Whole-Body Dynamic Obstacle Avoidance with Humanoid Skin

Niraj Pudasaini, Carson Kohlbrenner, and William Xie
{nipu2200, cako7425, wixi6454}@colorado.edu
University of Colorado Boulder

***Abstract*—In order for humanoid robots to safely inhabit and operate in human environments and beyond, they must be able to quickly and robustly react to dynamic and numerous obstacles. Existing methods for dynamic collision avoidance utilize task-constrained whole-body control, which accommodates slow-moving obstacles that do not require rapid, large bodily adjustments. We propose an end-to-end deep reinforcement learning (RL) method for whole-body collision avoidance of high-velocity, dynamic obstacles on humanoid robots equipped with proximity-sensing skins. We compare our method with a model predictive control (MPC) baseline and evaluate on two different sensing modalities–time-of-flight (ToF) and self-capacitance sensing (CPS)–and a variety of sensing ranges. Within a simplified simulation environment, our approach with combined self-capacitive sensing and time-of-flight proximity sensing achieves 97% avoidance success, outperforming MPC's 31% avoidance success. Our RL implementation[1] and MuJoCo MPC implementation frameworks[2] are made open-source.**

I. Introduction

Humanoid robots must rapidly and robustly avoid collisions with dynamic obstacles for safe deployment in everyday human environments that are unpredictable, cluttered, and unstructured. Then, they must extend this capability to truly adversarial environments—climate events, the deep sea, dense flora—if they are to advance frontiers of exploration and operation.

Traditional collision avoidance strategies rely on explicitly defined constraints and often analytically-derived motions. Whole body control (WBC) and model predictive control (MPC) solved with quadratic programming (QP) or differential dynamic programming (DDP) [4, 9, 24, 32] provide provable constraint satisfaction and explicit modeling of contact dynamics for robust and interpretable avoidance behavior. Modern algorithmic and computation speedups (GPU-parallelization [14], Moore's law) have made the real-time deployment of such control methods on complex humanoid robots increasingly tractable.

However, these approaches are still beholden to increasing task complexity and modeling, which can quickly render online control infeasible. When an obstacle approaches at high velocity, violating potentially dozens of collision constraints, the time between detection and impact may be insufficient for trajectory re-planning and constraint resolution. As such, no method has been demonstrated yet for avoidance of a single high-velocity, direct-impact obstacle, let alone multiple and successive obstacles, requiring a drastic body reconfiguration within tens or hundreds of milliseconds.

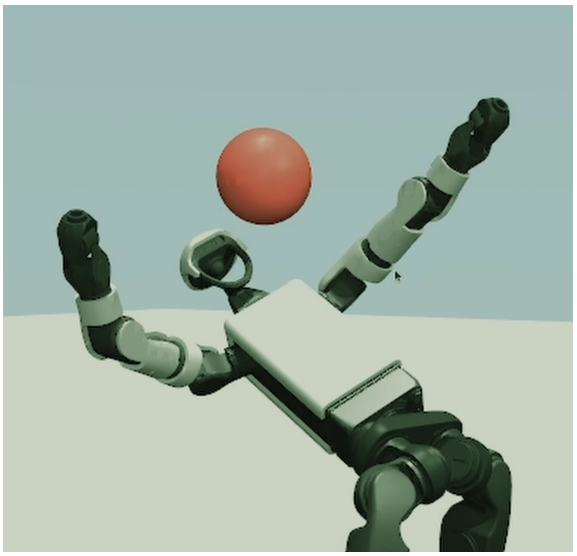


Fig. 1: Humans react to quickly-moving dynamic obstacles with agility, coordination, and subconscious reflexes. Can robots equipped with similar whole-body sensing suites do the same?

This limitation motivates our primary research inquiry: *Can humanoid robots leverage Reinforcement Learning to synthesize robust, reflexive whole-body avoidance behaviors directly from distributed Time-of-Flight (ToF) and capacitive sensing data, thereby bypassing the latency of online modeling?*

The fundamental challenge lies in the trade-off between reactivity and safety guarantees. Reinforcement learning (RL) policies cannot provide explicit constraint satisfaction, but by offloading learning of complex sensorimotor control to offline, simulated interactions, they can provide rapid and fluent reactions. By directly optimizing for cumulative reward through repeated interaction, RL policies can discover non-intuitive motor strategies that exploit the full operational space of the robot. Unlike supervised or imitation learning, which require difficult or impossible to collect expert demonstrations of evasive maneuvers, RL naturally handles the exploration-exploitation trade-off inherent in collision avoidance. Thus, with RL, robots can learn a robust representation mapping numerous diverse and noisy sensor observations to dynamic

[1] https://github.com/NirajPudasaini/H12_Bullet_Time

[2] https://github.com/badinkajink/mujoco_mpc/tree/humanoidbench/mjpc/tasks/humanoid_bench/avoid

and reflexive actions through task-driven optimization [37, 35].

We propose leveraging deep reinforcement learning to train whole-body collision avoidance policies on humanoid robots equipped with distributed proximity sensing skins. Specifically, we investigate:

1. **Body-centric proximity sensing efficacy**: How can body-centric proximity sensing leveraging short-range CAP and long range ToF sensors enable dynamic and reactive collision avoidance?
2. **Learned vs. analytical control**: How do learned reflex policies compare to optimal control baselines in terms of collision avoidance rate, stability, and generalization to novel obstacle trajectories?
3. **Reward Design**: How can we structure a reward function to balance the competing objectives of aggressive collision avoidance and bipedal stability, while regularizing for natural, human-like motion patterns?
4. **Sim-to-real transferability**: Can policies trained in simulation with simplified sensor models transfer to physical capacitive and ToF proximity sensing?

II. Related Works

A. Whole-Body Control via Optimization

Model predictive control has become a common approach for whole-body control of legged robots. MuJoCo MPC (MJPC) [10] provides a practical sampling-based implementation that combines trajectory optimization with contact-rich dynamics. The framework uses iterative linear-quadratic regulator (iLQR and iLQG) methods with finite-difference derivatives, enabling real-time operation at 20-50Hz on modern hardware. More recent work has demonstrated MPC on the torque-controlled Unitree H1 humanoid [36], showing robust locomotion and manipulation capabilities.

Alternative approaches use analytical derivatives through rigid-body dynamics libraries like Pinocchio [2]. Crocoddyl [21] builds on this foundation to provide differential dynamic programming (DDP) with explicit contact models. These methods offer faster convergence through exact gradients, though they require more careful contact model specification and may be less robust to model mismatch learning methods. Furthermore, recent methods from Armleder et al. [1] which leverage whole-body skin for control and collision-avoidance only evaluate slow-moving (0.2 ms^{-1}) obstacles moving toward auxiliary limbs like the arm. The proposed method also requires hand-designed parameters for each task and specifically highlights that aggressive parameters for large reactions can cause hysteresis—rapid, unstable constraint activations and deactivations.

