

Adaptive Safety Margins for Quadruped Navigation with Calibrated Velocity Interfaces

Junho Kang • Hyesun Bae • Daewon Ryu • Minju Seo

Legged Locomotion and Control Lab, KAIST, Daejeon, South Korea

PUBLISHED

September 28, 2026

REGISTERED

October 1, 2026

RECORD TYPE

Preprint

ABSTRACT

Fixed safety margins for legged navigation are tuned once, so they ignore both how well the robot's velocity interface is currently calibrated and how much free space the robot is about to enter. This paper presents an adaptive safety margin for quadruped navigation that combines the two signals. A calibrated interface model obtained with GAUGE supplies a high-confidence upper bound on the lateral deviation caused by command lag, and the control-aligned future-range predictor FutureRay supplies a lower confidence bound on the free range along the directions the controller can actually execute. The margin is the sum of a kinematic term and a lag-error term, each blended between its model-based value and its worst-case value by an explicit confidence weight. We prove that the margin is monotone nondecreasing in both uncertainty sources, that it never falls below the model-based bounds, and that it reduces exactly to the worst-case fixed margin when both confidences vanish. On a Unitree Go2 evaluated on 12 obstacle-adjacent segments at speeds from 1.0 to 4.0 m/s, with matching Isaac Lab simulation, the adaptive margin raises the collision-free speed from 3.10 m/s for the worst-case fixed margin to 3.40 m/s and task success from 78.4 to 90.8 percent, while lowering the 95th-percentile margin from 42.0 to 38.2 cm. When the interface model becomes stale after a payload change, the margin inflates toward the worst-case value without manual intervention.

INDEX TERMS quadruped navigation, adaptive safety margins, velocity interface calibration, future range prediction, legged robots, Unitree Go2

KAGG NOTE A safety margin should be as large as the robot's current uncertainty requires and no larger; calibrated confidence makes that size computable online, and an explicit worst-case ceiling makes it safe to lose that confidence.

01 Introduction

Learned locomotion controllers have turned quadrupeds into fast, robust mobile platforms. Policies trained in massively parallel simulation now track body-velocity commands over rough terrain, stairs, and deformable ground, and commercial robots such as the Unitree Go2 expose exactly this abstraction to the navigation stack: the planner sends a planar twist, and an on-board controller decides how the legs realize it [1, 2, 3, 4, 5]. The abstraction is convenient, but it is not transparent. The realized motion lags the command, the lag depends on the gait, the commanded speed, and the payload, and the controller is usually opaque to the planner that relies on it [6, 7, 8].

Local planners for legged robots protect against this mismatch, and against imperfect perception, with a safety margin: obstacles are inflated, or the robot footprint is padded, by a clearance that the planned trajectory must respect [9, 10, 11]. In nearly all deployed systems that margin is fixed, or at most scheduled on commanded speed, and it is tuned once. A fixed margin must then be chosen for the worst condition it will ever meet, which wastes clearance on straight, well-observed corridors and still fails when an unmodeled condition, such as an occluded corner approached at speed or a payload that changes the interface response, exceeds the tuning envelope. The trade-off is familiar: conservative margins cause detours, slowdowns, and infeasible narrow passages; permissive margins cause contact near obstacle boundaries [12].

Two recent components make a better choice possible on quadrupeds. GAUGE treats the velocity controller as an opaque interface and calibrates it actively, issuing planner-conditioned probe commands and fitting a model of the command-to-motion response online [13]. FutureRay predicts free range along the robot's own predicted, control-aligned directions, so that obstacle queries are aligned with motions the controller can actually execute [14]. Each component answers one half of the margin question. GAUGE indicates how far the robot will deviate from the command it is given; FutureRay indicates how much free space lies along the directions it is about to take. Neither, by itself, specifies how large the margin should be.

This paper combines the two into an adaptive safety margin. The margin is the sum of a body padding, a kinematic term driven by the predicted future range, and a lag-error term driven by the calibrated interface model. Each adaptive term carries an explicit confidence weight that interpolates between the model-based value and a worst-case value, so that the margin shrinks only when the evidence supports shrinking it. The construction is simple enough to analyze: the margin is monotone in both uncertainty sources, never falls below the model bounds it is built from, and reduces exactly to the worst-case fixed margin when both confidences vanish.

The contributions are as follows.

1. A decomposition of the quadruped navigation safety margin into a kinematic term, computed from control-aligned lower-confidence free range, and a lag-error term, computed from

a calibrated upper bound on command-lag deviation, each with an explicit confidence weight derived from the predictors' own calibration statistics.

2. An analysis showing that the margin is nondecreasing in the uncertainty of both predictors, dominates the model-based bounds, is bounded above by the worst-case fixed margin, and coincides with it when both confidences are zero; under the predictors' coverage assumptions this yields a probabilistic clearance statement.
3. An integration of the margin into a sampling-based local planner on the Unitree Go2, in which each candidate command sequence is evaluated with its own margin so that turning and accelerating candidates pay for the lag error they induce.
4. An evaluation on 12 obstacle-adjacent segments of a lab and campus course at speeds from 1.0 to 4.0 m/s, with matching Isaac Lab simulation, against fixed worst-case, fixed nominal, and single-term baselines, together with an ablation and an analysis of recorded failure and degradation cases.

Section 2 reviews related work, Section 3 formalizes the problem, Section 4 develops the margin and its properties, Section 5 reports the experiments, and Sections 6 and 7 discuss limitations and conclude.

02 Related Work

2.1 Velocity interfaces and their calibration

Reinforcement-learned locomotion policies are typically trained to track a commanded body twist. Hwangbo et al. combined a learned actuator model with simulation training to obtain agile and dynamic skills on ANYmal [1]; Lee et al. and Miki et al. extended this line to blind and perceptive locomotion over challenging natural terrain [2, 3]. Rudin et al. showed that such policies can be trained in minutes with massively parallel simulation [5], which made velocity-tracking controllers a commodity. RMA adapts a base policy online through an inferred extrinsics vector [4], and DreamWaQ learns robust command tracking with implicit terrain imagination [8]. Margolis et al. pushed commanded speeds on small quadrupeds toward their physical limits [6] and exposed gait parameters as additional commands [7]. These works optimize how well a controller tracks a command, and adaptation methods such as RMA reduce tracking error internally; they do not, however, publish to the planner a calibrated estimate of the residual error. From the planner's perspective, the controller remains an opaque map from commands to motion.

