Online Adaptive Learning for Robust LiDAR Perception in High-Performance Autonomous Racing

Raphael Blanchard[1], Haeyoon Han[1], Thomas A. Berrueta[1], Soon-Jo Chung[1]

***Abstract*— Deploying robust perception systems in the field is inherently difficult due to the highly variable geometric structure of real-world environments. Such variability introduces a strict operational trade-off for deployed perception pipelines: configurations robust enough to handle severe perceptual degradation are often computationally expensive for real-time applications, while static perception pipelines frequently fail to maintain the necessary balance, resulting in a single point of failure. In this work, we bridge the gap between reactive estimation and offline hyperparameter optimization by deriving an online adaptive learning framework for adaptive perception. We present a novel real-time metric of point cloud degeneracy that enables an online Soft Actor-Critic (SAC) policy to continuously learn from real-time observations. This allows the system to adapt to previously unseen environments by proactively adjusting perception hyperparameters on the fly. We validate our approach on an autonomous IndyCar at speeds ranging from 45 mph to 120 mph. Our results demonstrate that online parameter adaptation reduces operational failures by 84.6% compared to static-hyperparameter baselines, improving robustness against environmental degeneracies without the computational burden of static, high-fidelity configurations.**

I. Introduction

AUTONOMOUS robots are increasingly becoming ubiquitous in real-world field operations such as industrial inspection [1], search and rescue [2], and autonomous driving [3]. While reliable operation in these domains is critical, field deployments remain fundamentally challenging because environments are inherently uncertain, diverse, and continuously changing. A quadruped navigating natural landscapes must handle abrupt shifts in terrain [4], just as autonomous vehicles must contend with evolving scene geometry [5], weather [6], and unpredictable road conditions [3]. Because these environmental factors are highly variable and rarely known *a priori*, static system configurations frequently fail. Consequently, field deployments demand robots capable of online adaptation. By continuously estimating changes in the environment and adjusting their perception, planning, or control pipelines accordingly [7], adaptive systems can bridge the gap between static design and real-world variability to ensure robust performance.

While control [8], [9] and motion planning [10], [11] modules have become increasingly adept at online adaptation, perception systems have historically relied on pre-deployment tuning [12], [13]. A primary barrier to adapting perception pipelines on the fly is the severe trade-off between robustness and computational efficiency. In LiDAR-based pose estimation, navigating environments with weak geometric structure often requires retaining more measurements, expanding the search for scan-to-map correspondences, and allocating greater optimization effort [14], [15]. Although these measures improve robustness against failure, they incur a computational burden that is unsustainable during routine, nominal operation. The core challenge is dynamically allocating computational effort only when environmental conditions demand it.

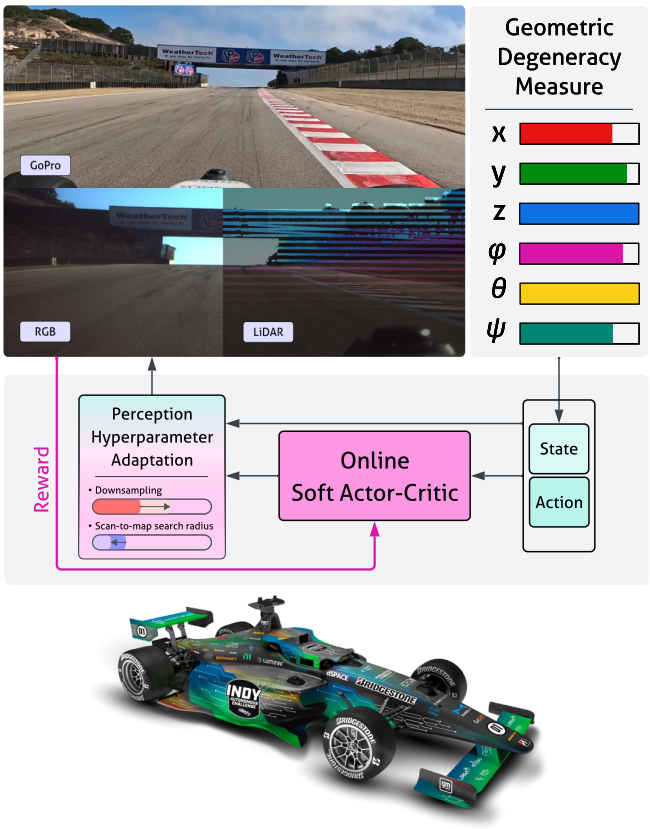

Fig. 1: Our perception hyperparameter adaptation framework measures point cloud degeneracy and tunes front-end perception parameters using an online Soft Actor–Critic policy. This framework improves the robustness of the perception system under geometric degeneracy while balancing computational cost.

