# Perception Debt: Monitoring Safety-Margin Consumption in Embodied Autonomy

Stavan Dholakia*, Shivani Shukla†, Abhishek Singh‡, Aditya Gazta‡
*Microsoft †University of San Francisco ‡Independent Researcher
github.com/stavanio/perception-debt

***Abstract*. Standard robot perception metrics evaluate accuracy at individual timesteps. They do not capture how localization and detection errors accumulate over long-horizon operation and consume available safety margin. We introduce the Debt-to-Margin Ratio (DMR), a runtime metric that tracks the fraction of a system's safety budget consumed by accumulated perception error. DMR is computed from a time-integral of safety-weighted perception error and requires only the error signal and a deployment-specific safety sensitivity. We show that any system with persistent nonzero perception error will exhaust a finite safety margin in bounded time, even when every frame passes its accuracy specification. We validate with two case studies: (1) a controlled numerical study showing that identical synthetic error traces produce a 5.7× difference in safety-margin exhaustion time under different deployment assumptions, and (2) a real-hardware detection study on a Yahboom ROSMASTER M1 with YOLOv5n showing a 2.3× difference under a PTZ viewpoint perturbation. Both cases illustrate a blind spot in robot perception benchmarking: the same per-frame accuracy can imply sharply different accumulated operational risk.**

***Index Terms*. perception evaluation, safety metrics, state estimation, long-horizon autonomy, object detection**

## I. Introduction

A robot perception pipeline can satisfy per-frame accuracy specifications at every timestep while accumulating enough error history to silently degrade its safety margin. Consider a warehouse navigation robot whose detector reports acceptable per-frame accuracy across a 90-second observation window. By any snapshot perception metric, this is within specification. But if the robot operates in a workspace where detection errors have high safety consequences, the accumulated safety-weighted error may exhaust the available margin well before the mission ends.

The root cause is a mismatch between how robot perception is evaluated and how perception errors affect long-horizon safety. As perception pipelines grow more capable through foundation models [3], [8], open-set detectors [15], and universal segmentation systems [12], the per-frame evaluation paradigm remains unchanged. Scaling efforts now span dozens of robot embodiments [16], and video-native perception models process temporal streams natively [17], yet existing benchmarks still summarize per-frame accuracy [14]. Deployed robots fail over time. The failure mode is not the snapshot. The failure mode is the integral.

We call the quantity that snapshot metrics miss *perception debt*: the time-integrated, safety-weighted divergence between estimated and true state. Analogous to technical debt in software engineering [6], perception debt arises when small, individually tolerable perceptual inaccuracies accumulate into a growing mismatch between the robot's believed state and the true state of the world.

To make this quantity operational, we define the Debt-to-Margin Ratio (DMR): the ratio of accumulated perception debt to the available safety margin. DMR is a complementary metric for robot perception evaluation, not a replacement for existing benchmarks. It answers a question that snapshot metrics do not: how much of the safety budget has been consumed so far?

This paper contributes: (1) a compact accumulated-risk metric for robot perception evaluation, (2) a finite-time expiration bound for snapshot-compliant perception under persistent error, and (3) two case studies, one controlled and one on real hardware, showing that identical local accuracy can imply sharply different long-horizon operational risk.

## II. DMR Formulation

Let $e(t) \geq 0$ be the instantaneous perception error (e.g., normalized detection center deviation). We define perception debt as:

$$D(t) = \int_0^t \omega(\tau) \cdot e(\tau)\, d\tau \tag{1}$$

where $\omega(t) \geq 0$ is the safety sensitivity, capturing how strongly the downstream pipeline amplifies perception error into safety-relevant consequences.

The sensitivity decomposes into three measurable quantities:

$$\omega(t) = \|J_P(t)\| \cdot K_c(t) \cdot \omega_e(t) \tag{2}$$

where $\|J_P(t)\|$ is the planner Jacobian norm, $K_c(t)$ is the controller gain, and $\omega_e(t)$ is environmental criticality (e.g., proximity to obstacles, derived from clearance or ISO/TS 15066 [10] force limits). For classical planners and PID controllers, all three are design constants or analytically computable.

Let $D_{\max}$ be the maximum tolerable accumulated debt for a given deployment. Then:

$$\mathrm{DMR}(t) = D(t)/D_{\max} \tag{3}$$

DMR is dimensionless and monotonically non-decreasing. In discrete implementation: $D_k = D_{k-1} + \omega_k \cdot e_k \cdot \Delta t$, $\mathrm{DMR}_k = D_k/D_{\max}$.

