Adaptive Dual Uncertainty Optimization: Boosting Monocular 3D Object Detection under Test-Time Shifts

————Appendix————

The structure of Appendix is as follows:

- Appendix A contains all missing proofs in the main manuscript.
- Appendix B presents further experimental results on more corruption levels.
- Appendix C provides additional ablation studies to validate the robustness and efficiency of our method.
- Appendix D details the compared methods, model architecture, and datasets used for comparison.

A. Theoretical Proof

Below, we provide detailed proofs of the theoretical results presented in Sec. 5.1 of the main paper.

Notation. First, we recall the notation that we used in the main paper as well as this appendix: x denotes an inputting test image and y denotes the one-hot coding of the ground-truth label. h_θ denotes the model with its parameter set θ and $h \triangleq h_\theta(x)$. $s \triangleq e^{h_1} + ... + e^{h_c}$ is the sum over the exponential outputs of the model and c is the number of classes. $p \triangleq \texttt{softmax}(h)$ is the normalized probability over the classes. $\text{diag}(\cdot)$ denotes the diagonal matrix and I denotes the identity matrix. We define the following two functions: $f(h) = \alpha \log s$, $g(h) = \alpha h + \alpha((1-p)^\gamma - 1)\log p$.

A.1. Legendre-Fenchel Structure

In this subsection, we demonstrate the equivalence of the vanilla focal loss with its Legendre-Fenchel structure.

$$\begin{aligned}
\mathcal{L}_{\text{FL}}(x, y) &= -\alpha(1-p)^\gamma \cdot y \log p \\
&= -\alpha(1-p)^\gamma \cdot y \left(\log \begin{pmatrix} e^{h_1} \\ \vdots \\ e^{h_c} \end{pmatrix} - \log \left(e^{h_1} + \cdots + e^{h_c} \right) \right) \\
&= -\alpha(1-p)^\gamma \cdot y \cdot \log \begin{pmatrix} e^{h_1} \\ \vdots \\ e^{h_c} \end{pmatrix} + \alpha(1-p)^\sigma y \log s \\
&= -\alpha(1-p)^\gamma \cdot y \cdot h + \alpha \cdot y \log s + \alpha \left((1-p)^\sigma - 1 \right) y \log s.
\end{aligned} \tag{15}$$

Duo to y is the one-hot vector, we have $y \log s = \log s$ and we further derive:

$$\begin{aligned}
\mathcal{L}_{\text{FL}}(x, y) &= \alpha \log s + \alpha y \left(-(1-p)^\gamma \cdot h + (1-p)^\gamma \log s - \log s \right) \\
&= \alpha \log s + \alpha y \left(-(1-p)^\gamma \cdot (h - \log s) - \log s \right) \\
&= \alpha \log s + \alpha y \left(-\log s - (1-p)^\gamma \log p \right) \\
&= \alpha \log s - y^\top \left(\alpha \log s + \alpha (1-p)^\gamma \log p \right) \\
&= \underbrace{\alpha \log s}_{f(h)} - y^\top \underbrace{\left(\alpha h + \alpha \left((1-p)^\gamma - 1 \right) \log p \right)}_{g(h)}.
\end{aligned} \tag{16}$$

Therefore, the focal loss can be formulated as a classical convex conjugate structure, allowing for further analysis in the conjugate optimization framework.

A.2. Problem Reconstruction.

In this section, we demonstrate the invertibility of function g to ensure the existence of a conjugate function and further reformulate the optimization problem into conjugate relationships. by the inverse function theorem, the local invertibility of g is guaranteed if its Jacobian is non-singular. For simplicity, we demonstrate the positive definiteness of the Jacobian under the default setting $\gamma = 2$:

Step 1: Jacobian of g. Taking the gradient of g with respect to h, we obtain:

$$\nabla_h g = I + \text{diag}(p - 2 - 2(1-p)\log p) \cdot (\text{diag}(p) - pp^\top) := I + D \cdot H. \tag{17}$$

where $D = \text{diag}(p - 2 - 2(1-p)\log p)$ and $H = \text{diag}(p) - pp^\top$.
Step 2: Positive Semidefiniteness of H. For any vector $\mathbf{v} \in \mathbb{R}^C$, the quadratic form of $\mathbf{H}$ is:

$$\mathbf{v}^\top \mathbf{H}\mathbf{v} = \sum_{i=1}^{C} p_i v_i^2 - \left(\sum_{i=1}^{C} p_i v_i\right)^2 \tag{18}$$

