Structured Action-Effect Observables for Residual RL under Hidden Actuator Drift

Sarvesh R. Babu

June 2026

Abstract

Motivation. Modern deep RL can train strong locomotion policies, but the usual robustness story depends on guessing deployment dynamics ahead of time. Dynamics randomization works when the real actuator failure, delay, gain drift, or nonlinear response lies inside the randomized support. If it does not, the expensive policy is out of distribution. This project asks how to keep that policy useful instead of replacing it. I assume a clean expert already exists, freeze it, and train a smaller residual policy whose job is to notice when expert commands no longer produce expected motion and add a bounded correction.

Problem. The clean expert produces $b_t = \pi_0(o_t)$. The residual observes features ϕ_t and outputs δ_t, giving command $u_t = \mathrm{clip}(b_t + \alpha\delta_t, a_{\min}, a_{\max})$. The simulator then applies a hidden actuator map $a_t = g_{z_t}(u_t, u_{t-1}, a_{t-1}, \epsilon_t)$, where z_t may be clean, delayed, mismatched, nonlinear, or stressed. Since z_t is unobserved, residual control is a POMDP: the policy must infer actuator mode from recent action-effect evidence before it can correct the expert.

Method. I train the residual with SAC while keeping π_0 frozen. The method contribution is the observation map, not the RL optimizer. At each step I store an action-effect vector $y_t = [b_t, \alpha\delta_t, u_t, \Delta o_t]$ and compute features over a causal window $Y_{t-W:t}$. I compare raw observations, histories, phase features, low-order actuator-map summaries, flat RG features, full pair-depth RG tensors, and temporal RG tensor sequences. The RG tensor is inspired by Ott's RG view of the CLT and Hamdan's fixed-point extension over spaces including $\mathbb{R}^2$. For each channel pair, I whiten bivariate action-effect samples, compare their empirical characteristic function to a standard 2D Gaussian, then repeat after adjacent-pair coarse-graining. This gives $R_t[k, i, j] \in \mathbb{R}^{K\times C\times C}$, a depth-indexed summary of pairwise action-effect geometry. The statistic is not used as a normality test; it is an online feature for hidden actuator inference.

Experiments and metric. I evaluate Hopper, Walker2d, and Ant in MuJoCo across fixed perturbations, held-out smooth ramps, and held-out abrupt switches. Training suites include clean, stress, mixed, safe-mix, RG-depth, RG-coverage, full tensor, temporal tensor, and low-order system-ID variants. The metric is diff $= R(\pi_0 + \pi_\theta) - R(\pi_0)$, so positive values mean the residual improved the frozen expert under the same perturbation. Units are undiscounted MuJoCo return points.

Regime	**Representative results**	**Interpretation**
Nonlinear drift	Hopper: RG+actmap $+111.3 \pm 151.0$, temporal RG $+103.7 \pm 77.6$, full RG tensor $+91.1 \pm 79.2$, coverage RG $+46.2 \pm 6.4$ with CI $[+30.3, +62.1]$. Ant action-RG: $+414.6 \pm 285.9$ and $+400.4 \pm 203.7$.	Best evidence for action-effect geometry. Simple gain correction is not enough here.
Mismatch / ramps	Walker2d fixed mismatch: full RG tensor $+335.9 \pm 214.7$, moments $+388.9 \pm 258.1$, RG+actmap $+408.3 \pm 729.9$. Walker2d ramp mismatch: full RG tensor $+1142.2 \pm 24.5$ with CI $[+1081.3, +1203.1]$.	Residual works when deployment drift is smooth or near training support.
Abrupt switches	Hopper switch delay -82.3 ± 39.1; Walker2d switch nonlinear -79.9 ± 92.4; Ant switch stress -344.4 ± 263.9.	OOD temporal changes need explicit fast adaptation or change detection.

Takeaway. Residual RL can salvage an expensive frozen expert under hidden actuator drift, but only when the residual receives features that make the hidden dynamics observable. Structured action-effect features, especially RG tensors, help most for nonlinear drift and smooth mismatch drift. Robust response to abrupt out-of-distribution mode switches remains open.

1 Introduction

Modern deep RL can produce strong locomotion policies in simulation, but those policies are only reliable under the dynamics they were trained to handle [24, 2, 9]. The usual answer is dynamics randomization: train a large policy over many simulated dynamics so deployment looks like another sample from the training distribution [19, 23]. This works when the randomization actually covers deployment. But that assumption is also the weakness. If the real actuator failure, delay, gain drift, or nonlinear response was not in the randomization range, the expensive policy is suddenly out of distribution.

This project asks how to keep that policy useful instead of replacing it. I assume we already have a clean expert that is expensive to train and works well under nominal dynamics. I freeze that expert and train a smaller residual policy that adds a bounded correction [13]. The residual does not need to relearn locomotion from scratch. Its job is narrower: notice when the expert's commands are no longer producing the expected motion, then add the correction needed to keep the trajectory alive.

The hard part is that the actuator error is hidden. The environment applies an unobserved actuator transformation, such as delay, mismatch, nonlinear response, or compound stress, before the action reaches the simulator. The residual-control problem is therefore a POMDP: the policy does not observe the actuator mode directly, but must infer it from recent action-effect evidence.

