2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learnable Conformal Prediction for Safe and Efficient Robotics under Perception and Planning Uncertainties

Divake Kumar[1], Sina Tayebati[1], Francesco Migliarba[1], Ranganath Krishnan[2], Amit Ranjan Trivedi[1]

***Abstract*— Deep learning models in robotics often output point estimates with poorly calibrated confidences, offering no native mechanism to quantify predictive reliability under novel, noisy, or out-of-distribution inputs. Conformal prediction (CP) addresses this gap by providing distribution-free coverage guarantees, yet its reliance on fixed nonconformity scores ignores context and can yield intervals that are overly conservative or unsafe. We address this with Learnable Conformal Prediction (LCP), which replaces fixed scores with a lightweight neural function $s_\theta(x) = f_\theta(\phi(x))$ that leverages geometric, semantic, and model cues. Trained to balance coverage, efficiency, and calibration, LCP preserves CP's finite-sample guarantees while producing intervals that adapt to instance difficulty, achieving context-aware uncertainty without ensembles or repeated inference. On the MRPB benchmark, LCP raises navigation success to 91.5% versus 87.8% for Standard CP, while limiting path inflation to 4.5% compared to 12.2%. For object detection on COCO, BDD100K, and Cityscapes, it reduces mean interval width by 46–54% at 90% coverage, and on classification tasks (CIFAR-100, HAM10000, ImageNet) it shrinks prediction sets by 4.7–9.9%. The method is also computationally efficient, achieving real-time performance on resource-constrained edge hardware (Intel NUC with footprint 4.6×4.4 inch^2 and idle power < 30 W) while simultaneously providing uncertainty estimates along with the mean prediction.**

Project page: [Code, Video & Results]

I. Introduction and Prior Works

Learning from data is an inherently ill-conditioned problem that often admits multiple optimal solutions. Selecting a single solution while discarding others is theoretically unjustified and thus limits predictive robustness. Consequently, most learning models that output point predictions or poorly calibrated confidences [1] incur prediction errors that depend heavily on context such as occlusion, clutter, or distribution shift. Moreover, these models are typically optimized for average-case accuracy rather than worst-case reliability in deployment [2], and can fail catastrophically in rare yet safety-critical corner cases—a crucial limitation for mission-critical robotics.

Two primary sources of uncertainty exist: *epistemic uncertainty*, arising from limited data or model capacity, and *aleatoric uncertainty*, caused by inherent sensor noise or environmental ambiguity [3]–[5]. Existing UQ methods—Bayesian neural networks, deep ensembles, Monte Carlo dropout—either require multiple stochastic passes (impractical for real-time control) or yield poorly calibrated estimates, while post-hoc calibration lacks statistical guarantees [1].

Conformal prediction (CP) has recently attracted significant interest as a principled framework for uncertainty quantification [6]–[12]. Originating in statistical learning theory, CP constructs prediction sets calibrated on held-out data and provides distribution-free, finite-sample coverage guarantees under the assumption of exchangeability. Unlike many heuristic post-hoc calibration methods, CP offers explicit statistical guarantees, ensuring that true outcomes fall within the predicted sets at a user-specified confidence level. Moreover, CP is suited even for legacy prediction models that do not necessarily rely on learning from data. These properties make CP particularly appealing for robotics, where models must operate under distribution shift and safety requires formal reliability bounds on decision-making.

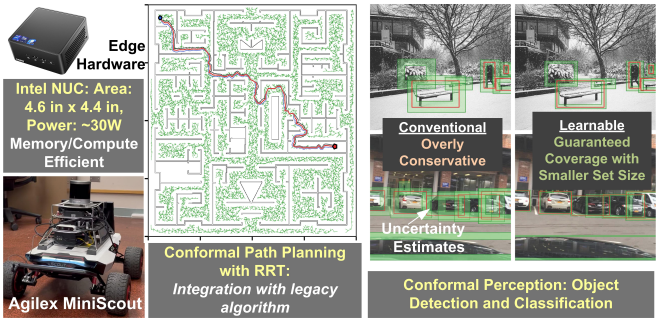

Fig. 1. **Real-time conformal prediction for safe and efficient robotics: Left:** The framework runs efficiently on an Intel NUC ($4.6\,\text{in} \times 4.4\,\text{in}$, $< 30\,\text{W}$) mounted on an Agilex MiniScout platform. **Center:** Conformal path planning integrates with legacy algorithms such as RRT. **Right:** Conformal perception improves object detection and classification, achieving guaranteed coverage with smaller set sizes.

Despite its generality, CP is most often implemented with fixed nonconformity functions that fail to capture the complex interplay of input data, application domain, and context in shaping uncertainty. For instance, in regression, standard CP with residual-based nonconformity produces intervals of constant width across all inputs, ignoring heteroscedasticity [13], [14]. As a result, CP provides valid coverage but does not account for how uncertainty emerges from the joint structure of observations and operating conditions. This limitation is especially critical in robotics, where risk depends not only on the raw input but also on situational context [15]: for example, a partially occluded object may be harmless clutter in a warehouse aisle yet represent a pedestrian entering a crosswalk in an urban scene. Treating both as equally uncertain either wastes efficiency in benign settings or under-protects in safety-critical ones.

Recent work has begun to address this gap by optimizing parameterized prediction regions. Cleaveland et al. [16] formulate temporal error allocation as a linear complementarity

[1]University of Illinois at Chicago
[2]Intel Labs

program for time-series forecasting, while Tumu et al. [17] optimize convex shape templates to reduce prediction region volume for trajectory prediction. However, these approaches rely on convex optimization over structured templates and have been demonstrated only in forecasting settings, leaving open the question of how to learn expressive, instance-adaptive nonconformity functions across diverse robotic tasks.

We address this with *Learnable Conformal Prediction (LCP)* (Fig. 1), which goes beyond template optimization by using neural networks to learn nonconformity scores from rich, task-specific feature representations. We introduce a feature-driven function $s_\theta(x) = f_\theta(\phi(x))$ that adapts to the structure of prediction errors. Features $\phi(x)$ encode geometric, semantic, and model-derived cues, while f_θ is a lightweight neural network trained to balance coverage, efficiency, and calibration. Calibration over these learned scores preserves the finite-sample coverage guarantees of CP [6], [14], while producing intervals that shrink in simple cases and expand in difficult ones.