B. Reinforcement Learning for Locomotion and Manipulation

Reinforcement learning (RL) has been successfully applied to achieve stable locomotion in quadrupeds [12, 18] and in humanoids [7]. Compared to classical optimal-control-problem (OCP) based controllers, which must solve high-dimensional trajectory optimizations online - inducing significant latency and exhibiting sensitivity to modeling errors [23, 33]. RL-trained policies learn end-to-end control mappings offline and execute in constant time via a single network inference [29, 8]. Low-latency RL policies have enabled agile and rapid whole-body control, but primarily in contact-free balance at the extremes of kinematic range of motion [37] or in dynamic single-contact reconfiguration (parkour) [35].

C. Hybrid and Residual RL architectures.

Combining control with learning has been increasingly popular. Residual reinforcement learning can refine MPC [3, 14, 11] or WBC [20] by learning corrective actions. Others embed MPC as a module inside a learned policy or meta-learn MPC hyperparameters [28]. Hierarchical control frameworks also decompose stabilization vs manipulation layers (e.g. balancing vs end-effector control) to handle complexity [38, 13].

D. Egocentric Sensing

Onboard sensors need to detect incoming impacts quickly and provide adequate sensing coverage for fast-pace collision avoidance [26]. Contact and proximity sensors distributed over the body of the robot (ego-centric) have advantages over cameras regarding data efficiency and sensing coverage of the close, pre-contact space around a robot. Additionally, recent works have found that ego-centric placement of sensors increased the performance of learned policies compared to exocentric placement [34, 16], and works which pair such control methods with whole-body distributed proximity sensing [25, 1] have proven effective for dynamic obstacle avoidance. Our study investigates the impact of egocentric sensing coverage on performance for reactive collision avoidance policies.

E. Gap and Contribution

Sensor modeling, simulation, and data representation are the fundamental challenges central to effective policy learning. Appropriately capturing and utilizing proximity data (across capacitive, ToF, and LIDAR sensors) has been explored across robot embodiments [5], but remains an open-ended challenge for bipedal whole-body control. High-velocity obstacle avoidance poses additional problems of proximity sensor distance resolution, measuring instantaneous velocity, or tracking obstacles across measurements, all of which can limit high-bandwidth anticipatory reflexes.

In our setting, we primarily compare policies achieved via end-to-end RL vs. MPC, and we do an initial study on trajectory-guided RL with MPC reference trajectories. We observe that MPC produces feasible avoidance behaviors but often fails to handle long-range proximity feedback or to maintain balance, and that trajectory-guided RL baseline performs very poorly with an initial, small collection of reference trajectories. The end-to-end RL in comparison demonstrates robustness, emergent avoidance behaviors inaccessible by optimal control methods, and quick, sample-efficient training. Our contributions are a) Bullet Time, an end-to-end humanoid RL architecture that integrates onboard sensors for high-speed collision avoidance, b) an evaluation on the emergent

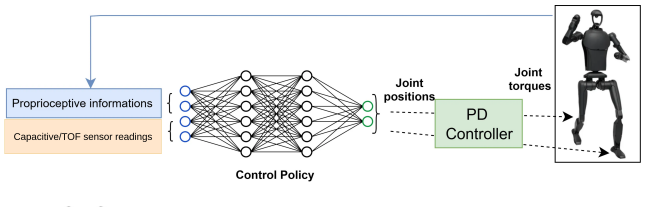


Fig. 2: Control Architecture. The policy network fuses history-stacked proprioceptive and exteroceptive (Capacitive/ToF) observations to output target joint positions. A PD controller converts these targets to torques for stable execution.

behaviors when using simulated sensor analogs in various configurations and sensing ranges, and c) comparison with optimal control baselines with multiple configurations.

III. Methodology

The objective is to train a reinforcement learning (RL) policy that enables the Unitree H12 humanoid robot to perform rapid, stable evasive maneuvers in response to approaching obstacles. The policy $\pi_\theta(a_t|o_t)$ outputs joint-level commands to maximize the expected cumulative reward:

$$\max_\theta \ \mathbb{E}\left[\sum_{t=0}^{T}\gamma^t r_t(o_t,a_t)\right], \tag{1}$$

where γ is the discount factor and r_t encodes both stability and safety. Training is conducted using the Proximal Policy Optimization (PPO) [29] algorithm, chosen for its robustness and scalability to high-dimensional continuous control.

A. Observation and Action Spaces

Observation Space ($\mathcal{O}$): Each observation o_t provides the policy with the required proprioceptive and exteroceptive information:

- **Proprioceptive states:** Joint positions and velocities, base angular velocity, and the gravity vector projected in the base frame to estimate torso orientation and stability.
- **Exteroceptive states:**
  1. *Capacitive proximity sensors* (0-15 cm range) distributed across the torso and upper limbs, providing dense local awareness of nearby objects.
  2. *Time-of-Flight (ToF) cone* sensor capturing 8×8 distance readings (64 values) across a 45° forward field of view up to 4 m.

Action Space ($\mathcal{A}$): The policy outputs target joint positions $q_t^* \in \mathbb{R}^{21}$. These are converted into motor torques τ_t via a low-level PD controller:

$$\tau_t = K_p(q_t^* - q_t) + K_d(\dot{q}_t^* - \dot{q}_t), \tag{2}$$

where K_p and K_d are the proportional and derivative gains, and $\dot{q}_t^*$ is approximated by zero or finite differences. This action formulation favors compliant, stable motion compared to direct torque control.

B. Reward Formulation

Designing a reward function for humanoid evasion requires balancing immediate survival with long-term stability and motion quality. We formulate the reward r_t as a weighted sum of *task objectives* (e.g., survival bonus, obstacle clearance) and *regularization terms* (e.g., energy efficiency, action smoothness). While the task rewards drive the robot to discover agile dodging maneuvers, the regularization terms are critical for suppressing high-frequency jitter and ensuring the learned behaviors are fluid, energy-efficient, and visually natural. The precise mathematical formulations, weights, and parameter values for all reward components are detailed in Appendix B and Table II.

C. Network Architecture

We adopt an asymmetric actor-critic structure [27] to enhance training stability while ensuring deployability. In this framework, the critic is provided with privileged information available only during simulation (such as the projectile's global position and velocity), while the actor observes only onboard sensors.

- **The Actor** is a Multi-Layer Perceptron (MLP) with two hidden layers of 32 units each and ELU activations. It receives only the observation history $o_{t-k:t}$ available on the physical robot.
- **The Critic** also uses a $[32, 32]$ MLP structure but receives the privileged simulation states to form a more accurate value estimate.