This makes observability the main question. A history of observations may help, and recurrent policies are a natural baseline for partial observability [8]. But hidden actuator drift is specifically an action-effect problem: the policy needs to estimate how commanded actions are being transformed into state changes. I therefore test structured action-effect features, including phase features, action-history features, low-order actuator-map summaries, and a full RG tensor representation.

Empirically, residuals help most when the perturbation is fixed or changes smoothly and the training mixture contains the relevant actuator variation. The clearest gains are on nonlinear and mismatch-like settings. Full RG tensor variants improve Hopper nonlinear performance by about +91 to +111 MuJoCo return points, and Walker2d ramp mismatch improves by about +1142 return points. Mismatch/slippage-like actuator-map perturbations also show positive results in several settings when deployment is close to the training mixture.

The negative result is just as important. Held-out abrupt switch evaluations are much harder than fixed perturbations or smooth ramps. This setting tests emergent temporal generalization: sudden hidden-mode changes were not present in the same form during training. The policy often cannot detect the new mode and intervene quickly enough without damaging the trajectory. Thus, the results do not solve broad domain randomization. They separate two regimes: residual adaptation works when deployment is inside or near the training support, but abrupt out-of-distribution hidden-mode switches remain open.

The contributions are threefold. First, I formulate hidden actuator drift for residual locomotion control as a POMDP in which a frozen clean expert is corrected by a learned residual policy. Second, I implement structured phase and RG action-effect representations, including a full pair-depth RG tensor derived from recent action and transition statistics. Third, I evaluate these representations across fixed, ramp, and abrupt-switch perturbation suites, showing both strong positive results under nonlinear and mismatch-like perturbations and clear limitations under emergent hidden-mode switches.

2 Related Work and Background

Residual reinforcement learning. Residual reinforcement learning trains a corrective policy on top of an existing controller rather than learning the whole behavior from scratch. This is useful when a nominal controller is already competent but fails under shifted dynamics or unmodeled effects. Johannink et al. introduced residual RL for robot control, showing that learned residuals can improve classical controllers

while preserving useful prior structure [13]. My setting follows this residual-control view, but the base controller is a frozen clean expert and the residual must adapt to hidden actuator perturbations.

Robust RL, domain randomization, and online adaptation. Robust robot learning often trains over randomized dynamics so the learned policy transfers to many deployment conditions [19, 23]. Other work explicitly estimates or adapts to hidden environment parameters, as in universal policies with online system identification and rapid motor adaptation [25, 14, 17]. These ideas are central in legged locomotion [12, 15, 22]. My experiments support the same broad lesson: policies work best when the training distribution contains the relevant deployment variation. I also test a harder case, abrupt hidden-mode switches whose temporal structure is held out.

Partial observability, action delay, and phase. Because the actuator mode is hidden, the same proprioceptive observation and expert action can lead to different next states depending on whether the hidden actuator is clean, delayed, mismatched, nonlinear, or stressed. History and recurrence are standard tools for partial observability [8], and action delay is a known difficulty for RL controllers [5]. Locomotion is also cyclic, so phase is often a useful state variable. Phase-functioned neural networks and phase-conditioned locomotion controllers use phase to organize behavior over the gait cycle [11, 20, 21]. In this project, phase helps with timing, but phase alone does not identify the actuator transformation.

RG fixed points. The RG representation is inspired by the renormalization-group view of the central limit theorem. Ott presents the CLT as an RG flow: repeatedly standardize and average independent samples, and the resulting distribution flows toward a Gaussian fixed point under suitable moment assumptions [18]. In the scalar case, if $X \sim \mu$,

$$T\mu = \mathcal{L}\left(\frac{X_1 + X_2}{\sqrt{2}}\right), \qquad X_1, X_2 \overset{\text{i.i.d.}}{\sim} \mu.$$

Repeated application gives

$$T^k\mu = \mathcal{L}\left(\frac{X_1 + \cdots + X_{2^k}}{\sqrt{2^k}}\right).$$

The standard Gaussian is a fixed point because $(X_1 + X_2)/\sqrt{2}$ is standard Gaussian whenever X_1 and X_2 are independent standard Gaussians. In characteristic-function form,

$$\phi_{T\mu}(\xi) = \phi_\mu(\xi/\sqrt{2})^2, \qquad \phi_{\mathcal{N}(0,1)}(\xi) = e^{-\xi^2/2}.$$

The useful idea here is not the CLT itself, but the depth profile of this flow: simple or Gaussian-like distributions behave differently under coarse-graining than distributions with nonlinear, multimodal, or structured dependence.

From Ott and Hamdan to action-effect tensors. Hamdan extends the fixed-point/contraction perspective to quantitative non-commutative CLTs, including contractions over spaces of probability measures on $\mathbb{R}$ and $\mathbb{R}^2$ with the appropriate Gaussian analogue as the fixed point [7]. This motivates a two-dimensional lift: instead of applying a scalar RG statistic to one coordinate at a time, I apply a bivariate Gaussian-null discrepancy to pairs of action-effect channels. Classical empirical-characteristic-function tests use related quantities to test Gaussianity [4, 1, 10], and RG-style coarse variables have also been connected to representation learning [16]. I use the statistic differently: not as a hypothesis test, but as an online observation for a residual policy.

