

PAC-MAN: Perception-Aware CBF-RL for Whole-Body Safety in Humanoid Dodgeball

Lizhi Yang, Junheng Li and Aaron D. Ames

Abstract—We present PAC-MAN, a perception-aware CBF-RL framework that couples control-barrier safety with deployment-realistic onboard sensing for whole-body humanoid dodgeball. The deployed policy sees the ball only as segmentation-masked depth from a head-mounted camera, while training-time CBF guidance represents clearance to every body link, and an adversarial motion prior regularizes the resulting evasive reflexes. We evaluate on a controlled any-link contact benchmark with seeded throws in two regimes: single throws and a deployment loop in which the robot walks back to its station and recovers between throws. On this benchmark, the policy comes within a few points of a privileged state oracle: a fixed onboard camera alone is adequate for evasion. We find that usable barrier structure depends on perceptual observability: Joint-CBF gives the best performance with accurate ball states, degrades under fixed-camera observations when used only as training guidance, and recovers with a ball-tracking gimbal or privileged runtime filter. We therefore deploy a lightweight Link-CBF policy zero-shot on the Unitree G1 in the real world, where it tolerates imperfect perception, succeeds on 95% of throws, and uses semantic segmentation to dodge different balls.

I. INTRODUCTION

Humanoid robots operating around moving objects must react within a fraction of a second, and the evasion itself must not cost them their balance. Dodgeball isolates this safety problem in a short-horizon, whole-body setting: the robot must perceive an incoming ball, determine which links are threatened, and coordinate the torso, arms, and legs quickly enough to avoid contact while remaining upright. Humans solve the same problem by keeping their eyes on the ball and moving their whole body out of its path. Recent learning-based controllers can produce dynamic humanoid locomotion [1], [2], whole-body imitation [3]–[5], and visually conditioned skills [6], [7], but reactive whole-body safety from onboard vision remains difficult [8], [9], especially for a humanoid.

Control barrier functions (CBFs) encode collision-free sets and correct unsafe commands, provided the relevant state is available [10]. Onboard perception breaks this convenient assumption: the incoming object may be visible only briefly, may leave the camera field of view, or may appear as a few noisy depth pixels. CBF-RL can move barrier information into policy training [11], but a deployed policy can internalize only as much safety structure as its observations support. The threat representation and the barrier design therefore

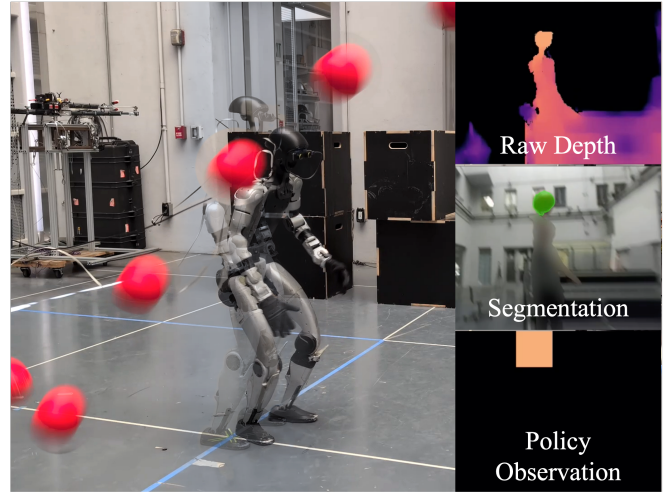


Fig. 1: Perception-aware dodgeball couples safety and sensing: a humanoid must infer an incoming threat from a head-mounted depth camera, move every threatened link out of its path, and preserve balance without privileged ball states at deployment.

have to be treated as one coupled safety problem rather than as independent modules.

We study this coupling through perception-aware humanoid dodgeball (Fig. 1). A ball is launched toward a humanoid robot, which must avoid contact with every body link and preserve balance using only proprioception and onboard depth at deployment. Unlike navigation around a mapped obstacle, the hazard is transient, and the robot has no nominal velocity command or reference motion that specifies when and how to evade. The short observation window and whole-body collision geometry make dodgeball a direct test of safety under partial observation.

We present PAC-MAN, a perception-aware CBF-RL framework for humanoid dodgeball. The central question is how much barrier structure a policy can internalize when it receives only deployment-realistic observations. PAC-MAN studies two levels: Link-CBF, a lightweight per-link reward used during training, and Joint-CBF, a stronger whole-body joint-space projection that can either guide training or stay active at test time as a privileged runtime filter. The two levels separate learned barrier-guided behavior from online barrier enforcement.

How well the learned reflex works depends on what perception tells the policy about the threat. The deployed system segments the ball in RGB and masks the depth image down to a compact ball-only observation. An active gimbal, studied in simulation, tests whether keeping the threat in

California Institute of Technology, CA 91106, USA

Corresponding author email: lzyang@caltech.edu. This work is supported by The Dow Chemical Company project #227027AW and in part by Technology Innovation Institute.

view for longer makes the stronger barrier structure usable. An adversarial motion prior (AMP) regularizes how the humanoid moves, so the evasions come out as crouches, leans, sidesteps, and leaps; the safety structure itself comes from the CBF terms.

We use a controlled, contact-based benchmark to ask three questions: whether barrier guidance improves whole-body avoidance, how much barrier structure a policy can internalize without a runtime filter, and how the available perception changes the answer. The benchmark calibrates on-target throws against a frozen statue and covers both single throws and a deployment loop in which the robot walks back to its station and recovers between throws. Across a privileged state oracle, a fixed onboard camera, and an oracle-aimed active gimbal, we find that stronger barrier structure needs more informative observations. The full Joint-CBF with runtime filtering is strongest when accurate ball states are available, whereas the lighter Link-CBF is the best deployable option under limited onboard perception. We validate this design choice by deploying the fixed-camera Link-CBF policy zero-shot on a Unitree G1 using only onboard depth and proprioception.