The architecture design is ablated in Appendix E, and we choose the normalized configuration. This asymmetry reduces variance during the learning process without imposing sensing requirements on the real hardware.

D. Sensor Modeling

The sensor observation vectors are modeled based on real capacitive and ToF sensors. We assume noiseless measurements of proximity in simulated environments to benchmark each of the proposed control algorithms.

1) Self-Capacitance Sensors: Proximity data from self-capacitance sensors (CPS) is modeled in the raw data format expected from real GenTact-Prox sensors using an analytically derived analog [17]. A CPS measures the electrostatic potential between an electrode and reference ground. For an array of n ideal CPSs, the capacitance is defined by the Maxwell capacitance matrix as the relation of charges q and potentials ϕ on the electrodes as follows [26, 31]:

$$q = C\phi = \begin{pmatrix} C_{1,1} & \cdots & -C_{n,1} \\ \vdots & \ddots & \vdots \\ -C_{1,n} & \cdots & C_{n,n} \end{pmatrix}\phi$$

where C is positive semi-definite, $q = [q_1, q_2, .., q_n]^T$, $\phi = [\phi_1, \phi_2, ..., \phi_n]^T$, and the off-diagonal elements $C_{i,j}$ represent the mutual capacitance between electrodes i and j. The observation of the array of CPSs z is the effective capacitive

measurements at a given timestep, defined as the sum of self-capacitance and mutual capacitance terms $z_i = \mathbf{1}^T c_i$, where $z = [z_1, z_2, ..., z_n]^T$. The observation vector can be extracted directly from a CPS by measuring the time Δt it takes to charge an RC circuit (measured in clock cycles) with known resistance R_i:

$$z_i^{(\text{real})} = \frac{-\Delta t}{nfR_i \log(1/2)}$$

where n is the number of samples and f is the clock frequency of the sensor [6]. If the Maxwell capacitance matrix and object positions are known in simulation, the simulated observation vector can be derived using a parallel plate assumption:

$$z_i^{(\text{sim})} = \frac{\alpha_i}{d_i} + \sum_{i \neq j} C_{ij}$$

where α is a vector of dynamic parameters associated with the permittivity and overlapping surface area of a detected object. The raw capacitance measurements can be simulated by through a digital twin of the distributed sensors that respects design parameters such as resistance and mutual coupling [17]. The CPS observation space is a 1D $1 \times s$ vector, where s is the total number of distributed sensors on the robot.

2) Time of Flight Sensing: We will simulate time of flight (ToF) sensors that observe distance measurements along a directional vector. We will model the SparkFun VL53L5CX ToF Imager as a depth camera with 8x8 resolution and a $45°$ field of view (FoV). As specified in the sensor's datasheet, each individual directional depth measurement will output in the following observation, where d is the true distance:

$$z_i = \begin{cases} 0, & d < 0.02\,\text{m or } d > 4\,\text{m} \\ d \pm 15\,\text{mm} & 0.02 < d \leq 0.2\,\text{m} \\ d \pm 0.05d & 0.2 < d \leq 4\,\text{m} \end{cases}$$

The ToF observation space is a 1D $1 \times 64s$ vector.

We additionally test a combination of CPS and ToF sensors, which observe a $1 \times 65s$ vector.

E. MPC based Whole-Body Control

To establish a performance benchmark grounded in analytical methods, we implement a MPC policy in Mujoco MPC (MJPC) [10]. At each control timestep, the planner solves a trajectory optimization problem over a finite-horizon T:

$$\min_{u_{0:T}} \sum_{t=0}^{T} c(x_t, u_t) \tag{3}$$

subject to dynamics $x_{t+1} = f(x_t, u_t, \Delta t)$, where $x_t \in \mathbb{R}^{n_x}$ is the state (positions and velocities), $u_t \in \mathbb{R}^{n_u}$ is the control (joint torques or position targets), and $c(\cdot)$ is the running cost over the finite horizon.

The base cost is defined as:

$$l(x, u) = \sum_{i=0}^{M} w_i \cdot \text{n}_i(r_i(x, u)) \tag{4}$$

Where the base cost is a sum of M terms, each comprising of a tunable weight $w \in \mathbb{R}_+$, a twice-differentiable norm function $\text{n}(\cdot)$, and residual terms $r \in \mathbb{R}^p$. Each residual r_i captures a specific objective (e.g., reaching a target, maintaining balance, minimizing control effort). The weights w_i are tuned to trade off competing objectives.

For this collision avoidance task, we establish the base task to be "standing," adopted from HumanoidBench [30, 22]. We adopt the base residual cost terms, described in App. G and add two simple repulsive cost terms: 1) $c_{\text{prox},i}$, a *n*-dimension CPS/ToF reading, normalized by the configured sensing range (Eqn. 5), and 2) $\mathbf{c}_{\text{CoM}}$, 2-dimension (x, y) distance between the robot center of mass and the estimated obstacle centroid, determined from the weighted average positions of active sensors (Eqn. 6). For policies using both CPS and ToF sensing, we use four cost terms, two for each sensing modality, and weigh CPS costs doubly compared to ToF sensing. More sophisticated cost design is beyond the scope of this project.

$$c_{\text{prox},i} = \begin{cases} 1 - \frac{d_i}{r_{\text{max}}} & \text{if } d_i \leq r_{\text{max}} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where d_i is the distance from sensor i to the obstacle and r_{max} is the configured sensing range. This cost encourages the robot to maintain distance from detected obstacles, with stronger penalties for closer proximity.

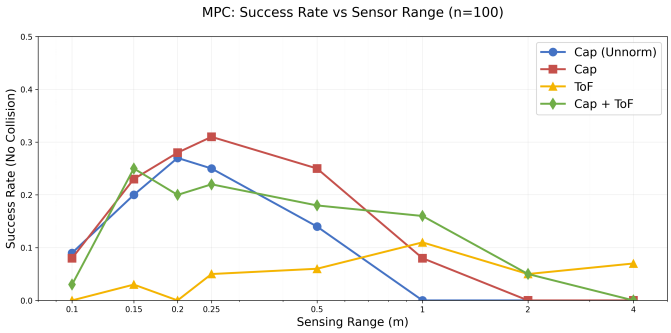
$$\mathbf{c}_{\text{CoM}} = (d_{\text{CoM}} - d_{\text{des}}) \cdot \frac{\mathbf{p}_{\text{CoM}} - \mathbf{p}_{\text{obs}}}{\|\mathbf{p}_{\text{CoM}} - \mathbf{p}_{\text{obs}}\|_2} \tag{6}$$

where $\mathbf{p}_{\text{CoM}} \in \mathbb{R}^2$ is the robot's center of mass position in the xy-plane, $\mathbf{p}_{\text{obs}}$ is the weighted centroid of active sensor detections, $d_{\text{CoM}} = \|\mathbf{p}_{\text{CoM}} - \mathbf{p}_{\text{obs}}\|_2$, and $d_{\text{des}} = 0.2$ m is the desired offset distance. This 2-dimensional cost guides the center of mass away from the estimated obstacle position.