We evaluate LCP for (i) robotic path planning under noisy and incomplete sensing on the MRPB benchmark, (ii) object detection with uncertainty calibration on COCO, BDD100K, and Cityscapes, and (iii) image classification on CIFAR-100, HAM10000, and ImageNet. Across these benchmarks, LCP consistently improves the safety–efficiency trade-off across planning, perception, and classification tasks. On the MRPB path-planning benchmark, LCP raises success rates to 91.5% while limiting path inflation to 4.5%, compared to 87.8% success and 12.2% inflation with standard CP. For object detection on COCO, BDD100K, and Cityscapes, LCP reduces mean interval width by 46–54% while sustaining ≈90% coverage. In classification (CIFAR-100, HAM10000, ImageNet), it cuts prediction set sizes by 4.7–9.9% relative to fixed baselines without losing validity. The proposed framework is also computationally efficient, achieving real-time performance on resource-constrained edge hardware (Intel NUC, area $4.6 \times 4.4\,\text{inch}^2$, idle power $< 30\,\text{W}$) while simultaneously extracting uncertainty estimates and prediction.

II. Learnable Conformal Prediction (LCP) by training Nonconformity Scoring Function

Conformal prediction (CP) is a distribution-free framework for constructing statistically valid prediction sets. Given a calibration dataset $\{(x_i, y_i)\}_{i=1}^n$ and a new test input x_{n+1}, the goal is to form a set $C(x_{n+1})$ that contains the true label y_{n+1} with probability at least $1-\alpha$. Formally, under exchangeability, CP guarantees $\mathbb{P}(Y \in C(X)) \geq 1-\alpha$ [18].

This guarantee is obtained by calibrating a nonconformity score $s(x, y)$, which measures how unusual a candidate label y is for input x. The prediction set contains labels with scores below the empirical quantile $\hat{q}$ from the calibration set:

$$C(x) = \{y \in \mathcal{Y} \mid s(x, y) \leq \hat{q}\},$$

$$\hat{q} = \text{Quantile}\Big(\{s(x_i, y_i)\}_{i=1}^n, \tfrac{\lceil (n+1)(1-\alpha) \rceil}{n}\Big).$$

While validity holds regardless of s, the usefulness of CP depends on the trade-off between prediction set size and informativeness. The expected size $\mathbb{E}[|C(X)|]$ is determined by both the base model's accuracy and the choice of $s(x, y)$. In practice, set size governs the safety–efficiency balance: larger sets provide more caution but reduce decisiveness, while smaller sets are efficient but risk undercoverage. Thus, the scoring function is a *key lever* for navigating this trade-off without compromising statistical rigor.

A. Limitations of Classical Nonconformity Scoring

Most CP methods use simple, hand-crafted rules that convert model outputs into nonconformity scores, such as:

- **Probability-based:** Scores such as $s(x, y) = 1 - p(y|x)$ assign low values to high-probability classes. Adaptive Prediction Sets (APS) [19] instead accumulate the probability mass of higher-ranked labels.
- **Margin-based:** Gap-based rules like $s(x, y) = \log(\max_j p(j|x)) - \log(p(y|x))$ measure separation between the top class and candidate y.
- **Logit-based:** Methods such as Sparsemax [20] operate in logit space by projecting onto a sparse simplex.

These scores are computationally efficient but rigid. For instance, softmax captures confidence yet ignores factors such as occlusion or distribution shift, often producing sets that are over-conservative in easy and unsafe in difficult ones.

B. Learning Nonconformity Scoring from Data

We introduce *learnable* nonconformity functions that adapt to the base model and dataset. The general procedure is domain-agnostic and consists of three steps: (1) design a task-specific feature extractor $\phi(x)$ that encodes relevant geometric, semantic, or model-derived cues; (2) train a lightweight neural network f_θ to map $\phi(x)$ to adaptive nonconformity scores via a composite loss that balances coverage, efficiency, and regularization; and (3) apply standard conformal calibration on a held-out split using the learned scores to restore finite-sample coverage guarantees. This procedure applies to any problem where conformal prediction is used—including trajectory prediction, pose estimation, or other structured outputs—provided that informative features can be extracted. Below, we instantiate this framework for three robotics use-cases:

Path planning. For each waypoint w, we construct a 20-dimensional feature vector $\phi(w)$ capturing geometry, uncertainty, and local context. Geometric features include minimum clearance $d_{\text{min}}(w, \hat{\mathcal{O}})$, average clearance at radii $\{1, 2\}$ m, passage width (largest inscribed circle), and obstacle density. Path context features include normalized progress $i/|p|$, distance to goal, curvature $\kappa(w) = \|\ddot{p}(s)\|$, velocity, and heading change $\Delta\theta$. An MLP (128-64-32-1, BatchNorm, 20% dropout) maps $\phi(w)$ to an adaptive margin:

$$\tau(w) = f_\theta(\phi(w)), \qquad \tau_{\text{final}}(w) = \max(r,\, \tau(w) + q^*), \quad (1)$$

$r = 0.17\,\text{m}$ is robot radius and q^* is the calibrated offset.

Training of the nonconformity function $s_\theta(x)$ balances coverage, efficiency, and task-specific goals through tailored objectives. Let $d = d_{\min}(w, \mathcal{O})$ denote the minimum distance from waypoint w to the nearest obstacle in the occupancy map $\mathcal{O}$. An asymmetric Huber loss penalizes unsafe margins (where the predicted margin τ is smaller than the true clearance d) more heavily:

$$\mathcal{L}_{\text{safety}} = \begin{cases} 0.5 \cdot \text{Huber}(\tau - d, 0), & \tau \geq d, \\ 2.0 \cdot \text{Huber}(\tau - d, 0), & \tau < d, \end{cases} \tag{2}$$

and the full path-planning loss adds efficiency, smoothness, and coverage terms:

$$\mathcal{L}_{\text{path}} = \mathcal{L}_{\text{safety}} + 0.3\|\tau - 0.3\| + 0.2\sum_i (\tau_{i+1} - \tau_i)^2 + \mathcal{L}_{\text{coverage}}, \tag{3}$$

where $\mathcal{L}_{\text{coverage}} = (\hat{C} - (1-\alpha))^2$ penalizes deviations of the empirical coverage $\hat{C}$ from the target $1-\alpha$, the efficiency term $0.3\|\tau - 0.3\|$ prevents excessive inflation, and the smoothness term discourages abrupt margin changes between consecutive waypoints.

Object detection. We also evaluated learnable nonconformity functions for object detection. Here, each bounding box b is represented by a 13-dimensional vector $\phi(b)$ including normalized coordinates, detector confidence, log area, aspect ratio, and distance to the image center. Coverage targets are scaled by size:

$$\text{target}_{\text{cov}} = \begin{cases} 0.90, & \sqrt{\text{area}} < 32, \\ 0.89, & 32 \leq \sqrt{\text{area}} < 96, \\ 0.85, & \sqrt{\text{area}} \geq 96, \end{cases} \tag{4}$$

so smaller objects receive higher coverage. For the nonconformity scoring $s_\theta(x)$, a network (256-128-64-4, ELU) outputs symmetric interval widths $\mathbf{w} = [w_{x_0}, w_{y_0}, w_{x_1}, w_{y_1}]$, later scaled by a calibrated factor τ.

