

Supplementary Tables

This supplement collects supporting tables that are referenced from the main body but moved here to keep the main text concise.

Table S1. Comparison of uncertainty quantification (UQ) methods considered in this work and their suitability in Uncertainty aware Fault Classification.

Method	Uncertainty type	Meta-learning (with MAML)	integration	Active sampling	Suitability
MC Dropout [24]	Epistemic	Post-hoc compatible (apply at meta-test / during adaptation)		No	Moderate
SVGD [26]	Epistemic	Nontrivial (requires particle / meta-training design)		No	Limited
S ² VGD [27]	Epistemic	Nontrivial (not plug-and-play with MAML)		No	Limited
EMP [30]	Epistemic	Post-hoc compatible (can wrap MAML)		No	Moderate
ECMP [30]	Epistemic	Post-hoc compatible		No	Moderate
Deep Ensembles [28]	Epistemic	Post-hoc compatible (ensemble of MAML initializations)		No	Moderate
Bayesian MAML [31]	Epistemic	Native (particle-based adaptation)		No	Moderate
Probabilistic initializations, e.g., PLATIPUS [32]	Epistemic	Native (distribution over initialization)		No	Moderate
Active-learning shot methods [35]	Varies			Yes (for labeling)	Limited
UDRML (Ours)	Aleatoric Epistemic	+ Native (designed into the meta-learning procedure)		Yes	Designed for this

Table S2. Synthetic dataset generation. Per-fault severity ranges, samples per severity, and simulation methods used in NVIDIA Omniverse Isaac Sim (v4.0.0).

Fault type	Severity levels		Samples / severity	Simulation method
Rain	Low, Medium, High		200	Particle systems + Mie scattering attenuation
Snow	Low, Medium, High		200	Particle systems + Mie scattering attenuation
Fog	Low, Medium, High		200	Volumetric fog with modulated density

Fault type	Severity levels	Samples severity	/ Simulation method
Signal Dropout	10%, 30%, 50%	200	Channel/azimuth nullification via Replicator API
Sensor Jitter	$\sigma = 0.02, 0.035, 0.05$ m	200	Gaussian noise on LiDAR Xform node
Ghosting	1, 3, 5 ghost objects	200	RTX multi-bounce ToF computation
Occlusion	15%, 30%, 50% area	200	Dynamic USD asset instantiation in FoV
Nominal (healthy)	-	-	Total 600

Table S3. Target-domain (mixed-origin) dataset generation. Base data sources and acquisition methods used to populate the eight fault classes.

Fault type	Base data	Fault source	Acquisition method
Rain	nuScenes (rainy)	Naturally occurring	-
Snow	Weather-NuScenes	Synthetically augmented	Calibrated snow augmentation with point-level labels
Fog	Weather-NuScenes	Synthetically augmented	Beer-Lambert attenuation with calibrated backscatter
Signal Dropout	nuScenes	Synthetically injected	Structured removal of scan lines and azimuth sectors
Sensor Jitter	nuScenes	Synthetically injected	Gaussian perturbation on 3D coordinates ($\sigma = 0.03$ m)
Ghosting	nuScenes	Synthetically injected	Duplicate point insertion at plausible reflection offsets
Occlusion	nuScenes	Naturally occurring	-
Nominal	nuScenes	No fault	-

Table S4. Comparative method overview. Categories, uncertainty quantification type, and ensemble size of all baselines used in Section 5.

Method	Category	UQ type	Ensemble size
MAML [13]	Meta-learning	None	1
Reptile [47]	Meta-learning	None	1
Matching Networks [48]	Meta-learning	None	1
ProtoNet+ [49]	Meta-learning	None	1
MC Dropout [24]	Bayesian approx.	Epistemic	$T = 20$ passes
SVGD [26]	Particle-based	Epistemic	$n = 10$ particles
S ² VGD [27]	Particle-based	Epistemic	$n = 10$ particles
EMP [30]	Ensemble	Epistemic	$M = 4$

Method	Category	UQ type	Ensemble size
ECMP [30]	Ensemble	Epistemic	$M = 4$
UDRML (Ours)	Meta + UQ	Both	$M = 4$

Table S5. CNN feature extraction backbone shared by all meta-learning baselines and the MTDA module. The network has approximately 113K parameters.

