Learning Structured Locomotion References for Optimization-Based Whole-Body Control

Anonymous Authors

***Abstract*— Quadruped locomotion requires coordinating body motion, contact forces, foothold placement, and gait timing. We present a hierarchical controller in which reinforcement learning adapts these structured locomotion references while a gait scheduler, reactive footstep planner, and whole-body impulse controller retain responsibility for high-frequency execution. The model-based backbone can sustain a nominal trot independently, allowing the policy to focus on when and how to modify an already functional locomotion controller. We train the complete hierarchy in a vectorized controller-in-the-loop environment using a six-direction disturbance curriculum. Under a common simulator and evaluation protocol, the proposed controller achieves 77.29% mean survival across the tested disturbance grid, compared with 50.35% for the model-based reference controller. Training-time channel ablations identify adaptive foothold placement as the largest marginal contributor to grid survival and reveal direction-dependent effects of gait-rate adaptation. Together, these results demonstrate the viability of the proposed hierarchical controller architecture for integrating learned adaptation with optimization-based whole-body control through structured locomotion references.**

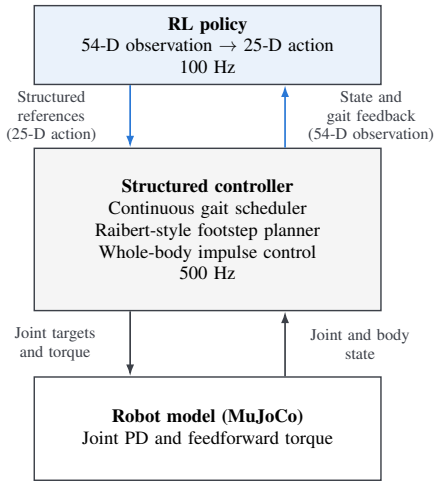

Fig. 1. Proposed controller. A 100 Hz policy supplies structured locomotion commands to a gait scheduler, footstep planner, and WBIC running at 500 Hz. Feedback provides the 54-dimensional observation.

I. Introduction

Quadruped locomotion is a hybrid control problem that couples continuous floating-base dynamics with discrete changes in unilateral contact [1]. The controller must coordinate body motion, ground-reaction forces, foothold placement, and contact timing while respecting friction and actuator limits. During a disturbance, several of these quantities may need to change simultaneously and within a short time.

Model-based locomotion controllers address this complexity by separating gait scheduling, footstep planning, force generation, whole-body control, and joint-level feedback [2], [3]. Template-based controllers likewise organize locomotion through low-dimensional gait variables [4]. Convex model predictive control (MPC), for example, plans ground-reaction forces (GRFs) using a reduced-order model [5]. Kim *et al.*'s whole-body impulse control (WBIC) is a specific optimization-based whole-body control (WBC) formulation that converts force and motion references into dynamically consistent joint commands at high frequency [6]. Related whole-body optimization methods reason more directly about contact and full-body dynamics [7]. These intermediate references provide a useful abstraction boundary: higher layers select behaviorally meaningful quantities, while lower layers resolve full-body dynamics and joint commands. This structure provides reliable and physically interpretable execution, but its adaptability depends on how the references are generated. Contact sequences and gait cadence are often prescribed, while footholds are commonly selected using engineered rules such as Raibert-style heuristics [8]. Extending the resulting behavior can therefore require additional modeling, planning, and manual tuning. More general motion-optimization methods expose footholds and contact timing as decision variables, but still require an explicit model and objective [9], [10].

Reinforcement learning (RL) offers a complementary way to adapt these decisions through interaction. Learned policies have demonstrated agile locomotion and robust terrain traversal in simulation and on hardware [11], [12], [13]. Here, we use *joint-space RL* to denote policies that output joint-position targets or residuals tracked by joint-level feedback, as in [14]. Such a policy must implicitly learn how joint commands produce coordinated body and foot motion. Training requires substantial experience, and the policy outputs do not by themselves enforce dynamic or contact feasibility. Task-space and hybrid approaches instead use learning to command physically meaningful variables while retaining a model-based execution layer [15], [16], [17], [18]. Related approaches augment MPC or WBC using learned corrections, optimal-control guidance, model-error compensation, or constrained execution layers [19], [20], [21], [22], [23].

We follow this principle with a quadruped locomotion stack comprising a continuous gait scheduler, Raibert-style footstep planning, WBIC, and joint-level feedback, as shown in Fig. 1. The model-based backbone can sustain a nominal trot from fixed motion and force references without learned modifications. RL replaces MPC as the adaptive reference

generator rather than replacing the low-level controller. Through a 25-dimensional structured action, the policy modifies commanded body motion, desired stance forces, foothold placement, body height, and gait progression. The policy can therefore coordinate balance, stepping, and cadence, while the model-based layers remain responsible for high-frequency execution.

The contribution is not the integration of RL with whole-body control alone. Instead, we give one policy broad supervisory authority over several coupled levels of an already functional locomotion controller. This allows learning to focus on adapting nominal locomotion rather than rediscovering joint coordination from scratch.