Prior work on Iterative-Closest-Point (ICP) uncertainty and dynamic covariance modeling has made progress in improving estimator consistency and handling ill-conditioned updates in LiDAR-based perception [16]–[20]. While these

This work was supported by the DSTA under contract number #DST000EC124000205.

[1]Authors are with the Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA. {rblanch2, hhan3, berrueta, sjchung} @caltech.edu

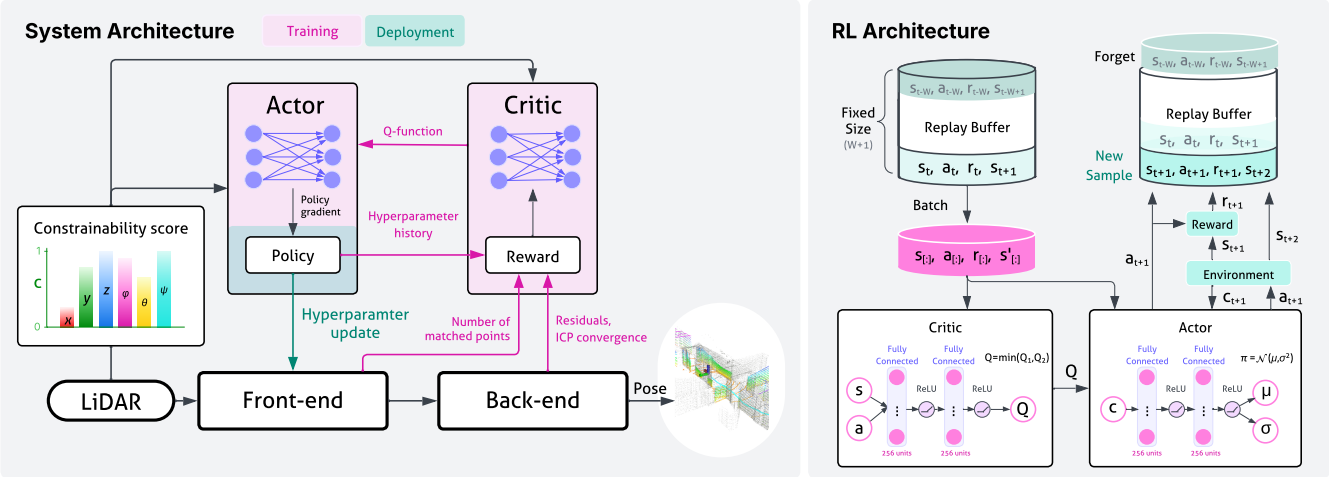

Fig. 2: **Overview of the proposed online adaptive hyperparameter adaptation framework.** Left: The adaptation policy receives the constrainability score and outputs front-end hyperparameters for downsampling, feature detection, and feature matching, which influence downstream back-end performance, including ICP convergence and feature alignment. Right: The policy is trained using off-policy SAC with a replay buffer and lightweight MLP-based actor and twin-critic networks.

methods quantify uncertainty from scan geometry and adjust estimator trust, they are inherently reactive. Because they rely on information extracted only after correspondences are formed and optimization is underway [21], they lack the proactive capability to adapt front-end runtime behavior before those degraded updates jeopardize high-risk field deployments.

To proactively adjust the front-end performance of perception pipelines, a common approach is hyperparameter optimization, where multiple configurations are evaluated using accuracy-based metrics [22], [23]. While effective for offline tuning on curated datasets, these methods assume stable environments and access to ground-truth scoring signals—none of which are available during live deployments. Even when parameter adaptation is learned [24], it typically relies on offline training with privileged signals, leaving the system vulnerable to domain gaps. While an offline-trained policy can adjust parameters on the fly, it remains restricted by its prior training distribution. These limitations underscore the critical need for an online adaptive learning formulation capable of continuously adapting perception parameters during deployment in previously unseen environments.

