

Beyond Dodge Rate: Testing Humanoid Dodgeball Under Camera Failures

Kingston Kuan Brandon Ong
Hebbian Robotics

<https://hebbian-robotics.github.io/beyond-dodge-rate/>

Abstract—A humanoid robot can dodge a ball and still leave its playing area. We study this gap under depth-camera failures, measuring contact, balance, and area compliance throughout each four-second trial. Two saved G1 policies, π_1 and π_2 , use the same training method and recorded settings but respond differently to stale and blank images across 192 paired test seeds per policy. For π_1 , all 122 area exits under stale images occur after the throw resolves. In a separate fixed paired recovery test with 64 seeds, restoring fresh images at two seconds increases success from 50.0% to 95.3%, while dodge rate remains 100%. Maximum excursion also decreases without an area cutoff. The same intervention does not improve overall success for π_2 . Shared ball trajectories preserve the π_1 recovery benefit. These results support evaluating motion after an encounter and testing camera handling on the policy in use. They concern two fixed models in simulation and do not establish a universal camera fallback or hardware safety.

I. INTRODUCTION

Dodge rate measures whether a robot avoids an incoming ball. It does not show whether the robot stays upright and inside its playing area afterward. A camera failure can therefore leave dodge rate nearly unchanged while causing failure after the encounter. We measure all three outcomes and compare camera inputs within two fixed policies trained with the same method and recorded settings.

Our central test restores fresh images after the simulator resolves the throw. Matched trials share the same physical throw and camera history up to restoration, so the comparison measures a change in subsequent motion. A control gives both policies identical recorded throws; distance analyses check dependence on the area requirement. Building on humanoid dodgeball [13, 6] and camera-fault evaluation [12], this study tests post-encounter area compliance and visual recovery without a new controller or training objective.

II. METHODS

A. Policies and simulator

We use Russi’s G1 Depth Dodge code [9], based on ViBe/ORCS [6], with LeRobot G1 deployment assets [5]. The controller combines frozen SONIC [7] and Theia [11] bases with depth adaptation trained by PPO [10] through RSL-RL in mjlab/MuJoCo Warp. π_1 is the released iteration-4000 model [1]; π_2 was trained separately by our team with the same method, architecture, and objective. Its saved configuration matches the released reference: seed 42, 4,000 updates, 512 environments, and about 49.15 million transitions, distributed across eight GPUs in our run. The weights

differ. Initialization, random-number sequences, and resume histories are incomplete, so these models do not constitute a controlled comparison of training methods or repeated training runs.

Evaluation fixes the weights and native gains, torque limits, depth noise, active-ball collisions, and fall checks. MuJoCo 3.3.7 and ONNX Runtime 1.23.2 run on CPU, with 50 Hz control and 25 Hz images. We correct an inactive-ball bug by disabling collisions while the ball is parked below the floor and restoring them at launch. Twelve exact replays found no inactive-ball robot contacts or camera visibility. All reported trials use this correction.

B. Task, inputs, and matching

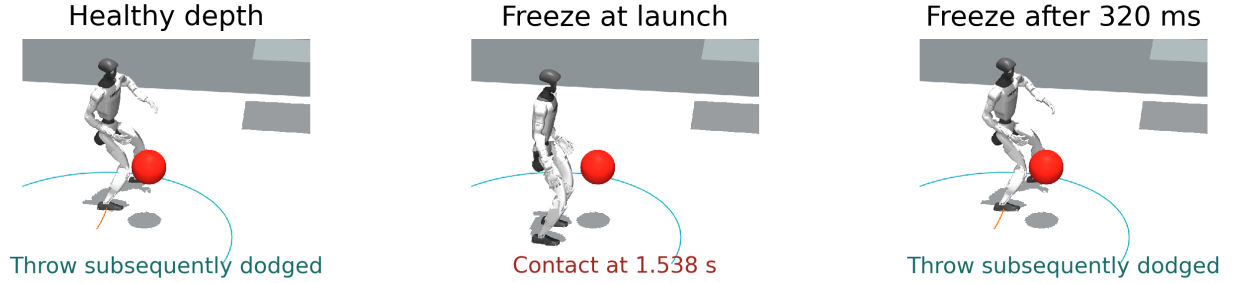
Each trial lasts four seconds unless the robot falls. One ball launches at one second. The native generator chooses among six body targets, adds jitter and $\pm 30^\circ$ direction variation, and uses a nominal flight time of 0.55–0.62 s. Healthy depth supplies normal images; stale depth repeats the last image before the interruption; blank depth supplies zeros. Policy history is never reset. Fault onset is known; there is no learned failure detector.