Our contributions are a hierarchical controller architecture that exposes body motion, contact forces, foothold placement, body height, and continuous gait timing to one learned supervisor; a vectorized MuJoCo–EnvPool environment for training the complete controller–simulator loop; and an empirical characterization through disturbance-survival evaluation and training-time action-channel ablations. We evaluate the resulting configured system under the same simulator and protocol as MPC+WBIC [6] and joint-space RL [14]. The disturbance experiments test the resulting architecture under large perturbations; they are not intended to establish the structured interface as a universally optimal action representation.

II. Proposed Hierarchical Control Architecture

A. Architecture Overview

The controller consists of a learned policy, a gait scheduler, a reactive footstep planner, and a whole-body impulse controller (WBIC), as shown in Fig. 1. The policy sets body-motion references, desired stance forces, foothold offsets, body height, and gait cadence. The scheduler determines the contact sequence, the planner generates swing trajectories, and WBIC computes joint commands from the motion and force references.

The policy runs at 100 Hz. Each action is held for five 2 ms updates of the scheduler, planner, and WBIC, which operate at 500 Hz with state feedback. Desired stance forces are supplied directly by the policy.

B. Policy Formulation

The policy maps an observation $\mathbf{o}_t \in \mathbb{R}^{54}$ to an action $\mathbf{a}_t \in [-1,1]^{25}$, partitioned as

$$\mathbf{a}_t = [\mathbf{a}_v^\top \ \mathbf{a}_f^\top \ \mathbf{a}_p^\top \ a_h \ a_\theta]^\top, \tag{1}$$

where the components respectively command body-motion residuals, stance-force targets, planar foothold residuals, body height, and gait progression. Table I gives their physical ranges.

Table II lists the observation channels. Actor and critic use the same current observation, without a history buffer or privileged disturbance information. The contact indicators describe the scheduled gait, not measured contact.

The force channels are absolute targets; the velocity, foothold, and height channels are residuals. The first three actions modify the external velocity command. The resulting horizontal velocity is integrated in world coordinates to obtain the desired body position, with position error limited to 0.1 m. The height action offsets a nominal stance height of 0.32 m. Desired reaction forces are applied to stance legs, while swing legs track the planner's Cartesian trajectories.

TABLE I

Action components; leg order is front-right (FR), front-left (FL), hind-right (HR), and hind-left (HL).

Index	Command	Decoded range
0–1	body v_x, v_y residual	± 0.5 m/s
2	yaw-rate residual	± 0.3 rad/s
3–14	four forces (f_x, f_y, f_z)	$\pm 50, \pm 50, [0, 250]$ N
15–22	four foothold offsets (x, y)	± 0.5 m
23	body-height residual	± 0.05 m
24	phase increment	$[0, 0.01]$ per tick

TABLE II

Observation components.

Signal	Dimension
Commanded forward velocity v_x^{cmd}	1
Body linear velocity $\mathbf{v}_b^b$	3
Body angular velocity $\boldsymbol{\omega}_b^b$	3
Body orientation $\boldsymbol{\phi}_b$ (roll, pitch, yaw)	3
Joint positions $\mathbf{q}_j$	12
Joint velocities $\dot{\mathbf{q}}_j$	12
Scheduled contacts $\mathbf{c}$	4
Body-frame foot positions $\{\mathbf{p}_{f,\ell}^b\}_{\ell=1}^4$	12
Mean swing-phase encoding $(\sin 2\pi\bar{s}, \cos 2\pi\bar{s})$	2
Measured and nominal body height (z_b, z^{ref})	2
Total	54

The training reward uses bounded task-space terms [15],

$$R_t = \sum_i w_i \exp(-p_i), \tag{2}$$

with the penalties and weights in Table III. The set $\mathcal{A} = \{0, \ldots, 23\} \setminus \{5, 8, 11, 14\}$ excludes the four vertical force channels and the phase action from the magnitude penalty. A fall terminates training with reward -10.

C. Continuous Gait Scheduling

A normalized phase $\theta \in [0, 1)$ and fixed per-leg offsets o_ℓ define a diagonal trot. At each 500 Hz update,

$$\begin{aligned} \theta_{k+1} &= \mathrm{mod}(\theta_k + \delta\theta_k, 1), \qquad \delta\theta_k = 0.005(a_{\theta,k} + 1), \\ \theta_{\ell,k} &= \mathrm{mod}(\theta_k - o_\ell, 1), \qquad c_{\ell,k} = \mathbb{I}[\theta_{\ell,k} < d_\ell], \end{aligned} \tag{3}$$

where $c_{\ell,k}$ denotes scheduled contact and $d_\ell = 0.5$ is the duty fraction. The offsets are 0 for FR and HL and 0.5 for FL and HR. A neutral action gives a 0.4 s cycle; the policy can accelerate or pause progression without changing the diagonal contact order. The scheduler also estimates stance and swing durations for the footstep planner.

D. Footstep Planning

Foot placement follows the Raibert-style rule used by Kim *et al.* [8], [6]. For swing progress s_ℓ, the remaining swing

TABLE III

REWARD PENALTIES AND WEIGHTS. EACH HEALTHY-STATE COMPONENT IS $w_i \exp(-p_i)$. SUPERSCRIPT w DENOTES THE WORLD FRAME, AND $\mathbf{q}_0$ IS THE NEUTRAL-ORIENTATION QUATERNION.