3 Problem Formulation

3.1 Hidden-actuator residual POMDP

Let $x_t \in \mathcal{S}$ denote the full simulator state and $o_t \in \mathcal{O}$ the policy observation. A clean expert policy π_0 maps the observation to a base action

$$b_t = \pi_0(o_t).$$

The residual policy produces a corrective action

$$\delta_t \sim \pi_\theta(\cdot \mid \phi_t),$$

where ϕ_t is the residual policy input. The commanded action is

$$u_t = \mathrm{clip}\left(b_t + \alpha\delta_t, a_{\min}, a_{\max}\right),$$

where α controls the residual scale. A hidden actuator map transforms the command into the applied action:

$$a_t = g_{z_t}(u_t, u_{t-1}, a_{t-1}, \epsilon_t),$$

and the environment transitions as

$$x_{t+1} \sim P(\cdot \mid x_t, a_t), \qquad r_t = r(x_t, a_t).$$

The hidden mode z_t changes the effective transition dynamics. Since z_t is not directly observed, the residual-control problem is a POMDP.

3.2 Actuator perturbations

The clean setting applies the command directly:

$$g_{\text{clean}}(u_t) = u_t.$$

The delay setting applies an older command,

$$g_{\text{delay}}(u_t) = u_{t-d}.$$

The mismatch setting changes actuator gain or bias,

$$g_{\text{mismatch}}(u_t) = Mu_t + c,$$

capturing calibration-like or slippage-like failures where the same command no longer produces the expected physical effect. The nonlinear setting applies a nonlinear actuator response,

$$g_{\text{nonlinear}}(u_t) = h(u_t),$$

where h may include saturation, deadzone, or nonlinear gain. The stress setting combines multiple effects:

$$g_{\text{stress}} = g_{\text{noise}} \circ g_{\text{nonlinear}} \circ g_{\text{mismatch}} \circ g_{\text{delay}}.$$

Stress is harder because the hidden actuator state is not a single scalar disturbance; delay, gain mismatch, bias, motor lag, stochastic noise, and deadzone are partially confounded.

3.3 Learning objective and metric

The expert is fixed and only θ is updated. The residual is trained with Soft Actor-Critic [6], using the maximum-entropy objective

$$J(\theta) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(r_t + \lambda\mathcal{H}\left(\pi_\theta(\cdot \mid \phi_t)\right)\right)\right].$$

SAC is the optimizer, not the methodological novelty. The research question is what representation ϕ_t makes the hidden actuator mode observable enough for the residual to improve the frozen expert.

For every evaluation, I report

$$\text{diff} = R(\pi_0 + \pi_\theta) - R(\pi_0),$$

where $R(\pi_0 + \pi_\theta)$ is the undiscounted episode return of the expert plus residual and $R(\pi_0)$ is the return of the frozen expert under the same perturbation. Positive means the residual helped; negative means it hurt. The unit is MuJoCo return points, not percent or normalized score.

4 Method

4.1 Representation variants

The method keeps the base locomotion controller fixed and changes only the residual policy input ϕ_t. I compare raw observations, short histories, phase features, low-order system-identification features, and RG action-effect tensors.

Raw observation. The raw baseline receives the current MuJoCo observation, $\phi_t = o_t$. This tests whether the current state alone is enough to correct hidden actuator drift.

History. The history baseline concatenates recent observations and actions,

$$\phi_t = [o_{t-H:t}, u_{t-H:t}] .$$

This is the generic POMDP baseline. If enough history is available, the residual may infer the actuator mode from delayed consequences of its actions.

Phase. For cyclic locomotion, I include phase features. For a selected coordinate q_t and local velocity estimate $\dot{q}_t$,

$$\varphi_t = \operatorname{atan2}\left(\dot{q}_t/s_{\dot{q}}, q_t/s_q\right), \qquad p_t = [\cos\varphi_t, \sin\varphi_t].$$

Phase helps condition corrections on gait timing, but does not identify the actuator map by itself.

Action-effect features. At each step I store

$$y_t = [b_t,\ \alpha\delta_t,\ u_t,\ \Delta o_t], \qquad \Delta o_t = o_t - o_{t-1}.$$

The window $Y_{t-W:t}$ contains evidence about how expert actions and residual corrections are mapping into motion. If the command is delayed, mismatched, or nonlinear, the relationship between u_t and Δo_t changes.

Low-order system identification. Low-order controls summarize the recent action-effect window using means, variances, correlations, and approximate actuator-map terms. Conceptually they estimate

$$\Delta o_t \approx A u_t + c.$$

These baselines test whether RG gains can be explained by simpler statistics.