Within each policy and seed, we require identical prelaunch robot positions and velocities and matching launch time, target, ball position, velocity, and flight time. We also verify camera schedules; missing or mismatched trials fail analysis. The policies can move differently before launch and change the simulator’s aim, so equal seed numbers do not ensure equal throws across policies. Comparisons therefore concern camera inputs within each policy, not a policy ranking. The input-handling test fixes 192 seeds (7300–7491) per policy before observing outcomes. Interruption starts 160 ms after launch and lasts to the endpoint.

C. Outcomes and uncertainty

Success requires dodging, staying upright, and keeping the pelvis within 1 m of its starting position throughout the trial. We report each outcome separately. The radius is an illustrative task requirement, not a court rule or injury threshold. A fall uses the native checks: pelvis height below 0.45 m or upright-axis cosine below 0.4 (about 66.4° tilt). Aborted throws remain failures in the total trial count. Paired 95% intervals use 10,000 bootstrap resamples of whole evaluation seeds, retaining their conditions together, with seed 20261014. Intervals are per comparison, without multiplicity adjustment; per-condition

A Just before the first hit ($t = 1.520$ s)



B End of the four-second trial: dodging can hide leaving the area

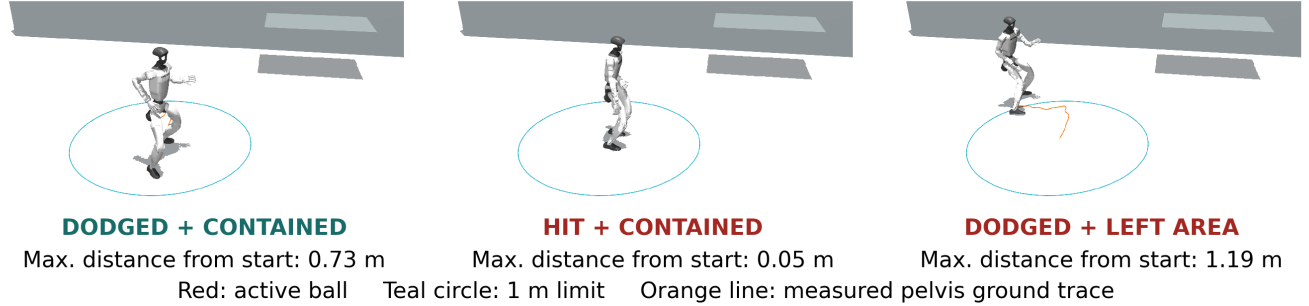


Fig. 1. What dodge rate misses. Exact replays of the first test seed (3000) for π_1 , with matched views and colors changed for visibility. Top: just before the middle robot is hit. Bottom: the four-second endpoint. The late camera freeze allows a dodge but the robot leaves the 1 m area. The circle marks the boundary; orange shows the pelvis path. This example illustrates failure timing, not average performance.

intervals are Wilson intervals. Evaluation seeds are not training replications.

D. Post-encounter motion and restoration

For all 1,152 input-handling trials, a post-hoc reanalysis locates the first sampled area exit relative to native throw resolution: contact for hits and native termination for misses. This is an accounting event, not closest approach. Exits strictly afterward are late; ties count as by-resolution exits. Samples are 20 ms apart, without interpolation; seeds and outcomes remain unchanged.

The recovery test uses five conditions: healthy, continuous stale, stale with restoration, continuous blank, and blank with restoration. Faults start at 1.16 s; fresh images resume at 2 s without a history reset. After a separate exploratory 16-seed pilot per policy (8000–8015), we fixed 64 new seeds (9000–9063) and the primary contrast: restored versus continuous stale input for π_1 . The other policy and blank-input contrasts are secondary. Pilot outcomes are excluded. All conditions retain the four-second horizon; there is no outcome-dependent stopping. Matching checks verify identical physical traces before 2 s in each restoration pair. All stale-input throws resolve by restoration within numerical tolerance, so the intervention tests post-resolution motion at a fixed time.

E. Shared-throw control

An exploratory control replays 64 healthy-condition ball trajectories, taking 32 donors from each policy without outcome filtering. Both policies face the same initial ball positions

and velocities in all five camera conditions (seeds 10000–10063; 640 trials). Native collision settings and accounting remain unchanged; robot states can still differ. Twenty separate preflight trials and two exact healthy-condition replays check implementation. We resample common-throw seeds with both policies and all conditions together. This control is not pooled into the fixed recovery estimate; it reuses recorded trajectories and does not add independent training runs. The supplement gives donor selection and all rows.