Let V be a random variable that takes the value v_i with probability p_i. The quadratic form simplifies to:

$$\mathbf{v}^\top \mathbf{H}\mathbf{v} = \mathbb{E}\left[V^2\right] - (\mathbb{E}[V])^2 = \text{Var}(V) \tag{19}$$

Since variance is always non-negative, we have:

$$\mathbf{v}^\top \mathbf{H}\mathbf{v} \geq 0 \quad \forall \mathbf{v} \in \mathbb{R}^C \tag{20}$$

Thus, $\mathbf{H}$ is positive semi-definite. Furthermore, using Gershgorin's circle theorem, all eigenvalues of H satisfy:

$$0 \leq \lambda(H) \leq \max_i \{2p_i\,(1-p_i)\} \leq \frac{1}{2}, \tag{21}$$

where $2p_i\,(1-p_i)$ attains its maximum $\frac{1}{2}$ when $p_i = \frac{1}{2}$.
Step 3: Bounding the Eigenvalues of $\nabla_h g$. Since D is diagonal, the matrix $D \cdot H$ can be seen as a row-scaled version of H. And the element of H follows:

$$D_{ii} = p_i - 2 - 2\,(1-p_i)\log p_i. \tag{22}$$

Numerical analysis of the function $\phi(p_i) = p_i - 2 - 2\,(1-p_i)\log p_i$ shows $D_{ii} > -2$ for $p_i \in (0,1)$. Therefore, the eigenvalues of $\nabla_h g$ satisfy:

$$\lambda\left(\nabla_h g\right) \geq 1 + \lambda_{\min}(D) \cdot \lambda_{\min}(H). \tag{23}$$

Since $\lambda_{\min}(H) \geq 0$ and $D_{ii} > -2$, we can obtain a tighter bound:

$$\lambda\left(\nabla_h g\right) \geq 1 - 2 \cdot \lambda(H) \geq 1 - 2 \cdot \frac{1}{2} = 0. \tag{24}$$

Step 4: Conclusion. As all eigenvalues of $\nabla_h g$ are non-negative, the Jacobian is positive semidefinite. In particular, as long as p is not degenerate (i.e., no prediction has probability exactly 0 or 1), H is full rank on its subspace, ensuring that $\nabla_h g$ is non-singular in a neighborhood of h. Thus, by the inverse function theorem, g is locally invertible.

This invertibility guarantees the existence of a conjugate function f^*, which equals to the minimization value of the objective:

$$\min_h \{f(h) - y^\top g(h)\} = \min_{z=g(h)} \{f \circ g^{-1}(z) - y^\top z\} = f^*(y). \tag{25}$$

Under the common assumption that the representation h pre-trained from the large source dataset is already close to a local optimal solution h_0, we can convert the problem into the following conjugate relationships:

$$f \circ g^{-1}(z) - y^\top z = f^*(y),\ \nabla_z (f \circ g^{-1}) = y. \tag{26}$$

A.3. Conjugate Focal Loss

In this subsection, we need to derive the estimation of y from the conjugate conditions in Equ. 26. For the derivative of g, we differentiate the two summation components separately:

$$g(h) = \alpha\, h + \alpha\, \phi(p), \quad \text{with} \quad \phi(p) = ((1-p)^\gamma - 1) \cdot \log p. \tag{27}$$

For the first part, we have $\nabla_h h = I$. For the second part, we utilize the chain rule to derive the following format:

$$\nabla_h \phi = \nabla_p \phi \cdot \nabla_h p. \tag{28}$$

We calculate the $\nabla_p \phi$ as follows:

$$\nabla_p \Big(((1-p)^\gamma - 1) \log p \Big) = \frac{(1-p)^\gamma - 1}{p} - \gamma \, (1-p)^{\gamma-1} \log p. \tag{29}$$