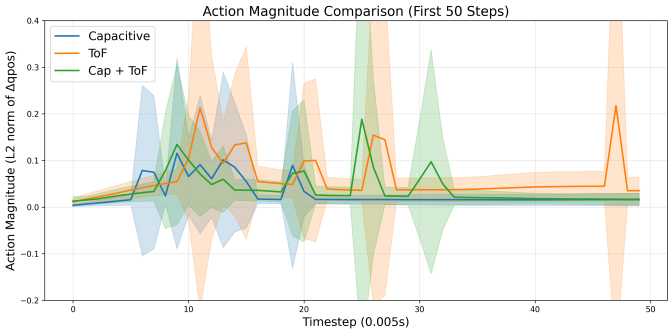
We use the iLQR algorithm [19] to evaluate multiple candidate control trajectories u in parallel with forward differencing (for greater detail, refer to [10]) over a finite horizon T of 1s, producing a control trajectory $u_{0:T}$ which is executed in a receding horizon fashion. MJPC is exclusively CPU-dependent, and we run all experiments on an AMD Ryzen 3700x CPU at 5% simulation speed.

F. Residual RL

Residual reinforcement learning combines the stability and interpretability of model-based control with the adaptability and optimality of learned policies [3, 14, 20, 15]. We implement a reference-tracking "residual" RL policy which learns to track MPC generated trajectories a_t^{ref} and the underlying avoidance task under domain randomization, perturbation, [20]. The best performing policy achieved poor tracking accuracy (average of 45% joint position tracking error) and collision avoidance (4% avoidance). Initial exploration shows partial model collapse with the policy performing only a few, low-action-magnitude evasive behaviors. Investigating and improving this poor performance is beyond this submission's scope,



(a) MPC with simple repulsive costs on sensor values



(b) Average joint action magnitude for first 50 timesteps (0.002s).

Fig. 3: Experimental results comparing sensor configurations. (a) Collision avoidance success rates across varying sensing ranges (n=100 trials each). Capacitance-only MPC achieves the best avoidance of 31% at a sensing range of 0.25 m. All sensor configurations except ToF alone perform worse at longer ranges, though capacitance normalization improves viable range. (a) Action smoothness measured by L2 norm of joint position changes. ToF signals are denser and earlier, disrupting the base standing policy and reducing success.

so we omit it from our results and provide the architectural details and discussion in App. H.

IV. Experiments and Results

We compare our learned RL policies against MPC in an obstacle avoidance task as our primary baseline.

A. Obstacle Avoidance Task

We launch obstacles with randomly generated trajectories to train and quantify the performance of our obstacle avoidance agent. Each projectile is 30 cm in diameter, weighs 100 g (see section F), and launched with an initial velocity between 4-6 m/s at a sampled joint on the robot. Only one projectile is launched at the robot per episode and starts from a random position between 2-3 m away from the robot's center. Success is defined as the percentage of agents that successfully avoid the launched projectile and maintain an upright standing posture until episode termination.

1) MPC Evaluation: Our MPC baseline follows the setup described in section III-E. We evaluate four sensor configurations in Fig. 3 on the same 100 obstacle trajectories: CPS sensing, ToF sensing, combined CPS and ToF sensing, and unnormalized CPS sensing across sensing ranges of: {0.1, 0.15, 0.20, 0.25, 0.50, 1, 2, 4}m. As shown in 3a, capacitance-only MPC achieves the best avoidance overall of 31% at a sensing range of 0.25m, 23% at a realistic sensing range of 0.15m, and an average of 15.4% across all ranges. ToF sensing alone achieves its best avoidance of 11% at a sensing range of 1m, combined CPS and ToF sensing has its best avoidance of 25% at a range of 0.15m, and the unnormalized CPS policy achieves 27% at a range of 0.2m. All sensor configurations perform worse as range increases, though capacitance normalization improves viable range. This is due to the simple MPC cost design: with greater sensing ranges, the controller receives proximity feedback even when the obstacle is quite far away.

Rather than executing large-action-magnitude avoidance motions when an obstacle is near-collision, such as in the CPS-only policies, the robot with long-range proximity sensing performs lower action-magnitude avoidance motions much earlier and continuously. These avoidance motions often conflict with standing costs (posture, height, balance) and MPC often struggles to optimize these conflicting costs for prolonged durations, leading to degenerate behaviors that are not able to avoid the obstacle when it is actually near and almost always lead to a fallen posture. This is evidenced by the fact that combining ToF sensing with CPS sensing yields lower avoidance success than the CPS policy alone and more empirically in Fig. 3b and 7, in which we show the mean action magnitude (L2 norm of change in 21 actuated joints) per sensor configuration over the first 50 and 220 (median trajectory length) timesteps (0.002s), respectively. Across the first 220 steps, the respective mean and standard deviation of action magnitude (μ, σ) for CPS, ToF, and CPS + ToF policies are as follows: (0.029, 0.022), (0.056, 0.036), and (0.039, 0.029). ToF sensing and CPS + ToF sensing policies result in a 93% and 35% increase in mean action magnitude with more extreme initial maneuvers relative to a CPS-only policy; this behavior continually compounds, resulting in an over-sensitive policy that can neither avoid nor stand well.

2) RL Evaluation: The RL policy was able to learn how to successfully dodge incoming obstacles using our approach and achieved a top success rate of 96%. We ablated the performance by sensor type and range and achieved the results seen in fig. 4. The best performance was observed when using CPS sensors with detection ranges greater than 1 cm, reaching success rates above 90% and avoiding termination for around 170s on average. Additionally, we noticed an increase in training speed using CPS sensors compared to ToF sensors, where a policy would only require < 500 steps to converge with capacitive sensors, ≈ 1000 steps for ToF sensors, and ≈ 1700 steps with a combination of both. Although the CPS range surpasses its practical sensor limitations, it reveals that the lower dimensional relative distance measurements from CPS's (which are orientation agnostic) may be more practically efficient for obstacle avoidance than the dense, orientation locked relative distance measurements from ToF sensors. Since ToF sensors were anticipated to perform better