A. Contributions

Our contributions are as follows:

- we introduce perception-aware CBF-RL for whole-body humanoid reaction, pairing deployment-realistic observations with Link-CBF and Joint-CBF safe sets over every body link; the policy can internalize these structures during training, or a runtime filter can enforce them when state information permits.
- we evaluate under an any-link contact criterion and show that the appropriate barrier mechanism depends on the threat representation: Joint-CBF benefits from accurate ball state and oracle-aimed tracking, while Link-CBF remains effective with a fixed camera and no runtime filter.
- we deploy the fixed-camera Link-CBF policy zero-shot on the Unitree G1 using only onboard depth and proprioception; the robot succeeds on 95% of throws, and semantic segmentation lets the same policy evade different balls. We will release the benchmark and training pipeline.

B. Related Work

Motion imitation and priors for humanoid control. Motion imitation has long provided a practical route to naturalistic, high-dimensional control. [12] showed that RL can track reference motions reliably, [13] replaced explicit tracking with an adversarial discriminator over motion features, which yields a style reward compatible with a task objective, and [14], [15] learned reusable skill and style embeddings over large motion datasets. These build on a broader body of legged learning [16]–[18] trained from task rewards, command tracking, and domain randomization. Recent whole-body tracking systems move closer to our humanoid setting:

[19] investigates dexterous whole-body control, [20] uses optimized trajectories to train policies, and [5] constructs a standardized whole-body imitation training pipeline. Retargeting quality also matters here, both for general humanoid tracking [21] and for interaction-preserving loco-manipulation [22]. For dodgeball, a motion prior is attractive because the reward for “not being hit” is too semantically sparse and geometric to specify how a humanoid should move. We select human dodging motions as priors and also condition the policy on a perception stream, thus combining the motion prior with exteroceptive information to produce whole-body evasive reflexes.

Perception-aware safety, parkour, and fast-object tasks.

The closest conceptual neighbor to perception-aware dodgeball is safe agility under partial exteroception. Quadruped parkour [23], [24] and agile-but-safe locomotion [8] show how perception, skill selection, recovery, and safety layers can support fast motion around obstacles, while humanoid parkour and perception-aware locomotion systems [2], [25]–[27] extend this to higher-dimensional bodies with depth- or elevation-conditioned policies for jumping, leaping, and terrain traversal. Related work studies the same perception–speed–stability tradeoff through depth conditioning [7], online policy switching [28], and active gaze [29]. Dodgeball differs from these terrain-centered settings because the hazard is a fast-moving object rather than traversable terrain. The ball may be visible only briefly, so the policy must forecast imminent body contact and produce an evasive whole-body response instead of merely choosing footholds. Task-centered fast-object work makes this timing pressure explicit: [30] uses humanoid table tennis to study sub-second ball prediction, task planning, and whole-body RL control. Dodgeball shares the timing but changes the objective from striking to avoiding contact. This shift stresses egocentric observability: an onboard depth camera must provide enough information for the body to leave the ball’s path before impact.

Whole-body avoidance and safety. Reactive safety spans both recovering balance after a disturbance via capture-point and push-recovery control [31], [32], and avoiding contact in the first place via control barrier functions [10], collision-free MPC [33], and fast collision-free motion generation [34]. Most related to our safety treatment, [11] injects safety structure during humanoid training and removes the runtime filter at deployment, and [9] concurrently explores egocentric tactile and proximity sensing as a modality complementary to the vision used here. We study depth-based dodgeball on the Unitree G1, using perception-aware CBF-RL to couple onboard visual observations with per-link safety guidance and motion-regularized whole-body evasion.

II. SAFETY-CRITICAL PERCEPTION-AWARE DODGEBALL

We formulate perception-aware dodgeball as a safety-critical whole-body reaction problem under partial observation. The robot must avoid contact with an incoming object while preserving balance, yet its deployed policy has only

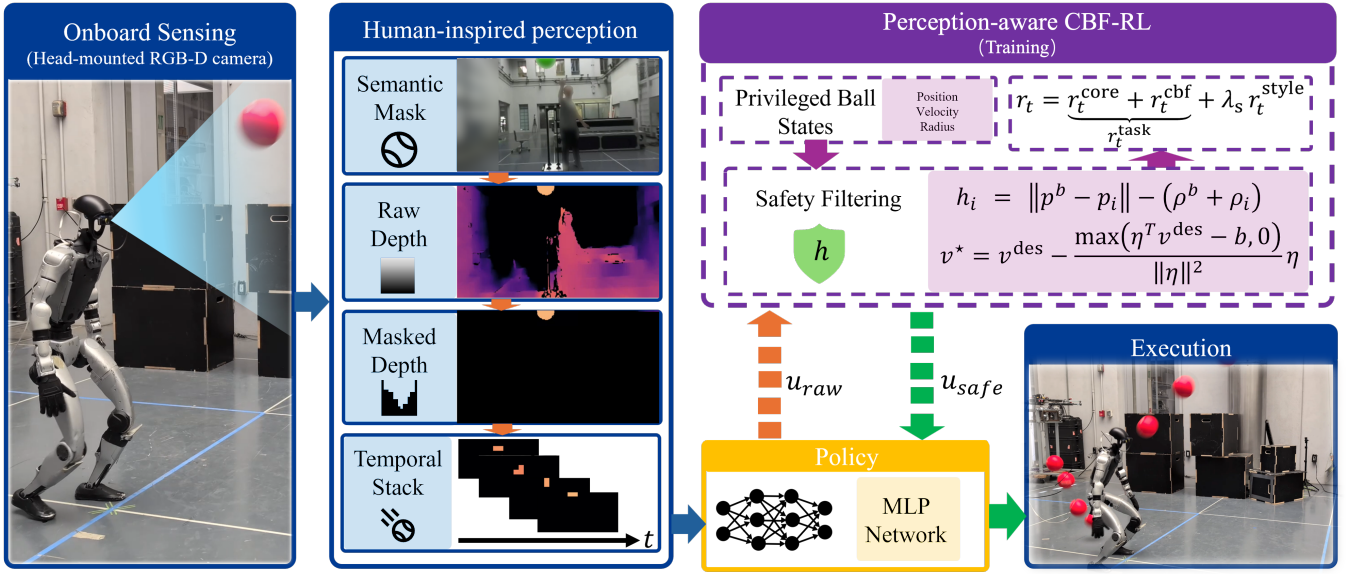


Fig. 2: Pipeline overview. The policy maps temporally stacked ball-only depth and proprioception to 29-D joint targets; during training, ball-to-link geometry supplies whole-body CBF guidance that the policy internalizes for deployment.

limited onboard sensing. This information constraint affects both which safety mechanisms we can use and how we measure success.