GAUGE addresses exactly this gap [13]. It treats the quadruped velocity controller as an opaque interface and calibrates it actively: probe commands are chosen conditioned on what the planner is likely to request, and a calibrated model of the command-to-motion response is fit online from the resulting motion. The key property for the present work is that GAUGE yields a predictive model whose errors are calibrated, so that the residual between commanded and realized motion can be bounded with stated confidence. GAUGE itself is concerned with obtaining and maintaining that model; it does not prescribe how the planner should use it for safety. This paper uses GAUGE's calibrated predictions as the input to the lag-error term of the margin and, in addition, uses the consistency of those predictions with measured motion as a confidence signal that detects when the calibration has become stale.

2.2 Perception and range prediction for legged navigation

Legged navigation stacks typically build a local elevation or occupancy representation from LiDAR or depth and odometry [15, 16] and then plan over it, either with explicit planners such as ArtPlanner [10] and collision-aware MPC [11], or with learned planners and policies [9, 17, 18, 19]. Fu et al. coupled vision and proprioception to navigate legged robots in cluttered spaces [20]; Frey et al. learned traversability online from the robot's own experience [21]; Miki et al. trained locomotion in confined spaces from a 3D representation [22]. Parkour systems demonstrate that legged robots can exploit obstacles at high speed when perception and control are tightly coupled [23, 24, 25]. In most of these systems, the planner consumes geometry that is expressed in a fixed world or body frame and is independent of what the controller can execute next.

FutureRay changes the query rather than the map [14]. It predicts the future free range along the robot's own predicted, control-aligned directions, that is, along the headings the controller will actually reach under candidate commands, so that obstacle queries are aligned with executable motion. This alignment is what makes FutureRay useful for a margin: a range predicted along the direction the robot will travel in the next second is directly comparable to a stopping or swerving distance. FutureRay is designed to support agile navigation and does not itself set a safety margin; this work turns its per-ray predictions, together with an uncertainty estimate attached to each ray, into the kinematic term of the margin.

2.3 Safety margins, tubes, and safety filters

Inflating obstacles by a clearance is the oldest form of robust planning, and modern formulations make the inflation principled. Tube-based methods bound the tracking error between a planning model and the true system: FaSTrack computes such a bound offline through Hamilton-Jacobi reachability [26, 27], and funnel libraries precompute verified regions around motion primitives [28]. Control barrier functions enforce forward invariance of a safe set through a quadratic program on the control input [29], and predictive safety filters minimally modify a learning-based controller to keep it within a certified safe set [30, 31]. These methods provide strong guarantees when the bound on the model error is valid, and they are typically conservative because the bound must hold for every condition considered offline. The proposed margin can be read as an online, confidence-weighted tube: its lag-error term plays the role of the tracking-error bound, but it is recomputed from a calibrated interface model at every planning cycle and falls back to the offline worst-case bound when the model loses confidence.

On legged robots specifically, Agile But Safe learns a reach-avoid value network that switches between an agile policy and a recovery policy for collision-free high-speed locomotion [32], and collision-free MPC formulations add obstacle constraints directly to the whole-body or centroidal problem [11]. Hu et al. learn context-dependent safety margins for visual navigation [12], showing that a clearance budget can be predicted from observations rather than fixed. The present work shares the goal of a context-dependent margin, but derives it from two calibrated predictors rather than learning it end to end, which makes its behavior under predictor failure analyzable: the margin cannot collapse below the model bounds and cannot exceed the worst-case value.

2.4 Uncertainty quantification for planning

Uncertainty-aware collision avoidance has been studied with model ensembles and bootstrapping [33], with Gaussian-process models that certify safe exploration [34], and with sampling-based predictive control that evaluates many rollouts in parallel [35]. Conformal prediction offers distribution-free coverage guarantees from held-out calibration data [36]; Lindemann et al. and Dixit et al. used it to bound the prediction error of other agents' trajectories in motion planning, including an adaptive variant that updates coverage online [37, 38]. High-speed flight in the wild shows that a learned perception-to-action mapping can operate near the limit when its training distribution matches deployment [39]. This work uses split-conformal scaling to calibrate the bounds that GAUGE and FutureRay supply and uses online innovation statistics to detect when the calibration no longer holds, which is the case that adaptive conformal methods address by re-estimating coverage.

2.5 Simulation and sim-to-real

Isaac Gym and its successor frameworks, including Orbit and Isaac Lab, made it practical to train and evaluate locomotion and navigation at scale [40, 41]. Domain randomization over appearance and dynamics remains the standard tool for bridging the reality gap [42, 43]. In the experiments of this paper, simulation is used for repeated-seed evaluation of the same segment protocol that is run on hardware; the margin itself is not trained and requires no simulation-to-reality transfer beyond that of the underlying locomotion controller and predictors.

03 Problem Formulation

3.1 Robot, interface, and environment

The robot moves in the plane with pose $x_t = (p_t, \psi_t) \in SE(2)$, where $p_t \in \mathbb{R}^2$ is the base position and ψ_t the yaw, sampled at the control period Δt . The navigation stack issues body-frame twist commands $u_t = (v_t^x, v_t^y, \omega_t) \in \mathcal{U}$ to an opaque velocity interface, which produces the realized body twist $\nu_t \in \mathbb{R}^3$. The interface is unknown but is described by a calibrated model, introduced in Section 4.2. The robot's collision footprint is $\mathcal{B}(x) \subset \mathbb{R}^2$, and the static obstacle set is $\mathcal{O} \subset \mathbb{R}^2$. The clearance of a pose is

$$d(x, \mathcal{O}) = \min_{a \in \mathcal{B}(x), b \in \mathcal{O}} \|a - b\|_2. \quad (1)$$