4.2 RG action-effect tensor

For each action-effect channel pair (i, j), I form bivariate samples

$$y_\tau^{(i,j)} = \begin{bmatrix} Y_{\tau,i} \\ Y_{\tau,j} \end{bmatrix}, \qquad \tau = t - W, \ldots, t.$$

I whiten them,

$$z_\tau^{(i,j)} = \Sigma_{ij}^{-1/2} \left(y_\tau^{(i,j)} - \bar{y}^{(i,j)} \right),$$

then compute the empirical characteristic function

$$\hat{\phi}_{ij}(\xi) = \frac{1}{W} \sum_{\tau=t-W}^{t} \exp(i\xi^\top z_\tau^{(i,j)}),$$

and compare it to the standard 2D Gaussian characteristic function

$$\phi_{\mathcal{N}(0,I)}(\xi) = \exp\left(-\frac{1}{2} \|\xi\|_2^2 \right).$$

The discrepancy is

$$D_{ij} = \max_{\xi \in \Xi} \frac{\left| \hat{\phi}_{ij}(\xi) - \phi_{\mathcal{N}(0,I)}(\xi) \right|}{\|\xi\|_2^3 + \epsilon}.$$

To obtain multiple RG depths, I repeatedly coarse-grain adjacent samples,

$$z_m^{(k+1)} = \frac{z_{2m}^{(k)} + z_{2m+1}^{(k)}}{\sqrt{2}},$$

and recompute the discrepancy at each depth. This produces

$$R_t[k, i, j] = D_{ij}^{(k)} \in \mathbb{R}^{K \times C \times C}.$$

Flat RG features select a limited number of entries and feed them to a standard MLP. The full tensor version preserves all pair-depth structure and uses a convolutional feature extractor before the SAC actor and critic. The temporal sequence version stores several recent RG tensors and uses a recurrent encoder to summarize their evolution.

4.3 Training mixtures

I train residual policies under several perturbation mixtures. Clean runs train without perturbation. Stress runs train under compound actuator stress. Mixed runs include multiple perturbation types. Safe-mix runs include clean episodes along with perturbed episodes, encouraging the residual to learn both when to correct and when to leave the expert alone. This matters because an aggressive residual can damage clean or mildly perturbed trajectories.

5 Experimental Setup

5.1 Environments and evaluation regimes

I evaluate in MuJoCo Hopper, Walker2d, and Ant [24, 3]. Hopper is lower-dimensional and often exposes actuator errors clearly. Walker2d has more coordinated bipedal dynamics. Ant is higher-dimensional and was generally harder for the residual policies in these experiments. Each environment uses a frozen clean expert; the residual only learns an additive correction.

There are three evaluation regimes. Fixed perturbations keep the perturbation type constant throughout the episode. Held-out ramps change the perturbation smoothly over the episode. Held-out switches change the perturbation abruptly. Switches are not the same temporal structure seen during training, so they test out-of-distribution temporal generalization.

5.2 Experiment inventory

The full experiment set includes early sanity checks, algorithm feasibility runs, main stress baselines, train-mixture sweeps, representation ablations, RG-depth and coverage ablations, full tensor models, sequence models, and low-order system-identification controls.

The grouped result tables contain 417 fixed rows, 120 ramp rows, and 72 switch rows with standard deviations and confidence intervals. Many key groups have $n = 3$, so I interpret results by looking for large effect sizes and consistency rather than treating every positive mean as definitive.

6 Results

6.1 Overview

The main positive result is nonlinear actuator drift. Across the structured-feature runs, nonlinear perturbations repeatedly produce positive residual gains, especially on Hopper and Ant. This matters because

Suite	Meta runs	Normal evals	Purpose
poster_fast	13	7	Early Hopper sanity and poster runs
algos_fast	9	9	Quick algorithm feasibility checks
main	10	0	Initial MuJoCo main attempts
final	54	54	Main stress baseline across Hopper, Walker2d, and Ant
mixed	27	27	Clean vs. stress vs. mixed train perturbations
repr	27	27	Phase and RG representation ablation under stress
safe_mix	27	27	Clean-inclusive training mixture
helpful_combo	43	43	Phase, RG, and action-feature combinations
rg_depths	71	71	RG contraction-depth ablation
rg_coverage	16	16	RG pair/channel coverage ablation
rg_tensor_cnn	12	12	Full pair-depth RG tensor with CNN extractor
rg_sequence_bilstm	12	6	Temporal RG tensor sequence model
loworder_sysid	18	18	Moment and actuator-map controls
actuator_map_stress	6	6	Full RG tensor plus actuator-map features

Table 1: Experiment suites. Meta runs are trained residual policies; normal evals are fixed-perturbation evaluation summaries. Held-out ramp and switch evaluations are reported separately.

Held-out evaluation set	Completed summaries
All switch/ramp summaries	234
rg_tensor_cnn ramp eval	12/12
rg_sequence_bilstm ramp eval	12/12
loworder_sysid ramp eval	18/18
actuator_map_stress ramp eval	6/6

Table 2: Held-out evaluation coverage.

nonlinear actuation is exactly the setting where a simple gain correction should be insufficient: saturation, deadzones, and nonlinear motor response change the geometry of the action-effect distribution. The RG/action-effect features were designed for that case, and the best nonlinear rows support that motivation.

The second positive result is mismatch and ramp mismatch, especially on Walker2d. Smooth ramps are learnable when the train and deployment perturbations are aligned. The main negative result is abrupt held-out switches: the residual must detect a sudden hidden-mode change that was not present in the same form during training.