Term	Penalty p_i	w_i	Term	Penalty p_i	w_i
v_x	$3\lvert v_x^w - v_x^{\mathrm{ref}}\rvert$	0.25	Straightness	$5\lvert y_b^w\rvert + 3\lvert v_y^w\rvert$	0.15
v_z	$3\lvert v_z^w\rvert$	0.10	Linear acceleration	$\lVert\mathbf{a}_b^w\rVert_2^2$	0.025
Height	$20\lvert z_b - z^{\mathrm{ref}}\rvert$	0.35	Angular velocity	$\lVert\boldsymbol{\omega}_b\rVert_2^2$	0.025
Orientation	$50(1 - \langle\mathbf{q}_b, \mathbf{q}_0\rangle)$	0.15	Action variation	$3\lVert\mathbf{a}_t - \mathbf{a}_{t-1}\rVert_2^2$	0.10
Phase progression	$\max(0, a_\theta + 1)$	0.05	Action magnitude	$\sum_{j\in\mathcal{A}} a_{t,j}^2$	0.05

time is $t_\ell^{\mathrm{rem}} = (1 - s_\ell)T_{\mathrm{sw}}$. The predicted foothold before feedback correction is

$$\bar{\mathbf{p}}_\ell^w = \mathbf{p}_b^w + \mathbf{R}_{wb}\left[\mathbf{R}_z\left(-\tfrac{1}{2}\dot{\psi}^{\mathrm{des}}T_{\mathrm{st}}\right)\mathbf{p}_{\mathrm{hip},\ell}^b + \mathbf{v}_{\mathrm{cmd}}^b t_\ell^{\mathrm{rem}}\right]. \tag{4}$$

The planar Raibert correction is

$$\mathbf{r}_{\mathrm{R}} = \begin{bmatrix} \frac{1}{2}v_x^w T_{\mathrm{st}} + k_v(v_x^w - v_{x,\mathrm{cmd}}^w) + \frac{z_b}{2g}v_y^w\dot{\psi}^{\mathrm{des}} \\ \frac{1}{2}v_y^w T_{\mathrm{st}} + k_v(v_y^w - v_{y,\mathrm{cmd}}^w) - \frac{z_b}{2g}v_x^w\dot{\psi}^{\mathrm{des}} \end{bmatrix}, \tag{5}$$

and the policy modifies the touchdown location according to

$$\mathbf{p}_{\ell,xy}^{w,*} = \bar{\mathbf{p}}_{\ell,xy}^w + \mathrm{clip}(\mathbf{r}_{\mathrm{R}}, -p_{\max}, p_{\max}) + \Delta\mathbf{p}_\ell^{\mathrm{pol}}. \tag{6}$$

Here $k_v = 0.03$, $p_{\max} = 0.30$ m, and clipping is componentwise. The learned residual is added after clipping, so $p_{\max}$ bounds the heuristic rather than the final target. The MPC+WBIC baseline uses the same heuristic bound in its separate planner. A Bézier trajectory with 0.10 m clearance connects lift-off to the target, whose world-frame height is -0.003 m.

E. Whole-Body Control

WBIC follows the formulation of Kim *et al.* [6]. For generalized coordinates $\mathbf{q}$, contact forces $\mathbf{f}_c$, and joint torques $\boldsymbol{\tau}$, the dynamics and stance constraints are

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top\boldsymbol{\tau} + \mathbf{J}_c^\top\mathbf{f}_c, \tag{7}$$

$$\mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = \mathbf{0}, \tag{8}$$

where $\mathbf{S}$ selects actuated joints and $\mathbf{J}_c$ is the contact Jacobian. Body and swing-foot objectives define the motion tasks. WBIC adjusts the nominal acceleration and desired reaction forces subject to floating-base dynamics, unilateral contact, and a friction pyramid with coefficient $\mu = 0.45$. The optimized forces can therefore differ from the policy's desired forces.

Inverse-dynamics feedforward and joint feedback give the torque command,

$$\boldsymbol{\tau}_{\mathrm{cmd}} = \boldsymbol{\tau}_{\mathrm{ff}} + \mathbf{K}_p(\mathbf{q}_j^{\mathrm{des}} - \mathbf{q}_j) + \mathbf{K}_d(\dot{\mathbf{q}}_j^{\mathrm{des}} - \dot{\mathbf{q}}_j). \tag{9}$$

Actuator limits are enforced during execution. The optimization imposes dynamic and contact constraints on each update; it does not provide a closed-loop stability guarantee.

III. TRAINING AND VALIDATION

A. Vectorized Controller-in-the-Loop Environment

We developed a C++ EnvPool environment [24] in which each instance executes the complete controller–simulator closed loop. MuJoCo [25] advances the floating-base dynamics, contacts, and actuators, while the embedded locomotion stack performs state processing, gait and foothold generation, whole-body control, and joint-command execution. Policy actions are held between policy updates while the 500 Hz lower-level loop continues to run. The environment also defines initialization, observation noise, command and disturbance generation, reward, and termination. EnvPool parallelizes complete copies of this loop rather than replacing the controller with a reduced training model. This design makes the proposed hierarchical interface trainable at high throughput while preserving its model-based execution path.