To train $s_\theta(x)$, we minimize the Mean Prediction Interval Width (MPIW), normalized by box size:

$$\text{MPIW} = \tfrac{1}{4} \sum_{j \in \{x_0, y_0, x_1, y_1\}} 2w_j\tau, \mathcal{L}_{\text{MPIW}} = \frac{\text{MPIW}}{(w_{\text{box}} + h_{\text{box}})/2}. \tag{5}$$

A coverage penalty enforces the target range $[0.88, 0.92]$:

$$\mathcal{L}_{\text{penalty}} = \begin{cases} 5(\hat{C} - 0.89)^2, & \hat{C} > 0.905, \\ 10(0.89 - \hat{C})^2, & \hat{C} < 0.88, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Classification. Extending the evaluations of $s_\theta(x)$ for classification, we extract the following features for each class c: probability $p(c|x)$, normalized rank $\text{rank}(p_c)/K$, margin to the top class, top-k indicators for $k \in \{1, 3, 5\}$, entropy contribution $-p(c|x)\log p(c|x)$, and global maximum probability. The multi-layer perceptron width for $s_\theta(x)$ scales with the number of classes to balance capacity and regularization:

$$(H_1, H_2) = \begin{cases} (32, 16), & K \leq 10, \\ (64, 32), & 10 < K \leq 100, \\ (128, 64), & 100 < K \leq 1000, \\ (256, 128), & K > 1000. \end{cases} \tag{7}$$

A three-phase schedule guides training:

Phase 1 (epochs 1–10) uses margin loss:

$$\mathcal{L}_{\text{margin}} = \text{ReLU}(s_{\text{true}} - \text{mean}(s_{\text{false}}) + \delta), \quad \delta = 0.8. \tag{8}$$

Phase 2 (epochs 11–20) adds coverage $\mathcal{L}_{\text{cov}} = (\hat{C} - (1-\alpha))^2$,

Phase 3 (epochs 21+) introduces set size minimization:

$$\mathcal{L}_{\text{size}} = \frac{1}{n}\sum_{i=1}^{n} \frac{|C(x_i)|}{K} + \lambda_{\text{empty}} \mathbf{1}[|C(x_i)| = 0]. \tag{9}$$

C. Calibration and Adaptive Thresholds

A key practical consideration is that learning s_θ requires splitting the available data into three disjoint sets: training (for f_θ), calibration (for $\hat{q}$), and evaluation. This additional split—absent in standard CP, which needs only calibration and test sets—reduces the data available for each purpose. In our experiments, we use a 60/20/20 split (train/calibrate/test). To assess sensitivity, we varied the calibration fraction from 10% to 30% on COCO: coverage remained within 89–91% across all splits, with MPIW varying by <3%, indicating that 20% calibration data suffices for stable performance. In data-scarce regimes, cross-validation-based calibration [18] could further mitigate this cost.

Post-training calibration restores coverage guarantees while maintaining efficiency. For path planning, we compute an additive offset:

$$q^* = \text{Quantile}_{1-\alpha}\left(\{\tau_{\text{pred}}(w_i) - d_{\text{true}}(w_i)\}_{i=1}^m\right). \tag{10}$$

For object detection, calibration uses a multiplicative factor based on the infinity norm of prediction errors:

$$\tau = \text{Quantile}_{1-\alpha}\left(\left\{\frac{\|\mathbf{b}_i^* - \hat{\mathbf{b}}_i\|_\infty}{f_\theta(\phi(\hat{\mathbf{b}}_i))}\right\}_{i=1}^m\right). \tag{11}$$

During training, thresholds are updated with an exponential moving average:

$$\tau_t = \beta\tau_{t-1} + (1-\beta)\hat{q}_t, \quad \beta = 0.95, \tag{12}$$

which balances stability (β close to 1) with adaptability to distribution shifts. For classification, we apply smoothed quantile estimation with asymmetric windows:

$$\hat{q} = \frac{\sum_{i=k-3}^{k+1} w_i s_i}{\sum_{i=k-3}^{k+1} w_i}, \quad w_i = 1.5 - 0.1|i-k|, \tag{13}$$

where k indexes the $(1-\alpha)$ quantile. The asymmetric window, with greater weight below the quantile, yields conservative yet stable estimates.

D. Optimization and Implementation Details

We optimize training of nonconformity scoring functions ($s_\theta(x)$) using AdamW with weight decay $\lambda_{\text{wd}} = 10^{-5}$ and a cosine annealing schedule. For classification, we apply CosineAnnealingWarmRestarts with $T_0 = 5$ epochs and $\eta_{\min} = 10^{-5}$. Gradient clipping at norm 0.5 prevents early-stage instability when scores are poorly calibrated. Batch sizes are task-specific: 256 for classification (stable statistics), 512 for detection (memory–efficiency trade-off), and 1024 for path planning (trajectory-level parallelism).

Loss weights adapt dynamically to coverage performance. When empirical coverage $\hat{C} < 1 - \alpha - \epsilon$ with $\epsilon = 0.02$, coverage weight increases ($w_c = 2.0$, $w_s = 1.0$). Otherwise, efficiency is prioritized ($w_c = 1.0$, $w_s = 1.5$). This guarantees coverage before minimizing set sizes. For detection, size-stratified metrics are also monitored, and weights are adjusted per category to balance performance across scales.

Running statistics are maintained per feature dimension, and inputs are standardized as $\tilde{\phi}_i = (\phi_i - \mu_i)/\sigma_i$. Path planning uses per-environment normalization to account for map variation, object detection uses global dataset statistics, and classification applies per-dataset normalization to handle differing class distributions.

E. Computational Complexity and Efficiency

Despite a more elaborate scoring function design, the overhead of learnable CP remains minimal relative to the base model. Feature extraction costs $O(d)$ per instance, where d (8–20 depending on task) is the feature dimension. The MLP forward pass requires $O(H_1 d + H_1 H_2 + H_2)$ operations, with H_1, H_2 denoting hidden layer sizes. Even for the largest network (256–128 neurons), this adds under 0.5 ms on GPU and under 5 ms on CPU per prediction.

Memory use is also modest. The largest model (classification on ImageNet) stores only 100 KB of parameters, while path planning models require about 42 KB. Feature caching during training costs $O(nd)$, where n is the dataset size, but inference needs only $O(1)$ memory per instance. Calibration involves sorting n scores ($O(n \log n)$) once during setup, with no additional overhead at deployment.