III. RESULTS

A. Dodges can hide late area exits

Under stale input, all 122 π_1 area exits occur after throw resolution; 121 follow a dodge. All 11 exits under blank input are also late. For π_2 , stale input gives seven exits by resolution and three afterward; both blank-input exits are late. These are descriptive timings, not explanations of the weights.

B. Fresh images change post-encounter motion

For π_1 , restoring fresh images changes overall success from 32/64 to 61/64, a difference of +45.3 percentage points (95% paired interval +32.8 to +57.8). There are 29 improved seeds and 0 worsened seeds.

For π_2 , restoring fresh images changes overall success from 53/64 to 51/64, a difference of -3.1 percentage points (95% paired interval -7.8 to +0.0). There are 0 improved seeds and 2 worsened seeds. Both π_1 stale conditions dodge 64/64 throws; both π_2 stale conditions dodge 54/64. No recovery condition has falls or aborted throws. The intervention therefore changes

TABLE I
FIXED RECOVERY TEST: 64 SEEDS PER POLICY AND CONDITION. FRESH IMAGES RESUME AT 2 S. FALLS AND ABORTED THROWS ARE ZERO. COMPARE CONDITIONS WITHIN EACH POLICY.

Policy	Camera input	Dodge	In area	Success
π_1	Healthy	64/64	64/64	64/64
	Stale	64/64	32/64	32/64
	Stale + recovery	64/64	61/64	61/64
	Blank	60/64	57/64	54/64
	Blank + recovery	60/64	60/64	56/64
π_2	Healthy	57/64	62/64	57/64
	Stale	54/64	61/64	53/64
	Stale + recovery	54/64	59/64	51/64
	Blank	36/64	63/64	36/64
	Blank + recovery	36/64	63/64	36/64

area compliance without rescuing the initial dodge. Table I includes all five conditions, including the smaller or absent effects under blank input. Figure 2 shows when exits occur. For this π_1 and throw distribution, restored visual input reduces later area failures; it does not establish a neural mechanism or a universal fallback.

C. Scoring choices and continuous distance

A post-hoc check uses six radii (0.5–2 m) and four windows (2–4 s). Recovery gains accumulate late and shrink at wider boundaries: widening from 1 to 1.5 m at four seconds reduces the π_1 gain from 29/64 to 2/64 in the fixed cohort and from 37/64 to 1/64 in the control. Without an area cutoff, restoring vision reduces π_1 mean maximum pelvis radius from 1.043 to 0.860 m (paired mean difference -0.184 m; nominal 95% interval -0.226 to -0.146). The shared-trajectory difference is -0.179 m. π_2 ends closer to the station but shows little improvement in maximum excursion. Both distance summaries include all 64 paired seeds, including hits. These post-hoc reanalyses are not independent confirmations.

D. Stale versus blank input

For π_1 , stale images gave 70/192 successful trials, compared with 171/192 for blank images. The blank-minus-stale difference was +52.6 percentage points (95% paired interval +43.8 to +60.9). The robot dodged 191/192 balls with stale images and 182/192 with blank images. Falls were 0/192 and 0/192, respectively. For π_2 , stale images gave 158/192 successful trials, compared with 100/192 for blank images. The blank-minus-stale difference was -30.2 percentage points (95% paired interval -37.0 to -23.4). The robot dodged 165/192 balls with stale images and 101/192 with blank images. Falls were 0/192 and 0/192, respectively. Table II also shows healthy-camera performance and area counts. The same input choice therefore has different effects on these two policies. These are within-policy comparisons; they do not establish which policy is better overall. For π_1 , blank input improves area compliance from 70/192 to 181/192 but reduces dodges from 191/192 to 182/192. For π_2 , area compliance improves from 182/192 to 190/192 while dodges fall from 165/192 to 101/192. With no falls, these outcomes explain the opposite

TABLE II
INPUT-HANDLING TEST: 192 SEEDS PER POLICY AND CONDITION. SUCCESS REQUIRES DODGING, STAYING UPRIGHT, AND STAYING INSIDE THE 1 M AREA. ALL TRIALS COUNT.

Policy	Depth	Dodged	Fell	In area	Success
π_1	Healthy	192	0	192	192
π_1	Stale	191	0	70	70
π_1	Blank	182	0	181	171
π_2	Healthy	162	0	189	161
π_2	Stale	165	0	182	158
π_2	Blank	101	0	190	100

success effects (Table II), but not their cause inside the learned policies.

