Symmetric Conformal Uncertainty Quantification for Dense Multimodal Semantic Segmentation in Planetary Rover Perception

Théo Gachet[1,2] and Giovanni Beltrame[1,2]

[1]*Mila, Quebec AI Institute*
[2]*MIST Lab, Polytechnique Montréal, Montréal, QC, Canada*
theo.gachet.pro@outlook.fr

A planetary rover acts on every segmentation label as if it were certain, so its safety depends less on average accuracy than on knowing when a prediction cannot be trusted. We present a conformal pipeline that attaches a finite-sample coverage guarantee to dense semantic segmentation on a rover's two modalities, camera images and LiDAR point clouds. Each emits an uncertainty card, a prediction set with an accept-or-reject mask on a common scale, built by four post-hoc components applied identically to a vision transformer and a point-cloud network, so nothing is retrained and the rover carries only a handful of scalars. On the LuSNAR lunar benchmark the sets cover at the target rate while remaining almost always singletons, a class-conditional step repairs the coverage a global threshold misses on crater, and rejection improves accuracy and error ranking. Because both branches are calibrated identically their disagreement becomes measurable: the camera abstains on craters far more often than the LiDAR, so the calibration names which sensor should carry the decision.

Keywords: Conformal prediction, uncertainty quantification, semantic segmentation, multimodal perception, planetary rover

1 Introduction

Semantic segmentation is how a planetary rover reads the terrain ahead, and models such as SegFormer [1] on images and PointNet++ [2] on point clouds now do so accurately on planetary benchmarks. Accuracy, however, is an average over a test set, and a traverse is decided one prediction at a time. The rover acts on each label as if it were true, so what threatens it is not error as such but error delivered with confidence.

Deep networks are systematically overconfident [3]: loose regolith read as solid rock at 0.99 is not a statistical blemish but a kinematic trap, and one such mistake can immobilize the platform for good. Safety therefore begins with frequentist calibration: a reported 0.8 should be correct about eight times in ten.

Yet even a well-calibrated probability is a score, not a certificate, and a usable guarantee needs three further properties. It should hold without parametric assumptions, no model of an unexplored terrain being safe to presume. It should hold class by class, because a target met on average can still fail on a rare hazard such as a crater, exactly where it matters. And it should take the same form for every sensor, since otherwise an uncertain pixel cannot be weighed against an uncertain LiDAR point.

Conformal prediction [4, 5] delivers the first by construction: a held-out calibration set turns any trained predictor into prediction sets covering the true label at a user-chosen rate. Class-conditional variants such as Kandinsky conformal prediction [6] extend it to each class. We secure the third property by design, chaining four standard components: Monte Carlo dropout [7] for epistemic uncertainty, temperature scaling [3] to calibrate it, dense adaptive prediction sets [8] for coverage, and a class-wise Kandinsky step for the per-class target. Run on both branches of the LuSNAR lunar benchmark [9], the pipeline gives each modality an uncertainty card: at every pixel and point, a prediction set whose size grades local ambiguity and whose members name the plausible classes, the calibrated scores behind it, and a class-conditional accept or reject decision graded by the margin to its threshold, commensurable across sensors by construction.

Conformal methods have reached dense image segmentation [10, 6], robot navigation, and camera-based 3D semantic occupancy [11], always for a single modality. To our knowledge, no prior work applies one conformal procedure symmetrically to paired image and LiDAR segmentation, with class-conditional calibration, in a planetary setting. That symmetry lets the streams disagree meaningfully, and our experiments show that the sensors fail in different places, each covering the other's weaknesses.

This paper makes four contributions.

- A modality-symmetric pipeline that calibrates 2D image and 3D LiDAR segmentation with one identical procedure, placing their uncertainty on a common scale (C1).
- Class-conditional coverage for rare hazardous classes, raising 2D crater coverage from $64.50\,\%$ under a single global threshold to $90.25\,\%$ (C2).
- A cross-modal rejection analysis quantifying the complementarity of the sensors: crater is rejected $84.3\,\%$ of the time in 2D but only $4.1\,\%$ in 3D (C3).
- An evaluation of selective prediction under the pipeline's accept or reject rule, with mean accuracy up 3.54 points in 2D and 10.70 in 3D, the area under the risk-coverage curve down $45\,\%$ and $85\,\%$, and a study of the rejection rate under input corruption (C4).

This paper stops at the per-modality uncertainty cards. The conservative operator that fuses them and the risk-aware planner AURA* that consumes them are developed in a companion paper that rests on the perception guarantees established here.

2 Related Work

The most direct response to overconfidence is recalibration. Temperature scaling and its relatives rescale logits on held-out data until confidences track observed frequencies [3], the expected calibration error [12] measuring the residual gap, and the approach now covers dense tasks such as segmentation and depth estimation [13]. Two limits matter here. A calibrated probability, however faithful, remains a pointwise score with no coverage certificate, and marginal assessment lets a good aggregate over all pixels hide severe miscalibration of one rare class.

A complementary line estimates what the model does not know instead of rescaling what it reports. Bayesian analyses separate aleatoric from epistemic uncertainty [14], deep ensembles read the epistemic term off the disagreement of independently trained networks [15], and Monte Carlo dropout draws a comparable signal from one trained model by sampling sub-networks at inference [7]. We adopt the latter, which needs only the model we train. All, however, return scores that rank predictions but certify none, and a certificate is what a safety argument needs.

Conformal prediction supplies that certificate: wrapped around any point predictor with a held-out calibration set, it yields prediction sets with finite-sample, distribution-free coverage [4, 5], and adaptive prediction sets [8] let set size track input difficulty. The framework now reaches dense prediction, with per-pixel sets for segmentation [10] and a class-by-class Kandinsky calibration restoring coverage for categories a marginal procedure would starve [6]. In three dimensions the literature is thinner: hierarchical conformal prediction protects rare safety-critical classes in camera-based semantic occupancy [11], and robotics has used conformal sets for navigation and safe planning in dynamic environments [16]. To our knowledge, dense conformal segmentation of raw LiDAR point clouds is unreported, and no procedure runs symmetrically on paired image and LiDAR streams: our multimodal guarantee needs both, and that is the gap this paper fills.