A. Task and observations

A ball is launched toward a standing humanoid, which must avoid being struck while preserving balance. We model the task as a partially observed Markov decision process. At control step t , the policy $\pi_\theta(a_t | o_t)$ emits joint-position targets $a_t \in \mathbb{R}^{29}$ (offsets from a nominal pose, tracked by PD actuators at 50 Hz).

The defining constraint is deployability: the policy observation o_t contains only signals computable on hardware from onboard sensing,

$$o_t = (\mathcal{D}_t, q_t, \dot{q}_t, \omega_t, g_t, a_{t-1}), \quad (1)$$

where $\mathcal{D}_t$ is the depth observation, $q_t, \dot{q}_t$ are joint positions/velocities, ω_t is the base angular velocity, g_t is the projected gravity, and a_{t-1} is the previous action. For the camera-based policies, the ball's state is never part of o_t ; whatever the policy knows about the ball, it must infer from depth.

B. Projectile threat model

Balls are thrown on an intermittent timer that leaves a recovery window between throws. The default distribution launches from a frontal cone (within $\pm 25^\circ$ of heading, 2-3 m ahead) with a flight time of roughly 0.6 s, which sets the reaction window; the aim point leads the robot's velocity. We mix two threat types equally so the policy learns both evasions: a descending ball that falls across the legs (sidestep/step-over) and a low-arc ball that rises to torso/head height (duck/lean). Throw heights are calibrated against a frozen-statue control (§IV) so that essentially every throw is on-target. We partition environments at reset into 20% ball-free standing anchors and 80% ball-thrown evade episodes.

C. Whole-body safety metric

The robot is hit if the ball contacts any body link and falls if the torso orientation, base height, or sustained crouch crosses a threshold. A trial succeeds if neither event occurs within the reaction window. The benchmark reports reset and deployment regimes in §IV; reset success is the primary single-throw metric. This contact-based score is intentionally stricter than measuring the distance from the ball to the pelvis alone: many failures are limb grazes, so a method that only protects the base can appear safe while still being struck.

III. METHODS

A. Safety-guided learning

PAC-MAN trains a perception-aware policy to map partial onboard observations to whole-body actions shaped by barrier information available only during training. Privileged ball and link states instantiate the barrier conditions, while the policy receives only the onboard observation o_t of Eq. (1). We use CBF-RL to inject violations of these conditions into policy optimization [11], so the policy internalizes the safety structure during training and needs no privileged geometry at deployment. Link-CBF is a lightweight per-link reward that extends the pelvis-centered task objective to every body link, while Joint-CBF is a stronger joint-space projection that can shape training or remain active as a privileged runtime filter. Segmentation-masked depth is the threat representation we deploy, and an adversarial motion prior regularizes the resulting reactions without defining the safe set.

Figure 2 shows the full pipeline. The full reward is

$$r_t = \underbrace{r_t^{\text{core}} + r_t^{\text{cbf}}}_{r_t^{\text{task}}} + \lambda_s r_t^{\text{style}}, \quad (2)$$

where the task reward pairs the distance-to-core evasion term r_t^{core} of prior humanoid dodgeball work [15] with the

control-barrier term r^{cbf} . The reward also contains station-hold, posture, and smoothness terms that we omit from the notation, and r^{style} regularizes the form of the motion (§III-D). Table I gives the full reward stack with formulations and weights. The distance-to-core term rewards keeping the ball far from the pelvis while otherwise standing still,

$$r_t^{\text{core}} = w_{\text{pos}} \left(1 - e^{-k_{\text{pos}} \|p_t^b - p_t^{\text{pel}}\|} \right) + w_{\text{vel}} e^{-k_{\text{vel}} \|v_{xy,t}^{\text{pel}}\|^2}, \quad (3)$$

where p^b, p^{pel} are the ball and pelvis positions and v_{xy}^{pel} is the horizontal base velocity. The distance term saturates once the ball is far, so the robot settles rather than flees, and the stillness term makes it move only when it must evade. The CBF term is the safety axis we vary in §IV: the no-barrier baseline sets $r^{\text{cbf}} = 0$, and the Link-CBF and Joint-CBF forms of §III-B add increasing barrier structure. The hit termination is active from the first training iteration and carries no explicit penalty: the only cost of being hit is the future reward foregone by an early episode end. Training follows the asymmetric actor-critic paradigm: the value function $V_\phi(s_t)$ observes a privileged state s_t that augments the policy observation o_t of Eq. (1) with the ground-truth ball relative position/velocity, radius, a visibility gate, and full-body link kinematics. These privileged signals support training and barrier computation but are discarded at deployment, so the policy must recover the relevant threat information from o_t alone.