E. Effects under shared throws

For π_1 , blank rather than stale images changes success from 23/64 to 53/64 (+46.9 percentage points; nominal 95% paired interval +31.2 to +62.5).

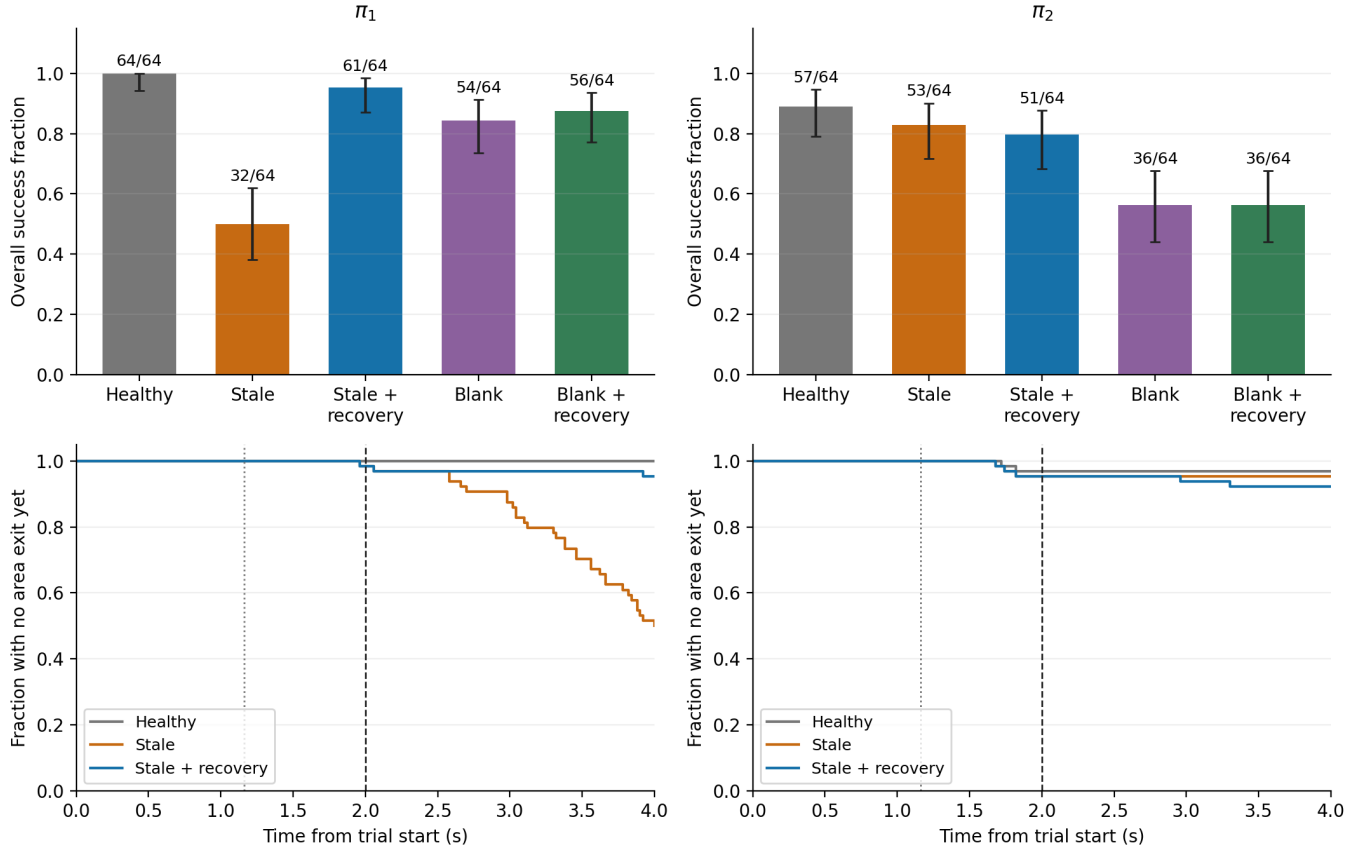
For π_2 , blank rather than stale images changes success from 46/64 to 39/64 (-10.9 percentage points; nominal 95% paired interval -21.9 to +0.0).

The π_1 input effect minus the π_2 input effect is +57.8 percentage points (nominal 95% paired interval +40.6 to +75.0). This comparison resamples each common-throw seed with both policies and all conditions together. The π_2 interval reaches zero, so this control alone does not establish a harmful blank-input effect for that policy. Restoring fresh images also improves π_1 stale-input success from 23/64 to 60/64, with 60/64 dodges in both conditions. π_2 changes from 46/64 to 45/64, with 52/64 dodges in both. No falls or aborted throws occur. The response difference therefore persists under identical ball trajectories, although robot states, weights, and incomplete training histories still differ.