Planetary exploration imposes constraints of its own: segmentation must run on limited onboard hardware [17], and the planners it feeds increasingly reason about risk, for instance fusing probabilistic traversability estimates over heterogeneous terrain [18]. Our experiments use LuSNAR [9], a simulated lunar benchmark pairing camera images with LiDAR scans and labeling both domains. This paper builds the perception side, a full pipeline over both modalities, and leaves the decision side, from fusion to planning, to the companion paper.

3 Multimodal Perception Setup

This section describes the benchmark, the alignment of its two label spaces, and the two models the pipeline wraps.

LuSNAR is a simulated benchmark for autonomous lunar navigation [9]. Its Unreal Engine scenes, built from lunar altimetry, span controlled combinations of topographic relief and object density, and at every timestamp along a traverse the virtual rover records camera images with dense semantic masks and a 360-degree LiDAR point cloud, all tied to one master clock. The benchmark suits our study because both domains are labeled on synchronized acquisitions, so the same scene is read by both sensors at once. Figure 1 shows a single frame in both domains, from the image mask and its overlay to the LiDAR points projected in the image plane and in three dimensions. The label spaces are:

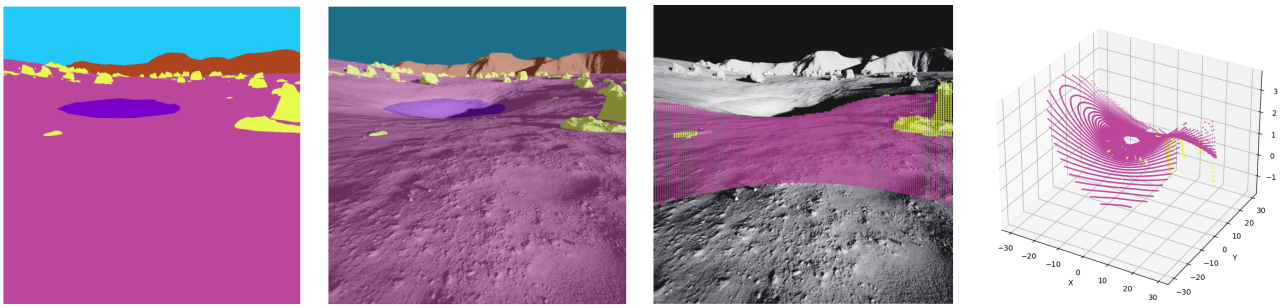
$$\mathcal{C}_{2D} = \{\text{regolith}, \text{crater}, \text{rock}, \text{mountain}, \text{sky}\}$$
$$\mathcal{C}_{3D} = \{\text{regolith}, \text{crater}, \text{rock}\} \subset \mathcal{C}_{2D}$$

The image branch is SegFormer-B0 [1], a hierarchical vision transformer chosen for its favorable accuracy-to-compute ratio. The LiDAR branch is the single-scale-grouping variant of PointNet++ [2] and classifies each point from its coordinates alone. The class distribution is skewed toward regolith, so training minimizes a cross-entropy weighted inversely to class frequency.

Both branches reach the aggregate quality expected of healthy baselines (Table 1): a mIoU of 0.951 for the image branch, and 0.920 for the LiDAR branch. The row that matters is crater: comfortable in 2D at 0.937, yet the weakest class in 3D at 0.832. Section 5 shows that this comfort does not survive calibration: crater is the class whose probability coverage fails first, and the first beneficiary of the class-conditional step of Section 4.

Table 1: Per-class IoU of the two branches on the validation set.

Class	IoU (2D)	IoU (3D)
Regolith	0.953	0.982
Crater	0.937	0.832
Rock	0.902	0.947
Mountain	0.972	—
Sky	0.990	—
mIoU	0.951	0.920



(a) Semantic mask **(b)** RGB + mask overlay **(c)** LiDAR projected **(d)** 3D point cloud

Figure 1: Synchronized multimodal views of a single LuSNAR frame. Classes: lunar regolith, impact crater, rock, mountain, sky [9].

4 Symmetric Conformal Pipeline

The pipeline is specified in Figure 2, and its shape follows from a single organising decision: everything statistical is settled before deployment. A held-out calibration set yields the temperature and the conformal thresholds once and for all, so that the rover carries no calibration data and computes no quantile, only a handful of stored scalars that it applies unchanged. There are seven of them in the image branch and five in the LiDAR branch: one temperature, one marginal threshold, and one threshold per class. What the pipeline then adds onboard, at each unit, is a sort over the classes, a cumulative sum, and two comparisons against those scalars, so the stochastic forward passes described below remain by far the dominant cost of quantifying uncertainty. Both bands are built from the same components, but a component does two different things depending on which band it sits in: offline it estimates a parameter from labelled units, and online it consumes that parameter to transform an input whose labels are unknown. For example, temperature scaling fits the scalar s in the first case and divides by it in the second.

Notation. The pipeline wraps a trained segmentation network f_θ, one per modality, and never retrains it. We denote by u a prediction unit, which is a pixel in the image or a point in the LiDAR cloud, depending on which model is being calibrated. C is the size of the corresponding label space, and $z^{(u)}$ represents a calibrated probability vector over those C classes. Calibration draws on a held-out set $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{n}$, disjoint from the data used for training, and α is the miscoverage we accept, fixed at 0.1 for a $90\,\%$ target. This risk level is set according to established standards [8], balancing the reliability demands of autonomous navigation against the informativeness of prediction sets while avoiding trivially large ensembles.

Calibrated probabilities. The calibrated distribution z is produced by two components in sequence. Monte Carlo dropout [7] replaces the network's single deterministic output by an ensemble, and temperature scaling [3] converts that ensemble into probabilities whose confidence can be read at face value. Applying the same two components to a vision transformer and to a point-cloud network is what puts the two modalities on one scale in the first place.

We keep dropout active whenever the network is run, in either phase, so that each pass evaluates a different sub-network and turns regularisation into a source of controlled stochasticity at inference. The T_{MC} sets of logits so obtained are averaged,

$$z_{\text{mean}}(x) = \frac{1}{T_{\text{MC}}} \sum_{t=1}^{T_{\text{MC}}} f_\theta^{(t)}(x), \tag{1}$$

with $T_{\text{MC}} = 25$, a value chosen empirically as the point beyond which the estimates no longer moved appreciably while the extra passes still fitted a plausible onboard budget. Averaging logits rather than softmax outputs is deliberate, since the next component rescales logits and would have nothing to act on once the nonlinearity had been applied.