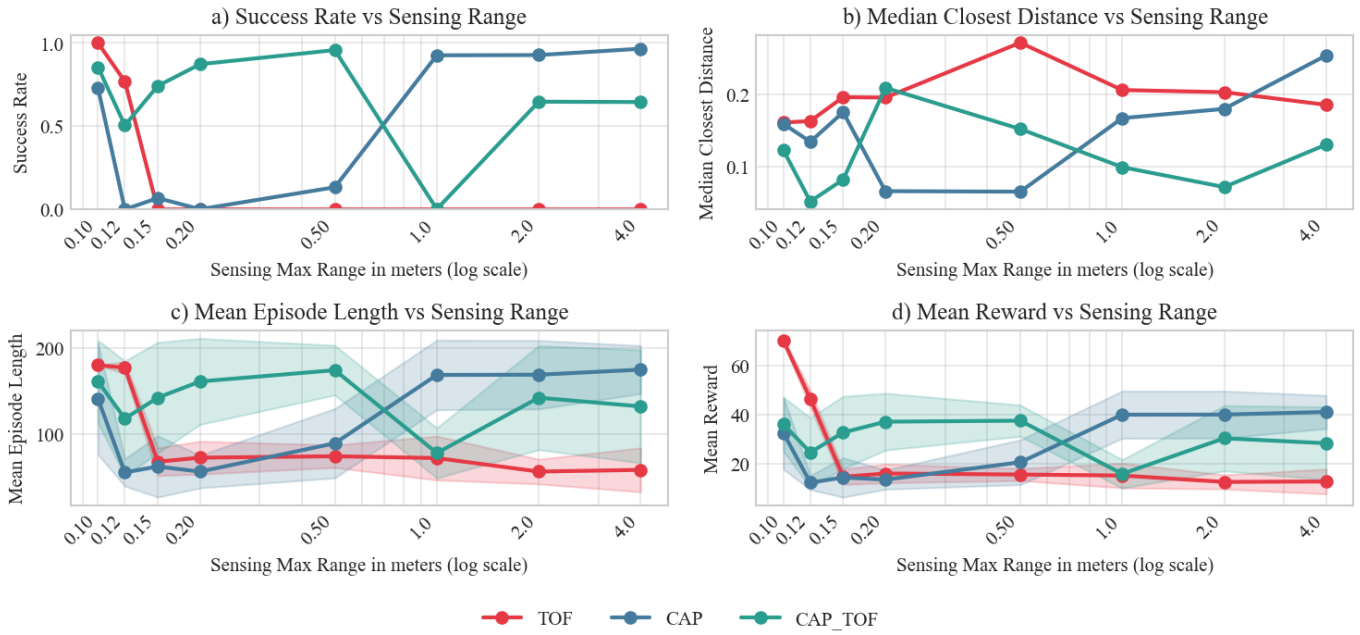


Fig. 4: Collision avoidance performance of policies trained on ToF, capacitive, and a combination of ToF and capacitive sensors at varying sensing ranges (4096 trials each). a) The percentage of trials that ended without termination due to falling or getting hit by the projectile. b) The median (by trials) of closest distance to the obstacles observed by the sensors. c) The length of time the agent survived until termination. d) The accumulated reward of each trial.

at longer distances, we only varied the CPS sensing range when using a combination of both and kept the ToF range at 4.0 m. Although neither CPS or ToF sensors performed particularly well in ranges under 1 m range, training agents on a combination of both achieved success rates greater than 75% in the 0.15-0.5 m CPS sensing range. CPS sensors reliably detect objects under 0.15 m in practice, so a combination of capacitive and ToF sensors may be the optimal coverage for deploying the policy on a real robot.

B. Discussion

One interpretation of the RL results compared to MPC is that the RL policy strongly benefits from having information about the incoming projectile far in advance whereas MPC is advantageous in quick, reactive motions. The RL policy commonly learned multi-step motions such as planting its feet in an advantaged positions to jump out of the way, whereas MPC would generate highly unstable reactions if it avoided the obstacle too early. These results support our original hypothesis that the RL agent learned fluent motions that are difficult to describe using optimal control constraints. Although both control methods have strengths at different sensing ranges, they both suffered in performance when using ToF sensing compared to capacitive sensing.

V. Conclusion

a) Future Work: A potential limitation of the ToF sensors may have been that information was too densely reported, slowing down the training speed and quality of the agent for the RL policy. Future work should include methods to process egocentric proximity data prior to being passed into the RL model and MPC module to reduce the data size and only extract necessary features for planning. This may include using spatially aware convolutions, Kalman filtering, and tracking temporal data such as rate of measurement changes to better estimate the state of incoming objects.

Additionally, we hope to explore sensing modalities such as RGB cameras or LIDAR sensors mounted in a similar distributed fashion. These sensors capture much denser and richer detail but also require complex signal processing and representation methods. Future work should explore the trade-offs and emergent behaviors of sensors at varying spatial, temporal resolution.

b) Bridging the Sim-to-Real Gap: The largest and most imminent goal is to transfer the trained RL policies and ideally MPC to a real robot. This process entails (non-exhaustive):

- Sensor layout design and fabrication: The sensors used in this study were randomly generated, however, more considerations into wiring and deployment are needed.
- Real sensor characterization: This study assumed noiseless sensors that report distance directly. For real capacitive sensors, an accurate distance measurement requires a well calibrated profile of the parasitic capacitance between sensors and some preset bias about the detected obstacle such as material and shape.
- Construction of a safe experiment apparatus: cushioned cage and floor, ceiling-mounted belay, a configurable and portable method for obstacle launching

References

[1] Simon Armleder, Florian Bergner, Julio Rogelio Guadarrama-Olvera, Jun Nakanishi, and Gordon Cheng. Real-time control of a humanoid robot for whole-body tactile interaction. *Advanced Intelligent Systems*, n/a(n/a):2500149. doi: https://doi.org/10.1002/aisy.202500149. URL https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202500149.

[2] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE International Symposium on System Integrations (SII)*, pages 614–619, 2019. URL https://ieeexplore.ieee.org/document/8700380.

[3] Jin Cheng, Dongho Kang, Gabriele Fadini, Guanya Shi, and Stelian Coros. Rambo: Rl-augmented model-based whole-body control for loco-manipulation, 2025. URL https://arxiv.org/abs/2504.06662.

[4] Andrea Del Prete, Nicolas Mansard, Oscar E Ramos, Olivier Stasse, and Francesco Nori. Implementing torque control with high-ratio gear boxes and without joint-torque sensors. *International Journal of Humanoid Robotics*, 13(01), 2016. URL https://hal.science/hal-01136936v2/file/paper_201403_hal.pdf.

[5] Caleb Escobedo, Matthew Strong, Mary West, Ander Aramburu, and Alessandro Roncone. Contact anticipation for physical humanrobot interaction with robotic manipulators using onboard proximity sensors. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7255–7262, 2021. doi: 10.1109/IROS51168.2021.9636130.

[6] Diogo Fonseca, Mohammad Safeea, and Pedro Neto. A flexible piezoresistive/self-capacitive hybrid force and proximity sensor to interface collaborative robots. *IEEE Transactions on Industrial Informatics*, 19(3):2485–2495, 2022.