3.2 Planner and margin

A local planner replans every T_p seconds. At replanning time t it evaluates candidate command sequences $\mathbf{u}^{(j)} = (u_t^{(j)}, \dots, u_{t+N-1}^{(j)})$ over a horizon of N control steps. For each candidate, the commanded pose sequence $\hat{x}_{t+h}^{(j)}$, $h = 0, \dots, N$, is obtained by integrating the commanded twist kinematically. The planner admits a candidate only if

$$d(\hat{x}_{t+h}^{(j)}, \hat{\mathcal{O}}_t) \geq m_t^{(j)} \quad h = 0, \dots, N, \quad (2)$$

where $\hat{\mathcal{O}}_t$ is the obstacle set in the current local map and $m_t^{(j)} \geq 0$ is the safety margin for that candidate. A *margin rule* is a map from the information available at time t (the candidate, the interface model, and the perception outputs) to $m_t^{(j)}$. A *fixed* rule depends only on the candidate's speed and turn rate,

$$m^{\text{fix}} = \mu_0 + \mu_{\text{kin}}(v) + \mu_{\text{lag}}(v, \omega), \quad (3)$$

with $v = \max_h \|(v_{t+h}^x, v_{t+h}^y)\|$ and $\omega = \max_h |\omega_{t+h}|$ over the candidate, and with schedules $\mu_{\text{kin}}, \mu_{\text{lag}}$ chosen offline. The *worst-case* fixed rule m^{wc} uses schedules that cover the full envelope of interface conditions and perception errors considered during design; the *nominal* fixed rule m^{nom} uses schedules evaluated at the interface model and range error identified at the start of a deployment.

3.3 Objective

Let $\mathcal{E}$ denote the event that the realized footprint keeps a non-negative clearance from $\mathcal{O}$ over a trial. The goal is a margin rule that (i) admits higher speeds and more feasible candidates than m^{wc} when the predictors are reliable, (ii) preserves a stated clearance with high probability when their coverage assumptions hold, and (iii) degrades to m^{wc} , rather than to an unsafe value, when they are not reliable. Requirement (iii) is the one most often violated by learned or purely adaptive margins, and it motivates the explicit confidence weights below.

04 Method

4.1 Overview

For a candidate j , the adaptive margin is

$$m_t^{(j)} = m_0 + m_t^{\text{kin},(j)} + m_t^{\text{lag},(j)}, \quad (4)$$

where $m_0 = \mu_0$ is a constant body padding shared with the fixed rules, m^{kin} is a kinematic term that accounts for the geometric clearance ahead, and m^{lag} is a lag-error term that accounts for the deviation of the realized motion from the commanded path. Both adaptive terms have the same structure: a model-based value, bounded above by the corresponding worst-case schedule, blended with that schedule by a confidence weight in $[0, 1]$. Sections 4.2 and 4.3 define the model-based values, Section 4.4 defines the confidence weights, and Section 4.6 states the resulting properties.

4.2 Lag-error term from the calibrated interface model

GAUGE supplies a calibrated model of the command-to-motion response [13]. In our implementation the model is used in the discrete-time form

$$\nu_{t+1} = A_\theta \nu_t + B_\theta u_{t-d_\theta} + w_t, \quad w_t \sim \mathcal{N}(0, Q_\theta), \quad (5)$$

where A_θ and B_θ encode per-axis lag and gain together with cross-coupling between axes, d_θ is an input delay in control steps, and the parameter estimate $\hat{\theta}_t$ carries a covariance $\Sigma_{\theta,t}$. Given a candidate $\mathbf{u}^{(j)}$, propagating Eq. (5) from the current measured twist yields a predicted realized pose sequence $\hat{x}_{t+h}^{(j)}$. The lateral lag deviation at step h is the displacement of the realized position from the commanded path, measured along the commanded path normal $n_{t+h}^{(j)}$,

$$e_{t+h}^{\perp,(j)} = n_{t+h}^{(j)\top} (\hat{p}_{t+h}^{(j)} - \bar{p}_{t+h}^{(j)}). \quad (6)$$

Linearized propagation of $\Sigma_{\theta,t}$ and Q_θ through the rollout gives a predictive mean $\mu_h^\perp$ and standard deviation $\sigma_h^\perp$ of $e^\perp$. The model-based lag bound for the candidate is

$$\bar{e}_t^{(j)} = \min \left\{ \max_{0 \leq h \leq N} (|\mu_h^\perp| + z_G \sigma_h^\perp), \mu_{\text{lag}}(v^{(j)}, \omega^{(j)}) \right\}, \quad (7)$$

where μ_{lag} is the worst-case lag schedule of m^{wc} and z_G is a scale calibrated by split conformal prediction [36] on held-out

rollouts so that the true lateral deviation satisfies $\max_h (|e_h^\perp| - |\mu_h^\perp| - z_G \sigma_h^\perp) \leq 0$ with probability at least $1 - \delta_G$. The clip to μ_{lag} encodes the design assumption that the worst-case schedule already covers every interface condition the robot is permitted to operate in.

The maximum over the horizon is what gives the lag term its anticipatory character. Lateral lag deviation is small on straight segments at constant speed and grows when the commanded heading or speed changes, because the realized twist approaches the command with a delay. A candidate that turns sharply in the second half of its horizon therefore receives a larger lag bound already at the moment it is evaluated, before the turn begins.

4.3 Kinematic term from control-aligned future range

FutureRay predicts the free range along a set of control-aligned rays [14]. For candidate j , let $\mathcal{K}_t^{(j)}$ index the rays whose directions are aligned with the candidate's predicted headings over the horizon, and let $\hat{r}_k$ and σ_k^R be the predicted range and its predictive standard deviation for ray k . As for the lag bound, σ_k^R is rescaled by split conformal calibration on held-out data so that a one-sided lower confidence bound

$$r_t^{(j)} = \min_{k \in \mathcal{K}_t^{(j)}} (\hat{r}_k - z_R \sigma_k^R)_+ \quad (8)$$

does not exceed the true free range along the candidate's directions with probability at least $1 - \delta_R$, where $(\cdot)_+ = \max\{\cdot, 0\}$.