Averaging reduces the noise in those logits but leaves their scale untouched, so the softmax remains too peaked. Temperature scaling divides every logit by one positive scalar s before the softmax, which flattens the distribution when $s > 1$ and sharpens it when $s < 1$. Because that single parameter is shared by all classes, the ordering of the logits is preserved, so recalibration moves confidence without changing a single prediction.

The scalar s is the first quantity estimated offline, chosen to make the calibration labels as likely as possible under the rescaled distribution, that is by minimising the negative log-likelihood

$$\mathcal{L}(s) = -\sum_{i=1}^{n} \log \frac{\exp\left(z_{\text{mean},i}^{\,y_i}/s\right)}{\sum_{c=1}^{C} \exp\left(z_{\text{mean},i}^{\,c}/s\right)} \tag{2}$$

over the labelled calibration units. Fitting returns one temperature per branch, whose values Section 5 reports together with the correction they imply.

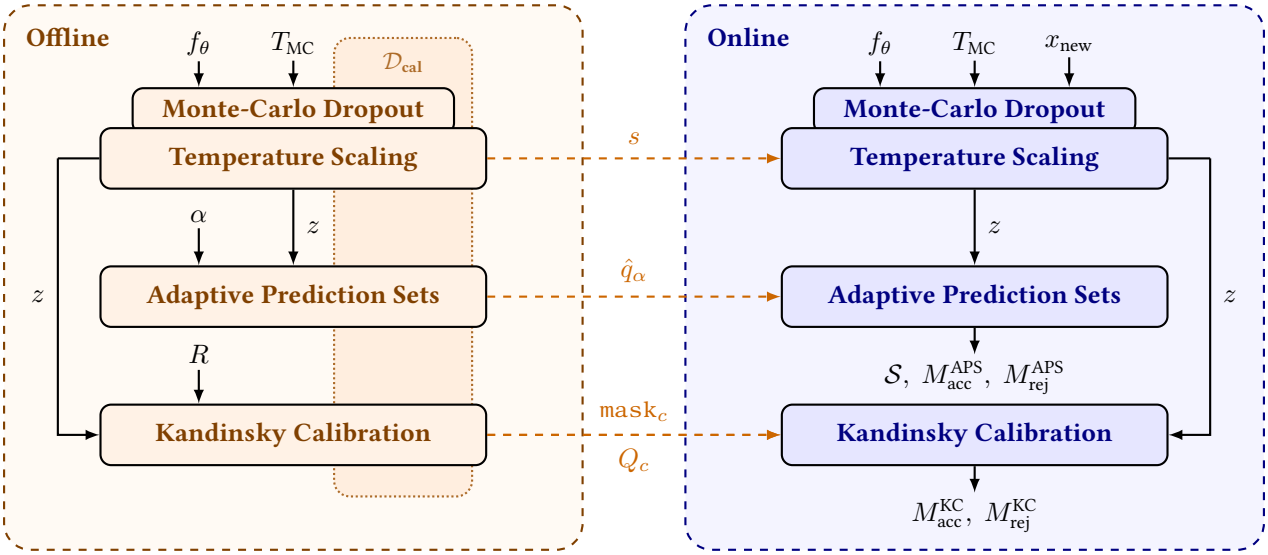


Figure 2: Unified calibration pipeline, run unchanged on each modality. Offline, a held-out set yields the temperature s, the marginal threshold $\hat{q}_\alpha$, and the class-wise thresholds. Dashed arrows carry those parameters into online inference, where the two heads emit the uncertainty card. In both phases the heads read the same calibrated probabilities z.

The scalar is then frozen and serves in both phases, since each needs calibrated probabilities for a different purpose: offline they are the input from which the thresholds below are estimated, online they are the input to which those thresholds are applied, and in either case they are obtained as $z = \mathrm{softmax}(z_{\mathrm{mean}}/s)$.

Scores and thresholds. The remaining two components are instances of one recipe, worth stating once. A nonconformity score maps $z^{(u)}$ and a reference class to a single number measuring how surprising that class is under the calibrated distribution. A threshold is the empirical $(1-\alpha)$ quantile of those scores over $\mathcal{D}_{\mathrm{cal}}$, and comparing a test score against that stored threshold is what turns a probability into a guarantee. We call each such rule a conformal head, and the two differ in only two respects: the score they compute, and how finely the threshold is resolved, either one value for the whole population or one value per class. They are not chained, so the second does not refine the output of the first, and both read the same z, which is what allows their guarantees to be compared rather than composed.

Conformal validity requires calibration and test units to be exchangeable, which a random image-level partition of the benchmark secures between frames. Within a frame it does not, since neighbouring pixels carry nearly the same score, so pooling them into one quantile treats correlated units as exchangeable. Two consequences follow: the guarantee holds for the pooled population rather than for any individual frame, which is the case that matters on a traverse, and the effective sample size is closer to the frame count than to the unit count. We therefore measure coverage in Section 5 rather than rest on its nominal level, and Section 6 returns to the assumption.

Adaptive prediction sets. The marginal head instantiates that recipe with a single threshold for the whole population. We write $y^{(u)}$ for the label of a unit u, and π for the permutation of the C class indices that sorts them by decreasing calibrated probability:

$$z^{(u)}\big[\pi[1]\big] \;\geq\; z^{(u)}\big[\pi[2]\big] \;\geq\; \cdots \;\geq\; z^{(u)}\big[\pi[C]\big] \tag{3}$$

The permutation depends on u, a dependence we leave implicit.

Offline the label is available, and the score of a calibration unit is the probability mass its distribution places on that label together with every class ranked above it [8],

$$\sigma^{\mathrm{APS}}(u) = \sum_{k=1}^{m} z^{(u)}\big[\pi[k]\big], \qquad \pi[m] = y^{(u)}, \tag{4}$$

where m is the rank of the label in the ordering of Equation 3. The threshold is the $(1-\alpha)$ quantile of those scores,

$$\hat{q}_\alpha = \underset{1-\alpha}{\mathrm{Quantile}}\Big(\big\{\sigma^{\mathrm{APS}}(u) : u \in \mathcal{U}_{\mathrm{cal}}\big\}\Big) \tag{5}$$

and reads as the amount of accumulated mass that suffices to reach the label in a proportion $1-\alpha$ of the calibration units.