[7] Xinyang Gu, Yen-Jen Wang, and Jianyu Chen. Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer. *arXiv preprint arXiv:2404.05695*, 2024.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

[9] Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimminger, Stefan Schaal, and Ludovic Righetti. Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. In *Autonomous Robots*, volume 40, pages 473–491, 2015. URL https://arxiv.org/abs/1410.7284.

[10] Taylor Howell, Nimrod Gileadi, Saran Tunyasuvunakool, Kevin Zakka, Tom Erez, and Yuval Tassa. Predictive sampling: Real-time behaviour synthesis with mujoco, 2022. URL https://arxiv.org/abs/2212.00541.

[11] Wei-Cheng Huang, Alp Aydinoglu, Wanxin Jin, and Michael Posa. Adaptive contact-implicit model predictive control with online residual learning, 2024. URL https://arxiv.org/abs/2310.09893.

[12] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), January 2019. ISSN 2470-9476. doi: 10.1126/scirobotics.aau5872. URL http://dx.doi.org/10.1126/scirobotics.aau5872.

[13] Koji Ishihara, Hiroaki Gomi, and Jun Morimoto. Hierarchical learning framework for whole-body model predictive control of a real humanoid robot, 2024. URL https://arxiv.org/abs/2409.08488.

[14] Se Hwan Jeon, Ho Jae Lee, Seungwoo Hong, and Sangbae Kim. Residual mpc: Blending reinforcement learning with gpu-parallelized model predictive control, 2025. URL https://arxiv.org/abs/2510.12717.

[15] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control, 2018. URL https://arxiv.org/abs/1812.03201.

[16] Daehwa Kim, Mario Srouji, Chen Chen, and Jian Zhang. Armor: Egocentric perception for humanoid robot collision avoidance and motion planning. *arXiv preprint arXiv:2412.00396*, 2024.

[17] Carson Kohlbrenner, Caleb Escobedo, S Sandra Bae, Alexander Dickhans, and Alessandro Roncone. Gentact toolbox: A computational design pipeline to procedurally generate context-driven 3d printed whole-body artificial skins. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[18] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), October 2020. ISSN 2470-9476. doi: 10.1126/scirobotics.abc5986. URL http://dx.doi.org/10.1126/scirobotics.abc5986.

[19] Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *International Conference on Informatics in Control, Automation and Robotics*, 2004. URL https://api.semanticscholar.org/CorpusID:19300.

[20] Fukang Liu, Zhaoyuan Gu, Yilin Cai, Ziyi Zhou, Hyunyoung Jung, Jaehwi Jang, Shijie Zhao, Sehoon Ha, Yue Chen, Danfei Xu, and Ye Zhao. Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation, 2025. URL https://arxiv.org/abs/2409.20514.

[21] Carlos Mastalli, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hammoud, Maximilien Naveau, Justin Carpentier, Ludovic Righetti, Sethu Vijayakumar,

and Nicolas Mansard. Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[22] Moritz Meser, Aditya Bhatt, Boris Belousov, and Jan Peters. Mujoco mpc for humanoid control: Evaluation on humanoidbench, 2024. URL https://arxiv.org/abs/2408.00342.

[23] Hirofumi Miura and Isao Shimoyama. Dynamic walk of a biped. *The International Journal of Robotics Research*, 3(2):60–74, 1984. doi: 10.1177/027836498400300206. URL https://doi.org/10.1177/027836498400300206.

[24] Masaki Murooka, Mitsuharu Morisawa, and Fumio Kanehiro. Centroidal trajectory generation and stabilization based on preview control for humanoid multi-contact motion. *IEEE Robotics and Automation Letters*, 7(3):8225–8232, 2022. URL https://ieeexplore.ieee.org/document/9807369/.

[25] Masaki Murooka, Kensuke Fukumitsu, Marwan Hamze, Mitsuharu Morisawa, Hiroshi Kaminaga, Fumio Kanehiro, and Eiichi Yoshida. Whole-body multi-contact motion control for humanoid robots based on distributed tactile sensors. *IEEE Robotics and Automation Letters*, 9(11):1062010627, November 2024. ISSN 2377-3774. doi: 10.1109/lra.2024.3475052. URL http://dx.doi.org/10.1109/LRA.2024.3475052.

[26] Stefan Escaida Navarro, Stephan Mühlbacher-Karrer, Hosam Alagi, Hubert Zangl, Keisuke Koyama, Björn Hein, Christian Duriez, and Joshua R Smith. Proximity perception in human-centered robotics: A survey on sensing systems and applications. *IEEE Transactions on Robotics*, 38(3):1599–1620, 2021.

[27] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning, 2017. URL https://arxiv.org/abs/1710.06542.

[28] Angel Romero, Elie Aljalbout, Yunlong Song, and Davide Scaramuzza. Actor-critic model predictive control: Differentiable optimization meets reinforcement learning, 2025. URL https://arxiv.org/abs/2306.09852.

[29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[30] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation, 2024. URL https://arxiv.org/abs/2403.10506.

[31] Ivica Smolić and Bruno Klajn. Capacitance matrix revisited. *arXiv preprint arXiv:2007.10251*, 2020.

[32] Yuquan Wang and Lihui Wang. Whole-body collision avoidance control design using quadratic programming with strict and soft task priorities. *Robotics and Computer-Integrated Manufacturing*, 62:101882, 2019.

[33] Patrick M. Wensing, Michael Posa, Yue Hu, Adrien Escande, Nicolas Mansard, and Andrea Del Prete. Optimization-based control for dynamic legged robots. *IEEE Transactions on Robotics*, 40:43–63, 2024. doi: 10.1109/TRO.2023.3324580.

[34] Xiaomeng Xu, Dominik Bauer, and Shuran Song. Robopanoptes: The all-seeing robot with whole-body dexterity, 2025. URL https://arxiv.org/abs/2501.05420.

[35] Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C. Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction, 2025. URL https://arxiv.org/abs/2509.26633.

[36] John Z. Zhang, Taylor A. Howell, Zeji Yi, Chaoyi Pan, Guanya Shi, Guannan Qu, Tom Erez, Yuval Tassa, and Zachary Manchester. Whole-body model-predictive control of legged robots with mujoco, 2025. URL https://arxiv.org/abs/2503.04613.

[37] Tong Zhang, Boyuan Zheng, Ruiqian Nai, Yingdong Hu, Yen-Jen Wang, Geng Chen, Fanqi Lin, Jiongye Li, Chuye Hong, Koushil Sreenath, and Yang Gao. Hub: Learning extreme humanoid balance, 2025. URL https://arxiv.org/abs/2505.07294.