The kinematic term expresses how much lateral clearance the robot needs given how soon it may have to react. If the free range ahead is large compared with the distance needed to stop, the robot can correct late and a small lateral buffer suffices; if the range approaches the stopping distance, the full kinematic buffer is required. With the stopping distance

$$d_{\text{stop}}(v) = v \tau_{\text{rx}} + \frac{v^2}{2a_b}, \quad (9)$$

where τ_{rx} is the reaction latency of the stack and a_b a conservative braking deceleration, the model-based kinematic value is

$$g(\underline{r}, v) = \mu_{\text{kin}}(v) \exp\left(-\frac{(\underline{r} - d_{\text{stop}}(v))_+}{\lambda}\right), \quad (10)$$

with length scale $\lambda > 0$ and the worst-case kinematic schedule $\mu_{\text{kin}}(v)$ of m^{wc} . The schedule $\mu_{\text{kin}}(v)$ is affine in speed, $\mu_{\text{kin}}(v) = k_0 + k_1 v$, and is identified as the upper envelope of the lateral body sway and path-tracking deviation observed during calibration at each speed tier. The function g is nonincreasing in $\underline{r}$, nondecreasing in v (through both μ_{kin} and d_{stop}), and attains $\mu_{\text{kin}}(v)$ whenever $\underline{r} \leq d_{\text{stop}}(v)$.

Using the lower confidence bound rather than the point prediction matters most at occlusions. Behind an occluded corner the predicted range along the robot's future heading is uncertain, so σ_k^R grows, $\underline{r}$ drops toward the stopping distance, and the kinematic term rises even though the point prediction may still report open space.

4.4 Confidence weights

The bounds in Eqs. (7) and (8) are only as good as the calibration behind them. Each predictor therefore carries a confidence weight that measures whether its recent predictions are consistent with what was subsequently observed.

Interface confidence. Let $r_\tau = \nu_\tau - \hat{\nu}_\tau|_{\tau-1}$ be the one-step innovation of the interface model, with predicted covariance S_τ . The windowed normalized innovation statistic over the last W control steps is

$$s_t = \frac{1}{3W} \sum_{\tau=t-W+1}^t r_\tau^\top S_\tau^{-1} r_\tau, \quad (11)$$

which has expectation one when the model is consistent. The interface confidence is

$$c_t^G = \exp(-\kappa_G (s_t - 1)_+) \cdot \mathbb{I}[\text{GAUGE model valid}], \quad (12)$$

where $\kappa_G > 0$ sets the sensitivity and the indicator is zero when no calibrated model is available, for example immediately after start-up or after an operator declares a configuration change. A payload swap biases the innovations, raises s_t , and drives c_t^G toward zero without requiring the change to be declared.

Range confidence. Let $\bar{\sigma}_t^{R,(j)}$ be the mean conformally scaled standard deviation over the rays in $\mathcal{K}_t^{(j)}$. The range confidence is

$$c_t^{R,(j)} = \left(1 - \frac{\bar{\sigma}_t^{R,(j)}}{\sigma_{\text{max}}^R}\right)_+, \quad (13)$$

where σ_{max}^R is the predictive spread at which the range prediction is considered uninformative. Occlusion and sparse returns raise $\bar{\sigma}^R$ and reduce c^R .

Blending. The adaptive terms interpolate between the model-based values and the worst-case schedules,

$$m_t^{\text{lag},(j)} = c_t^G \bar{e}_t^{(j)} + (1 - c_t^G) \mu_{\text{lag}}(v^{(j)}, \omega^{(j)}), \quad (14)$$

$$m_t^{\text{kin},(j)} = c_t^{R,(j)} g(r_t^{(j)}, v^{(j)}) + (1 - c_t^{R,(j)}) \mu_{\text{kin}}(v^{(j)}). \quad (15)$$

A convex combination is used rather than, for example, dividing the bound by the confidence, because it keeps each term inside the interval between its model-based value and its worst-case value. This makes the worst-case schedule an explicit ceiling and the model-based bound an explicit floor, which is what Section 4.6 exploits.

4.5 Planner integration

The margin is evaluated per candidate inside a sampling-based local planner in the spirit of predictive path-integral control [35]. At each replanning cycle, N_c candidate command sequences are sampled around the previous solution and around a set of constant-curvature primitives spanning speeds from zero to the speed ceiling of the current trial. For each candidate the planner computes $\bar{x}^{(j)}$, $\hat{x}^{(j)}$, $\bar{e}^{(j)}$, $\underline{r}^{(j)}$, and $m_t^{(j)}$, discards candidates that violate Eq. (2), and selects the admissible candidate that minimizes

$$J^{(j)} = -w_p \Delta s^{(j)} + w_u \sum_h \|u_{t+h}^{(j)} - u_{t+h-1}^{(j)}\|^2 + w_g \angle(\bar{p}_{t+N}^{(j)} - \bar{p}_t^{(j)}, g_t), \quad (16)$$

where $\Delta s^{(j)}$ is progress along the reference route, the second term penalizes command changes, and the third penalizes heading deviation from the local goal direction g_t . If no candidate

is admissible, the planner commands a braking sequence and replans. Because the margin is evaluated per candidate, a candidate that turns aggressively at high speed is charged for the lag deviation it would induce, and the planner tends to begin turns earlier and at lower speed when the interface is slow, rather than discovering the lag after the turn is underway.

To avoid chattering between consecutive cycles, the executed margin is rate-limited on decrease only, $m_t \geq m_{t-T_p} - \dot{m}_\downarrow T_p$, which enlarges the admissibility test of Eq. (2) and never reduces the margin below Eq. (4). The executed margin trajectory m_t reported in Section 5 is the margin of the selected candidate after this limiter.

4.6 Properties

The following statements hold for a fixed candidate and are therefore stated without the index j . Write $\sigma^\perp = (\sigma_h^\perp)_h$ for the predictive standard deviations of the lag deviation and $\sigma^R = (\sigma_k^R)_k$ for those of the ranges.