We calculate the $\nabla_h p$ as follows:

$$\nabla_h p = \text{diag}(p) - pp^\top. \tag{30}$$

Thus, the Jacobian of g is given by

$$\nabla_h g(h) = \alpha \Big[I + (\frac{(1-p)^\gamma - 1}{p} - \gamma (1-p)^{\gamma-1} \log p) \cdot (\text{diag}(p) - pp^\top) \Big]. \tag{31}$$

For the composite function $f \circ g^{-1}$, the chain rule gives

$$\nabla_z (f \circ g^{-1})(z) = \nabla_h f(h) \left(\nabla_h g(h) \right)^{-1}. \tag{32}$$

Evaluating at h_0 (with $p = \text{softmax}(h_0)$) leads to

$$y_0 = \left(\nabla_h g(h_0) \right)^{-1} \nabla_h f(h_0) = \Big[I + (\frac{(1-p)^\gamma - 1}{p} - \gamma (1-p)^{\gamma-1} \log p) \cdot (\text{diag}(p) - pp^\top) \Big]^{-1} p. \tag{33}$$

And we utilize the Taylor's Formula to approximate the value (neglecting higher-order terms of p):

$$(1-p)^\gamma \approx 1 - \gamma p + \frac{\gamma(\gamma-1)}{2} \text{diag}(pp^T), (1-p)^{\gamma-1} \log p \approx (1 - (\gamma - 1)p) \log p. \tag{34}$$

After algebraic manipulation (and neglecting higher-order terms in p and $\log p$), we obtain

$$y_0 \approx \Big(I + \gamma \Big((1 - \log p) \, pp^\top - \log p \;\; \text{diag}(p) \Big) \Big)^{-1} p. \tag{35}$$

Thus, by applying the chain rule and approximating the Jacobian of g to higher order, we obtain

$$y_0 \triangleq \left. \frac{\nabla_h (f \circ g^{-1})}{\nabla_h z} \right|_{z = g(h_0)} \approx \Big(I + \gamma (1 - \log p) \, pp^\top - \gamma \log p \;\; \text{diag}(p) \Big)^{-1} p. \tag{36}$$

Finally, we substitute this estimation into Equ. 26, yielding Conjugate Focal Loss:

$$\mathcal{L}_{\text{CFL}}(x) = f(h) - y_0^\top g(h) = -\alpha (1-p)^\gamma (I + \gamma (1 - \log p) \cdot p^\top p - \gamma \log p \cdot \text{diag}(p))^{-1} p \log p. \tag{37}$$

B. Further Experiments

In this section, we broaden our investigation by evaluating our method across a variety of shift severity levels. To this end, we conduct experiments on corruption scenarios with shift level 1, 3, 5. These experiments allow us to thoroughly examine the robustness of our approach under different severity levels of distribution shifts.

B.1. Different Severity Level Corruption

We further provide more discussions surrounding high-severity data corruptions (*i.e.* 3 and 1) based on the experimental results shown in Tab. 5&6, which clearly gives additional observations: 1) With the escalation of severity level, the source models suffer a larger performance decline within various corruptions. For instance, the pre-trained models of MonoFlex and MonoGround achieve obvious performance drop from level 1 to level 3, which significantly heightens the challenge for test-time adaptation. 2) Existing TTA methods struggle to recover performance under such extreme corruptions, highlighting the limitations of conventional uncertainty optimization approaches. 3) Despite these challenges, our DUO framework consistently achieves the best average performance across all corruption types. This robust performance demonstrates that our dual uncertainty optimization framework effectively stabilizes both the semantic classification and spatial perception branches, providing reliable adaptation even under different shift levels.

C. Additional Ablation Study

In Sec. 6.3 of the main paper, we provide a comprehensive analysis of each component's effectiveness and their complementary interactions. In this section, we extend our analysis by examining the sensitivity of key parameters and comparing running times, offering further insights into the robustness and efficiency of our method.

C.1. Hyperparameter Robustness

Our method involves two key hyperparameters: the coefficient λ, which determines the trade-off of the semantic and geometric uncertainty optimization, and the coefficient α, which controls the weighting scale in conjugate focal loss. We conduct ablation experiments on these two key coefficients independently:

As shown in Fig. 7(a), the different strengths of geometric constant yields stable performance gains. However, when λ exceeds the optimal range (*e.g.*, 1.1), the model tends to over-prioritize geometric consistency over uncertainty optimization, leading to worse performance. To balance the effects of two components in our method, we set λ to 0.7 by default. In Fig. 7(b), we observe that the weighting coefficient α consistently outperforms prior SOTA methods, demonstrating the robustness of our Conjugate Focal Loss weighting scheme. Empirically, we set α to 4 by default.