B. Policy Training

The proposed policies are trained from scratch with PPO [26]. The primary results use three independent policies different seeds. Each completed 85 million transitions; the training curves plateau by this horizon, with no sustained improvement near the end of training. The final checkpoint and its corresponding normalization statistics are used for every evaluation condition.

Training uses 256 parallel environments and rollouts of 512 steps per environment. Each PPO update performs ten epochs with minibatches of 1024. The actor and critic are separate $[256, 128]$ tanh networks. The learning rate is 10^{-5}, the entropy coefficient is 0.05, $\gamma = 0.995$, and $\lambda = 0.97$. Policy and value clipping parameters are 0.1 and 0.2, respectively; the target Kullback–Leibler (KL) divergence is 0.01, the gradient norm is limited to 0.1, and policy standard deviations are bounded to $[0.05, 0.30]$.

The curriculum exposes the policy to smaller perturbations during initial learning and then progressively expands the recovery task. Training disturbances are additive velocity changes in the six signed cardinal directions. Their ceiling increases from 1 to 2 m/s at 8.1 million transitions and to 3 m/s at 24.3 million transitions. Every 1.6 s, a disturbance is skipped with probability 0.2; otherwise its magnitude is drawn from the upper half of the current range with probability 0.75 and from the lower half otherwise. Initial conditions include a ± 0.03 m spawn-height perturbation.

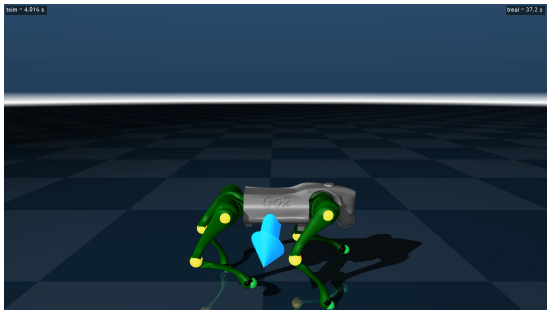

Fig. 2. Go2 simulation environment. The arrow illustrates the disturbance direction; evaluation applies instantaneous velocity changes.

Commands are sampled per episode with $v_x \in [0, 1]$ m/s, $v_y \in [-0.2, 0.2]$ m/s, and yaw rate in $[-0.2, 0.2]$ rad/s.

C. Validation Protocol

All experiments use a 15 kg Go2 model on flat ground (Fig. 2), a 2 ms simulation step, ground friction 1.0, and joint torque limits of $[23.7, 23.7, 45.43]$ N·m for the abduction, hip, and knee joints. Training-time friction randomization is disabled during evaluation. Uniform observation noise has half-widths of 0.05 rad for orientation, 0.1 m/s for linear velocity, 0.2 rad/s for angular velocity, 0.01 rad for joint position, and 1.5 rad/s for joint velocity; body position is unperturbed.

To contextualize survival, we include two reference configurations. The model-based *MPC+WBIC* controller is the Mini-Cheetah architecture described by Kim *et al.* [6], combining convex MPC reaction-force planning [5], reactive foot placement, and WBIC. The *source-recipe-based joint-space RL* reference follows a Unitree Go2 recipe adapted from Rudin *et al.* [14] and implemented with `rsl_rl`. Its 50 Hz policy produces 12 joint-position offsets, scaled by 0.25 rad and tracked with fixed gains $K_p = 20$ and $K_d = 0.5$. We enable its existing body-height objective with the term $-30(z-0.30)^2$ after 1500 iterations with the source recipe and continue training for 500 iterations. With 4096 environments and 24 rollout steps per iteration, this totals 196.608 million transitions for each of three policies. Both references are reimplemented in the same MuJoCo–EnvPool framework and share the Go2 plant, physics, actuator limits, initialization and noise models, disturbance generator, and evaluation code with the proposed controller. They retain their architecture-specific observations, actions, objectives, and update rates; the comparison therefore contextualizes complete configured systems rather than isolating their action representations.

The evaluation grid comprises six directions (forward, backward, left, right, up, and down) and ten magnitudes from 0.5 to 5.0 m/s in 0.5 m/s increments. Each of the 60 cells contains 100 episodes per policy. The proposed and source-recipe-based joint-space results each cover three independently trained policies; MPC+WBIC is one fixed model-based controller. An additional undisturbed condition is excluded from the grid score.

As in Rudin *et al.* [14], a single disturbance is applied as an instantaneous change in base velocity,

$$\mathbf{v}^+ = \mathbf{v}^- + \mathbf{R}_{wb}(t_0)\Delta\mathbf{v}_b, \tag{10}$$

where $\Delta\mathbf{v}_b$ is specified in the body frame at onset t_0. The perturbation changes translational velocity without directly changing angular velocity. Onset is sampled over gait phase after three cycles. MPC+WBIC and Ours use their gait schedulers; joint-space RL uses an external 0.4 s phase clock for onset sampling. Initial pose and joint angles are randomized, and the forward command is sampled from 0.8 ± 0.05 m/s. As a sensitivity check, we additionally evaluated the proposed controller using constant 0.2 s external forces with impulses matched to the velocity changes, $F\Delta t = m\Delta v$. Its survival trends were qualitatively consistent with the instantaneous-velocity results, which we report to maintain alignment with prior evaluation protocols.