Proposition 1 (monotonicity). With all other quantities fixed, the margin m_t in Eq. (4) is nondecreasing in each component of $\sigma^\perp$, in the innovation statistic s_t , and in each component of σ^R .

Proof. Consider the lag term. By Eq. (7), $\bar{e}$ is a minimum of a maximum of functions that are nondecreasing in each $\sigma_h^\perp$, hence nondecreasing in $\sigma^\perp$, and $\bar{e} \leq \mu_{\text{lag}}$. By Eq. (14), $\partial m^{\text{lag}} / \partial \bar{e} = c^G \geq 0$, so m^{lag} is nondecreasing in $\sigma^\perp$. As a function of the confidence, $\partial m^{\text{lag}} / \partial c^G = \bar{e} - \mu_{\text{lag}} \leq 0$, and by Eq. (12) c^G is nonincreasing in s_t ; hence m^{lag} is nondecreasing in s_t . For the kinematic term, $\underline{r}$ in Eq. (8) is nonincreasing in each σ_k^R , and g is nonincreasing in $\underline{r}$, so g is nondecreasing in σ^R ; moreover $g \leq \mu_{\text{kin}}$, so $\partial m^{\text{kin}} / \partial c^R = g - \mu_{\text{kin}} \leq 0$, and c^R in Eq. (13) is nonincreasing in each σ_k^R . Since m^{kin} is nondecreasing in g and nonincreasing in c^R , and both effects of an increase in σ_k^R push m^{kin} upward, m^{kin} is nondecreasing in σ^R . The sum in Eq. (4) preserves monotonicity, and the rate limiter, being a maximum with a quantity that does not depend on the current uncertainties, does as well. $\square$

Proposition 2 (reduction and envelope). For every candidate, $m_0 + g + \bar{e} \leq m_t \leq m^{\text{wc}}$ before rate limiting. Moreover, $m_t = m^{\text{wc}}$ exactly whenever $c_t^G = c_t^R = 0$, which is the case when the GAUGE model is flagged invalid and $\bar{\sigma}^R \geq \sigma_{\text{max}}^R$, and $m_t \rightarrow m^{\text{wc}}$ as $s_t \rightarrow \infty$ with $\bar{\sigma}^R \geq \sigma_{\text{max}}^R$.

Proof. Each adaptive term is a convex combination of a model-based value and the corresponding worst-case schedule, and the model-based value never exceeds the schedule ($\bar{e} \leq \mu_{\text{lag}}$ by the clip in Eq. (7); $g \leq \mu_{\text{kin}}$ because the exponential in Eq. (10) is at most one). A convex combination of two numbers lies between them, which gives both inequalities after adding $m_0 = \mu_0$ and comparing with Eq. (3). With both weights zero, Eqs. (14)–(15) return the worst-case schedules and Eq. (4) equals m^{wc} . $\square$

Propositions 1 and 2 together say that the margin moves continuously between two well-defined rules: it equals the fixed worst-case margin when nothing is trusted, it can never be larger than that margin, and it approaches the model-based bound only as confidence grows.

Proposition 3 (conditional clearance). Let E_G be the event that the realized lateral deviation from the commanded path sat-

isfies $|e_{t+h}^\perp| \leq |\mu_h^\perp| + z_G \sigma_h^\perp$ for all h , and let E_R be the event that the true free range r^* along the candidate's directions satisfies $r^* \geq \underline{r}$. Suppose the planner executes a candidate admissible under Eq. (2) and that the local map satisfies $\hat{\mathcal{O}}_t \supseteq \mathcal{O}$ in the region swept by the candidate. Then, on $E_G \cap E_R$, which has probability at least $1 - \delta_G - \delta_R$ by the union bound, the realized footprint keeps a clearance of at least $m_0 + g(r^*, v)$ from $\mathcal{O}$ over the horizon, provided the GAUGE bound is not clipped in Eq. (7).

Proof. On E_G , and without clipping, the realized position lies within $\bar{e} \leq m^{\text{lag}}$ of the commanded path along its normal (Proposition 2), so the realized clearance is at least $m_t - m^{\text{lag}} = m_0 + m^{\text{kin}}$. On E_R , $g(\underline{r}, v) \geq g(r^*, v)$ because g is nonincreasing in range, and $m^{\text{kin}} \geq g(\underline{r}, v)$ by Proposition 2. Combining the two inequalities gives the claim. $\square$

The statement is deliberately modest. It is conditional on the coverage of the two calibrated predictors, on the map containing the obstacles in the swept region, and on a small-angle approximation in which the lateral deviation dominates the footprint displacement. It does not claim that the kinematic buffer g is sufficient to avoid every collision, only that the buffer computed from the true geometry is not consumed by command lag.

4.7 Implementation details

The interface runs at $\Delta t = 20$ ms (50 Hz), and the planner replans at 10 Hz ($T_p = 0.1$ s) over a 1.0 s horizon ($N = 50$) with $N_c = 256$ candidates. The coverage levels are $\delta_G = \delta_R = 0.05$, the innovation window is $W = 25$ steps (0.5 s), the reaction latency is $\tau_{\text{rx}} = 0.1$ s, the braking deceleration is $a_b = 3.0$ m/s², and the length scale is $\lambda = 1.5$ m. At the top speed tier of 4.0 m/s these values give a stopping distance of a few meters, well inside the range of the onboard LiDAR. The worst-case schedules μ_{kin} and μ_{lag} are identified once from calibration runs that span the payload and terrain conditions of the course and are shared by the worst-case fixed baseline, so that Proposition 2 compares the adaptive margin with the same worst-case rule that is evaluated in Section 5. The conformal scales z_G and z_R are computed from held-out simulation rollouts and the calibration runs.

REPRODUCIBILITY NOTE

The margin adds no learned parameters beyond those of the interface model and the range predictor. All quantities in Eqs. (4)–(15) are computed per candidate from the predictors' outputs, so the margin can be logged and replayed offline from the planner's candidate set.