Notably, the default choice of α and γ not only yields strong empirical performance but also aligns with the standard settings used in vanilla focal loss during source training. This compatibility echoes our theoretical analysis in Sec.5.1, suggesting that hyperparameters can remain unchanged from the source phase. Such consistency removes the need for extensive hyperparameter tuning, significantly improving the efficiency and practicality of our adaptation strategy.

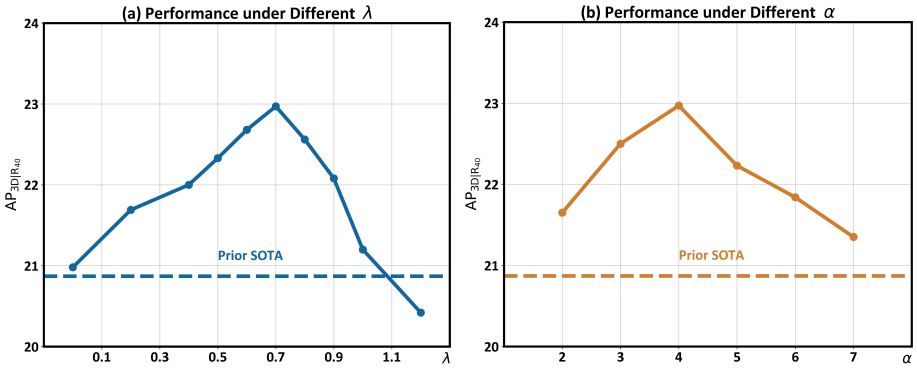

Figure 7. (*a*) Performance with varying strengths λ of the normal field constraint. (*b*) Performance with different weighting scare α of the conjugate focal loss.

Table 5. Comparisons with state-of-the-art methods on the KITTI-C *validation* set **(severity level 3)** in Car category. We highlight the best and second results with **bold** and underline respectively.

Method	Reference	Noise			Blur			Weather				Digital			Avg
		Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Snow	Frost	Fog	Brit.	Contr.	Pixel	Sat.	
MonoFlex	CVPR'21	0.63	0.49	0.63	1.09	26.10	0.71	14.21	15.88	10.16	27.88	4.41	11.61	39.25	11.77
• TENT	ICLR'21	17.99	26.99	22.29	13.46	35.73	9.36	32.52	30.99	38.13	40.67	39.28	34.46	43.37	29.63
• EATA	ICML'22	18.21	27.52	22.83	14.86	36.01	13.98	33.11	31.45	38.35	40.62	39.55	35.23	43.44	30.39
• DeYO	ICLR'24	18.36	28.49	23.15	15.04	36.44	16.38	33.67	31.32	38.57	40.75	39.93	35.81	43.58	30.89
• MonoTTA	ECCV'24	19.64	28.37	24.45	17.79	35.91	17.20	34.11	31.78	39.45	40.83	40.74	36.27	43.46	31.54
• Ours	This paper	**21.18**	**29.43**	**25.43**	**19.09**	**36.85**	**18.88**	**35.32**	**31.96**	**39.77**	**41.64**	**41.71**	**36.73**	**43.93**	**32.46**
MonoGround	CVPR'22	0.51	0.52	0.86	2.47	25.71	0.35	10.68	9.99	5.59	32.31	0.81	14.94	36.06	10.83
• TENT	ICLR'21	20.01	31.16	25.56	17.72	38.63	10.47	33.58	30.83	38.06	42.78	40.11	39.56	**45.49**	31.84
• EATA	ICML'22	20.36	31.84	26.63	18.39	38.77	14.21	34.03	31.08	37.93	42.32	40.32	39.57	45.30	32.37
• DeYO	ICLR'24	20.80	32.31	27.32	19.33	38.63	15.37	34.58	31.43	33.95	**42.97**	40.33	39.83	45.20	32.47
• MonoTTA	ECCV'24	22.10	33.93	28.35	22.49	39.88	16.64	32.71	32.42	39.93	42.69	40.61	39.92	44.85	33.58
• Ours	This paperX	**23.65**	**34.79**	**29.23**	**23.08**	**40.63**	**18.66**	**34.75**	**33.38**	**40.39**	**42.97**	**41.64**	**40.53**	44.91	**34.51**

Table 6. Comparisons with state-of-the-art methods on the KITTI-C *validation* set **(severity level 1)** in Car category. We highlight the best and second results with **bold** and underline respectively.