6.2 Nonlinear actuator drift

Nonlinear actuator drift is the cleanest evidence that structured action-effect features are useful beyond simple gain/bias correction. On Hopper, multiple independent feature families improve the frozen expert under fixed nonlinear perturbations: full RG tensor, temporal RG tensor, RG plus actuator-map history, phase/action RG, low-order actuator-map controls, and coverage-expanded RG. Ant also shows large positive nonlinear means for compact action-RG features, although with high variance. Walker2d nonlinear is weaker, which is consistent with Walker2d being more sensitive to destabilizing residual corrections in our runs.

The strongest statistically clean nonlinear row is the Hopper RG-coverage result, $+46.2 \pm 6.4$ with CI $[+30.3, +62.1]$. The larger Hopper full-tensor and sequence rows have wider intervals but consistent positive means. This pattern is useful: the effect is not one isolated run. Nonlinear drift keeps showing up as a setting where action-effect observability helps.

6.3 Fixed perturbations

Under fixed perturbations, the residual has an entire episode to infer a stable hidden actuator mode. Beyond the nonlinear successes in Table 3, the strongest fixed improvements appear for mismatch-like actuator changes in Walker2d. The full action-effect RG tensor improves Walker2d mismatch by +335.9 return

Environment / suite	Feature	Eval	n	Mean diff ± SD
Ant / helpful_combo	action_rg	nonlinear	2	$+414.6 \pm 285.9$
Ant / rg_depths	action_rg, depth 2	nonlinear	3	$+400.4 \pm 203.7$
Ant / rg_depths	phase_action_rg, depth 3	nonlinear	3	$+386.4 \pm 796.0$
Ant / rg_depths	action_rg, depth 3	nonlinear	3	$+367.0 \pm 489.5$
Hopper / mixed	history	nonlinear	3	$+135.8 \pm 174.1$
Hopper / safe_mix	history, stress-heavy	nonlinear	3	$+123.4 \pm 130.9$
Hopper / actuator_map_stress	RG tensor + actmap	nonlinear	3	$+111.3 \pm 151.0$
Hopper / rg_sequence_bilstm	action_rg2d_seq	nonlinear	3	$+103.7 \pm 77.6$
Hopper / rg_tensor_cnn	phase_action_rg2d_full	nonlinear	3	$+91.1 \pm 79.2$
Hopper / helpful_combo	phase_action_rg	nonlinear	2	$+73.3 \pm 56.9$
Hopper / loworder_sysid	phase_actuator_map	nonlinear	3	$+63.2 \pm 61.6$
Hopper / rg_coverage	phase_action_rg, keep 12	nonlinear	3	$+46.2 \pm 6.4$

Table 3: Nonlinear fixed-perturbation successes. These rows are important because nonlinear drift is where action-effect geometry should matter most. Hopper shows repeated positive gains across independent representation families; Ant has large positive means but wider uncertainty.

Environment / perturbation	Suite	Feature	Mean diff ± SD
Hopper nonlinear	rg_tensor_cnn	phase_action_rg2d_full	$+91.1 \pm 79.2$
Hopper nonlinear	rg_sequence_bilstm	action_rg2d_seq	$+103.7 \pm 77.6$
Hopper nonlinear	actuator_map_stress	phase_action_rg2d_full_actmap	$+111.3 \pm 151.0$
Hopper mismatch	helpful_combo	phase_rg	$+216.8 \pm 309.9$
Walker2d mismatch	rg_tensor_cnn	action_rg2d_full	$+335.9 \pm 214.7$
Walker2d mismatch	loworder_sysid	phase_action_moments	$+388.9 \pm 258.1$
Walker2d mismatch	actuator_map_stress	phase_action_rg2d_full_actmap	$+408.3 \pm 729.9$

Table 4: Representative fixed-perturbation improvements. Positive values mean the residual improved the frozen expert. Most rows have $n = 3$, so wide confidence intervals should be read as uncertainty, not hidden significance.

points on average. Low-order system-identification features also perform well, reaching +388.9, and RG plus actuator-map features reach +408.3. Hopper mismatch also improves with phase-RG at +216.8.

These results support the action-effect observability hypothesis. The residual is most useful when the perturbation changes the command-to-motion map in a way that can be inferred from recent action-effect history. Nonlinear and mismatch perturbations create this signal. The full RG tensor is especially relevant for nonlinear drift because it preserves pairwise action-effect geometry instead of compressing the window into only low-order moments. However, compound stress remains difficult. Hopper stress shows small gains, such as history SAC at $+11.4 \pm 16.5$ and RG/actuator-map variants around +6 to +10, but Walker2d stress is near zero or negative. Compound stress is not solved by simply adding a residual.

6.4 Held-out ramp generalization

Ramps evaluate gradual actuator drift. This is harder than a fixed perturbation but easier than an abrupt switch because the hidden mode changes smoothly. The clearest result in the project is Walker2d ramp mismatch. The full action-effect RG tensor achieves

$$+1142.2 \pm 24.5$$

return points, with a 95% confidence interval of approximately $[+1081.3, +1203.1]$. The flat action-RG feature also performs well, reaching $+564.9 \pm 110.0$, with a positive interval $[+291.6, +838.2]$.

These ramp results are important because they are not just memorization of a single fixed perturbation. They suggest that when deployment changes smoothly and remains within the support of the training mixture, the residual can track the actuator change. This is the regime where domain randomization should help. I did not run a full domain-randomization study, so the claim is narrower: support-matched smooth drift is learnable in several cases.