III. Robust Path Planning with Noisy and Incomplete Perception

Coverage target convention. Throughout all experiments, we set $\alpha = 0.1$, so the target coverage is $1-\alpha = 90\%$. A method achieving $\approx$90% coverage is well-calibrated; coverage significantly above 90% indicates over-conservatism (unnecessarily large prediction sets or margins), while coverage below 90% signals under-coverage. Thus, the ideal method matches the target while minimizing set size or interval width.

A. Task, Environments, and Metrics

We evaluate Learnable Conformal Prediction (LCP) on the MRPB benchmark [21] across five environments of varying complexity (some representative ones are shown in Fig. 2). To emulate practical constraints, we augment the benchmark with three empirically grounded sensing degradations [23]: (i) LiDAR transparency on glass with an 18.8% miss rate [24], (ii) partial occlusion hiding 57.5% of obstacles [25], and (iii) localization drift with 0.5 m standard deviation [26]. We also evaluate a combined setting including all three. Each method is tested over 1,250 Monte Carlo trials using RRT* [22].

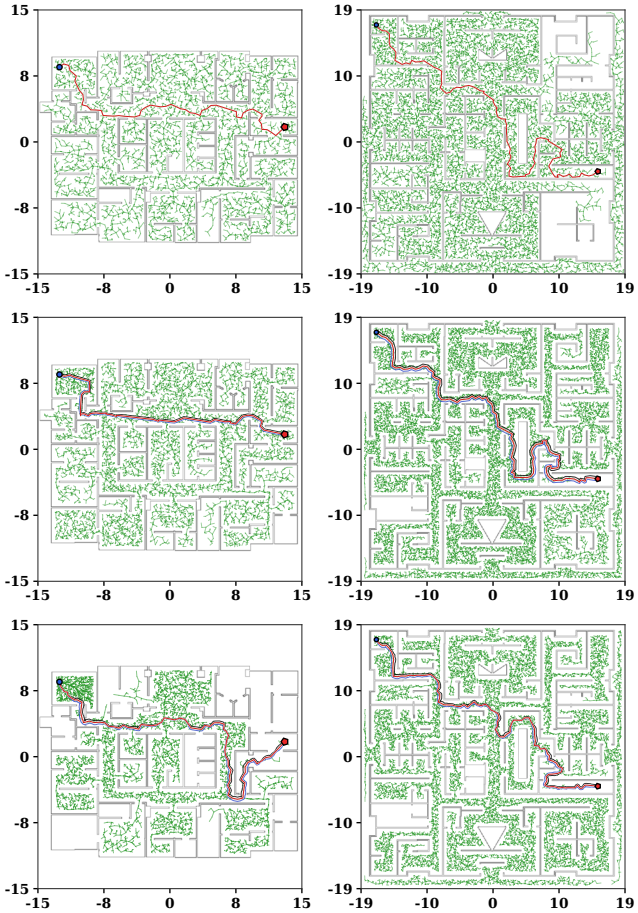

Fig. 2. Path planning benchmarks on MRPB [21]. Rows correspond to methods (Naive, Standard CP, Learnable CP) and columns to environments (Office02, Shopping Mall). In Standard CP, red shows the naive path, while black and blue denote fixed safety margins. In Learnable CP, the margin adapts, wider near obstacles and tighter in open areas. All methods are evaluated over 1250 Monte Carlo trials using RRT* [22].

We use the metrics in Table I for characterization. Safety is measured by success rate. Path quality is assessed by path length L and the number of waypoints. Safety margins are quantified by initial clearance d_0 (minimum obstacle distance along the planned path) and average clearance d_{avg}. Risk exposure is measured by danger-zone occupancy p_0, the fraction of the path within radius $r_0 = 0.20$ m. Execution time T captures efficiency. Conformal methods use a held-out calibration set disjoint from training and evaluation. Standard CP applies a single global quantile $\hat{q}$ across all environments and noise levels. LCP learns a parametric score s_θ and then applies the same calibration with $\hat{q}$ on the disjoint set.

B. Results and Insights

Key metrics to focus on: **Success rate** (primary safety measure) and **path length inflation** relative to naive planning

Table I: Path Planning Results on MRPB across Five Environments Comparing Naive, Standard CP, and LCP.

Environment	Method	Success Rate ↑	Path Length (m) ↓	Waypoints ↓	d_0 (m) ↑	d_{avg} (m) ↑	p_0 (%) ↓	T (s) ↓
office01add	Naive	81.62%	**21.92 ± 2.15**	**35 ± 3**	0.212 ± 0.048	0.637 ± 0.085	4.92 ± 1.23	**18.54 ± 2.35**
	Standard CP	89.24%	24.79 ± 2.38	39 ± 4	**0.342 ± 0.051**	**0.817 ± 0.092**	**1.48 ± 0.52**	20.98 ± 2.65
	Learnable CP	**92.78%**	22.82 ± 2.21	36 ± 4	0.285 ± 0.045	0.711 ± 0.088	2.56 ± 0.78	19.31 ± 2.42
office02	Naive	77.18%	**48.35 ± 4.62**	**85 ± 8**	0.206 ± 0.052	0.649 ± 0.098	5.28 ± 1.65	**42.65 ± 4.85**
	Standard CP	87.56%	51.28 ± 4.85	89 ± 9	**0.358 ± 0.058**	**0.832 ± 0.103**	**1.85 ± 0.68**	45.23 ± 5.12
	Learnable CP	**91.23%**	49.15 ± 4.71	86 ± 8	0.294 ± 0.052	0.744 ± 0.095	2.93 ± 0.95	43.35 ± 4.95
shopping_mall	Naive	75.64%	**62.45 ± 5.38**	**55 ± 5**	0.185 ± 0.045	0.753 ± 0.112	6.15 ± 1.82	**45.32 ± 5.25**
	Standard CP	86.81%	68.21 ± 5.95	65 ± 6	**0.365 ± 0.062**	**0.900 ± 0.118**	**1.23 ± 0.48**	49.50 ± 5.68
	Learnable CP	**90.37%**	64.87 ± 5.52	60 ± 6	0.308 ± 0.055	0.815 ± 0.108	2.45 ± 0.88	47.08 ± 5.42
room02	Naive	83.25%	**20.14 ± 1.95**	**30 ± 3**	0.233 ± 0.055	0.741 ± 0.089	3.85 ± 1.15	**16.26 ± 2.15**
	Standard CP	90.43%	22.15 ± 2.12	33 ± 3	**0.378 ± 0.055**	**0.921 ± 0.098**	**0.78 ± 0.32**	17.88 ± 2.35
	Learnable CP	**93.61%**	20.93 ± 2.03	31 ± 3	0.318 ± 0.048	0.816 ± 0.092	1.65 ± 0.58	16.89 ± 2.25
narrow_graph	Naive	71.97%	**31.39 ± 3.25**	**66 ± 6**	0.215 ± 0.051	0.441 ± 0.078	7.61 ± 2.12	**34.65 ± 3.85**
	Standard CP	85.18%	36.13 ± 3.48	75 ± 7	**0.335 ± 0.061**	**0.621 ± 0.085**	**3.03 ± 1.02**	39.88 ± 4.25
	Learnable CP	**89.59%**	33.68 ± 3.31	69 ± 7	0.274 ± 0.051	0.525 ± 0.081	4.64 ± 1.38	37.18 ± 4.05