Method	Reference	Noise			Blur			Weather				Digital			Avg
		Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Snow	Frost	Fog	Brit.	Contr.	Pixel	Sat.	
MonoFlex	CVPR'21	12.97	20.42	15.02	20.37	36.51	11.61	32.26	30.61	19.69	**45.33**	20.01	29.09	42.44	25.87
• TENT	ICLR'21	29.84	38.55	34.54	34.93	40.52	25.08	39.68	40.53	40.30	44.37	44.04	41.10	43.92	38.26
• EATA	ICML'22	30.13	38.69	34.77	35.43	40.16	27.93	39.85	40.38	40.60	44.81	44.43	41.39	44.30	38.68
• DeYO	ICLR'24	30.58	38.82	34.93	36.04	**41.00**	28.64	**39.96**	40.51	40.62	44.79	44.46	41.48	**44.90**	38.98
• MonoTTA	ECCV'24	**32.34**	39.05	35.68	36.58	40.69	30.25	39.70	40.01	41.22	44.76	**44.88**	41.91	44.20	39.33
• Ours	This paper	32.28	**39.42**	**36.78**	**36.87**	40.91	**31.33**	39.88	**40.71**	**41.23**	44.90	44.41	**42.47**	44.44	**39.66**
MonoGround	CVPR'22	13.05	22.05	19.41	20.75	38.72	8.40	30.65	27.66	14.56	46.22	14.95	33.40	36.29	25.08
• TENT	ICLR'21	34.94	42.76	37.93	37.79	44.95	25.15	40.67	42.77	41.26	**47.05**	45.12	43.73	46.56	40.82
• EATA	ICML'22	35.36	42.47	38.85	38.24	44.87	26.44	40.64	42.61	41.65	46.94	45.18	43.71	46.54	41.04
• DeYO	ICLR'24	35.88	42.07	39.86	38.51	44.81	28.01	40.60	42.49	41.95	46.83	45.25	43.67	46.54	41.27
• MonoTTA	ECCV'24	37.05	42.86	39.52	39.25	44.59	32.66	40.54	42.47	42.13	45.95	44.98	43.38	46.15	41.66
• Ours	This paper	**37.25**	**43.31**	**40.21**	**39.80**	**45.30**	**34.16**	**41.39**	**42.80**	**42.84**	46.61	**45.93**	**43.80**	**46.66**	**42.31**

Table 7. Comparisons with state-of-the-art methods on the KITTI-C *validation* set (severity level 5) with MonoGround. We highlight the best and second results with **bold** and underline respectively.

Car Category

Method	Reference	Noise			Blur			Weather				Digital			Avg
		Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Snow	Frost	Fog	Brit.	Contr.	Pixel	Sat.	
MonoGround	CVPR'22	0.00	0.00	0.00	0.00	11.63	0.29	1.95	6.59	3.14	19.25	0.00	4.66	3.74	3.94
• TENT	ICLR'21	6.82	14.81	8.21	4.88	28.38	2.65	23.92	28.08	33.06	36.70	20.22	30.63	33.27	20.90
• EATA	ICML'22	7.12	15.26	8.81	5.09	29.08	2.52	24.18	28.03	33.43	36.78	21.61	30.50	33.42	21.22
• DeYO	ICLR'24	7.35	15.72	9.38	5.74	30.01	2.99	25.03	28.55	34.32	37.31	23.41	30.99	34.16	21.92
• MonoTTA	ECCV'24	7.88	16.73	10.35	5.97	31.19	3.06	25.24	28.99	34.85	37.82	25.00	31.61	34.79	22.57
• Ours	This paper	**9.72**	**18.88**	**12.74**	**7.24**	**33.02**	**5.24**	**28.50**	**30.73**	**37.27**	**39.40**	**28.34**	**33.22**	**37.24**	**24.73**