## III. Finite-Time Exhaustion Bound

**Proposition 1.** *If $e(t) \geq \varepsilon > 0$ and $\omega(t) \geq \omega_{\min} > 0$ for all $t$, then for any finite $D_{\max}$:*

$$T^* \leq D_{\max}/(\omega_{\min} \cdot \varepsilon) \tag{4}$$

*where $T^*$ is the time at which $\mathrm{DMR}(T^*) = 1$.*

*Proof.* $D(t) \geq \omega_{\min} \cdot \varepsilon \cdot t$. Setting $D(T^*) = D_{\max}$ gives the bound. □

Proposition 1 states that any persistent nonzero perception error will eventually consume any finite safety margin. The expiration time $T^*$ depends on the deployment context through $\omega_{\min}$ and $D_{\max}$. Two systems with identical per-frame error but different deployment contexts can have very different $T^*$ values. Snapshot metrics cannot distinguish between them.

## IV. Case Study 1: Controlled Sensitivity

We generate a shared synthetic error signal over 120 seconds at $\Delta t = 0.1$ s (1,200 samples): $e(t) = 0.015 + 0.005\sin(0.12t) + \eta(t)$ where $\eta(t) \sim \mathcal{N}(0, 0.002^2)$. The error stays in $[0.8, 2.6]$ cm, within typical per-frame accuracy specifications.

We compute DMR under two deployment assumptions with $D_{\max} = 1.0$:

- **Open room:** $\|J_P\| = 1.0$, $K_c = 1.0$, $\omega_e = 1.0$, so $\omega = 1.0$.
- **Narrow corridor:** $\|J_P\| = 2.5$, $K_c = 1.0$, $\omega_e = 2.0$, so $\omega = 5.0$.

**Results.** Figure 1 (top) confirms both conditions share the same error trace with mean 1.5 cm. Under the open-room assumption, DMR reaches the violation threshold of 1.0 at $t = 64.5$ s. Under the corridor assumption, the same error trace drives DMR to 1.0 at $t = 11.3$ s, a **5.7× acceleration** in safety-margin exhaustion. A snapshot benchmark would report identical accuracy in both cases.

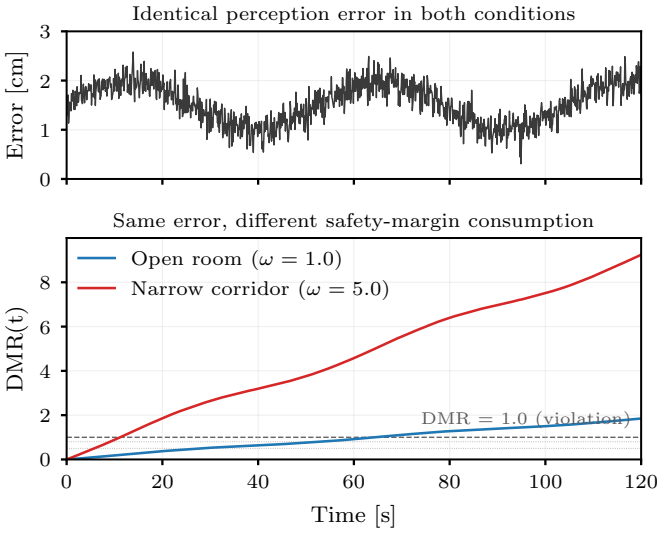


Fig. 1. Controlled study: identical error (top) produces divergent DMR trajectories (bottom). The corridor exhausts its safety budget 5.7× faster than the open room.

## V. Case Study 2: PTZ Perturbation Study on Temporal Detection Outputs

**Setup.** We evaluate DMR on a 90-second object detection sequence (343 frames, approximately 5 fps after initial warmup) captured by a YOLOv5n [11] detector running on a Yahboom ROSMASTER M1 robot (Jetson Nano B01) observing a static indoor scene at 1920 × 1080 resolution. The robot remains stationary throughout the sequence so that any observed change in the detection signal is attributable to the camera and the detector, not to platform motion. Four household objects placed at fixed positions in the scene serve as persistent scene anchors: a bottle, a shoe, a remote, and a book.

**Anchors and matching.** For each anchor object we define (i) a set of *allowed labels* that the detector might assign to it, and (ii) per-phase *reference centers* that specify where the anchor is expected to appear in the image. We use a label set rather than a single class because the detector exhibits label flicker: for example, the bottle is classified across frames as "fire hydrant," "bottle," or "vase" (Table I). For each frame, every detection whose predicted label belongs to an anchor's allowed set is considered a candidate match; among candidates, we assign the one closest in pixel space to that anchor's current reference center, with a maximum association distance of 300 pixels to suppress spurious matches. Anchors with no qualifying detection in a given frame are treated as missed.

**Perturbation protocol.** The 90-second sequence is divided into three phases:

- **Phase A** (0–30 s, stable viewpoint): the camera is held at its initial pose; reference centers are taken as the median detected position for each anchor over this window.
- **Perturbation** (30–60 s): a mild PTZ camera nudge is applied. The camera's pan/tilt mount is manually displaced by a small amount (a few degrees) and allowed to settle, simulating an inadvertent viewpoint disturbance such as a bumped mount or a low-amplitude vibration event.
- **Phase B** (60–90 s, new stable viewpoint): the camera holds the post-perturbation pose; new reference centers are computed by the same median procedure over this window.