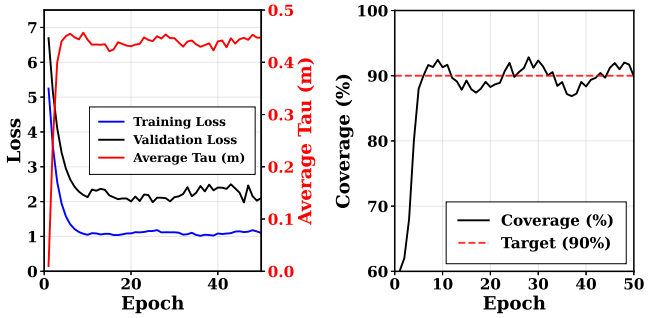

Fig. 3. Training of the scoring function over 50 epochs: **(a, left)** loss curves show fast convergence with a validation gap indicating conservative predictions, threshold τ converges to 0.44 m via automatic calibration, and **(b, right)** coverage stabilizes at 90±3%, confirming reliable safety ranking.

(efficiency cost of safety). Secondary metrics d_0, d_{avg}, and p_0 characterize how safety margins are allocated.

Table I compares naive planning, Standard CP, and LCP across environments. LCP achieves the highest average success rate (91.5%) while preserving near-optimal path efficiency. The safety-performance trade-off is inherently nonlinear: Standard CP ($d_0 \approx 0.35$ m) improves success by 10% over naive planning but inflates path length by 15.3%, whereas LCP's adaptive margin ($d_0 \approx 0.29$ m) captures 90% of the safety gain at only 40% of the efficiency cost.

Performance gaps widen with complexity. In the shopping_mall environment, naive planning fails in 24.4% of trials, Standard CP succeeds only by inflating path length by 9.2%, while LCP reaches 90.4% success with just 3.9% additional length. The p_0 metric reveals that LCP *redistributes rather than uniformly minimizes risk*, permitting closer proximity in wide passages while preserving clearance at critical points. LCP also reduces execution time by 7–8% versus Standard CP across environments.

C. Learning Dynamics and Safety Calibration

Fig. 3 tracks LCP training over 50 epochs. Training loss drops from 5.3 to 1.1 within 10 epochs, while the conformal threshold τ converges to 0.44 m (±0.02 m), about 38% above Standard CP's fixed 0.32 m. Coverage stabilizes

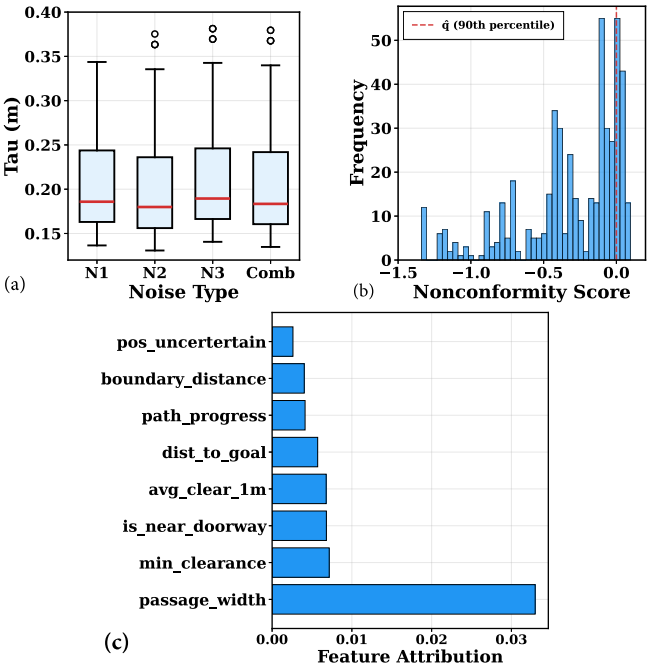

Fig. 4. **Ablation of Learnable CP: (a)** Thresholds adapt by noise, ranging 0.18–0.38 m. Here, N1 = transparency noise, N2 = occlusion noise, N3 = localization noise, and Comb = combined noise. **(b)** Bimodal score distributions reveal the gap between learned rankings and safety. **(c)** Feature importance shows geometry dominance, with passage width as key driver.

at 90±3% within 5 epochs, showing that LCP learns relative safety rankings rather than absolute thresholds, with coverage maintained through post-hoc calibration.

Ablation studies in Fig. 4 illustrate how LCP achieves context-aware safety. Threshold distributions vary by noise: occlusion yields narrow margins (0.16–0.24 m), localization spans 0.14–0.34 m, and rare outliers at 0.36–0.38 m act as crisis responses. Score histograms are bimodal, with peaks at -0.4 m (efficiency-oriented) and -0.05 m (safety-critical), confirming that the model learns ordinal safety rankings. Feature attribution highlights passage width (0.033) as the dominant causal driver, while negative attributions for density and occlusion ratios show that margins shrink in cluttered settings

Table II: Uncertainty Quantification for Object Localization across Models and Datasets (target: 0.90).