IV. RELATED WORK AND LIMITATIONS

ViBe adapts motion trackers with visual feedback [6]; our implementation’s released training already includes image corruption and repeated frames [1]. PAC-MAN evaluates contact and falls with control-barrier guidance and returns the robot to its station between throws [13]. Our area constraint applies throughout a single throw. Blackout-versus-freeze work evaluates physical camera-fault consequences in manipulation [12]. We extend this evaluation question to humanoid evasion and post-encounter area compliance. VB-Com switches between vision and blind policies [8]; CAP trains robustness to degraded depth [3]. Our tests keep each model fixed; blank images do not make a visual model a separately trained blind policy.

OopsieVerse separates completion from damage-free completion [2]; our area requirement is a task constraint, not a damage measure. LIBERO-MAX uses matched rollouts with execution-time changes [14]; we match camera conditions and verify the physical throw. Reinforcement-learning reproducibility work documents training variability [4], which our two models do not isolate.



π_1 : released checkpoint; π_2 : separately trained with the same method and recorded settings

Fig. 2. All 640 fixed recovery trials. Top: successes out of 64 with Wilson 95% intervals; paired differences are reported in the text. Bottom: fraction of seeds that have not yet left the area, including all trials. Curves drop at first exit even if the robot returns. Dotted: interruption at 1.16 s. Dashed: restoration at 2 s. Bottom panels show healthy and stale conditions; the top panels and Table I include all five. Throws can differ across policies.

The evidence covers two fixed policies, one simulated robot and starting posture, the native throw generator, and one recorded trajectory pool. It does not establish hardware safety, human/team interaction, a universal fallback, or training-seed reliability. Pelvis containment does not ensure whole-body containment. Trial archives and separately trained weights remain unreleased, limiting replication; code, configurations, and provenance are provided through the project website.

V. CONCLUSION

A successful dodge can hide a later failure to stay inside the playing area. For π_1 , restoring fresh images after throw resolution raises success from 32/64 to 61/64 without changing the dodge count and reduces maximum excursion without an area cutoff. π_2 responds differently. These simulation findings support measuring post-encounter motion alongside contact and falls. Future work should repeat the tests across independent training runs and varied throws, then assess transfer to physical robots.

REFERENCES

- [1] G1 Depth Dodge: released ONNX as-sets, 2026. Hugging Face revision 10e207cb734a6c3edf792d4995ca59ae4641bf29; accessed October 14, 2026.
- [2] A. Balaji, A. Bahety, S. Ambatipudi, D. Lam, J. Xu, and R. Martín-Martín. OopsieVerse: A safety benchmark with damage-aware simulation for robot manipulation, 2026. arXiv:2606.31993.
- [3] H. Chen et al. CAP: Continuously adaptive perception-blind humanoid locomotion via learned denoising, 2026. arXiv:2609.11553.
- [4] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters, 2017. arXiv:1709.06560.
- [5] Hugging Face. LeRobot: G1 robot assets and deployment integration, 2026.
- [6] L. Krishna, S. Venkatesan, A. Zhang, and Q. Nguyen. ViBe: Visual behavior adaptation for perceptive humanoid whole-body control, 2026. arXiv:2609.09918.
- [7] Z. Luo et al. SONIC: Supersizing motion tracking for natural humanoid whole-body control, 2025. arXiv:2511.07820.
- [8] J. Ren et al. VB-Com: Learning vision-blind composite humanoid locomotion against deficient perception, 2025.

arXiv:2502.14814.

- [9] M. Russi. G1 Depth Dodge: training and deployment code, 2026. Git revision 0df547b86578dcc8633f490784a14033f0189690.
- [10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. arXiv:1707.06347.
- [11] J. Shang et al. Theia: Distilling diverse vision foundation models for robot learning, 2024. arXiv:2407.20179.
- [12] H. Suh, J. Han, Z. Gholami, and Y. Jo. Blackout vs. freeze: Analyzing physical failure modes of VLAs under camera faults, 2026. arXiv:2609.39145.
- [13] L. Yang, J. Li, and A. D. Ames. PAC-MAN: Perception-aware CBF-RL for whole-body safety in humanoid dodgeball, 2026. arXiv:2607.28623, version 2.
- [14] Y. Zhang et al. LIBERO-MAX: Do robot policies adapt when the world changes?, 2026. arXiv:2609.36518.