B. Whole-body barriers

Control barrier functions encode a safe set through a function whose nonnegative superlevel set should remain invariant under an admissible controller [10]. PAC-MAN uses barrier information at two levels to study what a policy can internalize from partial perception and what still requires direct online enforcement. The lightest form sets the safety term of Eq. (2) to a per-link clearance reward, $r^{\text{cbf}} = \lambda_{\text{cbf}} r^{\text{clear}}$. We form the clearance for every robot link i ,

$$h_i = \|p^b - p_i\| - (\rho^b + \rho_i), \quad (4)$$

where p^b, p_i are the ball and link positions and ρ^b, ρ_i their effective radii (collision cross-section plus a small margin), and its closing rate $\dot{h}_i$ follows from the relative ball-link velocity. The corresponding whole-body collision-free set is $\mathcal{C} = \bigcap_i \{h_i \geq 0\}$, visualized in Fig. 3, so protecting only the pelvis is insufficient whenever the ball threatens an arm, leg, or torso link. The reward penalizes only violations of the barrier condition $\dot{h}_i + \alpha h_i \geq 0$ at the most-binding link:

$$r^{\text{clear}} = \min_i \text{clip}(\dot{h}_i + \alpha h_i, -c, 0), \quad (5)$$

gated to zero unless a thrown ball is airborne and moving, so the nominal stand is untouched. The class- $\mathcal{K}$ term αh_i ramps the penalty up over the final approach rather than only at contact, and the clip c keeps the term from acting like a hard terminal penalty. This lightweight per-link barrier, which we call Link-CBF, represents threats to limbs and other links that the pelvis-centered r^{core} does not see. Because Link-CBF enters only through the training reward, it guides the policy toward the safe set but does not enforce the barrier

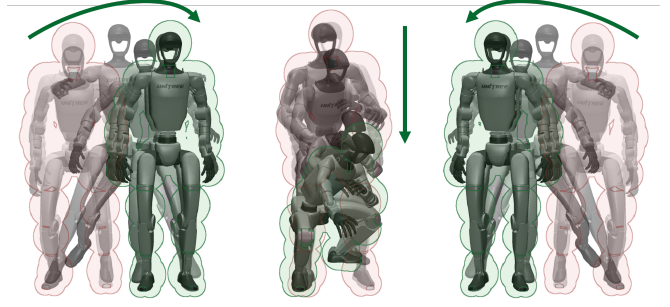


Fig. 3: Whole-body barrier geometry across representative motion-prior clips; the outlines show the union of per-link keep-out shells as the humanoid crouches, leaps, and sidesteps. The reference set contains 1 minute and 43 seconds of human motion retargeted to the Unitree G1 using [35].

condition online during deployment.

The Joint-CBF condition adds a stronger whole-body joint-space CBF module during training. The policy's joint-position target q^{des} (the nominal pose plus a_t) defines an implicit desired joint velocity $v^{\text{des}} = (q^{\text{des}} - q_t)/\Delta t$. The module defends every body point with the squared-distance barrier $h_i = \|p^b - p_i\|^2 - D_i^2$, where $D_i = \rho^b + \rho_i + d_{\text{buf}}$ is a keep-out radius, and at each step selects the most-threatened point $i^* = \arg \min_i h_i$. Enforcing $\dot{h}_{i^*} \geq -\alpha h_{i^*} + \delta$ then yields a single affine constraint on the joint velocity,

$$\eta^\top v \leq b, \quad \eta = 2 J_{i^*}^\top r, \quad b = \alpha h_{i^*} + 2 r^\top \dot{p}^b - \delta, \quad (6)$$

where $r = p^b - p_{i^*}$, J_{i^*} is the point's positional Jacobian, and $\delta = \delta_{\text{max}} [1 - T/T_{\text{alert}}]_+$ is an urgency margin that tightens the constraint as the time to contact T drops below T_{alert} . The barrier-projected velocity is the closed-form projection of v^{des} onto this half-space,

$$v^* = v^{\text{des}} - \frac{[\eta^\top v^{\text{des}} - b]_+}{\|\eta\|^2} \eta, \quad (7)$$

which equals v^{des} whenever the constraint already holds or no threat is sensed. Joint-CBF adds two shaping terms to r^{cbf} : a correction cost $-\lambda_{\text{corr}} \|v^* - v^{\text{des}}\|^2$ that penalizes how hard the filter has to intervene and a buffer cost $-\lambda_{\text{buf}} [d_{\text{safe}} - h_{\text{min}}]_+^2$ on the linear clearance h_{min} of the most-threatened point, which discourages skimming the keep-out boundary. At evaluation, we remove this training-time module, so the policy must execute its own learned avoidance from depth and proprioception alone. This reward-only Joint-CBF condition measures how much of the stronger barrier structure the perception-aware policy can internalize. We also report a privileged ceiling, denoted +filter, that keeps the projection of Eq. (7) active at test time and executes $q_t + v^* \Delta t$ in place of the policy target. The filter reads ground-truth ball states online, so it is not a deployable vision-only configuration; it measures what direct barrier enforcement achieves when accurate threat state is available.

C. Perception

The barriers above are computed from ball and link geometry during training, but the deployed policy must infer the relevant threat geometry from its observations. Perception,

therefore, controls how faithfully the policy can associate a visual threat with the barrier signal used to train it. We isolate this informational effect with three regimes that change the ball representation but leave the underlying control task unchanged:

State oracle. A symmetric actor-critic that observes the ground-truth ball states directly (no camera). This is the safety-information upper bound.

Fixed depth camera. The focusing stage alone: the ball is segmented from the RGB-D stream of a single rigidly mounted head camera (102° horizontal FoV, 57° vertical FoV, tilted 20° up). In simulation, we build this observation from the simulator’s segmentation: pixels on the ball retain their measured depth, every other pixel is set to the far plane, and the masked image is pooled to 16×9 . We stack the masked frame with older snapshots at sparse temporal offsets rather than consecutive frames (specifically the last [0, 3, 8, 18] frames with the camera stream at 50 Hz): they are far enough apart to encode looming and lateral motion while keeping the observation small enough for a feed-forward policy. The camera is rigidly mounted, so a fast ball that leaves its field of view or shrinks below the angular resolution at range is unobservable. During training, we randomize the masked observation with whole-ball dropout, ragged pixel dropout, depth jitter, and edge flicker to model failures of the real perception stack.

Active gimbal (gaze). The second part of human-like perception added on top of the focusing stage: we mount the same camera on a 1-DoF pitch joint ($\pm 30^\circ$, nominal position at 0°) so that it can track the ball through its flight, in analogy to human gaze, which orients toward a threat to keep it foveated. The robot also receives the aim angle as proprioception. We establish the ceiling of this design by driving the joint with an oracle that points the optical axis at the ball; the policy still dodges from depth alone, but the camera is perfectly aimed. This isolates the observability effect from the control problem: the policy receives the same low-resolution depth representation either way, but the ball remains in view for more of its flight.