In this work, we present an online learning framework that continuously adapts LiDAR perception during deployment. Central to our approach is the constrainability score, a novel real-time measure of scan degeneracy. This score informs a Soft Actor-Critic (SAC) Reinforcement Learning (RL) policy that learns from incoming data while dynamically adjusting runtime parameters across the perception pipeline. This simultaneous learning and adaptation allows the robot to balance the competing demands of maintaining robustness in degenerate environments and minimizing computational overhead during nominal operation.

The primary contributions of this manuscript are:

- **A directional metric for geometric degeneracy.** We propose the constrainability score, a 6-degree-of-freedom measure of geometric degeneracy derived from the sensitivities of point-to-plane and point-to-point ICP residuals to pose perturbations.
- **An online adaptive learning framework for hyperparameter adaptation.** We develop an online SAC framework that learns directly from live deployment data while simultaneously adjusting LiDAR perception hyperparameters.
- **Extensive validations across field deployments.** We validate the proposed method on an autonomous IndyCar under geometric degenerate environments, demonstrating improved robustness and favorable robustness/compute trade-offs relative to fixed-parameter baselines.

II. Method

To develop perception pipelines that are adaptive to dynamic environments, we require a means of estimating geometric degeneracies in real-time, as well as an adaptation mechanism for optimizing hyperparameters. In this section, we introduce a novel measure of geometric degeneracy for a given LiDAR scan and an online RL framework for adapting our perception pipeline. Figure 2 illustrates the high-level framework.

A. Environmental Constrainability Estimation

To quantify geometric degeneracy with low computational overhead, we define a constrainability score $\mathbf{c} \in [0,1]^6$ whose elements correspond to the translational and rotational components of the six-degree-of-freedom pose. The score is derived from the sensitivity of both point-to-point and point-to-plane ICP residuals to pose perturbations as follows.

Let $(\mathbf{p}_i, \mathbf{q}_i, \mathbf{n}_i) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3,\ i = 1, \cdots, N$ denote a correspondence. The point-to-plane and the point-to-point ICP residuals are defined as follows:

$$r_i^{\text{p2pl}} = \mathbf{n}_i^\top (R(\boldsymbol{\theta})\,\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i) \in \mathbb{R},$$

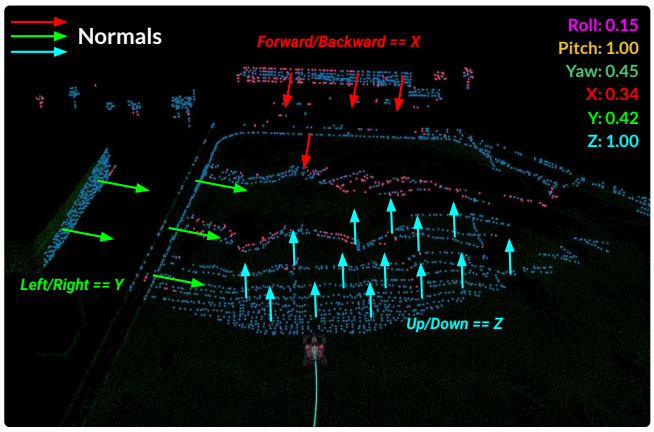

Fig. 3: **Constrainability score illustration on the Las Vegas Motor Speedway, Las Vegas, NV.** Arrows correspond to normals of LiDAR points. The current environment contains a high distribution of ground points, which constrain mostly the Z axis. However, there are fewer points on surfaces with normals in X and Y, which would cause ICP to be underconstrained in those directions.

$$r_i^{\text{p2p}} \;=\; R(\boldsymbol{\theta})\,\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i \in \mathbb{R}^3,$$

where $\boldsymbol{\theta}$ is the attitude of the robot, $R(\boldsymbol{\theta}) \simeq \mathbf{I}_3 + [\boldsymbol{\theta}]_\times$ for small $\boldsymbol{\theta}$, where $[\mathbf{a}]_\times$ is the skew-symmetric matrix representing the cross product (i.e., $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$, for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$), and $\boldsymbol{t}$ is position. The Jacobian at $\boldsymbol{\theta} = 0$ is given by

$$\mathbf{J}_i^{\text{p2pl}} \;=\; \mathbf{n}_i^\top \begin{bmatrix} -[\mathbf{p}_i]_\times & \mathbf{I}_3 \end{bmatrix},$$

$$\mathbf{J}_i^{\text{p2p}} \;=\; \begin{bmatrix} -[\mathbf{p}_i]_\times & \mathbf{I}_3 \end{bmatrix},$$