05 Experiments

5.1 Platform and protocol

All hardware experiments use a Unitree Go2 quadruped controlled through its body-velocity interface, with the GAUGE interface model and the FutureRay range predictor running onboard alongside the planner. The test course combines laboratory and campus sections and contains 12 obstacle-adjacent segments, numbered 1 to 12, each requiring the robot to pass close to obstacles, including an occluded corner (segment 7). Each segment is traversed at commanded speed ceilings from 1.0 to 4.0 m/s in 0.5 m/s steps, which gives seven speed tiers, with 20 trials per segment per speed; one full pass of the protocol therefore comprises $12 \times 7 \times 20 = 1,680$ trials per method.

The same segments are reproduced in Isaac Lab [41], where the protocol is repeated over 5 random seeds. The tables below report the final aggregated results of the study.

The fixed-margin baseline at the worst-case value, m^{wc} of Eq. (3), is the reference against which the adaptive margin is compared, as required by Proposition 2.

5.2 Baselines and variants

The main comparison contains five margin rules that share the planner, the perception stack, and the body padding m_0 .

- **Fixed margin (worst case):** m^{wc} , the worst-case schedules of Eq. (3).
- **Fixed margin (nominal):** m^{nom} , schedules evaluated with the interface model and range error identified at the start of the deployment and then frozen.
- **Lag-error term only:** the adaptive lag term of Eq. (14) with the nominal fixed kinematic schedule.
- **Kinematic term only:** the adaptive kinematic term of Eq. (15) with the nominal fixed lag schedule.
- **Ours:** the full adaptive margin of Eq. (4).

The ablation removes one component at a time from the full method without replacing it: **w/o confidence weighting** sets $c_t^G = c_t^R = 1$, so that the model-based bounds are used directly; **w/o kinematic term** sets $m^{\text{kin}} \equiv 0$; and **w/o lag-error term** sets $m^{\text{lag}} \equiv 0$. The distinction between the single-term baselines and the ablation is important when reading the tables: a single-term baseline still carries a nominal fixed allowance for the other source of error, whereas an ablation variant carries none.

5.3 Metrics

Collision-free speed summarizes how far up the speed sweep a rule remains safe. For each segment it is the highest speed at which no collision is recorded, obtained by linear interpolation of the per-tier collision rate between the last collision-free tier and the first tier with a collision; the reported value is the average over segments and evaluation runs. *Task success* is the percentage of trials in which the robot completes the segment within its time limit and without collision. *Mean margin* and *margin p95* are the mean and 95th percentile of the executed margin m_t , logged at every planning cycle within the obstacle-adjacent segments and pooled over all speed tiers. For the fixed rules, m_t varies with the executed speed and turn rate through the schedules of Eq. (3), so the margin statistics are informative for every rule.

5.4 Main comparison

Table 1 reports the main comparison.

The adaptive margin attains the highest collision-free speed, 3.40 m/s, and the highest task success, 90.8%, while using the smallest mean margin, 21.6 cm, and the smallest 95th-percentile margin, 38.2 cm. Relative to the worst-case fixed margin, the collision-free speed increases by 0.30 m/s (9.7%), task success increases by 12.4 percentage points, the mean margin decreases by 1.0 cm, and the margin p95 decreases by 3.8 cm. Relative to the nominal fixed margin, the collision-free speed increases by 0.53 m/s and task success by 6.4 points.

The two fixed rules fail in opposite ways. The worst-case rule reaches the higher collision-free speed of the two (3.10 versus 2.87 m/s) but the lower task success (78.4% versus

84.4%). This pattern is consistent with the expected cost of conservatism: a margin sized for the worst interface condition rules out candidates in narrow passages and at high turn rates, which leads to failed segment completions even when the robot does not collide. The nominal rule admits more of those candidates and completes more segments, but its margin is too small when the actual interface response or geometry departs from the start-of-deployment calibration, which lowers the speed at which it remains collision free (Section 5.6). The two rules have nearly equal mean margins (22.6 and 22.7 cm); their margin statistics differ mainly in the tail, where the nominal rule's p95 of 44.1 cm exceeds the worst-case rule's 42.0 cm. Because the fixed schedules are evaluated at the executed speed and turn rate, the nominal planner's acceptance of sharper, faster candidates moves its margin distribution upward in the tail even though its schedule is pointwise smaller.

Each single-term baseline recovers part of the improvement. Making only the kinematic term adaptive gives the second-highest collision-free speed (3.16 m/s) with 87.1% success; making only the lag-error term adaptive gives 2.93 m/s with 87.6% success and the largest mean margin of all rules, 24.9 cm. The full method improves on the kinematic-only baseline by 0.24 m/s and 3.7 points and on the lag-only baseline by 0.47 m/s and 3.2 points. The kinematic term therefore accounts for most of the speed gain, whereas both terms contribute similarly to success; the combination is better than either alone on every metric in Table 1.

The margin statistics clarify where the gain comes from. The adaptive margin is not uniformly smaller than the fixed margins: on average it is only 1.0 cm below the worst-case rule. Its advantage lies in where the clearance is spent. It is reduced on well-observed straight sections in which both predictors are confident and increased ahead of turns and occlusions, which lowers the tail of the distribution (38.2 cm p95, the lowest in the table) while raising the margin at the locations where collisions occur under the fixed rules.

5.5 Ablation

Table 2 removes one component of the full method at a time.

Every component is needed. Removing the kinematic term causes the largest degradation, to 2.76 m/s and 75.6% success (a loss of 0.64 m/s and 15.2 points), which is consistent with the main comparison, where the kinematic term carried most of the speed gain. Removing the lag-error term lowers the collision-free speed to 3.07 m/s and success to 81.3% (a loss of 0.33 m/s and 9.5 points). Both ablated variants are worse than the corresponding single-term baselines of Table 1 (2.93 m/s and 87.6% for lag-only; 3.16 m/s and 87.1% for kinematic-only), because the ablation leaves no allowance at all for the removed error source, whereas the single-term baselines retain a nominal fixed allowance.