Toward deployable active perception. The oracle-aimed gimbal requires the ground-truth ball states to aim, so it is a simulation ceiling. Closing the loop does not require the policy to learn the aim: a deployable version would drive the same pitch joint from the segmentation track already computed by the focusing stage, pointing the camera at the tracked ball rather than exposing the aim as a policy action. Realizing this tracker-aimed gimbal on the physical robot is future work, since it requires additional hardware; the system we deploy uses the fixed camera.

D. Motion prior and training

The CBF terms specify the safety structure of the task, but they do not prescribe which dynamically feasible whole-body motion should realize an evasion. We therefore add an adversarial motion prior (AMP) [13] as a motion regularizer. A discriminator D_ψ is trained to distinguish policy

TABLE I: Reward terms of the dodge task with formulations and training weights. $\mathbb{I}[\text{thr}]$ is the throw indicator function.

Term	Expression	Weight
<i>Task, safety, and style</i>		
Distance-to-core r^{core}	$0.9(1 - e^{-0.3\ p^b - p^{\text{pel}}\ }) + 0.1 e^{-\ v_{xy}^{\text{pel}}\ ^2}$	1.0
Link-CBF r^{clear}	$\min_i \text{clip}(\dot{h}_i + \alpha h_i, -c, 0)$ $\alpha=1, c=2$	0.27
Joint-CBF correction	$-\ v^* - v^{\text{des}}\ ^2 / n_j$	0.1
Joint-CBF buffer	$-\mathbb{I}[\text{thr}] [d_{\text{safe}} - h_{\min}]_+^2$ $d_{\text{safe}}=0.1$	1.0
AMP style r^{style}	$\max[0, 1 - \frac{1}{4}(D_\psi - 1)^2]$	0.5
<i>Station hold and posture</i>		
Station hold	$e^{-d_{\text{home}}^2 / 1.5^2}$	0.3
Stillness when safe	$(1 - \mathbb{I}[\text{thr}]) e^{-2\ v_{xy}^{\text{pel}}\ ^2}$	0.5
Action rate when safe	$(1 - \mathbb{I}[\text{thr}]) \ a_t - a_{t-1}\ ^2$	-0.05
Base height	$e^{-(z^{\text{des}} - z^{\text{pel}})^2 / 0.3^2}$	1.0
Angular stability	$e^{-\ \omega_{xy}\ ^2 / \pi^2}$	0.5
<i>Regularizers and terminations</i>		
Fall termination	$\mathbb{I}[\text{fall}]$	-200
Joint acceleration	$\ \ddot{q}\ ^2$	-2.5×10^{-7}
Joint limit violation	$\sum_j [q_j - q_j^{\text{lim}}]_+$	-10
Action rate	$\ a_t - a_{t-1}\ ^2$	-0.01
Foot slip	$\sum_{f \in \text{contact}} \ v_{xy}^f\ ^2$	-0.25
Self-collision	$\text{contacts} > 10 \text{ N}$	-0.1

state transitions from transitions drawn from a reference set $\mathcal{M}$ of human-derived whole-body clips (crouching, leaning, sidestepping, leaping, and standing), and the policy receives the style reward

$$r_t^{\text{style}} = \max\left[0, 1 - \frac{1}{4}(D_\psi(s_t, s_{t+1}) - 1)^2\right], \quad (8)$$

which favors reflex-like crouches, leans, and sidesteps over arbitrary joint excursions. Training uses PPO and the AMP discriminator from `rsl_rl` [36], with 8192 parallel environments for 20k iterations in `mjlab` [37], [38]. The depth feed is capped at 16×9 resolution. The full reward construction is shown in Table I.

IV. RESULTS

We first evaluate the PAC-MAN framework in simulation to quantify the impact of perception and safety structure; then we verify the full stack on the physical robot. Figure 4 shows emergent evasion modes from both.

A. Quantitative evaluation in simulation

In simulation, we compare the onboard fixed-camera policy against a gimbal-mounted camera and a privileged state oracle while varying the safety structure at each perception level, from no barrier to the full joint-space CBF with its runtime filter. All policies are evaluated on the same set of seeded throws and scored by the environment’s true terminations: success requires avoiding contact with every body link and remaining upright, rather than relying on pelvis clearance alone. A static robot survives only 4% of throws,