Layer	Operation	Output
Input	BEV projection	$3 \times 84 \times 84$
Block 1	(Conv 3×3, 64) + BN + ReLU + MaxPool 2×2	$64 \times 42 \times 42$
Block 2	(Conv 3×3, 64) + BN + ReLU + MaxPool 2×2	$64 \times 21 \times 21$
Block 3	(Conv 3×3, 64) + BN + ReLU + MaxPool 2×2	$64 \times 10 \times 10$
Block 4	(Conv 3×3, 64) + BN + ReLU + MaxPool 2×2	$64 \times 5 \times 5$
Flatten	-	1600
FC	Linear → logits	8

Full UDRML Algorithm Listing

Algorithm S1 UDRML — Uncertainty-Decomposed Reliability-Gated Meta-Learning

Require:

Task distribution $p(\mathcal{T})$ over simulated fault scenarios
 Target-domain dataset $\mathcal{D}_{\text{target}} = \{(x_j, y_j)\}_{j=1}^N$
 Inner-loop learning rate α , outer-loop learning rate β
 Number of ensemble members M , inner-loop steps G
 Number of candidate groups $L = 100/K$, per-class shot count K , number of classes C
 Uncertainty weights λ_1, λ_2 (subject to $\lambda_1 + \lambda_2 = 1$)
 Safety-gate decision thresholds $\tau_{\text{high}}, \tau_{\text{low}}$

Ensure:

Calibrated fault classification $\hat{y}$ and discrete system reliability action

Phase 1: MTDA $\times M$ members

- 1: Initialize M meta-parameter sets $\{\theta_1, \dots, \theta_M\}$
- 2: **for** each ensemble member $m = 1$ **to** M **do**
- 3: Meta-train θ_m via standard MTDA on tasks $\mathcal{T}_i \sim p(\mathcal{T})$:
 - Inner loop: $\theta'_{m,i} = \theta_m - \alpha \nabla_{\theta_m} \mathcal{L}_i^{\text{support}}(\theta_m)$ (G steps)
 - Task weights: $w_i = \mathcal{L}_i^{\text{query}}(\theta'_{m,i}) / \sum_j \mathcal{L}_j^{\text{query}}(\theta'_{m,j})$, fixed during differentiation [Eqs. (4b)–(4c), main text]
 - Outer loop: $\theta_m \leftarrow \theta_m - \beta \nabla_{\theta_m} \sum_i w_i \mathcal{L}_i^{\text{query}}(\theta'_{m,i})$
- 4: **end for**
- 5: **return** $\{\theta_1^*, \dots, \theta_M^*\}$
- Phase 2: Uncertainty-Aware Active Sampling (Target Domain)**
- 6: Partition each class's 100-sample candidate pool into $L = 100/K$ groups via LHS; combining one group per class forms candidate support sets $\{G_1, \dots, G_L\}$, each $|G_l| = 8K$
- 7: **for** each candidate group G_l , $l = 1$ **to** L **do**
- 8: **for** each sample $x_j \in G_l$ **do**
- 9: Compute pre-adaptation probabilities: $\pi_c(x_j) = \exp(z_c) / \sum_{k=1}^C \exp(z_k)$
- 10: Compute selection entropy: $H_{\text{select}}(x_j) = -\sum_{c=1}^C \pi_c(x_j) \log \pi_c(x_j)$
- 11: **end for**
- 12: Compute group uncertainty: $H(G_l) = \frac{1}{8K} \sum_{j=1}^{8K} H_{\text{select}}(x_j)$
- 13: **end for**
- 14: Select optimal support set: $\mathcal{S}^* = \arg \min_{G_l} H(G_l)$
- Phase 3: Few-Shot Adaptation (Sim-to-Real Transfer)**
- 15: **for** each ensemble member $m = 1$ **to** M **do**
- 16: Adapt θ_m^* on $\mathcal{S}^*$ via G gradient steps: $\theta'_m \leftarrow \theta_m^* - \alpha \nabla_{\theta} \mathcal{L}^{\text{adapt}}(\theta_m^*)$
- 17: **end for**
- 18: **return** adapted ensemble $\{f_{\theta'_1}, \dots, f_{\theta'_M}\}$
- Phase 4.1: Hyper-Ensemble Inference with Uncertainty Decomposition**
- 19: **for** each query sample x **do**
- 20: **for** $m = 1$ **to** M **do**
- 21: $p_c^{(m)} = \exp(z_c^{(m)}) / \sum_{k=1}^C \exp(z_k^{(m)})$
- 22: $H_m(x) = -\sum_{c=1}^C p_c^{(m)} \log(p_c^{(m)})$
- 23: **end for**
- 24: Ensemble mean: $\bar{p}_c = \frac{1}{M} \sum_{m=1}^M p_c^{(m)}$
- 25: Total uncertainty: $H_{\text{total}}(x) = -\sum_{c=1}^C \bar{p}_c \log(\bar{p}_c)$
- 26: Aleatoric uncertainty: $H_{\text{aleatoric}}(x) = \frac{1}{M} \sum_{m=1}^M H_m(x)$
- 27: Epistemic uncertainty: $I_{\text{epistemic}}(x) = H_{\text{total}}(x) - H_{\text{aleatoric}}(x)$
- 28: Predict: $\hat{y} = \arg \max_c \bar{p}_c$
- 29: **end for**
- Phase 4.2: Reliability-Aware Safety Gate**
- 30: Compute normalized uncertainties:
 - $\hat{H}_{\text{aleatoric}}(x) = H_{\text{aleatoric}}(x) / \log(C)$
 - $\hat{I}_{\text{epistemic}}(x) = I_{\text{epistemic}}(x) / \log(C)$
- 31: Compute reliability score [Eq. (17), main text]:

$$R(x) = 1 - [\lambda_1 \hat{H}_{\text{aleatoric}}(x) + \lambda_2 \hat{I}_{\text{epistemic}}(x)]$$
- 32: **if** $R(x) \geq \tau_{\text{high}}$ **then**
- 33: Accept $\hat{y}$ as reliable
- 34: **else if** $R(x) \geq \tau_{\text{low}}$ **then**
- 35: Flag $\hat{y}$ for secondary verification
- 36: **else**
- 37: Reject $\hat{y}$, trigger system fallback
- 38: **end if**

Algorithm S1. Complete pseudocode of the UDRML procedure. Phases 1–3 cover meta-training, uncertainty-aware active sampling, and few-shot adaptation; Phase 4 covers hyper-ensemble inference with uncertainty decomposition and reliability-aware decision gating.

Supplementary Table S6. Dataset composition

Table S6. Per-class dataset composition. Every class uses support sets of $K = 5, 10, 20$, or 50 samples drawn from its candidate pool, and a 100-sample query set. Scene counts per source: 12 independent Isaac Sim scenes per fault class (synthetic domain); 25 distinct nuScenes scenes for rain and 18 for occlusion; 30 Weather-NuScenes scenes for snow and fog; and 40 clean scenes used for controlled fault injection.

Fault class	Category	Origin	Synthetic samples	Target samples
Rain	Environmental	Naturally occurring	600	200
Snow	Environmental	Physically augmented	600	200
Fog	Environmental	Physically augmented	600	200
Signal dropout	Sensor-internal	Injected	600	200
Sensor jitter	Sensor-internal	Injected	600	200
Ghosting	Scene-interaction	Injected	600	200
Occlusion	Scene-interaction	Naturally occurring	600	200
Nominal	Baseline	Naturally occurring	600	200

Supplementary Table S7. Extended classification metrics

Table S7. Macro-F1, macro-precision, macro-recall, and balanced accuracy at 10-shot for all methods (five-run means). The target classes are balanced; the ranking matches the accuracy-based ordering of the main text.