Removing confidence weighting is instructive. The collision-free speed drops by 0.24 m/s, to 3.16 m/s, but task success drops by 11.5 points, to 79.3%, which is only 0.9 points above the worst-case fixed margin. Without the weights, the margin follows the model-based bounds directly even when those bounds are no longer calibrated, for example after the interface response changes or behind occlusions in which the range uncertainty is underestimated. The resulting margins are too small exactly in the conditions where the predictors are least reliable. This is the failure mode that the confidence weights, and the envelope of Proposition 2, are designed to prevent.

Table 1: Main comparison on the Unitree Go2 segment protocol (12 obstacle-adjacent segments, 1.0–4.0 m/s). The proposed method is shown in bold.

Method	Collision-free speed (m/s)	Task success (%)	Mean margin (cm)	Margin p95 (cm)
Fixed margin (worst case)	3.10	78.4	22.6	42.0
Fixed margin (nominal)	2.87	84.4	22.7	44.1
Lag-error term only	2.93	87.6	24.9	40.9
Kinematic term only	3.16	87.1	22.6	40.8
Ours	3.40	90.8	21.6	38.2

Table 2: Ablation of the adaptive margin. Each variant removes one component from the full method without replacing it.

Variant	Collision-free speed (m/s)	Task success (%)
Ours (adaptive, both terms)	3.40	90.8
w/o confidence weighting	3.16	79.3
w/o kinematic term	2.76	75.6
w/o lag-error term	3.07	81.3

5.6 Margin trajectories and failure cases

The executed margin m_t was logged over time for every trial. Three recorded cases illustrate how the margin evolves and where the fixed rules fail.

Occluded corner (segment 7, 3.5 m/s). Approaching the occluded corner on segment 7 at 3.5 m/s, the lag-error term widens the margin to 44 cm before the corner, 0.3 s earlier than the kinematic term alone would. At 3.5 m/s, 0.3 s corresponds to 1.05 m of travel. The ordering follows from the construction. The lag term reacts as soon as candidates that turn around the corner enter the planning horizon, because Eq. (7) takes the maximum predicted lateral deviation over the full horizon; the kinematic term reacts only when the lower confidence bound on range along the post-turn directions approaches the stopping distance, which happens later. The earlier widening gives the planner time to shift the approach line outward and reduce speed before entering the turn.

Nominal margin at high speed (segment 9, 4.0 m/s). The nominal fixed margin, which evaluates to 18 cm on segment 9 at 4.0 m/s, collides in 7 of 20 trials (35%). On the same segment and speed, the adaptive margin holds a mean of 31 cm, 13 cm more than the nominal value. The nominal rule has no mechanism to grow its margin in response to the elevated lag deviation and range uncertainty at the top of the speed range, whereas the adaptive margin increases through both terms.

Stale interface model (payload swap). When the interface model is stale, after a payload swap and before recalibration, the adaptive margin inflates toward the worst-case value automatically. This graceful-degradation case was recorded in 2 trials. The mechanism is the innovation statistic of Eq. (11): the changed mass and inertia bias the realized twist relative to the stale model, s_t rises, c_t^G decreases, and Eq. (14) shifts the lag term toward μ_{lag} , as predicted by Propositions 1 and 2. Two trials demonstrate the mechanism but do not establish its reliability across payloads; a systematic study is left for future work.

06 Limitations

Several limitations qualify the results. First, the guarantee in Proposition 3 is conditional: it assumes that the conformal cov-

erage of GAUGE and FutureRay holds at deployment, that the local map contains every obstacle in the swept region, and that lateral deviation dominates footprint displacement. Coverage computed from held-out data need not hold under distribution shift; the confidence weights mitigate this by detecting inconsistency, but they do so with a delay of up to the innovation window W and cannot detect a shift that leaves one-step innovations unchanged.

Second, the worst-case schedules μ_{kin} and μ_{lag} define both the ceiling of the margin and the baseline it is compared with. If they underestimate the true worst case, Proposition 2 still holds, but the fallback behavior is only as safe as the schedules. The schedules were identified for one platform, one set of payloads, and the terrain of one course.

Third, the evaluation is limited in scope. All experiments use a single robot, the Unitree Go2, on 12 segments of one lab and campus course with static obstacles. The tables report aggregate results, and this paper does not report separate breakdowns of simulation and hardware performance. The graceful-degradation behavior after a payload swap was observed in only 2 trials. Dynamic obstacles, which would require the range predictor to account for obstacle motion, are not considered.

Fourth, the confidence weights in Eqs. (12) and (13) are hand-designed functions with sensitivity parameters κ_G and σ_{max}^R . They are monotone, which is what the analysis requires, but other monotone choices could trade speed for conservatism differently, and the choice was not optimized systematically. Finally, the kinematic term in Eq. (10) is a heuristic shaping of the clearance requirement. It is designed to be monotone and to reach the worst-case schedule at the stopping distance, but it is not derived from a reachability computation.

07 Conclusion

This paper presented an adaptive safety margin for quadruped navigation that combines a calibrated velocity-interface model and a control-aligned future-range predictor. The margin is the sum of a body padding, a kinematic term derived from lower-confidence free range along the directions the controller will execute, and a lag-error term derived from a calibrated upper bound on command-lag deviation. Explicit confidence weights blend each term with its worst-case value. The construction makes the margin monotone in both uncertainty sources, bounds it between the model-based bounds and the worst-case fixed margin, and makes it reduce to the fixed margin exactly when confidence is lost. On a Unitree Go2 over 12 obstacle-adjacent segments from 1.0 to 4.0 m/s, the adaptive margin increased the collision-free speed from 3.10 to 3.40 m/s and task success from 78.4% to 90.8% relative to the worst-case fixed margin, while lowering the 95th-percentile margin from 42.0 to 38.2 cm. The ablation indicates that both terms and the confidence weights are necessary. Future work will evaluate the degradation behavior systematically across payloads, extend the range term to moving obstacles, and re-

place the heuristic kinematic shaping with a reachability-based requirement.