Base Model	Method	COCO		BDD100K		Cityscapes	
		Cov.	MPIW	Cov.	MPIW	Cov.	MPIW
ResNeXt-101-FPN*	Standard CP	(0.900 ± 0.013)	(90.6 ± 9.3)	(0.919 ± 0.009)	(59.8 ± 3.3)	(0.912 ± 0.029)	(100.0 ± 20.3)
	Ensemble	(0.927 ± 0.005)	(109.7 ± 3.7)	(0.900 ± 0.037)	(80.4 ± 7.1)	(0.906 ± 0.037)	(127.6 ± 16.1)
	CQR	(0.891 ± 0.010)	(87.7 ± 13.6)	(0.910 ± 0.006)	(71.0 ± 4.4)	(0.908 ± 0.063)	(110.0 ± 25.9)
	Learnable (Ours)	$\mathbf{(0.902 \pm 0.020)}$	$\mathbf{(41.9 \pm 1.8)}$	$\mathbf{(0.896 \pm 0.019)}$	$\mathbf{(28.8 \pm 1.2)}$	$\mathbf{(0.887 \pm 0.021)}$	$\mathbf{(53.8 \pm 2.1)}$
Cascade R-CNN	Standard CP	(0.927 ± 0.008)	(109.0 ± 9.6)	(0.912 ± 0.025)	(54.3 ± 2.3)	(0.924 ± 0.065)	(96.6 ± 12.3)
	Learnable (Ours)	$\mathbf{(0.898 \pm 0.018)}$	$\mathbf{(37.3 \pm 1.5)}$	$\mathbf{(0.892 \pm 0.017)}$	$\mathbf{(27.5 \pm 1.1)}$	$\mathbf{(0.903 \pm 0.019)}$	$\mathbf{(51.2 \pm 2.0)}$
ResNet-50-FPN	Standard CP	(0.900 ± 0.011)	(105.9 ± 11.3)	(0.901 ± 0.013)	(58.7 ± 3.6)	(0.900 ± 0.022)	(95.5 ± 18.7)
	Learnable (Ours)	$\mathbf{(0.891 \pm 0.020)}$	$\mathbf{(46.4 \pm 1.6)}$	$\mathbf{(0.885 \pm 0.018)}$	$\mathbf{(29.1 \pm 1.3)}$	$\mathbf{(0.896 \pm 0.022)}$	$\mathbf{(57.9 \pm 2.2)}$
ResNet-50-C4	Standard CP	(0.939 ± 0.016)	(120.7 ± 10.6)	(0.910 ± 0.048)	(62.8 ± 4.6)	(0.910 ± 0.038)	(128.3 ± 35.0)
	Learnable (Ours)	$\mathbf{(0.905 \pm 0.021)}$	$\mathbf{(51.2 \pm 1.9)}$	$\mathbf{(0.900 \pm 0.019)}$	$\mathbf{(32.2 \pm 1.4)}$	$\mathbf{(0.910 \pm 0.023)}$	$\mathbf{(62.4 \pm 2.4)}$
ResNeXt-INT8	Standard CP	(0.885 ± 0.015)	(95.2 ± 10.8)	(0.895 ± 0.020)	(65.5 ± 5.2)	(0.900 ± 0.025)	(115.8 ± 22.1)
	Learnable (Ours)	$\mathbf{(0.894 \pm 0.019)}$	$\mathbf{(55.8 \pm 2.1)}$	$\mathbf{(0.888 \pm 0.018)}$	$\mathbf{(35.7 \pm 1.5)}$	$\mathbf{(0.899 \pm 0.020)}$	$\mathbf{(66.7 \pm 2.6)}$

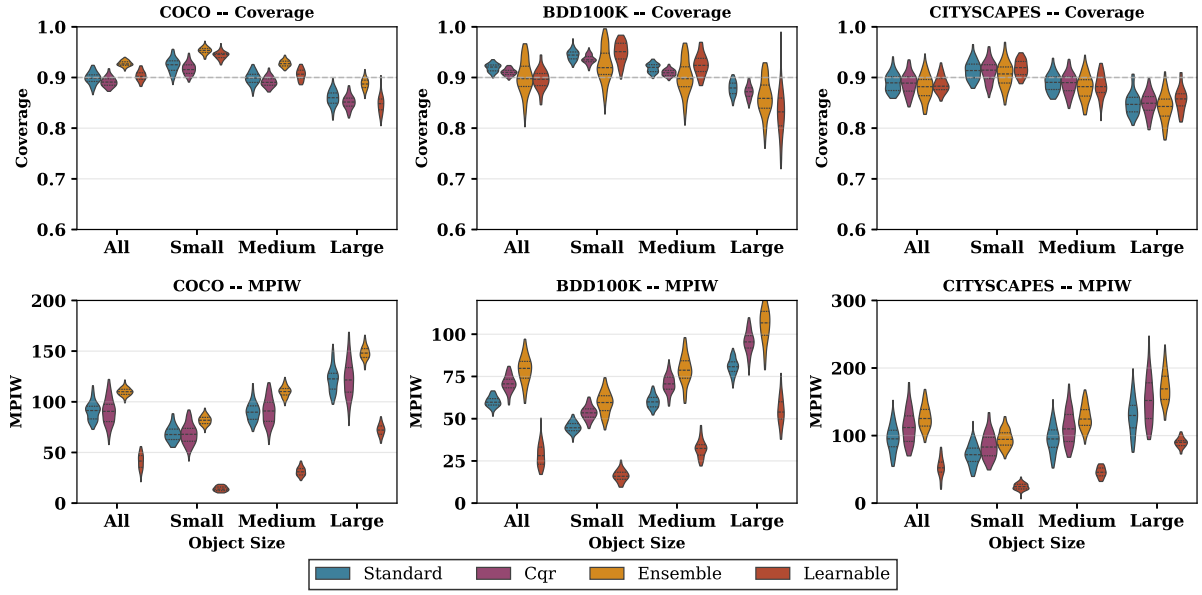

Fig. 5. Violin plots of coverage and MPIW on COCO, BDD100K, and Cityscapes. Our Learnable CP (red) achieves the tightest intervals with coverage near 90%, reducing small-object MPIW to 15 pixels (vs. 70) and large-object MPIW to 70 pixels (vs. 120–150), demonstrating superior coverage-efficiency.

to preserve navigability. By contrast, Standard CP adapts only coarsely (τ shifts between 0.32–0.37 m) and lacks the continuous, context-aware adaptation that enables LCP to sustain 91.5% success with near-optimal efficiency.

IV. Perception Robustness under Uncertainty

A. Task, Datasets, and Setup

We evaluate uncertainty-aware object detection on COCO [27], BDD100K [28], and Cityscapes [29], comparing LCP with Standard CP, deep ensembles, and conformalized quantile regression (CQR) [30]. In addition to these datasets, our hardware evaluation on Intel NUC (Core Ultra 7 165H, 64 GB DDR5, Intel Arc GPU, AI Boost NPU) shows that LCP adds $<1\%$ memory overhead and 15.9% inference overhead ($\sim$3.5 ms per frame), while maintaining 39 FPS for YOLOv8n object detection. Compared to Standard CP, LCP halves interval width (102 px $\rightarrow$ 51 px) at the same 90% coverage, with modest power increase (35 W $\rightarrow$ 38 W).

B. Results and Insights

Key metrics to focus on: **MPIW** (mean prediction interval width, lower is better) and **Coverage** (should be $\approx$90%, not higher). In Table II, the primary comparison is MPIW reduction at matched coverage. In Table III, **Set Size** (lower is better) is the primary efficiency metric.

As Table II and Fig. 5 show, LCP achieves scale-aware efficiency: average MPIW contracts to 41.9 px on COCO, 28.8 px on BDD100K, and 53.8 px on Cityscapes—46–54% smaller than Standard CP while maintaining $\approx$90% coverage. This improvement is adaptive redistribution, not uniform compression: small objects reach 94.2% coverage with only 14.7 px slack, whereas large objects settle at 84.7% coverage with 72.2 px slack, encoding task difficulty directly into interval allocation.