yielding the normal equations for the optimal pose increment $\boldsymbol{\xi}^* = [\boldsymbol{\theta}^{*\top} \;\; \boldsymbol{t}^{*\top}]^\top \in \mathbb{R}^6$:

$$\mathbf{H}\boldsymbol{\xi}^* \;=\; -\left(\sum_{i=1}^{N_{pl}} \mathbf{J}_i^{\text{p2pl}\top} r_i^{\text{p2pl}} + \sum_{i=1}^{N_p} \mathbf{J}_i^{\text{p2p}\top} r_i^{\text{p2p}} \right), \qquad (1)$$

$$\mathbf{H} = \sum_{i=1}^{N_{pl}} \mathbf{J}_i^{\text{p2pl}\top} \mathbf{J}_i^{\text{p2pl}} + \sum_{i=1}^{N_p} \mathbf{J}_i^{\text{p2p}\top} \mathbf{J}_i^{\text{p2p}}.$$

Although full condition-number evaluation of the normal matrix is computationally prohibitive for real-time systems, assessing constrainability through its diagonal components serves as a lightweight proxy, effectively capturing the strength of the geometric constraints. The diagonal component corresponding to the point-to-plane ICP residuals evaluates to:

$$C^{\text{p2pl}} = \sum_{i=1}^{N_{pl}} \begin{bmatrix} (\mathbf{p}_i \times \mathbf{n}_i) \odot (\mathbf{p}_i \times \mathbf{n}_i) \\ \mathbf{n}_i \odot \mathbf{n}_i \end{bmatrix},$$

Similarly, the effect of point-to-point ICP residuals is given by:

$$C^{\text{p2p}} = \sum_{i=1}^{N_p} \begin{bmatrix} (\mathbf{1}\mathbf{1}^\top - \mathbf{I})\,(\mathbf{p}_i \odot \mathbf{p}_i) \\ \mathbf{1} \end{bmatrix}.$$

To incorporate the sensitivity of both residuals, we define the combined measure $C^{\text{tot}} = C^{\text{p2pl}} + C^{\text{p2p}} \in \mathbb{R}^6$, where a low value indicates geometric degeneracy. To obtain a scale-invariant and bounded measure, each element is normalized to lie in $[0, 1]$:

$$\mathbf{c} = \begin{bmatrix} (\mathbf{c}^\theta)^\top & (\mathbf{c}^t)^\top \end{bmatrix}^\top, \qquad (2)$$

$$\mathbf{c}^\theta = \frac{C^{\text{tot},\theta}}{\|C^{\text{tot},\theta}\|_\infty}, \quad \mathbf{c}^t = \frac{C^{\text{tot},t}}{\|C^{\text{tot},t}\|_\infty}. \qquad (3)$$

The scores are computed during feature extraction, before ICP optimization, so that the parameters can adapt in a timely manner.

TABLE I: Representative LiDAR perception parameters adapted online.

Parameter	Role
Front-end Parameters	
Edge feature threshold	Accept sharper edge points
Surface/planarity threshold	Accept flatter planar points
Front-end voxel size	Downsample incoming scan
Mapping update interval	Set scan-to-map update rate
Keyframe spacing (distance)	Subsample keyframes
Local map search radius	Max neighbor range for ICP

B. On-The-Fly Hyperparameter Adaptation

We cast hyperparameter adaptation as an online adaptive learning problem formulated as a Markov Decision Process (MDP), $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s' \in \mathcal{S} \mid s \in \mathcal{S}, a \in \mathcal{A})$ is the transition model describing the probability density of the next state s' conditioned on the current state s and action a, $r(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. We define the state space $\mathcal{S}$ as the Cartesian product of the estimated robot pose and the constrainability score. Augmenting the pose with $\mathbf{c} \in [0, 1]^6$ ensures that the state $s \in \mathcal{S}$ provides a real-time representation of the scene's geometric degeneracy. The action space $\mathcal{A}$ is defined as the continuous set of incremental updates applied to preprocessing and front-end hyperparameters. These updates are bounded and applied incrementally, preventing abrupt changes that could destabilize the pipeline.