Online the label is unknown, so the sum has nothing to stop at. The construction is reversed instead, and classes are added in the same order until the accumulated mass reaches that stored amount,

$$\begin{aligned} \mathcal{S}(u) &= \big\{\pi[1], \ldots, \pi[m^*]\big\}, \\ m^* &= \min\Big\{m : \sum_{k=1}^{m} z^{(u)}\big[\pi[k]\big] \geq \hat{q}_\alpha\Big\}. \end{aligned} \tag{6}$$

The set therefore contains the label as often as $\hat{q}_\alpha$ was sufficient on the calibration data, which under exchangeability is the guarantee

$$\mathbb{P}\big(y^{(u)} \in \mathcal{S}(u)\big) \;\geq\; 1-\alpha, \tag{7}$$

the probability being taken over the unit population described earlier. Because the set grows only as far as the mass requires, it stays short where the distribution is peaked and lengthens where it is flat, and running the construction at every unit rather than once per input returns a spatially structured field of such sets.

A planner cannot act on a set. We therefore treat a unit as unambiguous only when its set is a singleton, since $|\mathcal{S}(u)| = 1$ is precisely the case where one class already carries the mass the guarantee requires. Rather than record whether that condition held, the two maps record how far the unit stands from it,

$$\begin{aligned} M^{\mathrm{APS}}_{\mathrm{acc}}(u) &= \begin{cases} z^{(u)}\big[\pi[1]\big] & \text{if } |\mathcal{S}(u)| = 1, \\ 0 & \text{otherwise,} \end{cases} \\ M^{\mathrm{APS}}_{\mathrm{rej}}(u) &= \max\big(0,\; \hat{q}_\alpha - z^{(u)}\big[\pi[1]\big]\big), \end{aligned} \tag{8}$$

so that a unit whose leading class falls just short of the threshold remains distinguishable from one that falls far short.

Kandinsky calibration. Calibrating a threshold for every unit is noisy, and calibrating one for an entire image is too coarse. Kandinsky conformal prediction [6] sits between the two, on the assumption that an image splits into regions within which uncertainty behaves alike, and it discovers those regions by clustering per-unit nonconformity curves. We retain the regional principle and change how the regions are obtained, taking them to be the semantic classes themselves, as class-conditional conformal prediction does [19, 20]. The premise is that uncertainty follows the semantics of the scene rather than a latent partition, and it pays twice here: the offline stage no longer builds a curve per unit, and each threshold becomes a tolerance attached to a terrain type, which is what makes the values of Section 5 readable at all.

Offline, this head scores a calibration unit by one minus the probability its distribution gives to the label,

$$\sigma^{\mathrm{KC}}(u) = 1 - z^{(u)}\big[y^{(u)}\big], \tag{9}$$

a simpler quantity than Equation 4, and using a different score here costs nothing, since conformal validity attaches to a score together with the quantile computed from it, so each head is valid on its own terms. Calibration units are then grouped by label, and each class c draws its own quantile,

$$\begin{aligned} Q_c(\alpha) &= \underset{1-\alpha}{\mathrm{Quantile}}\big(\mathcal{R}_c\big), \\ \mathcal{R}_c &= \big\{\sigma^{\mathrm{KC}}(u) : u \in \mathcal{U}_{\mathrm{cal}},\; y^{(u)} = c\big\}, \end{aligned} \tag{10}$$

so that $Q_c(\alpha)$ is the nonconformity level below which a proportion $1-\alpha$ of the calibration units of class c fall.

Online no label is available to index the threshold, and the leading class $\hat{y} = \pi[1]$ stands in for it. $\sigma^{\mathrm{KC}}(u)$ is now evaluated at $\hat{y}$ rather than at the unavailable label, but the comparison and the margin then follow the same pattern as before,

$$\begin{aligned} M^{\mathrm{KC}}_{\mathrm{acc}}(u) &= \begin{cases} \sigma^{\mathrm{KC}}(u) & \text{if } \sigma^{\mathrm{KC}}(u) \leq Q_{\hat{y}}(\alpha), \\ 0 & \text{otherwise,} \end{cases} \\ M^{\mathrm{KC}}_{\mathrm{rej}}(u) &= \max\big(0,\; \sigma^{\mathrm{KC}}(u) - Q_{\hat{y}}(\alpha)\big). \end{aligned} \tag{11}$$

That substitution deserves a word, because it is where the guarantee weakens. Equation 10 calibrates Q_c on units whose true class is c, and by exchangeability that quantile transfers to test units of the same true class. Indexing by $\hat{y}$ reproduces the calibrated statement exactly wherever the prediction is correct, and departs from it wherever the prediction is wrong, since both the score and the threshold then refer to a class the unit does not belong to. Per-class coverage is therefore something we measure rather than assert, and Section 5 reports it directly.

The uncertainty card. The two heads converge on a single output, and each modality emits, at every unit, the tuple

$$\left(z,\ \mathcal{S},\ M_{\text{acc}}^{\text{APS}},\ M_{\text{rej}}^{\text{APS}},\ M_{\text{acc}}^{\text{KC}},\ M_{\text{rej}}^{\text{KC}}\right), \tag{12}$$

to which the dispersion of the T passes adds a per-unit variance and mutual information that the card carries for the planner and that calibration itself never consumes. Nothing in Equation 12 is specific to a sensor, and both cards are produced by the same procedure on either sensor, from the same held-out protocol, which is what licenses comparing a pixel with a point in Section 5. Algorithm 1 states the construction end to end.