Environment / perturbation	Suite	Feature	Mean diff ± SD
Walker2d ramp mismatch	rg_tensor_cnn	action_rg2d_full	$+1142.2 \pm 24.5$
Walker2d ramp mismatch	helpful_combo	action_rg	$+564.9 \pm 110.0$
Walker2d ramp mismatch	rg_depths	action_rg	$+414.9 \pm 742.3$
Walker2d ramp mismatch	loworder_sysid	phase_action_moments_actmap	$+692.9 \pm 746.7$
Hopper ramp mismatch	helpful_combo	phase_rg	$+305.2 \pm 446.1$
Hopper ramp stress	rg_sequence_bilstm	action_rg2d_seq	$+32.8 \pm 12.8$
Walker2d ramp stress	rg_depths	action_rg	$+29.9 \pm 32.1$

Table 5: Representative held-out ramp results. Ramp mismatch, especially on Walker2d, is the strongest evidence that action-effect representations can track smooth actuator drift.

Environment / perturbation	Train / feature	Mean diff ± SD	Interpretation
Hopper switch mismatch	safe_mix history	$+126.8 \pm 432.0$	high variance
Hopper switch stress	safe_mix history	-16.8 ± 47.4	near zero / weakly negative
Hopper switch nonlinear	safe_mix history	-73.6 ± 111.6	negative
Hopper switch delay	safe_mix history	-82.3 ± 39.1	negative
Walker2d switch nonlinear	safe_mix history	-79.9 ± 92.4	negative
Ant switch stress	safe_mix history	-344.4 ± 263.9	strongly negative
Ant switch delay	safe_mix history	-346.8 ± 232.0	strongly negative

Table 6: Representative held-out switch results. Abrupt switches are mostly negative, even when fixed or ramped versions of related perturbations can be helped.

6.5 Held-out switch generalization

Held-out switches are the hardest evaluation regime. The hidden actuator mode changes abruptly, and that temporal structure was not present in the same form during training. The least-bad result is Hopper switch mismatch with safe-mix history, $+126.8 \pm 432.0$, but the interval crosses zero widely. Several switch settings are clearly harmful: Hopper switch delay gives -82.3 ± 39.1, and Ant switch stress gives -344.4 ± 263.9.

This does not mean the residual cannot handle these perturbation types when they are included in training. It means sudden hidden-mode changes are a distinct temporal generalization problem. The policy must detect that the actuator changed and correct quickly enough to avoid trajectory damage. A cautious residual reacts too slowly; an aggressive residual can destabilize the clean expert.

6.6 Representation ablations

The representation ablations show that observability matters. Raw and history features are reasonable baselines, but they do not always expose the right structure. Phase helps because locomotion is cyclic, but phase alone cannot identify the actuator map. Action-effect features are more targeted because they directly summarize how commands become motion.

Training mixture ablation. The mixed and safe-mix suites tested whether the train perturbation distribution controls generalization. The main pattern is that train/deployment match matters a lot. Mixed training produced very large Walker2d mismatch gains, but with high variance. Safe-mix helped in some mismatch and nonlinear settings while also training the residual not to damage clean behavior.

Phase/RG/action feature combinations. The early representation sweep under stress was not enough by itself: phase helped Ant stress, RG helped Hopper stress slightly, and Walker2d stress remained hard. The more useful sweep was ‘helpful_combo‘, which tested phase, RG, action-RG, and combinations under the safer training setup. Action-RG was strong on Walker2d ramp mismatch, while phase-RG helped Hopper mismatch.

Suite / setting	Train	Feature	n	Mean diff ± SD
mixed / Walker2d mismatch	mixed	history	3	$+1740.9 \pm 1109.5$
safe_mix / Walker2d mismatch	safe_mix	history	1	$+911.2 \pm 0.0$
safe_mix / Walker2d mismatch	stress-heavy	history	3	$+632.8 \pm 2443.8$
mixed / Hopper nonlinear	mixed	history	3	$+135.8 \pm 174.1$
safe_mix / Hopper nonlinear	stress-heavy	history	3	$+123.4 \pm 130.9$
safe_mix / Hopper mismatch	safe_mix	history	3	$+105.4 \pm 155.4$
final / Ant stress	stress	raw	9	$+78.5 \pm 105.9$
final / Ant stress	stress	history	9	$+62.8 \pm 110.4$
final / Hopper stress	stress	history	9	$+11.4 \pm 16.5$

Table 7: Training-mixture ablation. Large positive means appear when the evaluation perturbation is close to the training mixture, but high variance prevents overclaiming.