An episode survives if it completes the 5.5 s post-disturbance window without base contact force exceeding 1 N, absolute roll exceeding 0.8 rad, absolute pitch exceeding 1.0 rad, or nonfinite state. These criteria measure survival rather than return to the commanded motion.

D. Training-Time Channel-Ablation Protocol

To measure the contribution of each action group after the controller has adapted to its absence, we train five additional policies from scratch using the same PPO configuration, curriculum, and training horizon. For each policy, one action group is overwritten with its neutral value before every simulator step throughout training, and the same constraint is retained at evaluation. The remaining action groups can therefore adapt during training instead of encountering a channel removal only after the policy has converged.

Zero body-velocity residuals retain the external command, and zero foothold residuals retain the heuristic foothold. Fixed height retains the nominal 0.32 m body target, while fixed cadence retains the nominal phase increment of 0.005 per controller update. The fixed-GRF condition sets $f_x = f_y = 0$ and $f_z = 125$ N per stance leg. Because the GRF action supplies an absolute target rather than a residual, this condition removes learned force modulation but not the stance force itself. Each retrained controller and the corresponding unablated reference are evaluated on the complete 60-cell grid with 100 episodes per cell, plus an undisturbed control condition. The common grid spans 0.5–5.0 m/s. When backward survival remains above 50% at 5.0 m/s, evaluation continues to the first crossing solely to estimate c_{50}; these extension trials are excluded from the grid-survival score. The unablated and ablated arms use different evaluation seeds.

E. Evaluation Measures

Survival is the primary disturbance outcome because it is defined identically for model-based and learned controllers and is independent of their training objectives. The direction-specific survival curves retain the response over the complete disturbance range. The 50% survival threshold c_{50}

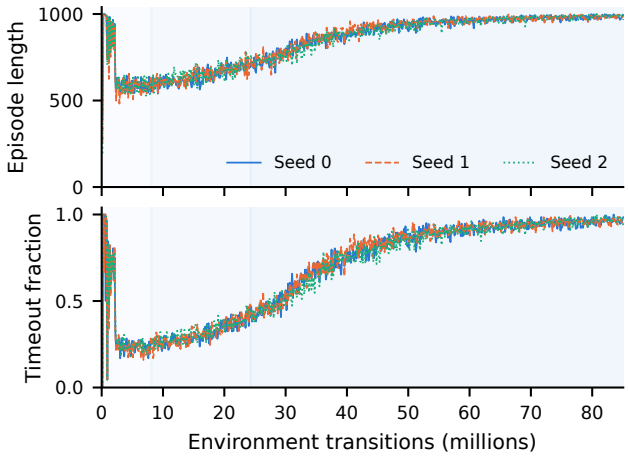

Fig. 3. Training episode length and timeout fraction for three seeds. Shading indicates disturbance ceilings of 1, 2, and 3 m/s, with transitions at 8.1 and 24.3 million transitions. Curves are unsmoothed.

provides a compact, physically interpretable survival limit and is obtained by linear interpolation at the first downward crossing of the mean curve. A curve above 50% throughout its evaluated range is right-censored at the largest tested magnitude; the full curves retain any later recrossings.

Let $s_{k,d,j}$ be the survival percentage for training seed k, direction d, and magnitude j. The grid score is

$$G_k = \frac{1}{60} \sum_{d=1}^{6} \sum_{j=1}^{10} s_{k,d,j}. \tag{11}$$

Each cell receives equal weight, so the score is the expected survival rate under a uniform selection from the tested direction–magnitude conditions. Because it depends on the chosen grid, it is reported as a secondary summary rather than a replacement for the curves or directional thresholds. We distinguish variation across independently trained policies from episode-level uncertainty for a fixed policy.

Separately, five undisturbed full-trajectory trials per controller characterize nominal tracking and mechanical effort. We report realized forward speed, joint-torque RMS, absolute mechanical power, and mechanical cost of transport, $\mathrm{CoT}_{\mathrm{abs}} = \int \sum_i |\tau_i \dot{q}_i| \, dt / (mgD_{xy})$, where D_{xy} is horizontal displacement. All controllers receive the same nominal 0.8 m/s forward command, while their mean realized speeds vary by approximately ± 0.1 m/s. Electrical losses are not included.

IV. Results

A. Training Viability

All three policies completed 85.1 million transitions without training collapse. Figure 3 shows that episode length and timeout fraction plateau near the end of training; final timeout fractions range from 92.9% to 98.4%. This demonstrates that the 25-dimensional supervisory interface can be optimized while executing the complete scheduler, planner, and WBIC loop.

TABLE IV

Mean survival over the 60 disturbance conditions for the complete configured systems. Learned-controller results report mean ± sample SD across three training seeds; MPC+WBIC is one fixed controller.