Method	Macro-F1	Macro-precision	Macro-recall	Balanced accuracy
MC Dropout	0.552	0.560	0.568	0.568
SVGD	0.578	0.585	0.592	0.592
S ² VGD	0.584	0.591	0.598	0.598
EMP	0.570	0.577	0.584	0.584
ECMP	0.600	0.606	0.613	0.613
Matching Nets	0.619	0.625	0.632	0.632
ProtoNet+	0.646	0.651	0.658	0.658
Reptile	0.651	0.656	0.663	0.663
MAML	0.662	0.667	0.674	0.674
UDRML (ours)	0.786	0.789	0.792	0.792

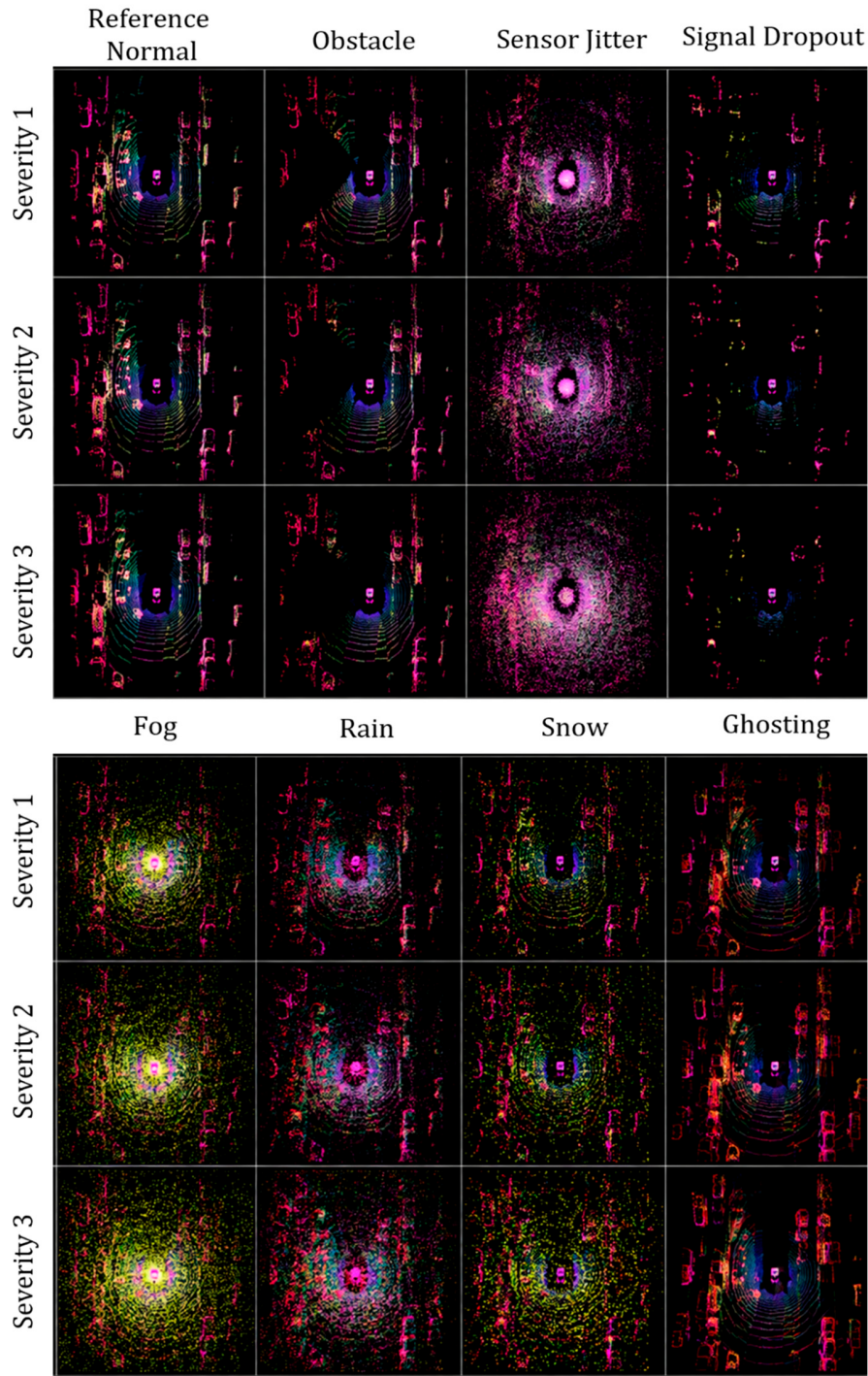


Figure S1. Sample BEV projections of a single street scene under the three severity levels defined in Table S2, for the seven fault classes and the clean reference. All tiles use the same projection and rendering as Figure 4 of the main text.