Since perception hyperparameters dictate how raw sensor data is processed, they directly control the trade-off between robustness and computational efficiency. Downsampling parameters, for instance, determine this balance: larger voxel sizes improve efficiency, whereas smaller voxel sizes preserve more geometric detail for residual minimization. Under our online RL paradigm, where ground-truth poses are not consistently available, the reward must act as a multi-objective proxy. To encourage accurate pose estimation, we define a residual-based reward:

$$r_{\text{res}} = \rho_{t-1}^{\text{surf}} - \rho_t^{\text{surf}},$$

where ρ_t^{surf} is the aggregated ICP residual at time t. To balance this with computational cost, we introduce a penalty

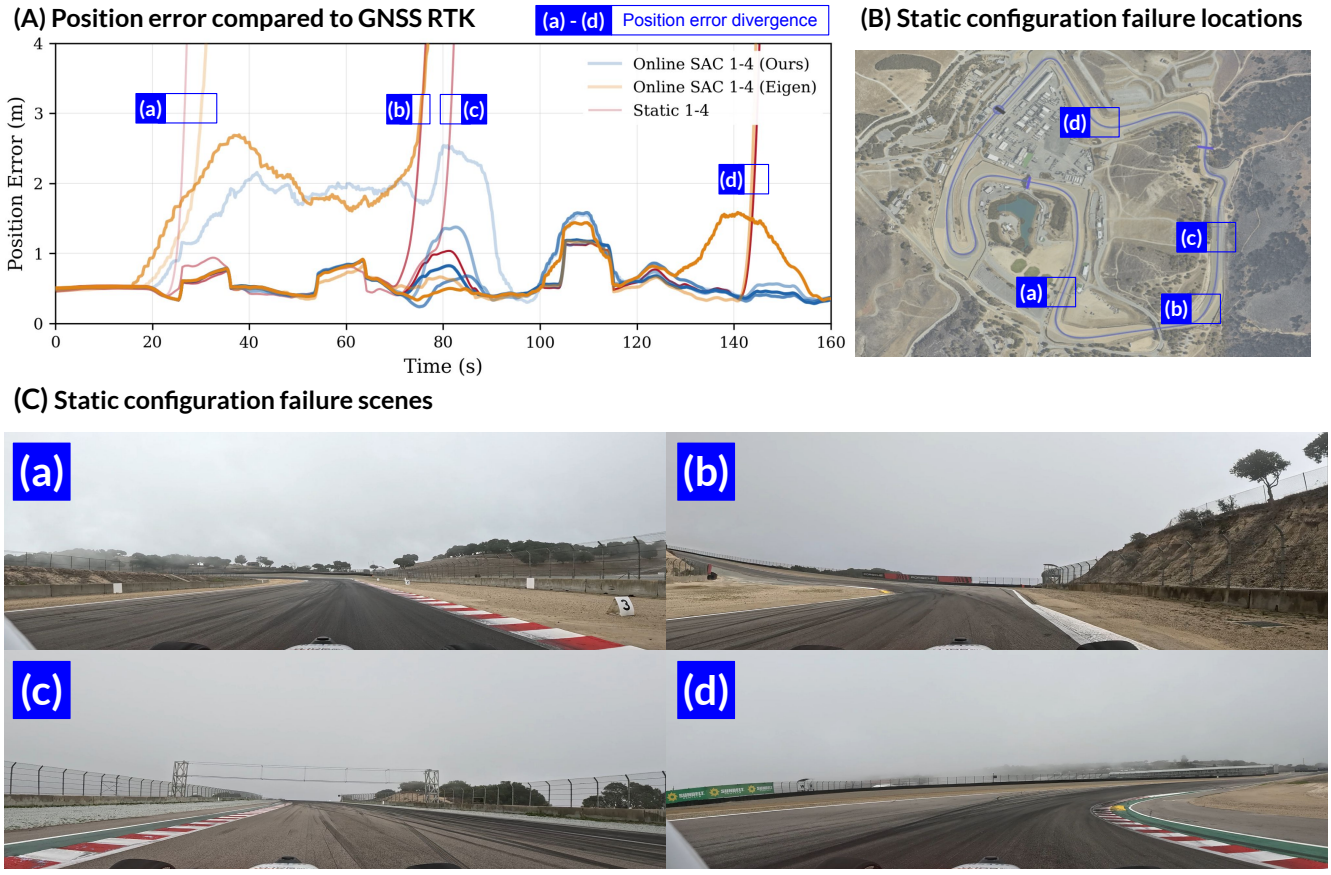

Fig. 4: **Performance comparison of our approach.** (A) Position error divergence locations for static configurations and the baseline online policy with ICP eigenvalue-based context, shown in (a)-(d). Our policy avoids divergence by producing more reliable configurations. (B) Satellite image of the WeatherTech Raceway Laguna Seca and highlighted locations where static configurations failed. (C) Onboard footage from the autonomous racecar highlighting open fields with low geometric features.