Pedestrian Category

Method	Reference	Noise			Blur			Weather				Digital			Avg
		Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Snow	Frost	Fog	Brit.	Contr.	Pixel	Sat.	
MonoGround	CVPR'22	0.00	0.00	0.00	0.00	14.76	0.00	0.28	0.74	0.68	4.63	0.00	0.34	1.80	1.79
• TENT	ICLR'21	1.47	2.91	1.01	1.19	15.19	0.66	6.98	10.44	14.95	17.49	11.10	10.72	8.72	7.91
• EATA	ICML'22	1.85	2.86	1.05	1.31	14.02	0.79	7.41	10.08	14.72	17.57	11.31	11.20	9.38	7.97
• DeYO	ICLR'24	2.25	2.81	1.08	1.46	13.28	0.92	7.75	9.74	14.45	17.64	11.49	11.68	9.99	8.04
• MonoTTA	ECCV'24	**2.40**	4.74	1.52	1.60	**16.31**	1.09	8.95	11.06	14.72	17.96	10.62	**12.39**	12.11	8.88
• Ours	This paper	2.26	**5.03**	**1.85**	**2.24**	16.26	**2.29**	**10.44**	**12.37**	**15.50**	**18.89**	**12.59**	12.35	**12.95**	**9.62**

Cyclist Category

Method	Reference	Noise			Blur			Weather				Digital			Avg
		Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Snow	Frost	Fog	Brit.	Contr.	Pixel	Sat.	
MonoGround	CVPR'22	0.00	0.00	0.00	0.00	0.47	0.00	0.10	1.20	0.21	3.85	0.00	0.76	0.19	0.52
• TENT	ICLR'21	**1.77**	**0.14**	0.04	0.07	2.92	0.31	1.92	2.70	6.90	8.14	1.08	1.51	2.71	2.32
• EATA	ICML'22	0.88	0.13	0.05	0.06	2.94	0.40	2.03	2.81	6.91	8.36	1.32	1.52	2.98	2.34
• DeYO	ICLR'24	0.00	0.12	**0.07**	0.06	2.96	0.49	2.13	2.93	6.91	**8.58**	1.57	1.54	3.25	2.35
• MonoTTA	ECCV'24	0.04	0.10	0.04	0.15	3.59	0.52	2.51	3.96	8.45	7.80	3.00	**2.90**	3.61	2.82
• Ours	This paper	0.05	**0.14**	**0.07**	**0.24**	**4.01**	**0.70**	**2.67**	**4.20**	**8.79**	8.22	**3.91**	2.55	**3.72**	**3.02**

C.2. Running Time Comparison

In our experiments, we have demonstrated the effectiveness of DUO in various scenarios. In this subsection, we focus on the computational efficiency of DUO. Although our method relies on dual-branch optimization, the computation of the geometric constraint with efficient operators incurs only a slight time cost. As shown in Table 8, DUO's running time is less than half that of DeYO, which relies heavily on data augmentation to optimize the uncertainty. Moreover, DUO's adaptation efficiency exceeds that of MonoTTA, which only optimizes semantic uncertainty without addressing geometric uncertainty. Notably,

processing 1k images with DUO adds only an extra 6 seconds compared to inference alone, underscoring the high efficiency of our dual-branch optimization framework.

Table 8. Running time comparison of various methods. We assess TTA approaches for processing 1k images in Gaussian corruption type, using a single Nvidia RTX 4090 GPU.

Metrics	Source Model	TENT	EATA	DeYO	MonoTTA	Ours
Running Time	26s	31s	29s	87s	33s	32s

D. More Implementation Details

D.1. Baseline Methods

We compare our DUO with several state-of-the-art methods. TENT [46] reduces the entropy of test samples to guide model updates, prompting the model to generate more confident predictions. Building on this, EATA [36] incorporates a sample selection mechanism based on low uncertainty to specifically minimize entropy for the most reliable samples, thereby further reducing semantic uncertainty. DeYO [22] prioritizes samples with dominant shape information and employs a dual semantic uncertainty criterion to identify reliable samples for adaptation. MonoTTA [24] introduces a negative regularization term on low-score objects, leveraging their negative class information to reduce uncertainty.