[38] Yuanhang Zhang, Yifu Yuan, Prajwal Gurunath, Tairan He, Shayegan Omidshafiei, Ali akbar Agha-mohammadi, Marcell Vazquez-Chanlatte, Liam Pedersen, and Guanya Shi. Falcon: Learning force-adaptive humanoid loco-manipulation, 2025. URL https://arxiv.org/abs/2505.06776.

APPENDIX A
HYPERPARAMETERS & SETTINGS FOR RL

This appendix details the hyperparameters used for the environment, the PPO algorithm, and the ablation study configuration.

TABLE I: Hyperparameter and Environment Settings

Simulation & Environment	
Physics Engine	PhysX (Isaac Lab)
Simulation dt	0.0083s (120 Hz)
Control Decimation	2
Control Policy Frequency	60 Hz
Episode Length	3.0s (180 steps)
Num. Environments	4096
Network Architecture (Baseline)	
Actor Network	MLP $[64, 64]$
Critic Network	MLP $[64, 64]$
Activation Function	ELU
PPO Optimization	
Optimizer	Adam
Learning Rate	1×10^{-3}
Discount Factor (γ)	0.99
Clip Range (ϵ)	0.2
Entropy Coefficient	0.0
Mini-batch Size	4096

APPENDIX B
REWARD FORMULATION

Table II lists the exact mathematical formulations for the reward terms defined in the MDP.

APPENDIX C
ABLATION CONFIGURATION

Table III describes the parameters varied to test the hypotheses regarding network complexity and sensor modality.

APPENDIX D
NEURAL NETWORK ARCHITECTURE ABLATION STUDY

An ablation study was conducted to evaluate five neural network architectures across four performance metrics. The tested variants are summarized in Table IV.

The baseline actor and critic networks consist of two fully-connected layers with 32 neurons each and ELU activation (see Appendix E for detailed architecture).

TABLE II: Reward Composition and Scaling

Component	**Expression**	**Weight**
Stabilization & Posture		
Base Height	$\exp(-5.0(z_{\text{base}} - 1.04)^2)$	10.0
Base Velocity	$\exp(-10.0\|v_{xy}\|^2)$	10.0
Stand Still	$\exp(-100.0\|v_{xy}\|^2)$	10.0
Alive Bonus	1.0	5.0
Collision Avoidance		
Projectile Proximity	$\min_{i \in \text{links}} \left[\beta_i \cdot \text{clip}\left(w_p(1 - \frac{d_i}{d_{\max}}), w_p, 0\right)\right]$	1.0
$d_{\max} = 2.0$m, $w_p = -1.0$. Multiplier $\beta_i = 3.0$ for Lidar, 1.0 otherwise.		
Regularization		
Joint Acceleration	$-\|\ddot{q}\|^2$	−2.5e-7
Action Rate	$-\|a_t - a_{t-1}\|^2$	−0.005
Joint Limits	Soft limit penalty	−3.0

TABLE III: Ablation Study Parameters

Parameter	**Default**	**Description**
`MAX_RANGE`	4.0 m	Max detection range
`SENSOR_TYPE`	TOF, CAP, Both	Active sensor modality
`Policy_Hidden`	$[64, 64]$	MLP Dimensions

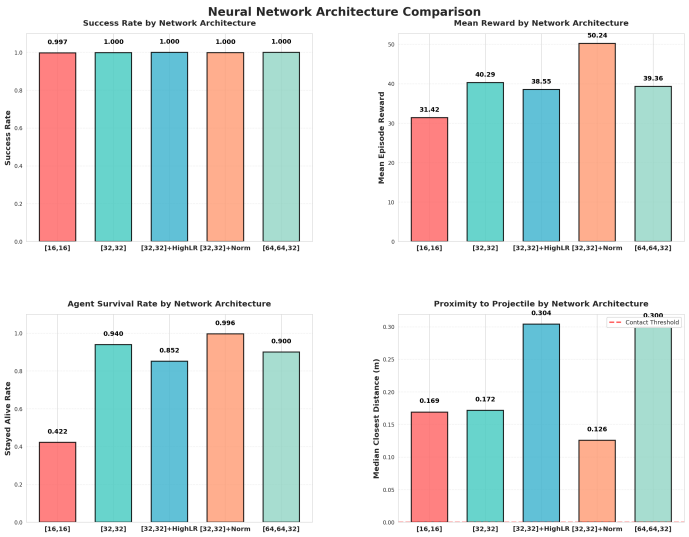


Fig. 5: Architecture comparison

APPENDIX E
NETWORK ARCHITECTURE DETAILS

```
ActorCritic Network Architecture

Actor MLP:
  Linear(in_features  32), ELU
  Linear(32  32), ELU
  Linear(32  21)  # joint position commands
```

TABLE IV: Neural Network Architecture Variants

Architecture Configuration	
Small	$[16, 16]$ neurons
Medium	$[32, 32]$ neurons
Large	$[64, 64, 32]$ neurons
Normalized	$[32, 32]$ + obs normalization
HighLR	$[32, 32]$ + LR 3.0×10^{-3}

```
Critic MLP (asymmetric):
  Linear(in_features + priv  32), ELU
  Linear(32  32), ELU
  Linear(32  1)
```

The critic network additionally incorporates privileged information (ground truth state) during training for improved value estimation.

APPENDIX F
PROJECTILE MASS ABLATION

The trained agents have a tendency to learn a planted, hard stance when the projectile mass is set too high. Although a strong stance may be preferable, we found that it negatively impacts the agent's training and preference for dodging obstacles as opposed to taking the hit. 100 g produced the strongest results and was used for this study.

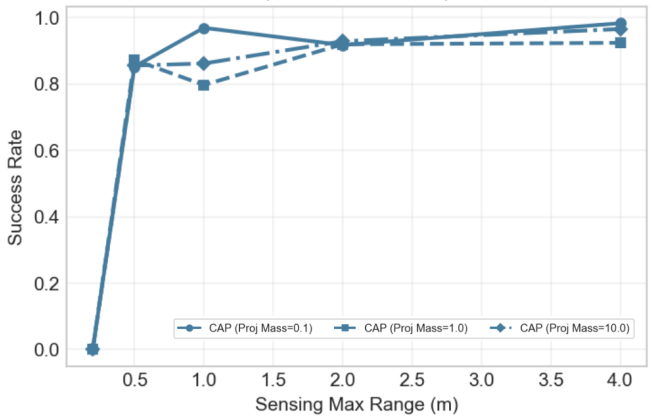


Fig. 6: Performance of the capacitance RL policy when trained on a variety of projectile masses. A projectile mass of 100 g was chosen both empirically and by inspection of policy quality.

APPENDIX G
MPC DESIGN

a) Cost Terms:

b) Robot Model: We use the Unitree H12 humanoid without hands or articulated wrist joints which has 21 controllable joints. The model can be accessed at this link, with 63 skin sensor sites. We disable five ToF sensor sites with XML configuration issues and leave re-integration to future work.