Calibration is stable across backbones (Fig. 6): coverage stabilizes in the 88–92% band, τ converges to dataset-specific optima (0.45 for COCO, 0.20 for BDD100K), and MPIW shrinks by 30–50% within 20–30 epochs. This pat-

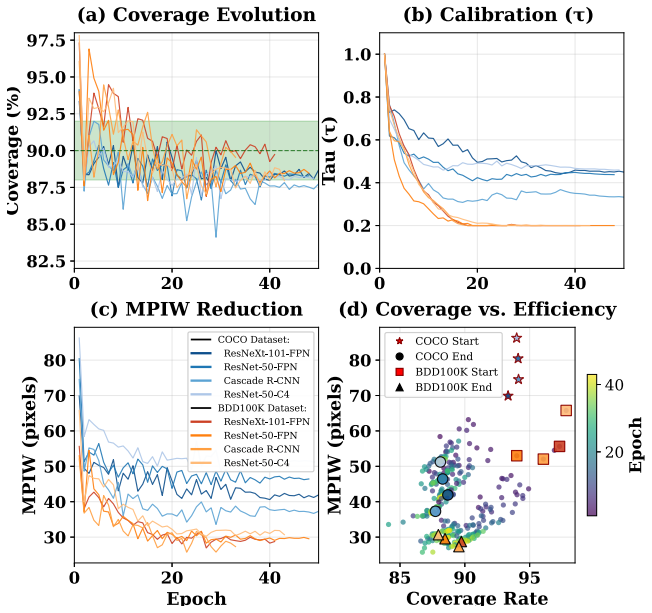

Fig. 6. Multi-architecture τ calibration analysis. **(a)** Coverage stabilizes in the 88–92% zone for COCO (blue) and BDD100K (orange). **(b)** τ adapts from 1.0 to dataset-specific optima (≈0.45, 0.2). **(c)** MPIW shrinks 30–50% (80→46 px). **(d)** Coverage-efficiency trajectories converge to Pareto-optimal fronts within 20–30 epochs, consistent across detector backbones.

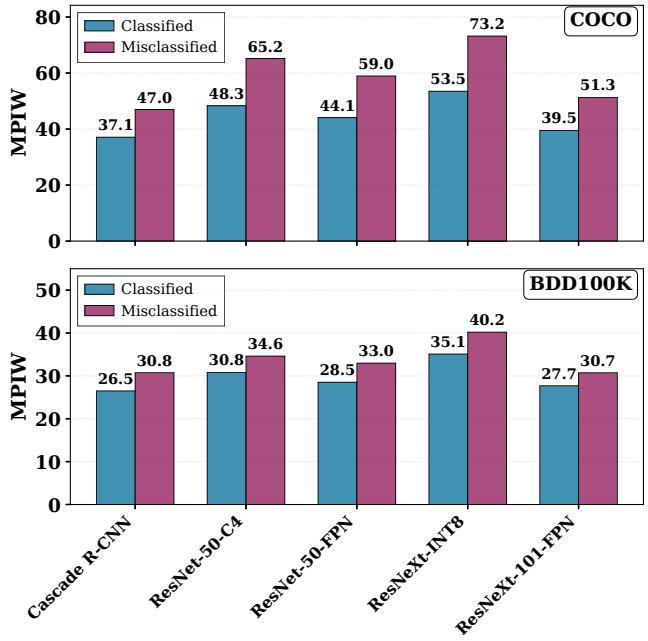

Fig. 7. MPIW comparison on COCO and BDD100K showing misclassified predictions receive wider intervals, confirming adaptive uncertainty quantification across architectures.

tern holds across ResNeXt-101, Cascade R-CNN, ResNet-50-FPN/C4, and quantized ResNeXt-INT8, confirming that learned nonconformity functions transfer across detectors without repeated calibration sweeps.

LCP also widens intervals when predictions are unreliable: as Fig. 7 shows, misclassified detections receive 33% wider intervals on COCO and 14% on BDD100K, enabling downstream planners to act selectively on uncertain detections. Comparatively, standard CP applies rigid margins, ensembles

Table III: Learnable vs. static scoring on classification.

	Method	Coverage ↑	Set Size ↓	AUROC ↑	ECE ↓
CIFAR-100	1-p (Baseline)	89.64	1.73	0.97	0.04
	APS	**90.38**	3.71	0.95	0.11
	LogMargin	90.14	2.47	0.96	0.06
	Sparsemax	88.44	2.21	0.92	0.03
	Ours	88.01	**1.56**	**0.98**	**0.02**
HAM10000	1-p (Baseline)	90.24	1.31	0.96	0.04
	APS	**90.34**	1.25	0.94	0.05
	LogMargin	89.94	1.51	0.94	0.03
	Sparsemax	90.04	1.12	0.88	0.02
	Ours	89.31	**1.06**	**0.97**	**0.01**
ImageNet	1-p (Baseline)	90.15	1.53	0.97	0.02
	APS	90.16	2.05	0.95	0.04
	LogMargin	90.06	1.66	0.96	0.02
	Sparsemax	**90.55**	1.80	0.93	0.01
	Ours	89.57	**1.46**	**0.98**	**0.01**
Places365	1-p (Baseline)	90.12	17.99	0.95	0.02
	APS	90.10	21.06	0.96	0.04
	LogMargin	90.16	16.64	0.97	0.03
	Sparsemax	**90.59**	22.55	0.94	0.06
	Ours	89.85	**14.97**	**0.97**	**0.01**
PlantNet	1-p (Baseline)	89.66	10.08	0.71	0.02
	APS	89.72	13.27	0.70	0.03
	LogMargin	89.42	12.36	0.78	**0.01**
	Sparsemax	**90.93**	12.42	0.79	0.14
	Ours	89.08	**6.73**	**0.94**	0.02

inflate excessively (often doubling box area), and CQR produces geometrically implausible shapes. LCP yields tight intervals that scale with object size: on COCO, small-object intervals shrink from ∼70 px to 15 px, while large-object intervals contract from 120–150 px to ∼70 px.

The classification results in Table III confirm that learned scoring generalizes beyond detection. Across CIFAR-100, HAM10000, ImageNet, Places365, and PlantNet, LCP consistently yields the smallest prediction sets, shrinking average size by 4.7–9.9% while maintaining coverage. In PlantNet, where baselines degenerate to AUROC ≈0.71, LCP recovers discriminative power (AUROC ≈0.94) with far leaner sets.