D.2. Detailed Model Architecture

Our framework is built on a widely-adopted multi-branch architecture for monocular 3D object detection, where separate branches predict various object properties to simultaneously achieve recognition and spatial localization. In 3D detection, accurate depth estimation is a critical factor that significantly influences overall performance [32]. To enhance depth prediction, many existing models adopt a multi-head strategy that integrates diverse depth estimates to reduce the individual bias. For example, MonoFlex [55] combines direct regression with multiple keypoint estimation; MonoGround leverages ground plane priors for refined depth predictions [40]; and MonoCD exploits the complementary strengths of multiple prediction heads [53].

To effectively integrate the multi-head predictions, these models includes an uncertainty estimation branch that quantifies the reliability of each depth prediction. The final depth estimation is computed as an uncertainty-weighted average, as shown in the following formulation:

$$z_{\text{soft}} = \left(\sum_{i=1}^{n} \frac{z_i}{\sigma_i} \right) / \left(\sum_{i=1}^{n} \frac{1}{\sigma_i} \right), \tag{38}$$

where the σ_i is the uncertainty of the corresponding depth estimation and n is the number of depth heads. In the main paper, we use the average of the $\log \sigma_i$ as the **depth uncertainty metric**.

Furthermore, the uncertainty regression loss for the entire depth branch is designed as:

$$L_{dep} = \sum_{i} \left[\frac{|z_i - z^*|}{\sigma_i} + \log\left(\sigma_i\right) \right], \tag{39}$$

where the z^* is the ground-truth depth.

For our TTA setting, we attempt to utilize the weighted average z_{soft} as a pseudo-label to optimize this loss, directly optimizing the depth uncertainties. However, this approach can lead to model collapse—a phenomenon we analyze in detail in Sec. 4 of the main paper.

For Monoflex [55] and MonoGround [40], we follow their original settings by using a randomly generated seed. Both Monoflex and MonoGround employ the same modified DLA-34 [54] as their backbone network, with input resolutions of 384×1280 for the KITTI-C and 928×1600 for nuScenes, respectively.

D.3. More Details on Dataset

KITTI-C Dataset. We follow the protocol from [24, 55] to partition the KITTI dataset into a training set (3712 images) and a validation set (3769 images) for model training and adaptation, respectively. For evaluation, we employ the KITTI-C version, which applies 13 distinct corruptions to the validation set—namely, Gaussian noise, shot noise, impulse noise,

defocus blur, glass blur, motion blur, snow, frost, fog, brightness, contrast, pixelation, and saturation [14], as shown in Fig. 8. Each corruption is further divided into five severity levels, with higher levels indicating more extreme perturbations and distribution shifts.