proportional to the ICP convergence time:

$$r_{\text{ICP}} = 0.005\left(20 - \max(T_t^{\text{ICP}}, 0)\right),$$

where T_t^{ICP} is the computation time in milliseconds. Next, to prevent the policy from achieving low compute times by over-downsampling, we enforce feature sufficiency via a high ratio of matched points:

$$r_{\text{ratio}} = \begin{cases} \dfrac{n_t^{\text{match}}}{n_t^{\text{down}}}, & \text{if } n_t^{\text{down}} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

To strictly discourage operating with an unsafely low number of matches, we apply a threshold penalty:

$$p_{\text{low}} = \begin{cases} P_{\text{low}}, & \text{if } n_t^{\text{match}} < N_{\text{low}}, \\ 0, & \text{otherwise,} \end{cases}$$

where N_{low} is the minimum acceptable number of correspondences. Lastly, we explicitly encourage the policy to select configurations that improve constrainability:

$$r_{\text{const}} = \frac{1}{6}\sum_{i=1}^{6}(c_{t,i} - c_{t-1,i}).$$

The total reward is then given by the bounded sum of these sub-objectives:

$$r_t = \text{clip}(w_{\text{res}}\, r_{\text{res}} + r_{\text{ICP}} + r_{\text{ratio}} - p_{\text{low}} + r_{\text{const}}, -1, 1),$$

where $\text{clip}(x, \underline{m}, \bar{m}) := \min(\max(x, \underline{m}), \bar{m})$ and w_{res} weights the residual improvement term.

Implementation details. The online RL algorithm employed in this paper is SAC [25] with automated temperature tuning. To ensure safe adaptation, we constrain the action space to parameter ranges that avoid parameter configurations known to cause catastrophic failures (e.g., excessively low feature extraction thresholds or overly aggressive downsampling). This allows SAC to preserve the flexibility of continuous hyperparameter adaptation while restricting the policy to a physically meaningful and operationally safe region of the parameter space. Additionally, we apply spectral normalization to the actor and critic networks, constraining layer-wise operator norms and yielding a tighter Lipschitz bound. In our online setting, this reduces sensitivity to input perturbations and improves adaptation stability.

Both the actor and the critic are implemented as 2-hidden-layer perceptrons with 256 units, making the framework lightweight. The actor maps the current constrainability score to a stochastic continuous action distribution representing

incremental hyperparameter updates. During deployment, the actor receives the current constrainability score and outputs incremental hyperparameter updates, which are regulated to a safe admissible range before being applied to the perception stack. This incremental formulation encourages smooth adaptation. The critic is not used directly for action generation but it operates as part of the off-policy learning process to improve the actor. In the current implementation, parameter updates are produced online while learning proceeds asynchronously at a slower optimization rate, enabling real-time adaptation without interrupting perception.

The reward weights and penalty constants were tuned to balance robustness and computational efficiency under real-time deployment constraints. In practice, increasing the weighting of residual minimization terms produced marginally lower localization error but encouraged computationally heavier configurations that increased onboard resource usage. Conversely, emphasizing compute-efficiency penalties reduced runtime overhead at the expense of more frequent localization degradation in geometrically degenerate environments. The selected reward configuration was chosen to achieve a stable trade-off between operational robustness and computational cost on shared onboard hardware.

III. Experiments on Autonomous IndyCar

We evaluate the proposed online adaptive parameterization approach by deploying it on an autonomous IndyCar and applying it to the LiDAR localization subsystem. We compare our online RL-based approach against static-parameter baselines fine-tuned with respect to the initial operating condition. We evaluate robustness using the squared ℓ_2 norm position error $e_{\text{pos}} = \|\mathbf{t}_{\text{RTK}} - \hat{\mathbf{t}}_{\text{LiDAR}}\|_2^2$, considering a run as failed when localization error exceeds 3 m.

Experiment details. Evaluation was conducted at the *WeatherTech Raceway Laguna Seca*. The platform is equipped with a Luminar Iris LiDAR operating at a 10 Hz scan rate and RTK GPS. In the online RL pipeline, parameter updates are performed at the same rate as the LiDAR scan, whereas gradient updates are performed at 100 Hz. We use a 5-second warm start while the car is stationary to expose the agent to a fixed local scene, allowing it to improve from early policy-driven interactions. To remain compatible with a compute-aware autonomy stack, localization inference is kept on the CPU while SAC is trained on the GPU.