Because the true anchor positions in image coordinates change between Phase A and Phase B, a single global reference would conflate genuine viewpoint shift with detector error. Phase-specific references isolate the latter. Within the 30–60 s perturbation window, we use Phase A references for frames before $t = 45$ s and Phase B references thereafter; this midpoint switch approximates the unknown moment at which the camera transitions from its old equilibrium to its new one, and any residual deviation from the (unknown) true geometry shows up as accumulated debt.

**Error computation.** For each frame, we compute the Euclidean pixel distance between each anchor's matched

TABLE I
PER-ANCHOR DETECTION RATES OVER 343 FRAMES.

| Anchor | Allowed labels | Det. rate | Primary error |
|---|---|---|---|
| Bottle | fire hydrant, bottle, vase | 97% | label flicker |
| Shoe | bird, skateboard, cat | 64% | misclassification |
| Remote | remote | 17% | frequent misses |
| Book | book | 0.3% | near-total miss |

detection center and its phase-appropriate reference center. Undetected anchors are assigned a fixed miss penalty of 150 pixels, chosen to be comparable in magnitude to a typical bounding-box width at this resolution so that a missed detection contributes error on the same order as a moderately misplaced one (rather than dominating the signal or being negligible). The per-frame error is the mean over the four anchors, normalized by image width (1920 px) so that error is expressed as a dimensionless fraction. When integrating debt, inter-frame time steps are capped at 1.0 s to prevent rare startup gaps during detector warmup (the YOLOv5n model takes several seconds to reach steady-state framerate on the Jetson Nano) from inflating $D(t)$ via spuriously large $\Delta t$ values.

Table I shows per-anchor detection rates. The bottle is detected in 97% of frames but under three different class labels. The shoe is detected 64% of the time, always misclassified. The remote is detected in only 17% of frames. The book is almost never detected. This mix of label instability, spatial jitter, and missed detections is representative of real detector behavior on edge hardware.

**DMR computation.** We set $D_{\max} = 3.0$ based on the following reasoning: at the observed mean normalized error of $\tilde{e} = 0.051$ and the lower sensitivity $\omega = 1.0$, this budget corresponds to approximately 60 seconds of continuous operation before exhaustion, consistent with a typical short-horizon pick-and-place cycle. Our claim is comparative rather than absolute: identical detector outputs can imply materially different safety-budget consumption under different deployment criticality assumptions. We evaluate under $\omega = 1.0$ (open layout) and $\omega = 5.0$ (tight layout).

**Results.** Figure 2 shows the anchor-based normalized detection error (top) and DMR trajectories (bottom). Under the open-layout assumption, DMR crosses 1.0 at $t = 77.9$ s. Under the tight-layout assumption, DMR crosses 1.0 at $t = 33.4$ s, a **2.3× acceleration** in safety-margin exhaustion.

Error remains elevated after the perturbation because the camera settles into a new viewpoint: detections stabilize but around shifted reference geometry, producing persistent anchor-wise deviation that continues to accumulate debt. In a deployed system, this means a robot that recovers perceptually (detections stabilize, per-frame accuracy returns to nominal) is still accumulating safety-margin debt from the viewpoint shift, invisible to any snapshot metric.

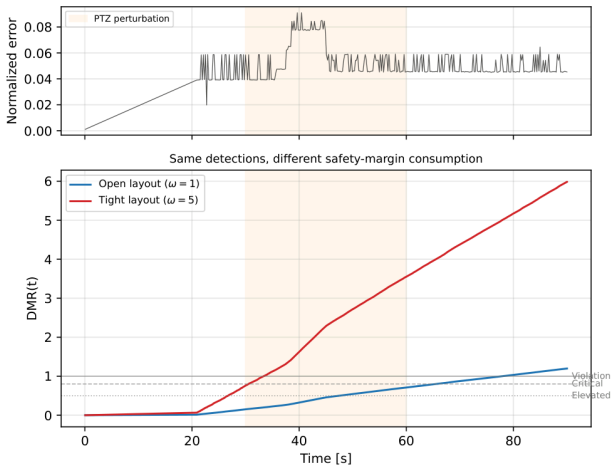


Fig. 2. PTZ perturbation study on real hardware: normalized detection error (top) and DMR (bottom). The tight layout exhausts its safety budget 2.3× faster.