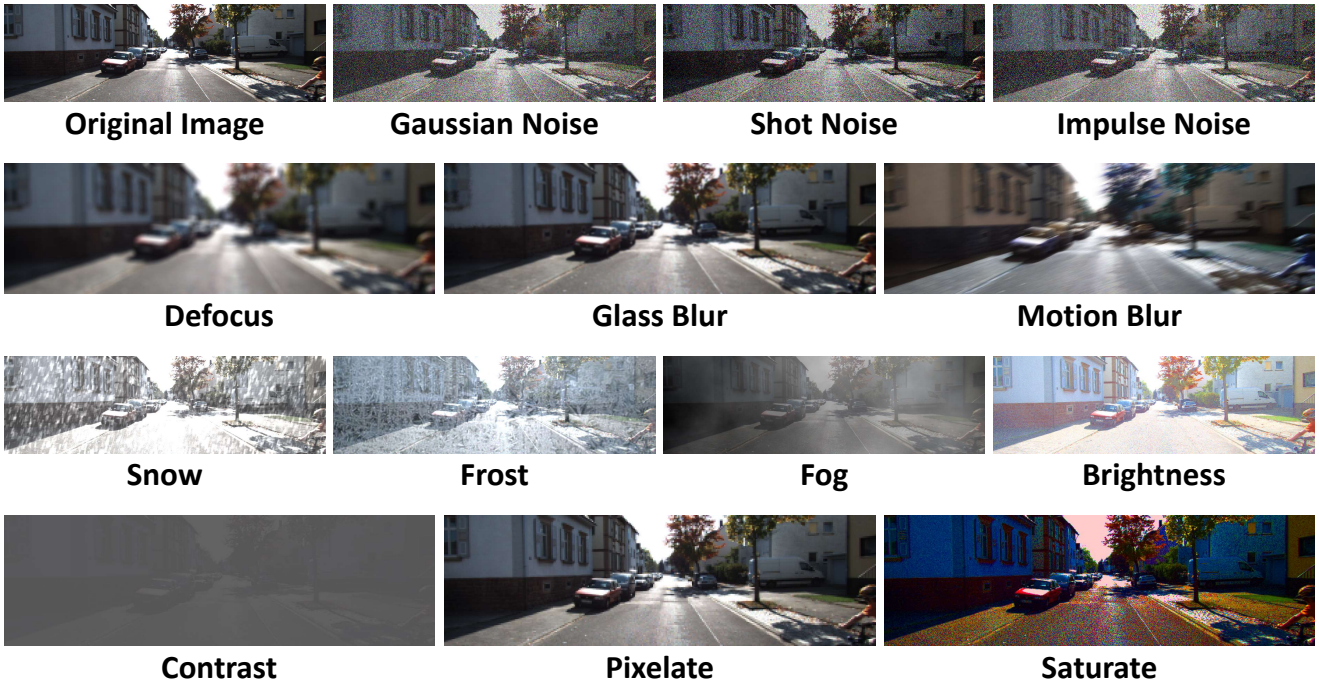

Figure 8. An illustration of 13 distinct types of corruptions in the severity level 3 of the KITTI-C dataset.

nuScenes Dataset. For the four real-world scenarios in the nuScenes dataset, we first extract all *front-view* images and convert them to KITTI format using the official devkit [5]. Following [29], we partition these images into Daytime, Night, Sunny, and Rainy scenarios based on their scene descriptions. For each scenario, we train our model on the training split and evaluate its performance on the validation split (the number of images per scenario is shown in Fig. 9). Since the Night scenario contains fewer than 4k images with fewer objects (e.g., pedestrians), we report results only for the *Car* category.

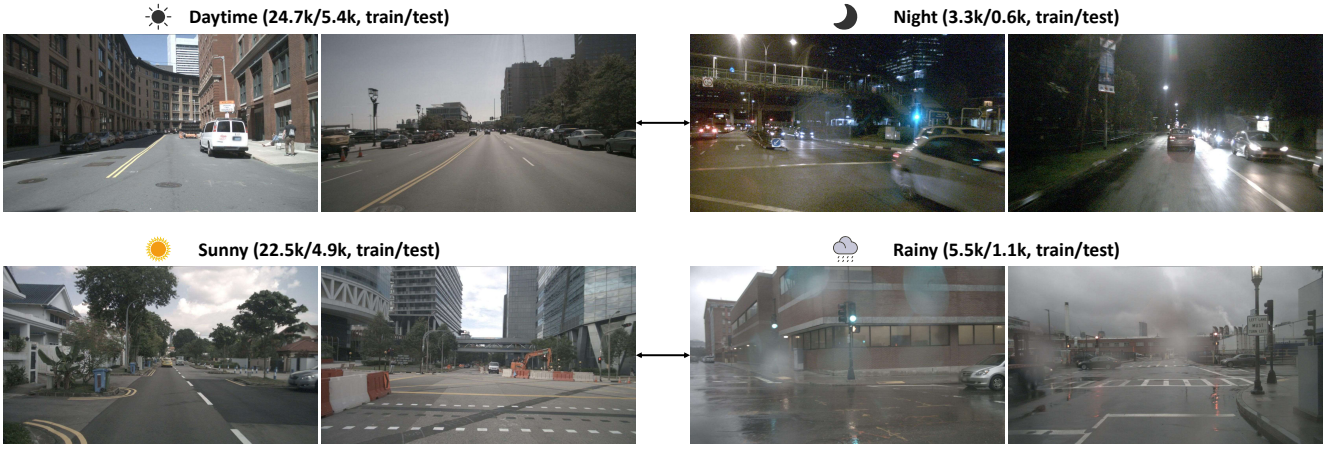

Figure 9. An illustration of the Daytime, Night, Sunny, and Rainy scenarios of the nuScenes dataset.