Operational robustness. The first experiment compares localization error over one lap. Figure 4 (A) shows the error comparison between the static-parameter baseline and our online approach. The static baseline fails in different sections of the track (Fig. 4 B), demonstrating it is unsuitable for dynamically evolving scenes. Furthermore, Online SAC using ICP eigenvalue-based context ($\mathbf{H} = \mathbf{J}^\top\mathbf{J}$) fails to adapt safely; raw eigenvalues lose interpretability under degeneracy, whereas our normalized constrainability scores provide actionable information. Figure 4 (C) shows failures occur in flat areas with fewer geometric cues, whereas our approach's adaptation remains effective. Figure 5 showcases how reduced geometric information in a straight line

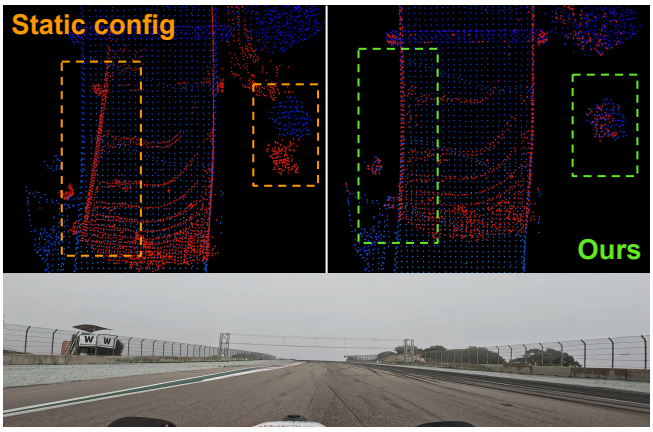

Fig. 5: **Real-time scan alignment on the Rahal straight of the WeatherTech Raceway Laguna Seca.** This segment contains low geometric features, causing static configurations to fail. The SAC-outputted configuration allows the system to remain operational.

triggers system failure for static configurations. Leveraging the constrainability scores, the SAC-outputted configuration allows the system to remain operational in previously unseen environments. This result highlights that, when the robot is deployed in a previously unseen type of environment, a static configuration may lead not only to performance degradation but also to system failure in the worst case, whereas online adaptations maintain operation by actively adapting the stack's hyperparameters.

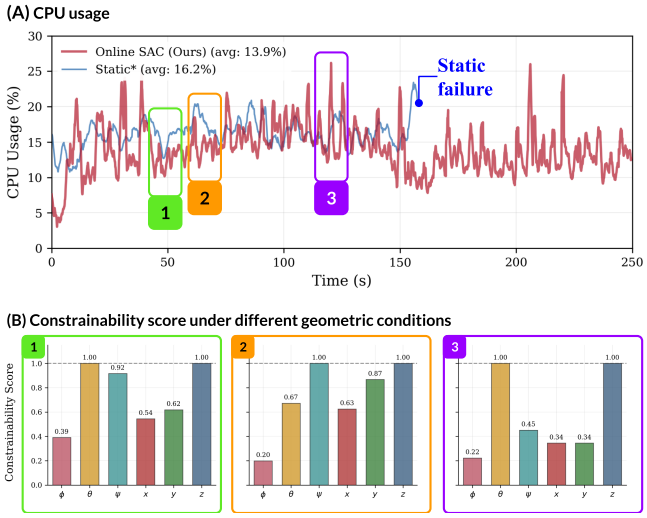

Fig. 6: **CPU performance in various environments.** (A) CPU usage recorded throughout deployment. $Static^*$ corresponds to the best performing static configuration. (B) Constrainability scores of various environments.

CPU usage on shared onboard hardware. Figure 6 illustrates CPU usage in environments with different geometric constraints. A conservative static configuration keeps localization stable but at a higher CPU cost. Our SAC-based approach allocates higher CPU only in hard segments where robustness is required, then reduces CPU in easier segments. This leads to lower average compute usage without

sacrificing operational robustness. This supports the system-level claim of this paper: online adaptation of perception parameters improves operational robustness while freeing compute budget to other autonomy systems on the same onboard platform.