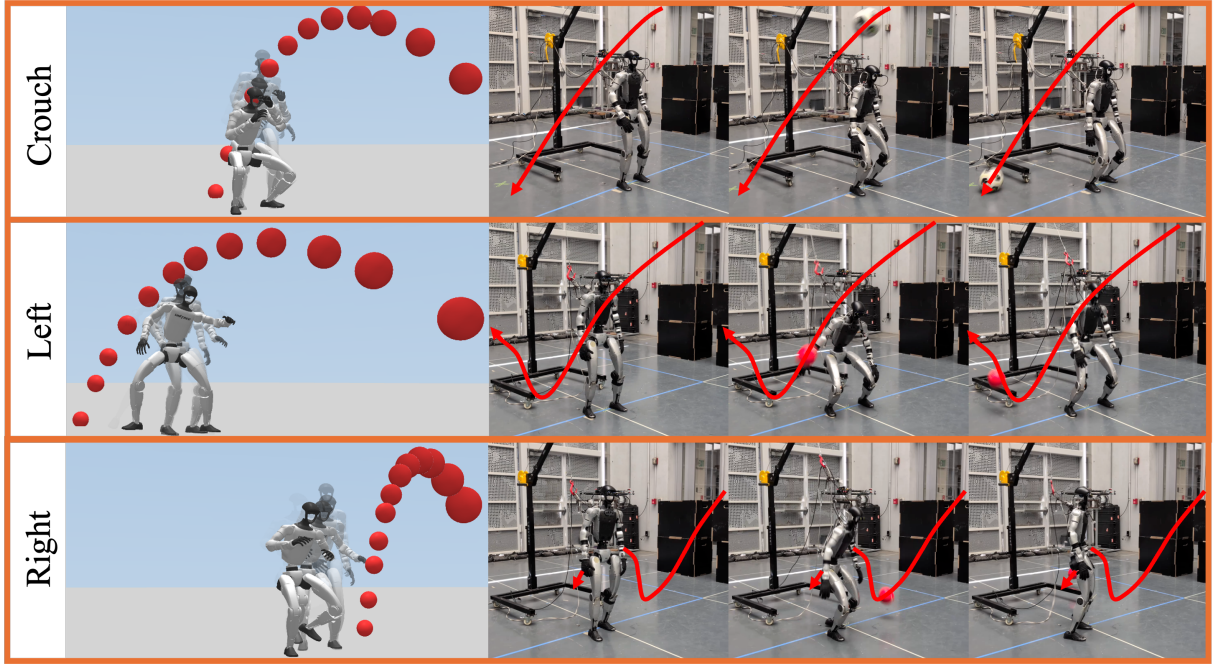


Fig. 4: Emergent whole-body evasion modes in simulation (left) and hardware (right). Crouching, leaning, and sidestepping coordinate the legs, torso, and arms to move threatened links rather than only translating the base. Semantic segmentation allows the same policy to dodge different balls, shown with a soccer ball (top row) and a foam ball (lower two rows).

which sets the on-target floor. We evaluate every policy at a matched 20k-iteration checkpoint. We report two throw regimes. **Reset:** We reset the robot between throws, and the robot dodges immediately. **Deployment:** A locomotion controller returns the robot to the origin between throws before the next dodge. We vary the safety reward along four modes:

- **No barrier** uses the distance-to-core reward r^{core} only ($r^{\text{cbf}} = 0$).
- **Link-CBF** adds the lightweight per-link clearance barrier r^{clear} as reward guidance (§III-B); this is the configuration we deploy on hardware.
- **Joint-CBF** additionally trains with the whole-body joint-space CBF module as reward guidance, evaluated without a runtime filter.
- **+filter** keeps the joint-space CBF as an online filter.¹

Onboard sensing costs only a few points of evasion. Trained with the same Link-CBF reward, the fixed-camera policy (90% reset, 89% deployment) is on par with the state oracle (88% and 93%) despite seeing the ball only as a conservative masked depth (Table II). The oracle’s best configurations (96-99%) sit fewer than ten points above the policy we deploy. The fixed onboard camera alone is therefore adequate for evasion; what perception changes is which safety structure works.

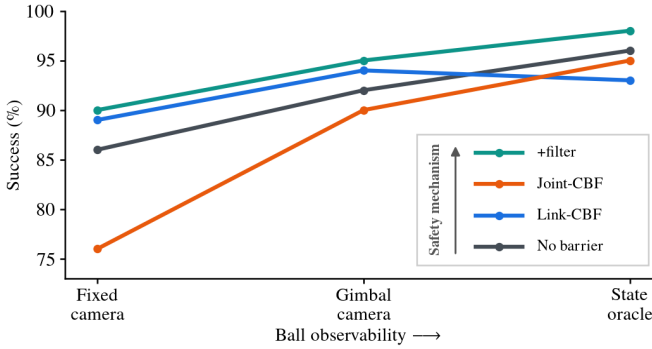
With privileged ball states, the strongest safety configuration achieves the highest success rates. The plain distance-to-core baseline is already strong (96%-98% across regimes),

¹The filter reads ground-truth ball states, so running it on hardware would require a highly accurate online ball state estimator that our vision stack does not provide.

and the full CBF-RL recipe, with the joint-space barrier kept as its runtime filter, outperforms it (98%-99%). The +filter column is the best mode in every row of Table II: given accurate online ball states, runtime whole-body barrier enforcement achieves the highest success rate at each perception level. This improvement requires ball states that our onboard stack currently cannot supply.

Among the modes that run without runtime ball states, the ranking flips. Link-CBF is best on the fixed camera in both regimes (reset 90%, deployment 89%), above the no-barrier baseline (86% in both) and well above the bare Joint-CBF, which drops to 76% in the deployment regime. Link-CBF extends safety guidance from pelvis clearance to every body link; under the any-link contact criterion, it improves fixed-camera success from 86% to 90% in reset and 89% in deployment. It is therefore the reward we deploy on hardware (§IV-B).

We theorize that the Joint-CBF gap comes from perception: the policy is asked to internalize a whole-body barrier that it cannot perceive accurately enough, and it learns worse evasions instead. Two changes repair this, and both supply the ball information that the fixed camera lacks. Keeping the joint-space CBF as an online filter hands exact ball states to the barrier itself and lifts the fixed camera to 90%-97%. Giving the policy a better view works as well: with an oracle-aimed, ball-tracking gimbal, the same Joint-CBF training no longer hurts (deployment 90% versus 76% on the fixed camera) and beats its no-barrier counterpart with the full safety structure (i.e. +filter). This shows that the whole-body barrier succeeds only when the ball is sufficiently observable. Fig. II (top) illustrates this: deployment success of Joint-



Regime	Perception	No barrier	Link-CBF	Joint-CBF	+filter	Fall
Reset	State oracle	98	88	96	99	0
	Fixed camera	86	90	83	97	0
	Gimbal camera	96	95	91	97	0
Deployment	State oracle	96	93	95	98	0
	Fixed camera	86	89	76	90	0
	Gimbal camera	92	94	90	95	0

TABLE II: Whole-body success (no link contact or fall) and fall rates (%) by throw regime, perception, and safety-reward mode (defined in §IV-A) at a matched 20k-iteration checkpoint. Bold marks the deployed fixed-camera Link-CBF cells. Top: deployment success versus ball observability, one line per safety mode; Joint-CBF improves sharply with observability, Link-CBF remains consistent, and +filter requires runtime ball states.