Algorithm 1: Symmetric conformal pipeline

Input: network f_θ, calibration set $\mathcal{D}_{\text{cal}}$, passes T_{MC}, level α, new input x_{new} (2D or 3D)
Output: uncertainty card of x_{new}

▷ *Offline: estimate the scalars, then freeze them*
foreach $(x_i, y_i) \in \mathcal{D}_{cal}$ **do**
  $z_{\text{mean},i} \leftarrow$ mean logits over T_{MC} passes ▷ (1)
$s \leftarrow \arg\min_s \mathcal{L}(s)$ ▷ (2)
$z_i \leftarrow \text{softmax}(z_{\text{mean},i}/s)$ for every i
foreach $u \in \mathcal{U}_{cal}$ **do**
  $\pi \leftarrow$ classes by decreasing $z^{(u)}$ ▷ (3)
  $\sigma^{\text{APS}}(u),\ \sigma^{\text{KC}}(u) \leftarrow$ scores at $y^{(u)}$ ▷ (4), (9)
$\hat{q}_\alpha \leftarrow \text{Quantile}_{1-\alpha}\left\{\sigma^{\text{APS}}(u)\right\}$ ▷ (5)
foreach *class* c **do**
  $Q_c \leftarrow \text{Quantile}_{1-\alpha}\left\{\sigma^{\text{KC}}(u) : y^{(u)} = c\right\}$ ▷ (10)

▷ *Online: apply the frozen scalars to* x_{new}
$z \leftarrow \text{softmax}\left(z_{\text{mean}}(x_{\text{new}})/s\right)$
foreach *unit* u *of* x_{new} **do**
  $\pi \leftarrow$ classes by decreasing $z^{(u)}$
  $\mathcal{S}(u) \leftarrow$ shortest prefix of π with mass $\geq \hat{q}_\alpha$ ▷ (6)
  $M_{\text{acc}}^{\text{APS}},\ M_{\text{rej}}^{\text{APS}} \leftarrow$ singleton test, margin ▷ (8)
  $\hat{y} \leftarrow \pi[1]$ ▷ *no label online*
  $M_{\text{acc}}^{\text{KC}},\ M_{\text{rej}}^{\text{KC}} \leftarrow \sigma^{\text{KC}}(u)$ against $Q_{\hat{y}}$, margin ▷ (11)
return *the card of* (12)

5 Experiments

What calibration returns. The offline stage of Section 4 compresses the calibration set into the scalars of Table 2, the complete set a rover carries on board. The count claimed in Section 4 can now be read off the columns: seven scalars govern the image branch and five the LiDAR branch, one temperature, one marginal quantile, and one threshold per class of each label space.

Table 2: Calibration artifacts per branch at $\alpha = 0.1$.

	2D	3D
Temperature s	1.297	1.987
Marginal quantile $\hat{q}_\alpha$	0.8931	0.9277
Regolith (class threshold)	0.0403	0.0182
Crater (class threshold)	0.9970	0.0471
Rock (class threshold)	0.7461	0.5417
Mountain (class threshold)	0.9678	—
Sky (class threshold)	0.0346	—

Fitting Equation 2 returns $s_{\text{2D}} = 1.297$ and $s_{\text{3D}} = 1.987$. Both exceed one, so both branches needed their distributions flattened, which is the signature of overconfidence, and the LiDAR branch needed the stronger flattening of the two. The correction they imply is not cosmetic. Expected calibration error falls from 9.47 % to 1.09 % on the image branch and from 11.50 % to 2.13 % on the LiDAR branch, and the Brier score follows, from 0.0928 to 0.0695 and from 0.1020 to 0.0317. Mean predictive entropy rises meanwhile, from 0.0612 to 0.3520 and from 0.0550 to 0.1033, which is the mechanics of flattening made visible: the argmax survives untouched while the excess confidence around it is released.

Figure 3 locates where the excess lived. Before scaling, an image pixel announced at 0.90 confidence is correct only 54 % of the time, a gap none of the aggregate scores of Table 1 betrays. After scaling that bucket sits on the diagonal, then tips into mild under-confidence, the conservative side to err on for a rover. The LiDAR branch starts further out, 65 % correct at the same confidence, consistent with the max-pooling of its architecture, which keeps only the largest activations and hands the softmax inflated logits [2]. Once scaled it lands within one point of the diagonal.

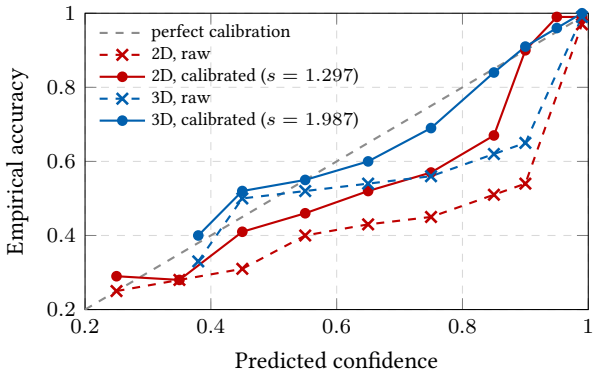


Figure 3: Reliability diagrams of both branches on the test set.

The class rows of Table 2 carry the more interesting spread. Each threshold is the tolerance a unit of that class must stay under to be kept, so a low value means the branch scores the class confidently enough to be strict with it, and a high value means the scores are uninformative and almost nothing can be turned away. Crater sits at both extremes, 0.9970 on the image branch against 0.0471 on the LiDAR branch. Yet the camera holds the better crater IoU of the two in Table 1, 0.937 against 0.832: the branch that finds craters more accurately is the one whose confidence has to be trusted the least.

Does the guarantee hold, and at what price. The reliability diagrams asked whether the announced probabilities are honest. Coverage asks something more operational, whether the true label lies inside the emitted set, and the two can disagree. A branch could be badly calibrated and still cover, by emitting large sets, which is why coverage has to be read together with set size. Figure 4 sweeps the threshold and reports both.

At the calibrated thresholds the image branch covers 93.6 % of test units and the LiDAR branch 95.1 %, against a target of 90 %. The overshoot is expected rather than alarming, since the finite-sample quantile of Equation 5 is deliberately conservative and the score distribution is discrete, so the attainable coverage moves in steps. The price is what makes the result usable. Mean set size stays at 1.07 and 1.05, so the overwhelming majority of units still carry a single label and the downstream consumer receives a segmentation map, not a cloud of alternatives. The two sweeps differ in shape. The image curve climbs almost linearly across the operating range, whereas the LiDAR curve saturates early, which follows from the low entropy of sparse point predictions: once the geometry is unambiguous, raising the threshold buys coverage the branch already had.