Controller	Grid survival (%)
MPC+WBIC	50.35
Source-recipe-based joint-space RL	54.01 ± 5.55
Proposed	77.29 ± 1.15

B. Disturbance Survival

All evaluated controller instances completed every episode in the additional undisturbed condition, confirming that the disturbance results distinguish disturbance survival rather than basic nominal locomotion. Under this experimental configuration, the proposed controller achieves a mean grid survival of $77.29 \pm 1.15\%$ across three training seeds, 26.94 percentage points above the 50.35% obtained by the model-based MPC+WBIC comparison controller (Table IV). Each proposed policy survives every episode in every direction through 2.0 m/s. For context, the source-recipe-based joint-space configuration achieves $54.01 \pm 5.55\%$. This result characterizes that configured policy rather than the attainable robustness of joint-space RL under task-specific disturbance training.

Relative to MPC+WBIC, the proposed controller's largest threshold gains are lateral: c_{50} increases from 2.26 to 4.19 m/s leftward and from 1.96 to 3.80 m/s rightward. Backward survival remains 70.3% at 5.0 m/s, so its threshold lies beyond the tested range. These results demonstrate that the proposed system has a wider disturbance envelope than the model-based comparison controller; they do not show that structured references maximize survival among learned control formulations. Learned-policy robustness depends strongly on the training objective and disturbance distribution, and a joint-space policy can likewise be optimized for this particular protocol. The comparison therefore contextualizes complete configurations rather than isolating the effect of the action representation.

Seed variation is concentrated near the transitions between survival and failure. The maximum per-cell standard deviation is 14.7 percentage points at 4.0 m/s leftward, while the corresponding per-seed thresholds span only 3.95–4.26 m/s. The unablated reference used in the ablation study obtains 77.43% grid survival, close to the 77.29% primary mean.

C. Training-Time Action-Channel Ablations

Figure 5 summarizes the pooled and directional effects of training with one action group held neutral. The foothold residual has the largest marginal contribution: without it, grid survival falls from 77.43% to 66.60%, a loss of 10.83 percentage points. The body-velocity, force-modulation, cadence, and height conditions reduce the pooled score by 2.37, 1.50, 1.12, and 0.57 points, respectively. All five ablated controllers survive every undisturbed control episode, so

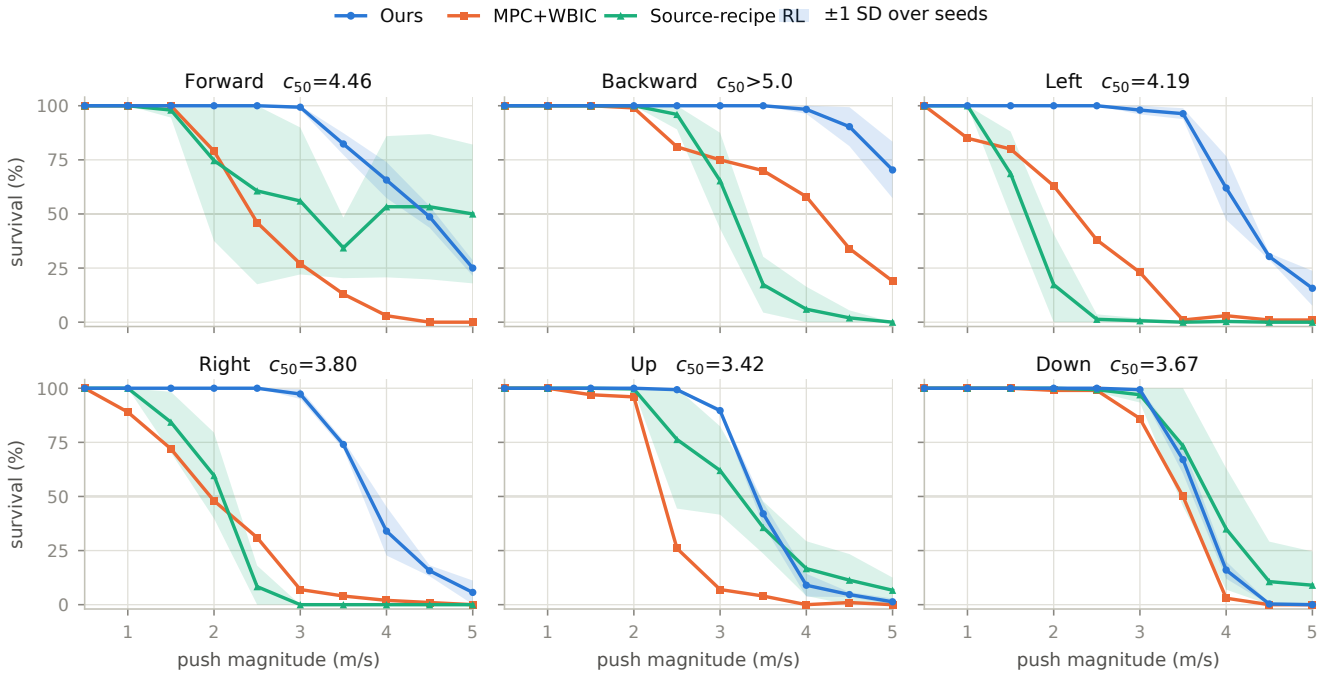

Fig. 4. Survival under additive velocity disturbances, with 100 episodes per cell and policy. Curves and bands show the mean and ± 1 sample standard deviation across three training seeds for the proposed and source-recipe-based joint-space controllers. MPC+WBIC is one fixed controller.

these differences reflect disturbance recovery rather than failure to learn nominal locomotion.

