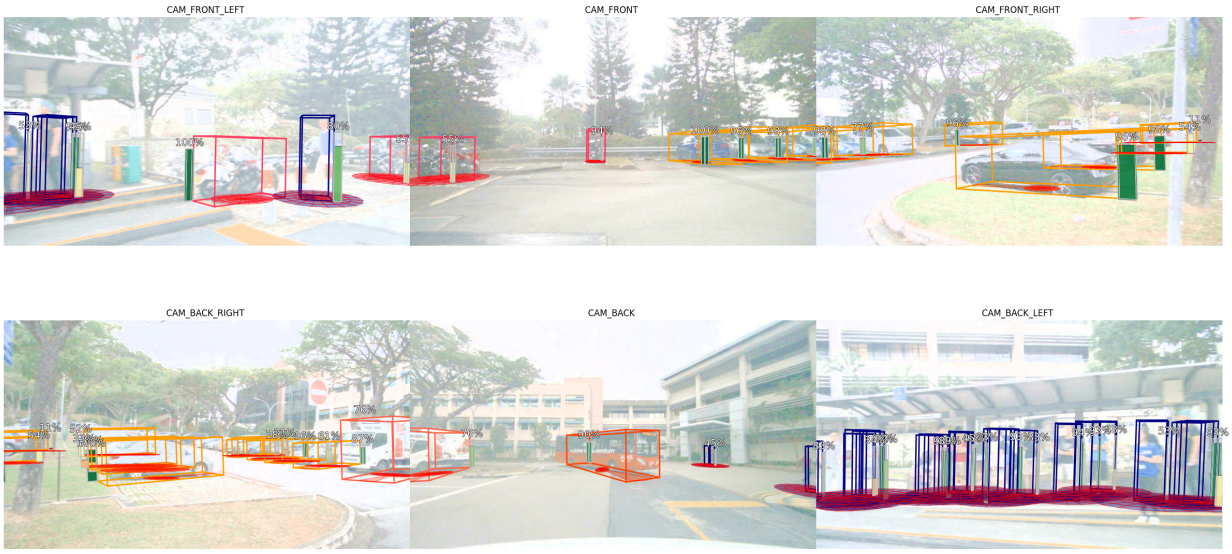


Figure 9. **Distribution Shift - Brightness Level 1** PETR with KL [15] calibrated with DA-TS for regression and DA-IR for classification.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*, 2020. 1

[2] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection, 2020. 1

[3] Di Feng, Lars Rosenbaum, Claudius Glaeser, Fabian Timm, and Klaus Dietmayer. Can we trust you? on calibration of a probabilistic object detector for autonomous driving. In *IROS*, 2019. 3, 5, 6

[4] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Int. Conf. Mach. Learn.*, 2017. 2, 3, 4, 8

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Int. Conf. Mach. Learn.*, page 1321–1330, 2017. 2, 3, 6

[6] Ali Harakeh and Steven L. Waslander. Estimating and Evaluating Regression Predictive Uncertainty in Deep Object Detectors, 2021. 3

[7] Selim Kuzucu, Kemal Oksuz, Jonathan Sadeghi, and Puneet K. Dokania. On calibration of object detectors: Pitfalls, evaluation and baselines. In *ECCV*, 2024. 1

[8] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *NeurIPS*, 2017. 2, 3, 4, 8

[9] Muhammad Akhtar Munir, Muhammad Haris Khan, M Sarfraz, and Mohsen Ali. Towards improving calibration in object detection under domain shift. *NeurIPS*, 35:38706–38718, 2022. 2, 4

[10] Muhammad Akhtar Munir, Salman Khan, Muhammad Haris Khan, Mohsen Ali, and Fahad Khan. Cal-DETR: Calibrated Detection Transformer. *NeurIPS*, 2023. 2, 3

[11] Matthew Pitropov, Chengjie Huang, Vahdat Abdelzad, Krzysztof Czarnecki, and Steven Waslander. LiDAR-MIMO: Efficient Uncertainty Estimation for LiDAR-based 3D Object Detection. In *IEEE Intell. Vehicles Symp.*, pages 813–820, 2022. 3

[12] John Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif.*, 10, 1999. 2, 3, 6

[13] Joachim Sicking, Alexander Kister, Matthias Fahrland, Stefan Eickeler, Fabian Hüger, Stefan Rüping, Peter Schlicht, and Tim Wirtz. Approaching Neural Network Uncertainty Realism. In *NeurIPS 2019 Workshop on Machine Learning for Autonomous Driving*, Vancouver, Canada, 2019. 2

[14] Bianca Zadrozny and Charles Elkan. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. *Int. Conf. Knowl. Discov. Data Min.*, 2002. 2, 3, 6

[15] Yuanxin Zhong, Minghan Zhu, and Huei Peng. Uncertainty-Aware Voxel based 3D Object Detection and Tracking with von-Mises Loss. *arXiv preprint arXiv:2011.02553*, 2020. 3, 5, 6, 7, 9, 10