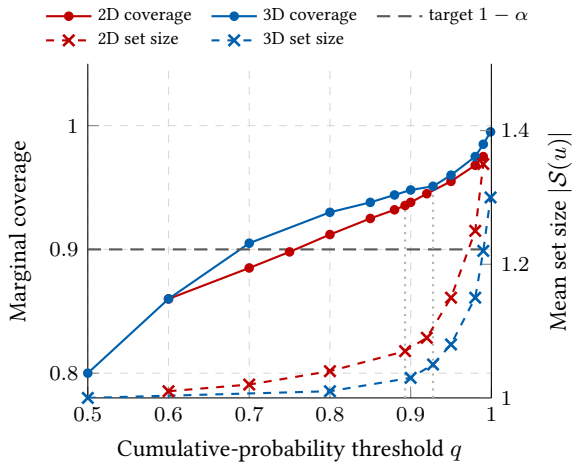


Figure 4: Coverage and set size as the threshold sweeps, both branches. Vertical marks give the calibrated $\hat{q}_\alpha$ of each branch.

Where the marginal guarantee fails. A marginal statement is an average over classes, and averages hide the case that matters. Table 3 splits the same test set by class. On the image branch, marginal calibration alone leaves crater at 64.50 % coverage while the fleet-wide number reads 93.6 %. The guarantee was honoured and the one class a rover cannot afford to miss was undercovered by twenty-five points, absorbed by regolith and sky at 98.40 % and 97.34 %. This is the concrete failure that motivates a per-class threshold, and it is invisible to every aggregate reported so far.

Kandinsky calibration repairs it, and the repair is not free. Crater coverage returns to 90.25 %, at the cost of rejecting 84.3 % of crater units. That figure should be read as a diagnosis rather than a defect. To cover craters at the target rate on a branch whose crater scores carry almost no information, the procedure has no option but to abstain on nearly all of them. The LiDAR branch shows the same mechanism running in the opposite direction. Its native coverage already exceeded the target everywhere, up to 100 % on regolith, so calibration tightens instead of loosening, trading unnecessary conservatism for precision while rejection stays below 5 % on every class. One procedure, one target, two opposite corrections, because both are driven by what the calibration data measured rather than by an assumption about which sensor should be trusted.

Table 3: Per-class coverage before and after class-conditional calibration, at $\alpha = 0.1$.

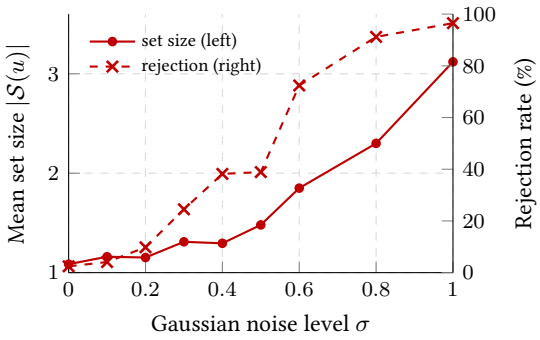
Class	Marginal (%)	Per-class (%)	Rejected (%)
Image branch			
Regolith	98.40	90.05	1.2
Rock	93.15	90.10	8.4
Crater	64.50	90.25	84.3
Mountain	83.99	89.95	3.1
Sky	97.34	90.03	1.6
LiDAR branch			
Regolith	100.00	90.02	0.5
Rock	98.88	90.15	2.8
Crater	95.20	90.12	4.1

What symmetry buys. The crater rows of Table 3 are the argument for building both branches the same way. The image branch abstains on 84.3 % of craters, the LiDAR branch on 4.1 %, for the same class, the same target, and the same definition of rejection. Because Section 4 applies one protocol to both sensors, the gap between those numbers is a measurement rather than an artefact of two differently tuned procedures, and a fusion stage can act on it directly. The pattern is legible: on craters the camera declares itself unfit and the geometry does not, so the sensor that should carry the decision is named by the calibration itself rather than by a hand-set prior. The cards therefore carry the information a fusion rule needs, which is the precondition the companion paper builds on.

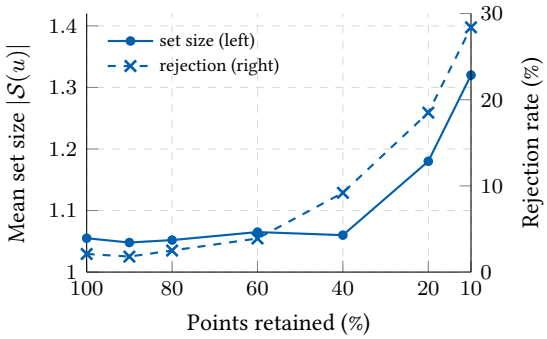
Rejection as a predictor of error. If the uncertainty is worth its cost, discarding the units it flags should leave a more accurate map behind. The baseline is the same trained networks stripped of the pipeline, ranking units by their raw softmax maximum, with no dropout sampling, temperature scaling, or conformal step. Both systems abstain by sweeping a threshold on their own score and emit the same labels, since Section 4 leaves the argmax untouched, so any gap is attributable to the ranking alone. Table 4 reports both at their operating point and Figure 5 shows the whole trade-off. Accuracy on retained units rises from 89.96 % to 93.50 % on the image branch and from 85.10 % to 95.80 % on the LiDAR branch. The second gain is the larger one because the baseline is worse, not because the geometry is harder.

Table 4: Selective prediction against the softmax baseline. The mAcc row is read at the class-conditional operating point, the AURC row grades the APS ranking over the whole sweep.

	Image branch		LiDAR branch	
	Baseline	Pipeline	Baseline	Pipeline
mAcc (%)	89.96	93.50	85.10	95.80
AURC	0.0584	0.0321	0.0825	0.0124



(a) Image branch under additive Gaussian noise.



(b) LiDAR branch under decimation, density decreasing rightwards.

Figure 6: Response to controlled corruption. Perturbation grows to the right in both panels.