The effects are strongly directional. Training without foothold residuals reduces c_{50} by 1.19 m/s leftward and 0.93 m/s rightward, but increases it by 0.31 m/s downward. This sign reversal shows that the effect of foothold adaptation is direction-dependent rather than uniformly beneficial. Although fixing cadence changes the pooled score only modestly, it reduces upward c_{50} by 0.62 m/s and is the only ablation that reduces backward c_{50} (-0.26 m/s). The smaller height and force effects likewise change sign by direction, so their pooled scores should not be interpreted as evidence that the policy never uses them. The extended backward evaluation gives an unablated c_{50} of 5.15 m/s; the fixed-GRF ablation produces the largest backward increase ($+0.40$ m/s).

D. Nominal Locomotion Characteristics

Figure 6 shows that the retrained fixed-cadence controller has lower nominal mechanical power and CoT than the unablated controller. Its pooled survival changes only modestly, with improvements in some directions and reductions in others. Removing the foothold or body-velocity residual likewise produces a lower-effort nominal gait. The foothold ablation combines the lowest torque RMS with the largest grid-survival loss, indicating that the additional actuation associated with learned foothold correction is functional rather than uniformly inefficient.

V. Discussion

The results validate the hierarchy as a controller architecture in simulation. Its distinguishing feature is the coordinated authority of one policy over several meaningful references, while the scheduler, planner, and WBIC retain high-frequency execution. Disturbance survival tests this interface; it does not establish a performance-optimal form of learned control.

The ablations distinguish a channel's contribution after the controller has adapted to its absence. Foothold adaptation is the least replaceable component: removing it produces the only broad degradation and the largest lateral losses. Its improvement under downward disturbances also shows that learned foothold corrections are not uniformly beneficial and motivates direction-dependent calibration. The fixed-cadence controller exhibits lower nominal mechanical effort than the adaptive controller while retaining similar pooled survival. Cadence adaptation instead changes how robustness is distributed across disturbance directions; its observed contribution is directional flexibility rather than demonstrated energy savings. The smaller, signed force and height effects likewise suggest compensatory pathways rather than uniformly inactive channels. Even without foothold residuals, the controller retains 66.60% grid survival, so no single action group accounts for the complete robustness advantage. The fixed-GRF condition remains specific to the 125 N target and does not test removal of stance force itself.

The reference controllers contextualize validation rather than isolate action representation. Their observations, actions, objectives, training, and update rates differ. In particular, the lower source-recipe-based joint-space result is not an inherent limit: learned robustness depends on training distribution, and joint-space policies can be optimized for this benchmark. Our claim is instead that substantial adaptation can be learned through explicit references while model-based

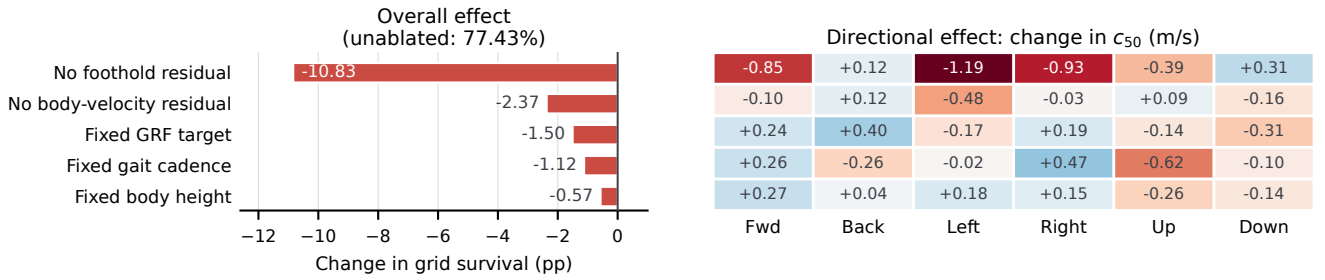

Fig. 5. Training-time action-channel ablations: grid-survival change (left) and directional c_{50} change (right; negative is worse). Backward c_{50} uses extensions above 5.0 m/s excluded from the grid score.