CBF climbs steeply with ball observability, while Link-CBF stays nearly flat. Therefore, the design implication is that the strength of the safety structure must be matched to the information available. A hardware gimbal aimed by the same segmentation-based focusing stage that already feeds the policy is left as future work. We also compare against prior results reported in SMP [15] with an omnidirectional throw distribution (360° , 8-10 m, 12-15 m/s in codebase) with oracle observations. Under this matched setting, PAC-MAN scores 98% reset, in line with their reported $\sim 99\%$.

B. Hardware experiments

We deploy the Link-CBF fixed-camera policy on the Unitree G1 with a head-mounted ZED Mini. On hardware, the perception node replaces the simulator’s segmentation with a learned tracker: an EfficientTAM track, initialized from a clicked foreground point, segments the ball in RGB at ~ 60 Hz, above the 50 Hz control rate, so the masked-depth observation is effectively fresh with each control step. This also gives us the ability to dodge different balls since the tracker can be initialized on any object of interest. An example of dodging a soccer ball instead of a foam ball is shown in the top row of Fig. 4. The tracked object’s ZED depth pixels are retained, and all others are set to far range before pooling to the same 16×9 stacked-depth observation used in training. Several filters keep this feed conservative: low-percentile pooling preserves the ball while rejecting isolated ZED flying pixels; distance-aware mask-size checks reject masks that are too tall to be a dodgeball at their measured range; and

an optional looming gate suppresses tracked clusters whose average depth is not closing quickly enough, preventing a static object or over-sensitive segmentation from triggering dodges. If the track is absent, rejected by these checks, or lost, the policy receives an all-far frame rather than a false incoming ball. Proprioception comes from the robot’s IMU and encoders and uses a walk $\leftrightarrow$ dodge mode switch to return the robot to a fixed location between throws. No ground-truth ball states, velocity estimates, or runtime CBF inputs are available at any point: the policy reacts purely to the conservative masked-depth feed, exactly as in the fixed-camera Link-CBF cells of Table II. We stress-test this stack by throwing balls at the robot from the front. A throw counts as dodged only if the ball contacts no body link. The policy dodges 19 of 20 throws (95%, Table III). Fig. III (top) shows the estimated barrier value h for each throw. The hit is shown at contact, and the remaining throws clear the per-link keep-out shell by 0.05 to 1.7 m. The sim-trained policy therefore transfers to the physical robot from onboard depth alone.

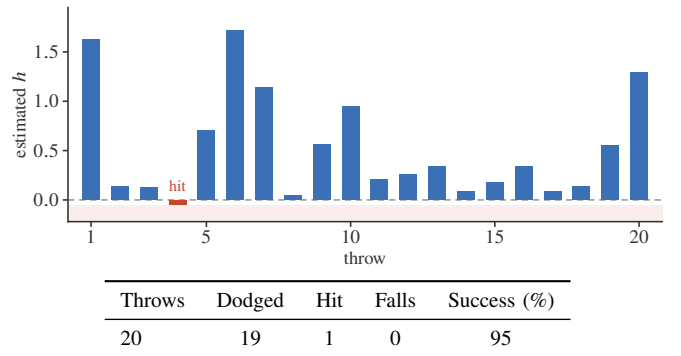


TABLE III: Hardware dodge outcome: 20 balls thrown by hand at the deployed Link-CBF fixed-camera policy on the Unitree G1 (policy input: onboard depth; RGB used only for segmentation), with contact on any body link counted as a hit. Top: estimated barrier value h for each throw; the dashed line is the per-link keep-out shell ($h = 0$), and the hit (throw 4, red) is shown at contact.

V. CONCLUSION AND FUTURE WORK

We presented PAC-MAN, a perception-aware CBF-RL framework for whole-body humanoid dodgeball, and deployed its fixed-camera Link-CBF policy on the Unitree G1 using only onboard depth and proprioception. PAC-MAN couples segmentation-masked depth with control-barrier guidance computed during training, and a human motion prior regularizes the form of the evasive reflexes. Our results show that perception-aware CBF-RL is a safety-information co-design problem: the appropriate barrier structure depends on the threat representation available to the policy or filter. With accurate ball states, the Joint-CBF runtime filter is the strongest configuration, whereas the same joint-space barrier used only as reward guidance degrades the fixed-camera policy. Link-CBF extends the pelvis-centered objective to every body link while tolerating limited onboard

perception, and the deployed policy succeeds on 95% of physical throws without privileged ball states or a runtime filter. The segmentation front end also lets one policy evade different balls. These results distinguish internalizing barrier structure in a learned policy from enforcing a barrier online: internalization is limited by what the policy can infer, while enforcement requires accurate threat state at runtime. Future work will close this perception-to-CBF loop with a physical tracker-aimed gimbal or an accurate online ball-state estimator and extend the task from dodging at a recovered stance to dodging while walking.