Two things are being measured here and they should not be conflated. The area under the risk-coverage curve grades the ranking, since it summarises the whole sweep and is therefore insensitive to where a threshold happens to sit, and it falls by 45 % on the image branch and by 85 % on the LiDAR branch. The accuracies above are instead read at the single operating point the class-conditional step selects, which is what an onboard system would actually run. That point is the more economical of the two policies: on the image branch it reaches 93.5 % while discarding 7.96 % of units, half the budget the marginal ranking needs for a comparable accuracy.

The shapes of the two sweeps differ in kind. The LiDAR ranking recovers most of its headroom within the first 5 % rejected, 96.40 % against 91.15 % for the softmax ranking at the same budget, because its errors are sparse and topologically isolated, mixed points at depth discontinuities and stray returns that the score separates cleanly. The image ranking climbs gradually instead, since its errors are structured and concentrated along semantic boundaries where a pixel genuinely mixes two surfaces, so no small rejection budget removes them. The same mechanism yields different returns because the error geometry differs, which an aggregate accuracy hides. On both branches the calibrated score ranks errors better than raw confidence at a modest budget. For a rover that is the useful form, since the rejected units are what a planner should treat as unknown, not free space.

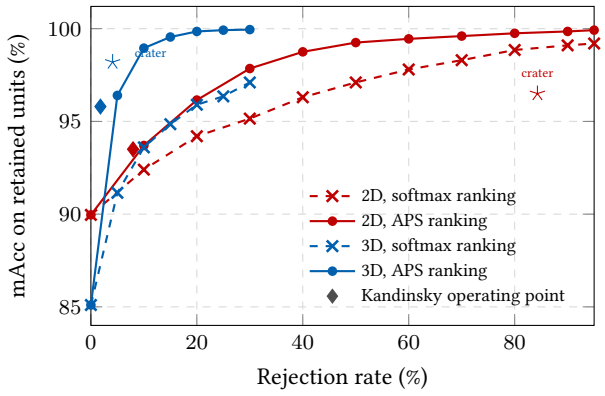


Figure 5: Accuracy against rejection rate for both rankings. Diamonds give the operating point Kandinsky selects, stars the crater point on each branch. The LiDAR sweep is drawn to 30 %, where accuracy has already saturated at 99.95 %.

Behaviour under controlled corruption. A calibrated score should degrade when the input degrades, and the last experiment checks that the pipeline reacts in the right direction rather than claiming robustness to distribution shift. The image branch receives additive Gaussian noise of increasing variance, with perceptual severity tracked by LPIPS [21] rather than a pixel metric (Figure 6a), and the LiDAR branch is decimated uniformly from full density down to a tenth of it (Figure 6b). The two panels use their own scales, since a noise level and a point density are not commensurable.

Monotonicity holds on both branches, with each correlation computed against its own perturbation variable. Spearman correlation with mean set size reaches $\rho = 0.967$ over the nine noise levels and $\rho = 0.857$ over the seven density levels, the latter taken against decreasing density. Because those sample sizes are small, the p-values come from exact one-sided permutation tests over all orderings rather than a large-sample approximation, and give $p = 8.3 \times 10^{-5}$ and $p = 1.2 \times 10^{-2}$. Both remain significant at the conventional level, with the LiDAR estimate resting on seven points and therefore stated with more reserve.

The two responses differ in kind, and the vertical scales make it plain. In Figure 6a the image branch holds steady up to about $\sigma = 0.4$, then breaks, its set size tripling and rejection reaching 96.5 % at full noise. That collapse is the desired behaviour, since a camera facing an unusable image should invalidate its output wholesale rather than degrade quietly. Figure 6b spans a far narrower range, and the LiDAR branch even moves the wrong way at first: rejection falls from 2.10 % to 1.80 % once the first tenth of the points goes. Decimation acts as a passive filter, discarding isolated returns that would otherwise dominate the max-pooling aggregation, so the branch grows more confident before sparsity costs it below 40 % density. Both effects follow from the architectures and establish what the experiment set out to check: the score tracks input quality, so a branch can flag its own perception as unusable before the errors reach a planner.

Taken together, the four experiments trace one procedure. Scaling makes the probabilities honest, the marginal quantile turns them into sets that cover at almost no cost in decisiveness, the class-conditional thresholds expose and repair what each sensor cannot score, and the corruption sweeps confirm the signal tracks the input rather than the network. Every number comes from the same offline stage, applied identically to a pixel and to a point, which is what makes the two branches comparable.

6 Discussion

Section 4 promised to return to the exchangeability assumption, and the coverage numbers cannot be read without it. Neighbouring pixels carry nearly the same score, so pooling them into one quantile treats strongly correlated units as independent draws. The guarantee does not fail, but it is weaker than its nominal statement: it holds for the pooled population of units, and the effective sample size behind each threshold is closer to the number of calibration frames than to the number of pixels. The distinction is operational, not merely formal, since a rover consumes a whole frame at a time and nothing bounds how badly a single unlucky frame may be covered. This is why Section 5 measures coverage on a held-out split, and why the measured values land above target rather than on it. That overshoot is the safe direction to err in, bought with sets slightly larger than the target requires.

The crater rejection rate on the image branch, $84.3\,\%$, is the most striking number of Section 5 and the easiest to misread. The calibration creates no detection failure, since the underlying scores never carried the information and a global threshold merely hid that behind a marginal average. What the class-conditional step adds is the ability to say so, naming the class and the sensor at fault instead of letting an overconfident crater label reach the planner. The lesson outlives the lunar setting: segmentation quality and score reliability are distinct properties, and the branch with the better crater intersection over union is the one whose crater confidence is least usable. A pipeline that ranks its sensors by accuracy alone will trust the wrong one exactly where the stakes are highest. On one sensor that is a diagnosis without a remedy. Beside a second modality that rejects the same class only $4.1\,\%$ of the time, it becomes a delegation signal.

Three boundaries mark what the evidence covers. LuSNAR is the reference benchmark for lunar multimodal segmentation and the natural place to establish the method, but its scenes are simulated, so transfer to flight imagery depends on how closely the simulator reproduces real appearance and geometry. That dependence is on the score distribution rather than on photorealism, a weaker requirement, and one a short in-flight calibration pass could satisfy. The corruption study sweeps two parametric families over nine and seven levels, which probes the direction of the response without characterising the shifts a mission would meet, such as dust on the optics, low sun angles, or specular returns. The temperature and thresholds are fitted once and never revisited, so drift would erode the guarantee silently, coverage being exactly the quantity no one can measure without labels.