Deploying our approach on 20 laps executed at speeds ranging from **45 to 120 mph**, static configurations caused the perception system to fail in 13/20 laps. Our approach failed in only **2 runs**. Through online adaptation of hyperparameters, our approach reduces the failure rate by **84.6%** compared to static configurations.

IV. Conclusion

In this paper, we presented an online adaptive learning framework for proactive parameter adaptation in field perception systems. By proposing the constrainability score as a real-time measure of geometric degeneracy, we enabled a SAC policy to continuously learn and adjust LiDAR perception hyperparameters on the fly without prior training data. The constrainability score provides a directional representation of geometric information in LiDAR scans, allowing the system to adjust perception behavior accordingly. Through extensive experiments on an autonomous IndyCar operating at speeds up to 120 mph, we showed that our online adaptation framework significantly improves operational robustness while maintaining efficient use of computational resources. These results demonstrate that concurrent learning and adaptation can effectively break the static trade-off between robustness and efficiency in unpredictable environments. Future work will extend this framework to additional perception modalities.

References

[1] R. A. Clark *et al.*, "Autonomous and scalable control for remote inspection with multiple aerial vehicles," *Robotics and Autonomous Systems*, vol. 87, pp. 258–268, 2017.

[2] B. Lindqvist *et al.*, "Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue," *Robotics and Autonomous Systems*, vol. 154, p. 104134, 2022.

[3] K. Cordes, C. Reinders, P. Hindricks, J. Lammers, B. Rosenhahn, and H. Broszio, "Roadsaw: A large-scale dataset for camera-based road surface and wetness estimation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 4439–4448.

[4] L. Wellhausen and M. Hutter, "Rough terrain navigation for legged robots using reachability planning and template learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6914–6921.

[5] S. Zuo, W. Zheng, Y. Huang, J. Zhou, and J. Lu, "Gaussianworld: Gaussian world model for streaming 3d occupancy prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 6772–6781.

[6] M. Hahner *et al.*, "Lidar snowfall simulation for robust 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 364–16 374.

[7] H. Azpúrua *et al.*, "A survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments," *Robotics and Autonomous Systems*, vol. 160, p. 104304, 2023.

[8] E. S. Lupu, F. Xie, J. A. Preiss, J. Alindogan, M. Anderson, and S.-J. Chung, "Magicvfm-meta-learning adaptation for ground interaction control with visual foundation models," *IEEE Transactions on Robotics*, vol. 41, pp. 180–199, 2025.

[9] Q. Yao *et al.*, "Adaptive legged manipulation: Versatile disturbance predictive control for quadruped robots with robotic arms," *Robotics and Autonomous Systems*, vol. 167, p. 104468, 2023.

[10] A. Patel *et al.*, "Towards field deployment of mavs in adaptive exploration of gps-denied subterranean environments," *Robotics and Autonomous Systems*, vol. 176, p. 104663, 2024.

[11] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, 2018.

[12] X. Wu, Z. Wu, H. Guo, L. Ju, and S. Wang, "Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 769–15 778.

[13] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[14] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[15] H. Teng, Y. Wang, D. Chatziparaschis, and K. Karydis, "Adaptive lidar odometry and mapping for autonomous agricultural mobile robots in unmanned farms," *Computers and Electronics in Agriculture*, vol. 232, p. 110023, 2025.

[16] D. Landry, F. Pomerleau, and P. Giguère, "Cello-3d: Estimating the covariance of icp in the real world," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8190–8196.

[17] A. De Maio and S. Lacroix, "Deep bayesian icp covariance estimation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6519–6525.

[18] W. Talbot *et al.*, "Principled icp covariance modelling in perceptually degraded environments for the eels mission concept," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 10 763–10 770.

[19] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 809–816.

[20] S. Zhao *et al.*, "Superloc: The key to robust lidar-inertial localization lies in predicting alignment risks," 2025. [Online]. Available: https://arxiv.org/abs/2412.02901

[21] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp – simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[22] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Automatic hyperparameter tuning for black-box lidar odometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5069–5074.

[23] A. Jacobson, Z. Chen, and M. Milford, "Online place recognition calibration for out-of-the-box slam," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1357–1364.

[24] N. Messikommer, G. Cioffi, M. Gehrig, and D. Scaramuzza, "Reinforcement learning meets visual odometry," 2024.

[25] T. Haarnoja *et al.*, "Soft actor-critic algorithms and applications," 2019.