Suite / setting	Feature	Eval	n	Mean diff ± SD
repr / Ant	phase	stress	3	$+98.3 \pm 108.3$
repr / Hopper	rg	stress	3	$+7.7 \pm 13.2$
helpful_combo / Ant	action_rg	nonlinear	2	$+414.6 \pm 285.9$
helpful_combo / Hopper	phase_rg	mismatch	3	$+216.8 \pm 309.9$
helpful_combo / Walker2d	phase_rg	mismatch	3	$+148.1 \pm 297.9$
helpful_combo / Walker2d	action_rg	ramp mismatch	3	$+564.9 \pm 110.0$
helpful_combo / Walker2d	phase_action_rg	ramp mismatch	3	$+417.7 \pm 423.6$
helpful_combo / Hopper	phase_rg	ramp mismatch	3	$+305.2 \pm 446.1$
helpful_combo / Hopper	rg	ramp stress	3	$+31.9 \pm 8.9$

Table 8: Feature-combination ablation. Action-conditioned RG features are the strongest compact representation in ramp mismatch, while phase/RG combinations help some fixed mismatch and nonlinear cases.

RG depth and coverage. The RG-depth runs asked whether more contraction depths help. The answer is mixed: useful rows appear at depths 1–4, but more depth is not uniformly better. The coverage sweep then tested whether compact RG kept too few channel pairs. On Hopper nonlinear, increasing coverage produced a clean positive phase-action RG result.

Full tensor, temporal tensor, and low-order controls. The full pair-depth tensor and the system-identification controls are the most important ablations. The full RG tensor gives the cleanest positive result on Walker2d ramp mismatch: $+1142.2 \pm 24.5$, with 95% CI $[+1081.3, +1203.1]$. The temporal BiLSTM tensor has an even larger mean on that setting, but huge variance. Low-order moments and actuator-map features are competitive in mismatch, which means the RG tensor is not the only useful action-effect representation.

Taken together, the ablations support a careful claim: RG does not universally dominate, but structured action-effect observability is the important ingredient. Low-order actuator-map features are often enough for simple mismatch. The full RG tensor is most compelling for nonlinear drift and smooth ramp mismatch, where preserving pairwise action-effect geometry across depths gives the policy a richer hidden-mode signal.

7 Discussion and Limitations

When residual adaptation works. Residual adaptation works best when the deployment perturbation is fixed or changes smoothly. In these cases, recent action-effect history contains usable evidence about the hidden actuator map. The nonlinear Hopper results and mismatch/ramp Walker2d results support this interpretation. The strongest result, Walker2d ramp mismatch with the full RG tensor, suggests that structured action-effect features can generalize beyond a single fixed perturbation when the drift is smooth.

Why switches are hard. Switches are a different temporal problem. The residual is not merely asked to adapt to a hidden actuator mode; it is asked to detect a sudden change and respond before the trajectory is damaged. This creates a tradeoff: a cautious residual preserves the expert on clean or mild perturbations but may react too slowly, while an aggressive residual may respond quickly but damage trajectories where the expert was still adequate. The switch failures should be read as out-of-distribution temporal generalization failures, not as evidence that the residual cannot handle the perturbation when trained on it.

Suite / setting	Feature / config	Eval	n	Mean diff ± SD
rg_depths / Ant	action_rg, depth 2	clean	3	$+472.0 \pm 136.8$
rg_depths / Ant	action_rg, depth 2	nonlinear	3	$+400.4 \pm 203.7$
rg_depths / Ant	action_rg, depth 3	nonlinear	3	$+367.0 \pm 489.5$
rg_depths / Walker2d	action_rg, depth 4	mismatch	3	$+286.1 \pm 1717.0$
rg_depths / Walker2d	action_rg, pooled	ramp mismatch	12	$+414.9 \pm 742.3$
rg_depths / Walker2d	action_rg, pooled	ramp stress	12	$+29.9 \pm 32.1$
rg_coverage / Hopper	phase_action_rg, keep 12	nonlinear	3	$+46.2 \pm 6.4$
rg_coverage / Hopper	action_rg, keep 16	nonlinear	3	$+33.6 \pm 52.3$
rg_coverage / Hopper	action_rg, keep 12	nonlinear	3	$+31.1 \pm 70.4$

Table 9: RG depth and coverage ablations. Depth is useful but not monotonic; preserving enough channel-pair coverage matters, especially for Hopper nonlinear.

Suite / setting	Feature	Eval	n	Mean diff ± SD
rg_tensor_cnn / Walker2d	action_rg2d_full	ramp mismatch	3	$+1142.2 \pm 24.5$
rg_sequence_bilstm / Walker2d	action_rg2d_seq	ramp mismatch	3	$+1317.0 \pm 1935.2$
rg_tensor_cnn / Walker2d	phase_action_rg2d_full	ramp mismatch	3	$+445.8 \pm 303.0$
rg_tensor_cnn / Walker2d	action_rg2d_full	mismatch	3	$+335.9 \pm 214.7$
loworder_sysid / Walker2d	phase_action_moments	mismatch	3	$+388.9 \pm 258.1$
loworder_sysid / Walker2d	moments + actmap	ramp mismatch	3	$+692.9 \pm 746.7$
loworder_sysid / Walker2d	moments + actmap	ramp stress	3	$+32.3 \pm 8.8$
actuator_map_stress / Walker2d	RG tensor + actmap	mismatch	3	$+408.3 \pm 729.9$
actuator_map_stress / Hopper	RG tensor + actmap	nonlinear	3	$+111.3 \pm 151.0$

Table 10: Full tensor and system-identification controls. The full RG tensor is the cleanest ramp-mismatch result; low-order controls show that simple actuator-map structure explains part of the mismatch gains.