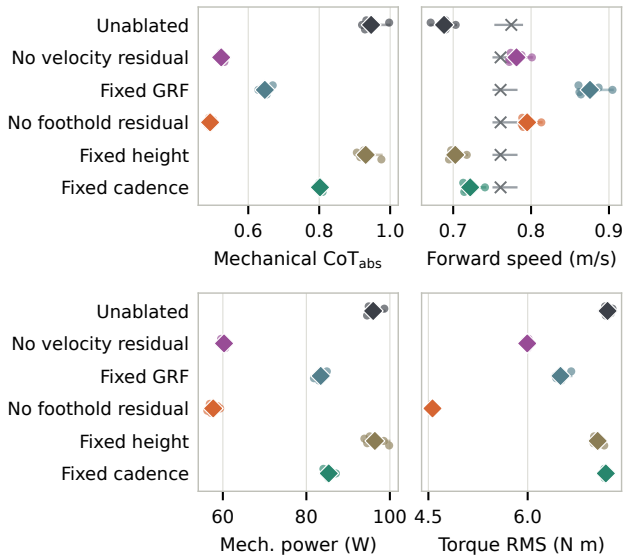

Fig. 6. Nominal behavior of the retrained ablation arms over five undisturbed episodes. Small circles denote episodes, diamonds denote means, and crosses in the speed panel denote commanded-speed means.

execution is retained. The hardware-demonstrated WBIC architecture [6] provides a credible basis for future sim-to-real work, but does not establish transfer of the complete hierarchy.

Finally, the single-channel ablations do not resolve interactions or mutual compensation between multiple action groups, and the source-recipe-based comparison is not training-matched. Evidence is limited to flat-ground simulation with noisy state. An impulse-matched 0.2 s force-pulse check produced qualitatively consistent results for the proposed controller, but was not conducted for the reference controllers; other push durations and rotational disturbances remain unevaluated. Survival does not establish return to commanded motion, and transfer, safety, and hardware reliability remain unevaluated.

VI. Conclusion

We introduced a learned supervisor over structured locomotion references while a scheduler, planner, and WBIC retain high-frequency execution. The hierarchy achieves 77.29% mean survival versus 50.35% for the model-based reference, and training-time ablations identify foothold adaptation as its largest marginal channel contribution and reveal direction-dependent effects of gait-rate adaptation. These results validate the architecture in simulation without claiming maximum benchmark performance; broader tasks, realistic estimation, and hardware experiments must determine its practical advantages.

Acknowledgment

OpenAI Codex assisted with language editing but generated no experimental data. The authors take responsibility for the final content.

References

[1] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Trans. Robotics*, vol. 40, pp. 43–63, 2024.

[2] M. Hutter, H. Sommer, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, "Quadrupedal locomotion using hierarchical operational space control," *The International Journal of Robotics Research*, vol. 33, no. 8, pp. 1047–1062, 2014.

[3] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2018, pp. 2245–2252.

[4] U. Saranli and D. E. Koditschek, "Template based control of hexapedal running," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, 2003, pp. 1374–1379.

[5] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2018, pp. 1–9.

[6] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," arXiv:1909.06586 [cs.RO], 2019.

[7] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[8] M. H. Raibert, *Legged Robots That Balance*. Cambridge, MA: MIT Press, 1986.

[9] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[10] F. Jenelten, R. Grandia, F. Farshidian, and M. Hutter, "TAMOLS: Terrain-aware motion optimization for legged systems," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3395–3413, 2022.

[11] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[12] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.
[13] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
[14] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. Conf. Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 164. PMLR, 2022, pp. 91–100.
[15] H. Duan, J. Dao, K. Green, T. Apgar, A. Fern, and J. Hurst, "Learning task space actions for bipedal locomotion," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2021, pp. 1276–1282.
[16] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "DeepGait: Planning and control of quadrupedal gaits using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3699–3706, 2020.
[17] Z. Xie, X. Da, B. Babich, A. Garg, and M. van de Panne, "GLiDE: Generalizable quadrupedal locomotion in diverse environments with a centroidal model," in *Algorithmic Foundations of Robotics XV*. Springer, 2023, pp. 523–539.
[18] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "RLOC: Terrain-aware legged locomotion using reinforcement learning and optimal control," *IEEE Trans. Robotics*, vol. 38, no. 5, pp. 2908–2927, 2022.
[19] Y. Chen and Q. Nguyen, "Learning agile locomotion and adaptive behaviors via RL-augmented MPC," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2024, pp. 11 436–11 442.
[20] M.-G. Kim, D. Kang, H. Kim, and H.-W. Park, "A modular residual learning framework to enhance model-based approach for robust locomotion," *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 9072–9079, 2025.
[21] D. Kang, J. Cheng, M. Zamora, F. Zargarbashi, and S. Coros, "RL + model-based control: Using on-demand optimal control to learn versatile legged locomotion," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6619–6626, 2023.
[22] A. Pandala, R. T. Fawcett, U. Rosolia, A. D. Ames, and K. Akbari Hamed, "Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced- and full-order models," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6622–6629, 2022.
[23] H. Wang, R. Zhou, L. Ding, T. Liu, Z. Zhang, P. Xu, H. Gao, and Z. Deng, "Whole-body constrained learning for legged locomotion via hierarchical optimization," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 7460–7467, 2025.
[24] J. Weng, M. Lin, S. Huang, B. Liu, D. Makoviichuk, V. Makoviychuk, Z. Liu, Y. Song, T. Luo, Y. Jiang, Z. Xu, and S. Yan, "EnvPool: A highly parallel reinforcement learning environment execution engine," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 22 409–22 421.
[25] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2012, pp. 5026–5033.
[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347 [cs.LG], 2017.