TABLE V: Cost terms and weights for humanoid standing and obstacle avoidance task in MPC. Standing terms are adopted from HumanoidBench [30, 22].

Category	Term	Expression	Weight
		Standing	
	Height	$\lvert h_{\text{torso}} - h_{\text{goal}} \rvert$	30.0
	CoM velocity	$\lVert \mathbf{v}_{\text{CoM},xy} \rVert_2$	10.0
	Joint velocity	$\lVert \dot{\mathbf{q}} \rVert_2$	0.01
	Balance	$\lVert \mathbf{p}_{\text{CP}} - \mathbf{p}_{\text{support}} \rVert_2$	10.0
	Upright	$\lVert \mathbf{R}_{\text{torso}} - \mathbf{R}_{\text{world}} \rVert_F$	5.0
	Posture	$\lVert \mathbf{q} - \mathbf{q}_{\text{home}} \rVert_2$	0.03
	Foot velocity	$\lVert \mathbf{v}_{\text{feet}} - \mathbf{v}_{\text{CoM}} \rVert_2$	0.625
	Control	$\lVert \mathbf{u} - \mathbf{q}_{\text{home}} \rVert_2$	0.1
		Obstacle Avoidance	
	CPS proximity	$\sum_{i=1}^{63} c_{\text{prox},i}^{\text{cap}}$ (Eq. 5)	200.0
	CPS CoM offset	$\lVert \mathbf{c}_{\text{CoM}}^{\text{cap}} \rVert_2$ (Eq. 6)	100.0
	ToF proximity	$\sum_{i=1}^{58} c_{\text{prox},i}^{\text{tof}}$ (Eq. 5)	100.0
	ToF CoM offset	$\lVert \mathbf{c}_{\text{CoM}}^{\text{tof}} \rVert_2$ (Eq. 6)	50.0

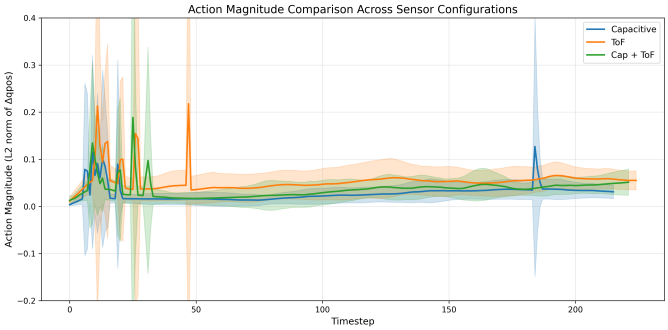


Fig. 7: L2 norm of action (change in 21 actuated joints) over the first 220 timesteps for each sensor configuration.

c) MPC Parameters.:

- Horizon: 66 timesteps 0.015s timestep = 1s
- Planner: iLQR
- Control rate: 0.002s (500Hz)
- Spline points: 3 (piecewise linear interpolation)

d) Action Magnitude Across First 250 Timesteps: Fig. 7

APPENDIX H
REFERENCE-TRACKING (RESIDUAL) RL

We adopt a trajectory-guided reinforcement learning approach inspired by Opt2Skill [20], which leverages model-based planning to generate high-quality demonstration data for training reactive policies. Unlike traditional residual RL methods that execute model-based control online and learn corrective actions [3], our approach uses MPC trajectories purely as training supervision, enabling the learned policy to operate independently without requiring online optimization at deployment.

Trajectory Generation. We use MuJoCo MPC to solve the obstacle avoidance task offline, generating an initial dataset of 25 dynamically feasible reference trajectories. Each trajectory

includes time-indexed sequences of joint positions $\mathbf{q}_t^{\text{ref}}$, velocities $\dot{\mathbf{q}}_t^{\text{ref}}$, base velocities $\mathbf{v}_t^{\text{ref}}$, and obstacle states. We augment these trajectories 100x by sampling new obstacle trajectories that constitute a valid, avoided obstacle: it comes within the sensing range of any sensor and does not collide at any point in the trajectory. These trajectories are stored in a database and randomly sampled during training to provide varied obstacle approach scenarios.

Policy Architecture. The learned policy π_θ directly outputs joint-level actions without relying on MPC at execution time. Following asymmetric actor-critic design, the actor observes realistic onboard sensor data:

- Proprioception: IMU orientation, joint positions/velocities, action history
- Exteroception: Capacitive skin sensor readings (63 sites), obstacle centroid estimated from capacitance spatial distribution
- Reference trajectory: Target joint positions $\mathbf{q}_t^{\text{ref}}$, velocities $\dot{\mathbf{q}}_t^{\text{ref}}$, and base velocities $\mathbf{v}_t^{\text{ref}}$ at the current timestep

The critic additionally receives privileged information unavailable during deployment: precise obstacle position, velocity, and ground-truth collision states. This asymmetry enables more stable value estimation while ensuring the policy remains deployable with only onboard sensing.

Reward Function. The reward combines trajectory tracking objectives with safety and regularization terms:

$$r_t = \underbrace{w_{\text{pos}} \cdot r_{\text{pos}}(\mathbf{q}_t, \mathbf{q}_t^{\text{ref}}) + w_{\text{vel}} \cdot r_{\text{vel}}(\dot{\mathbf{q}}_t, \dot{\mathbf{q}}_t^{\text{ref}})}_{\text{Trajectory tracking}} - \tag{7}$$

$$\underbrace{w_{\text{col}} \cdot \mathbb{I}_{\text{collision}} + w_{\text{reg}} \cdot \|\mathbf{a}_t\|^2}_{\text{Safety \& regularization}} \tag{8}$$

where r_{pos} and r_{vel} are exponential tracking rewards that encourage the policy to follow the reference trajectory, while collision penalties and action regularization maintain safety margins. After the reference trajectory completes, recovery rewards encourage the robot to return to a stable standing configuration.

Training We train policies following Opt2Skill: PPO optimization, 300 million steps, and 4-layer MLPs [512, 512, 256, 256] for both actor and critic. Training one policy takes approximately 30 minutes on a NVIDIA RTX4070.

This formulation ideally enables the policy to learn obstacle avoidance behaviors from MPC demonstrations while developing reactive responses based on real-time capacitive sensing, combining the motion quality of model-based planning with the adaptability of learned control. However, since only 25 original trajectories are tracked, this extreme data scarcity most likely led to poor data performance, as resulting policies had fairly uniform, non-diverse motions. Opt2Skill generates 2000 trajectories per skill (walking, leaning and pushing, box pick-up), and the avoidance task inherently has more diverse motions than those skills. Likely $>>$2000 high-quality trajectories may be required to learn robust avoidance behavior. The code for the MuJoCo Playground policy training is available at this link.