C. Sensitivity to Hyperparameters

We ablate key design choices on the COCO detection task (ResNeXt-101-FPN) and CIFAR-100 classification. **Loss weights:** Varying the coverage weight w_c from 1.0 to 3.0 (with w_s fixed at 1.5) changes MPIW by <5% while coverage remains within 88–92%, indicating robustness to this balance. Setting $w_c < 0.5$ causes undercoverage (<85%), confirming the dynamic weighting scheme is important but not sensitive within a reasonable range. **Network depth:** Reducing the classification MLP from (256,128) to (64,32) on CIFAR-100 increases set size by only 3.7% (from 1.56 to 1.62), while a single-layer model degrades set size by 12%, suggesting that moderate depth is needed to capture class interactions but exact architecture is not critical. **Feature ablation:** Removing geometric features from the detection scorer increases MPIW by 18%, while removing uncertainty features increases it by 9%, confirming that both feature groups contribute but geometry is more impactful. The method is most sensitive to the coverage penalty threshold (Eq. 6): widening the acceptable band from $[0.88, 0.92]$ to $[0.85, 0.95]$ reduces MPIW by 8% but drops coverage to 87%.

V. Conclusion

We introduced Learnable Conformal Prediction (LCP), a framework that learns context-aware nonconformity functions while preserving CP's coverage guarantees. LCP raised navigation success to 91.5% (vs. 87.8% for Standard CP) with only 4.5% path inflation, reduced object detection interval widths by 46–54% at 90% coverage, and cut classification set sizes by 4.7–9.9%. The method is lightweight ($\sim$4.8% total pipeline overhead, 42 KB memory) [31], [32], sustaining 39 FPS on Intel NUC with only 15.9% inference-only overhead ($\sim$3.5 ms) and halving interval width (102 px $\rightarrow$ 51 px) at modest power rise (35 W $\rightarrow$ 38 W).

Acknowledgment

This work was supported by the SRC/JUMP 2.0 program through the Center on Cognitive Multispectral Sensing (CogniSense), as well as by the National Science Foundation (NSF) under Grant No. 2329096.

References

[1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[2] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *arXiv preprint arXiv:1606.06565*, 2016.

[3] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?" *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[4] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, 2017, pp. 5574–5584.

[5] D. Kumar, P. Poggi, S. Tayebati, D. Naik, N. Ahuja, and A. R. Trivedi, "Calibrated decomposition of aleatoric and epistemic uncertainty in deep features for inference-time adaptation," *arXiv preprint arXiv:2511.12389*, 2025.

[6] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer, 2005.

[7] A. N. Angelopoulos and S. Bates, "Conformal prediction: A gentle introduction," *Foundations and Trends in Machine Learning*, vol. 16, no. 4, pp. 494–591, 2023.

[8] A. C. Stutts, D. Erricolo, T. Tulabandhula, and A. R. Trivedi, "Echoes of Socratic doubt: Embracing uncertainty in calibrated evidential reinforcement learning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 9651–9657.

[9] A. C. Stutts, D. Erricolo, S. Ravi, T. Tulabandhula, and A. R. Trivedi, "Mutual information-calibrated conformal feature fusion for uncertainty-aware multimodal 3D object detection at the edge," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2029–2035.

[10] A. C. Stutts, D. Erricolo, T. Tulabandhula, and A. R. Trivedi, "Lightweight, uncertainty-aware conformalized visual odometry," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7742–7749.

[11] D. Kumar, S. Tayebati, N. Darabi, V. P.-H. Hu, and A. R. Trivedi, "Uncertainty-aware LiDAR-camera autonomy via conformal prediction and principled abstention," in *2025 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE, 2025, pp. 1–6.

[12] S. Tayebati, D. Kumar, N. Darabi, D. Jayasuriya, T. Tulabandhula, R. Krishnan, and A. R. Trivedi, "CAP: Conformalized abstention policies for context-adaptive risk management for LLMs and VLMs," in *Proceedings of the 17th Asian Conference on Machine Learning (ACML)*, ser. Proceedings of Machine Learning Research. PMLR, 2025.

[13] H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman, "Inductive confidence machines for regression," in *European Conference on Machine Learning*. Springer, 2002, pp. 345–356.

[14] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1094–1111, 2018.

[15] D. Kumar, S. Tayebati, D. Naik, P. Poggi, A. S. Rios, N. Ahuja, and A. R. Trivedi, "TRIAGE: Type-routed interventions via aleatoric-epistemic gated estimation in robotic manipulation and adaptive perception," *arXiv preprint arXiv:2603.08128*, 2026.

[16] M. Cleaveland, I. Lee, G. J. Pappas, and L. Lindemann, "Conformal prediction regions for time series using linear complementarity programming," *arXiv preprint arXiv:2304.01075*, 2023.

[17] R. Tumu, M. Cleaveland, R. Mangharam, G. J. Pappas, and L. Lindemann, "Multi-modal conformal prediction regions with simple structures by optimizing convex shape templates," in *Proceedings of The 6th Annual Learning for Dynamics and Control Conference*, ser. Proceedings of Machine Learning Research, vol. 242. PMLR, 2024, pp. 1343–1356.

[18] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer, 2005.

[19] Y. Romano, E. Patterson, and E. Candès, "Conformalized quantile regression," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[20] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1614–1623.

[21] J. Wen, X. Zhang, Q. Bi, Z. Pan, Y. Feng, J. Yuan, and Y. Fang, "MRPB 1.0: A unified benchmark for the evaluation of mobile robot local planning approaches," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8238–8244.

[22] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[23] N. Darabi, D. Kumar, S. Tayebati, and A. R. Trivedi, "INTACT: Inducing noise tolerance through adversarial curriculum training for LiDAR-based safety-critical perception and autonomy," *arXiv preprint arXiv:2502.01896*, 2025.

[24] C. Glennie and D. D. Lichti, "Static calibration and analysis of the Velodyne HDL-64E S2 for high accuracy mobile scanning," *Remote Sensing*, vol. 2, no. 6, pp. 1610–1624, 2010.

[25] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[26] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.

[28] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2633–2642.

[29] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223.

[30] A. Timans, C.-N. Straehle, K. Sakmann, and E. Nalisnick, "Adaptive bounding box uncertainties via two-step conformal prediction," in *European Conference on Computer Vision*. Springer, 2024, pp. 363–398.

[31] N. Darabi, P. Shukla, D. Jayasuriya, D. Kumar, A. C. Stutts, and A. R. Trivedi, "Navigating the unknown: Uncertainty-aware compute-in-memory autonomy of edge robotics," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–6.

[32] A. R. Trivedi, S. Tayebati, H. Kumawat, N. Darabi, D. Kumar, A. K. Kosta, Y. Venkatesha, D. Jayasuriya, N. Jayasinghe, P. Panda, S. Mukhopadhyay, and K. Roy, "Intelligent sensing-to-action for robust autonomy at the edge: Opportunities and challenges," in *2025 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2025, pp. 1–10.