References

- [1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning Agile and Dynamic Motor Skills for Legged Robots. *Science Robotics*, 2019. arXiv:1901.08652.
- [2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. *Science Robotics*, 2020. arXiv:2010.11251.
- [3] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Robust Perceptive Locomotion for Quadrupedal Robots in the Wild. *Science Robotics*, 2022. arXiv:2201.08117.
- [4] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. *Robotics: Science and Systems*, 2021. arXiv:2107.04034.
- [5] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. *Conference on Robot Learning*, 2021. arXiv:2109.11978.
- [6] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid Locomotion via Reinforcement Learning. *Robotics: Science and Systems*, 2022. arXiv:2205.02824.
- [7] G. B. Margolis and P. Agrawal. Walk These Ways: Tuning Robot Control for Generalization with Multiplicity of Behavior. *Conference on Robot Learning*, 2022. arXiv:2212.03238.
- [8] I. M. A. Nahrendra, B. Yu, and H. Myung. DreamWaQ: Learning Robust Quadrupedal Locomotion with Implicit Terrain Imagination via Deep Reinforcement Learning. *IEEE International Conference on Robotics and Automation*, 2023. arXiv:2301.10602.
- [9] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. *IEEE Robotics and Automation Letters*, 2021. arXiv:2103.04351.
- [10] L. Wellhausen and M. Hutter. ArtPlanner: Robust Legged Robot Navigation in the Field. *Field Robotics*, 2023. arXiv:2303.01420.
- [11] M. Gaertner, M. Bjelonic, F. Farshidian, and M. Hutter. Collision-Free MPC for Legged Robots in Static and Dynamic Scenes. *IEEE International Conference on Robotics and Automation*, 2021. arXiv:2103.13987.
- [12] J. Hu, S. Yuan, G. C. R. Bethala, A. Tzes, and Y. Fang. Learning Adaptive Safety Margins for Visual Navigation. arXiv:2607.18200, 2026.
- [13] T. Zang, H. Huang, S. Cheng, and W. Zhang. GAUGE: Planner-Conditioned Active Calibration of Opaque Quadruped Velocity Interfaces. arXiv:2607.02841, 2026.
- [14] T. Zang, S. Wang, H. Huang, and S. Cheng. FutureRay: Control-Aligned Future Range for Agile Quadruped Navigation. arXiv:2608.01197, 2026.
- [15] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter. Elevation Mapping for Locomotion and Navigation using GPU. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022. arXiv:2204.12876.
- [16] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Transactions on Robotics*, 2022. arXiv:2107.06829.
- [17] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022. arXiv:2209.12827.
- [18] F. Yang, C. Wang, C. Cadena, and M. Hutter. iPlanner: Imperative Path Planning. *Robotics: Science and Systems*, 2023. arXiv:2302.11434.
- [19] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter. ViPlanner: Visual Semantic Imperative Learning for Local Navigation. *IEEE International Conference on Robotics and Automation*, 2024. arXiv:2310.00982.
- [20] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling Vision and Proprioception for Navigation of Legged Robots. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. arXiv:2112.02094.
- [21] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast Traversability Estimation for Wild Visual Navigation. *Robotics: Science and Systems*, 2023. arXiv:2305.08510.
- [22] T. Miki, J. Lee, L. Wellhausen, and M. Hutter. Learning to Walk in Confined Spaces Using 3D Representation. *IEEE International Conference on Robotics and Automation*, 2024. arXiv:2403.00187.
- [23] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. *Science Robotics*, 2024. arXiv:2306.14874.
- [24] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot Parkour Learning. *Conference on Robot Learning*, 2023. arXiv:2309.05665.
- [25] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. *IEEE International Conference on Robotics and Automation*, 2024. arXiv:2309.14341.
- [26] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin. FaSTrack: A Modular Framework for Fast and Guaranteed Safe Motion Planning. *IEEE Conference on Decision and Control*, 2017. arXiv:1703.07373.
- [27] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances. *IEEE Conference on Decision and Control*, 2017. arXiv:1709.07523.
- [28] A. Majumdar and R. Tedrake. Funnel Libraries for Real-Time Robust Feedback Motion Planning. *International Journal of Robotics Research*, 2017. arXiv:1601.04037.
- [29] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Transactions on Automatic Control*, 2017. arXiv:1609.06408.
- [30] K. P. Wabersich and M. N. Zeilinger. A Predictive Safety Filter for Learning-Based Control of Constrained Nonlinear Dynamical Systems. *Automatica*, 2021. arXiv:1812.05506.
- [31] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems. *IEEE Transactions on Automatic Control*, 2019. arXiv:1705.01292.
- [32] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi. Agile But Safe: Learning Collision-Free High-Speed Legged Locomotion. *Robotics: Science and Systems*, 2024. arXiv:2401.17583.
- [33] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine. Uncertainty-Aware Reinforcement Learning for Collision Avoidance. arXiv:1702.01182, 2017.
- [34] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause. Safe Model-Based Reinforcement Learning with Stability Guarantees. *Advances in Neural Information Processing Systems*, 2017. arXiv:1705.08551.

- [35] G. Williams, A. Aldrich, and E. A. Theodorou. Model Predictive Path Integral Control using Covariance Variable Importance Sampling. arXiv:1509.01149, 2015.
- [36] A. N. Angelopoulos and S. Bates. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. arXiv:2107.07511, 2021.
- [37] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas. Safe Planning in Dynamic Environments Using Conformal Prediction. IEEE Robotics and Automation Letters, 2023. arXiv:2210.10254.
- [38] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick. Adaptive Conformal Prediction for Motion Planning among Dynamic Agents. Learning for Dynamics and Control Conference, 2023. arXiv:2212.00278.
- [39] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. Learning High-Speed Flight in the Wild. Science Robotics, 2021. arXiv:2110.05113.
- [40] V. Makoviychuk et al. Isaac Gym: High Performance GPU-Based Physics Simulation for Robot Learning. arXiv:2108.10470, 2021.
- [41] M. Mittal et al. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. IEEE Robotics and Automation Letters, 2023. arXiv:2301.04195.
- [42] 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. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017. arXiv:1703.06907.
- [43] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. IEEE International Conference on Robotics and Automation, 2018. arXiv:1710.06537.