REFERENCES

- [1] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu, *et al.*, “Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning,” *IEEE/ASME Transactions on Mechatronics*, vol. 31, no. 2, pp. 2300–2330, 2026.
- [2] Z. Zhuang, S. Yao, and H. Zhao, “Humanoid parkour learning,” in *Proceedings of The 8th Conference on Robot Learning* (P. Agrawal, O. Kroemer, and W. Burgard, eds.), vol. 270 of *Proceedings of Machine Learning Research*, pp. 1975–1991, PMLR, 06–09 Nov 2025.
- [3] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang, “Gmt: General motion tracking for humanoid whole-body control,” *arXiv preprint arXiv:2506.14770*, 2025.
- [4] Z. Luo, Y. Yuan, T. Wang, C. Li, F. Castañeda, S. Chen, Z.-A. Cao, J. Li, D. Minor, Q. Ben, *et al.*, “Sonic: Supersizing motion tracking for natural humanoid whole-body control,” *arXiv preprint arXiv:2511.07820*, 2025.
- [5] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu, “Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion,” *arXiv preprint arXiv:2508.08241*, 2025.
- [6] S. Yin, Y. Ze, H.-X. Yu, C. K. Liu, and J. Wu, “Visualmimic: Visual humanoid loco-manipulation via motion tracking and generation,” *arXiv preprint arXiv:2509.20322*, 2025.
- [7] Y. Hao, R. Yu, S. Luo, G. Zhang, J. Wu, and Q. Zhu, “CReF: Cross-modal and recurrent fusion for depth-conditioned humanoid locomotion,” *arXiv preprint arXiv:2603.29452*, 2026.
- [8] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, “Agile but safe: Learning collision-free high-speed legged locomotion,” 2024.
- [9] C. Kohlbrenner, N. Pudasaini, W. Xie, N. Sivagnanasadan, N. Correll, and A. Roncone, “Egocentric tactile and proximity sensors as observation priors for humanoid collision avoidance,” *arXiv preprint arXiv:2604.25554*, 2026.
- [10] 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*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [11] L. Yang, B. Werner, M. de Sa, and A. D. Ames, “CBF-RL: Safety filtering reinforcement learning in training with control barrier functions,” *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.
- [12] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, “Deepmimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [13] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, “Amp: Adversarial motion priors for stylized physics-based character control,” *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–20, 2021.
- [14] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, “Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters,” *ACM Transactions On Graphics (TOG)*, vol. 41, no. 4, pp. 1–17, 2022.
- [15] Y. Mu, Z. Zhang, Y. Shi, D. Yang, M. Matsumoto, K. Imamura, G. Tevet, C. Guo, M. Taylor, C. Shu, P. Xi, and X. B. Peng, “Smp: Reusable score-matching motion priors for physics-based character control,” *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2026)*, 2026.
- [16] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, “Reinforcement learning for robust parameterized locomotion control of bipedal robots,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2811–2817, IEEE, 2021.
- [17] J. Dao, H. Duan, and A. Fern, “Sim-to-real learning for humanoid box loco-manipulation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16930–16936, IEEE, 2024.
- [18] M. Dai, W. D. Compton, J. Li, L. Yang, and A. D. Ames, “Walk the PLANC: Physics-guided rl for agile humanoid locomotion on constrained footholds,” *arXiv preprint arXiv:2601.06286*, 2026.
- [19] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang, “Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control,” in *Proceedings of Robotics: Science and Systems*, (Los Angeles, CA, USA), June 2025.
- [20] F. Liu, Z. Gu, Y. Cai, Z. Zhou, H. Jung, J. Jang, S. Zhao, S. Ha, Y. Chen, D. Xu, *et al.*, “Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation,” *IEEE Robotics and Automation Letters*, 2025.
- [21] J. P. Araujo, Y. Ze, P. Xu, J. Wu, and C. K. Liu, “Retargeting matters: General motion retargeting for humanoid motion tracking,” *arXiv preprint arXiv:2510.02252*, 2025.
- [22] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferazza, C. K. Liu, R. Duan, and G. Shi, “Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction,” *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.
- [23] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, “Extreme parkour with legged robots,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11443–11450, IEEE, 2024.
- [24] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, “Anymal parkour: Learning agile navigation for quadrupedal robots,” *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.
- [25] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang, “Learning humanoid locomotion with perceptive internal model,” pp. 9997–10003, 2025.
- [26] Z. Wu, X. Huang, L. Yang, Y. Zhang, X. Chen, P. Abbeel, R. Duan, A. Kanazawa, C. Sferazza, G. Shi, *et al.*, “Perceptive humanoid parkour: Chaining dynamic human skills via motion matching,” *arXiv preprint arXiv:2602.15827*, 2026.
- [27] Z. Zhuang, S. Zhu, M. Zhao, and H. Zhao, “Deep whole-body parkour,” *arXiv preprint arXiv:2601.07701*, 2026.
- [28] D. Baek, S.-H. Kim, and S. Ha, “BAT: Balancing agility and stability via online policy switching for long-horizon whole-body humanoid control,” *arXiv preprint arXiv:2604.01064*, 2026.
- [29] P. Li, H. Li, M. Fan, F. Xu, S. Liao, Y. Ma, Z. Zeng, Z. Wang, Y. Jin, Y. Cao, *et al.*, “TAGA: Terrain-aware active gaze learning for generalizable agile humanoid locomotion,” *arXiv preprint arXiv:2606.05880*, 2026.
- [30] Z. Su, B. Zhang, N. Rahmanian, Y. Gao, Q. Liao, C. Regan, K. Sreenath, and S. S. Sastry, “Hitter: A humanoid table tennis robot via hierarchical planning and learning,” *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.
- [31] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, “Capture point: A step toward humanoid push recovery,” in *2006 6th IEEE-RAS international conference on humanoid robots*, pp. 200–207, IEEE, 2006.
- [32] B. J. Stephens and C. G. Atkeson, “Dynamic balance force control for compliant humanoid robots,” in *2010 IEEE/RSJ international conference on intelligent robots and systems*, pp. 1248–1255, IEEE, 2010.
- [33] J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, “A collision-free mpc for whole-body dynamic locomotion and manipulation,” in *2022 international conference on robotics and automation (ICRA)*, pp. 4686–4693, IEEE, 2022.
- [34] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, *et al.*, “Curobo: Parallelized collision-free robot motion generation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8112–8119, IEEE, 2023.
- [35] Bones Studio, “BONES-SEED: Skeletal everyday embodiment dataset,” <https://bones.studio/datasets/seed>, 2026.
- [36] C. Schwärke, M. Mittal, N. Rudin, D. Hoeller, and M. Hutter, “Rsl-rl: A learning library for robotics research,” *arXiv preprint arXiv:2509.10771*, 2025.
- [37] K. Zakka, Q. Liao, B. Yi, L. L. Lay, K. Sreenath, and P. Abbeel, “mjlab: A lightweight framework for gpu-accelerated robot learning,” *arXiv preprint arXiv:2601.22074*, 2026.
- [38] ccrpRepo, “AMP-mjlab: G1 AMP motion control on mjlab + rsl.rl,” https://github.com/ccrpRepo/AMP_mjlab, 2025.