TABLE II
SUMMARY: WITHIN EACH EXPERIMENT, THE SAME UNDERLYING ERROR TRACE IS EVALUATED UNDER TWO DEPLOYMENT CRITICALITY SETTINGS. MEAN NORMALIZED ERROR THEREFORE REMAINS UNCHANGED WITHIN EACH EXPERIMENT PAIR, WHILE DMR THRESHOLD CROSSING TIMES DIFFER BECAUSE $\omega$ IS HIGHER IN THE CONSTRAINED CONDITION.

| Condition | Mean $\tilde{e}$ | $\omega$ | $t_{0.5}$ | $t_{1.0}$ | Ratio |
|---|---|---|---|---|---|
| Exp. 1 Open room | 0.015 | 1.0 | 28.0 s | 64.5 s | – |
| Exp. 1 Corridor | 0.015 | 5.0 | 6.0 s | 11.3 s | 5.7× |
| Exp. 2 Open layout | 0.051 | 1.0 | 47.5 s | 77.9 s | – |
| Exp. 2 Tight layout | 0.051 | 5.0 | 26.8 s | 33.4 s | 2.3× |

## VI. DISCUSSION AND LIMITATIONS

**What the case studies show.** Both studies demonstrate a single point: identical snapshot perception accuracy can imply sharply different accumulated safety consumption depending on deployment context. This is a blind spot in current robot perception evaluation. DMR makes it visible. More broadly, DMR offers a practical tool for studying robustness, failure analysis, and evaluation blind spots in long-horizon robot perception.

**Relation to existing work.** DMR is complementary to, not competitive with, existing metrics. Per-frame metrics (mAP [14]) evaluate perception accuracy. Robustness benchmarks [9] evaluate perception under distribution shift but at the per-frame level. Bayesian state estimation [18] and SLAM covariance propagation [4], [5] track uncertainty but report it as a confidence bound, not as consumed safety budget. ATE drift rate (cm/s) captures localization degradation but is not weighted by deployment criticality: the same drift rate has different safety implications in an open warehouse aisle versus a narrow inter-rack gap. Multi-object tracking metrics (CLEAR MOT [2], HOTA [13]) evaluate identity consistency but do not connect detection instability to downstream safety-margin consumption. Control barrier functions [1] enforce safety at the control layer. Runtime safety monitors [7] check whether the current state satisfies invariants but do not track

whether accumulated perception drift has corrupted the estimate feeding those checks. Foundation models that fuse vision, language, and action [8], [12] have made per-frame perception dramatically more capable, yet none of these systems monitor whether their temporal error integral remains within budget. DMR operates between perception and control: it tracks whether the accumulated error in the perception estimate has degraded the input quality that downstream modules depend on.

**Limitations.** We do not claim universal safety-budget or sensitivity values. The thresholds and sensitivity parameters used here are illustrative, chosen to demonstrate the evaluation blind spot under explicit and interpretable assumptions. In Case Study 2, the error signal represents anchor-based temporal detection inconsistency relative to phase-specific reference centers, not geometric ground truth. Calibrating $\omega(t)$ and $D_{\max}$ for a specific deployment requires domain knowledge of clearances, controller gains, and acceptable risk. Hardware validation on mobile platforms with geometric ground truth is the subject of companion work.

An open-source implementation of DMR computation, including both case studies, is available at https://github.com/stavanio/perception-debt.

## REFERENCES


[1] A.D. Ames *et al.*, "Control barrier functions: Theory and applications," in *European Control Conf.*, 2019.

[2] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, 2008.

[3] A. Brohan *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Conf. Robot Learning*, 2023.

[4] C. Cadena *et al.*, "Past, present, and future of SLAM," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[5] C. Campos *et al.*, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

[6] W. Cunningham, "The WyCash portfolio management system," in *OOPSLA Experience Report*, 1992.

[7] A. Desai, T. Dreossi, and S.A. Seshia, "Combining model checking and runtime verification for safe robotics," in *Intl. Conf. Runtime Verification*, 2017.

[8] D. Driess *et al.*, "PaLM-E: An embodied multimodal language model," in *ICML*, 2023.

[9] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *ICLR*, 2019.

[10] ISO/TS 15066:2016, "Robots and robotic devices: Collaborative robots," 2016.

[11] G. Jocher *et al.*, "ultralytics/yolov5: v7.0," Zenodo, 2022. doi:10.5281/zenodo.7347926.

[12] A. Kirillov *et al.*, "Segment Anything," in *ICCV*, 2023.

[13] J. Luiten *et al.*, "HOTA: A higher order metric for evaluating multi-object tracking," *Intl. J. Comput. Vis.*, vol. 129, pp. 548–578, 2021.

[14] T.Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.

[15] S. Liu *et al.*, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," arXiv:2303.05499, 2023.

[16] Open X-Embodiment Collaboration, "Open X-Embodiment: Robotic learning datasets and RT-X models," in *ICRA*, 2024.

[17] N. Ravi *et al.*, "SAM 2: Segment anything in images and videos," arXiv:2408.00714, 2024.

[18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2005.