Each boundary suggests its own next step. Calibrating on units that respect the correlation structure, one quantile per frame or per region rather than per pixel, would trade statistical efficiency for a guarantee closer to the per-frame statement a planner wants. Drift-aware recalibration answers the second, and a dataset with real sensor data the third. None of them touches the architecture, which is the practical merit of the approach: the method is post-hoc, it leaves the trained networks and their argmax untouched, and everything a rover carries reduces to twelve scalars. A validated perception stack can therefore be given a coverage guarantee without being retrained or requalified, which counts for more in a flight programme than on a benchmark. Nothing in the procedure is specific to lunar geometry, only to the availability of a held-out set from the deployment distribution.

7 Conclusion

We applied one conformal calibration procedure symmetrically to the image and LiDAR branches of a planetary segmentation stack, so that the uncertainty attached to a pixel and the uncertainty attached to a point are produced by the same protocol and can be compared rather than merely juxtaposed. The offline stage compresses to twelve scalars, and the online stage emits, at every unit, a prediction set, its calibrated scores, and a class-conditional accept or reject decision graded by the margin to its threshold. Nothing is retrained, and the argmax the networks were validated on survives untouched, so the guarantee is added to a perception stack rather than substituted for it.

The guarantee comes cheap. Marginal calibration reaches $93.6\,\%$ and $95.1\,\%$ coverage at a mean set size near one, so almost every unit still carries a single label and the map stays as decisive as the raw prediction. Class-conditional calibration then raises crater coverage on the image branch from $64.50\,\%$ to $90.25\,\%$, and tightens an already excessive coverage on the LiDAR branch, one procedure correcting in opposite directions because both corrections are driven by measurement rather than by a prior on which sensor to trust. Acting on the rejections raises accuracy from $89.96\,\%$ to $93.50\,\%$ and from $85.10\,\%$ to $95.80\,\%$, with the area under the risk-coverage curve down $45\,\%$ and $85\,\%$. Under controlled corruption the score grows with the damage on both branches, so the quantity that ranks errors also reports when a sensor should no longer be believed at all.

The cross-modal reading is the result we consider most useful. On the class a rover cannot afford to misjudge, the camera abstains $84.3\,\%$ of the time and the LiDAR $4.1\,\%$, and because both figures come from the same procedure the gap is a measurement rather than an artefact of two separately tuned methods. The calibration names which sensor should carry the decision, with no hand-set prior, and it names it on the one class where accuracy alone would have pointed the other way. That is the practical shape of the argument: a perception stack does not need a better model to become safer, it needs an honest account of where its confidence is worth acting on. Producing that account is inexpensive, and the account is legible, since every rejection carries the class and the margin that caused it rather than a single opaque score.

Two directions follow. The first is to make the guarantee match the granularity a planner consumes, by calibrating on frames or regions instead of individual units, and to keep it valid over a mission through drift-aware recalibration. The second is to use the abstentions rather than merely record them, fusing the two cards into a single traversability estimate and giving a planner a cost that treats an abstention as terrain to avoid rather than as data to ignore. Neither direction requires a different perception model, only a more careful account of what the present one knows, which is the position this work argues for.

References

[1] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 12 077–12 090.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, vol. 30, 2017.

[3] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[4] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

[5] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *arXiv preprint arXiv:2107.07511*, 2021.

[6] J. Brunekreef, E. Marcus, R. Sheombarsing, J.-J. Sonke, and J. Teuwen, "Kandinsky conformal prediction: Efficient calibration of image segmentation algorithms," 2023. [Online]. Available: https://arxiv.org/abs/2311.11837

[7] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning (ICML)*. PMLR, 2016, pp. 1050–1059.

[8] Y. Romano, M. Sesia, and E. J. Candès, "Classification with valid and adaptive coverage," 2020. [Online]. Available: https://arxiv.org/abs/2006.02544

[9] J. Liu, Q. Zhang, X. Wan, S. Zhang, Y. Tian, H. Han, Y. Zhao, B. Liu, Z. Zhao, and X. Luo, "Lusnar: A lunar segmentation, navigation and reconstruction dataset based on muti-sensor for autonomous exploration," 2024. [Online]. Available: https://arxiv.org/abs/2407.06512

[10] L. Mossina, J. Dalmau, and L. Andéol, "Conformal semantic image segmentation: Post-hoc quantification of predictive uncertainty," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 3574–3584.

[11] S. Su, N. Chen, C. Lin, F. Juefei-Xu, C. Feng, and F. Miao, "α-occ: Uncertainty-aware camera-based 3d semantic occupancy prediction," 2025. [Online]. Available: https://arxiv.org/abs/2406.11021

[12] M. Pakdaman Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, Feb. 2015.

[13] S. Landgraf, M. Hillemann, T. Kapler, and M. Ulrich, "A comparative study on multi-task uncertainty quantification in semantic segmentation and monocular depth estimation," 2025. [Online]. Available: https://arxiv.org/abs/2405.17097

[14] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems (NeurIPS)*, vol. 30, 2017.

[15] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems (NeurIPS)*, vol. 30, 2017.

[16] L. Lindemann, M. Cleaveland, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5263–5270, 2023.

[17] D. Gasperini, W. Masawat, S. Santra, and K. Yoshida, "Clovernet: A real-time network for semantic segmentation onboard edge devices towards planetary exploration," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2023, pp. 333–338.

[18] M. Endo, T. Taniai, R. Yonetani, and G. Ishigami, "Risk-aware path planning via probabilistic fusion of traversability prediction for planetary rovers on heterogeneous terrains," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 852–11 858.

[19] T. Ding, A. N. Angelopoulos, S. Bates, M. I. Jordan, and R. J. Tibshirani, "Class-conditional conformal prediction with many classes," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[20] Y. Shi, S. Ghosh, T. Belkhouja, J. R. Doppa, and Y. Yan, "Conformal prediction for class-wise coverage via augmented label rank calibration," 2024. [Online]. Available: https://arxiv.org/abs/2406.06818

[21] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.