What the RG tensor adds. The RG tensor preserves pairwise action-effect geometry across coarse-graining depths. Low-order features can capture simple gain or bias mismatch, explaining why those baselines are competitive in mismatch settings. Nonlinear actuator drift can change higher-order and pairwise structure in the action-effect distribution, which is where the RG tensor is most justified. The key result is that structured action-effect observability matters; RG is one useful way to provide it.

Limitations. Many experiment groups have small sample sizes, often $n = 3$. I report standard deviations, standard errors, and confidence intervals in the exported tables, but several intervals are wide. The largest and most consistent result is Walker2d ramp mismatch with the full RG tensor. Other positive results should be treated as suggestive rather than definitive. The experiments are also simulation-only. Real actuators may add thermal drift, backlash, contact-dependent friction, unmodeled compliance, sensor delay, and state-estimation error. Finally, I did not run exhaustive domain randomization over perturbation ranges or curricula, and compound stress remains unsolved.

8 Conclusion

I studied residual RL for locomotion under hidden actuator drift. A frozen clean expert provides the nominal action, while a learned SAC residual adds bounded corrections. Because the actuator mode is not directly observed, the residual-control problem is a POMDP: the policy must infer whether the actuator is clean, delayed, mismatched, nonlinear, or stressed from recent action-effect evidence.

The main finding is that representation matters. Residual policies can improve the frozen expert when the perturbation is fixed or changes smoothly and when the residual observes features that expose the command-to-motion relationship. The nonlinear results are the best evidence for the RG/action-effect idea: nonlinear actuator drift repeatedly benefits from structured features, especially on Hopper, where full RG tensor variants, temporal RG, RG plus actuator-map history, and coverage-expanded RG all improve the frozen expert. The full pair-depth RG tensor also performs especially strongly on Walker2d ramp mismatch, improving the frozen expert by over +1100 MuJoCo return points with low variance.

The negative result is also clear. Abrupt held-out switches are much harder than fixed perturbations or ramps. These switches test out-of-distribution temporal generalization: the policy must detect a sudden hidden-mode change and respond quickly without destabilizing the expert trajectory. Future work should

combine structured action-effect representations with explicit change-point detection, recurrent belief-state inference, or training distributions that include abrupt mode changes.

AI Tools Disclosure

I used AI tools, including ChatGPT/Codex, as coding and writing assistants during this project. AI assistance was used extensively for boilerplate engineering, experiment orchestration scripts, SLURM command generation and monitoring utilities, result aggregation scripts, table formatting, and drafting/editing support. The core research formulation, including the hidden-actuator residual-control problem as a POMDP, the RG/action-effect representation idea, the implementation of said feature and method code, the residual-control method code, experiment design decisions, interpretation of results, and final scientific claims were developed and verified by the project author. AI-generated code and text were reviewed, modified, and integrated by the author before inclusion.

References

[1] Ludwig Baringhaus and Norbert Henze. A consistent test for multivariate normality based on the empirical characteristic function. *Metrika*, 35:339–348, 1988. doi: 10.1007/BF02613322. URL https://doi.org/10.1007/BF02613322.

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

[3] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1329–1338. PMLR, 2016. URL https://arxiv.org/abs/1604.06778.

[4] T. W. Epps and Lawrence B. Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983. doi: 10.1093/biomet/70.3.723. URL https://doi.org/10.1093/biomet/70.3.723.

[5] Vlad Firoiu, Tina Ju, and Josh Tenenbaum. At human speed: Deep reinforcement learning with action delay, 2018. URL https://arxiv.org/abs/1810.07286.

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 2018. URL https://arxiv.org/abs/1801.01290.

[7] Jad Hamdan. A fixed-point approach to non-commutative central limit theorems, 2023. URL https://arxiv.org/abs/2305.06960.

[8] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable MDPs, 2015. URL https://arxiv.org/abs/1507.06527.

[9] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11694. URL https://arxiv.org/abs/1709.06560.

[10] Norbert Henze and Bernd Zirkler. A class of invariant consistent tests for multivariate normality. *Communications in Statistics - Theory and Methods*, 19(10):3595–3617, 1990. doi: 10.1080/03610929008830400. URL https://doi.org/10.1080/03610929008830400.

[11] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics*, 36(4):1–13, 2017. doi: 10.1145/3072959.3073663. URL https://doi.org/10.1145/3072959.3073663.

[12] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26): eaau5872, 2019. doi: 10.1126/scirobotics.aau5872. URL https://arxiv.org/abs/1901.08652.

[13] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6023–6029, 2019. doi: 10.1109/ICRA.2019.8794127. URL https://arxiv.org/abs/1812.03201.

[14] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021. URL https://arxiv.org/abs/2107.04034.

[15] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020. doi: 10.1126/scirobotics.abc5986. URL https://arxiv.org/abs/2010.11251.

[16] Pankaj Mehta and David J. Schwab. An exact mapping between the variational renormalization group and deep learning, 2014. URL https://arxiv.org/abs/1410.3831.

[17] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1803.11347.

[18] Sebastien Ott. A note on the renormalization group approach to the central limit theorem, 2023. URL https://arxiv.org/abs/2303.13905.

[19] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization, 2017. URL https://arxiv.org/abs/1710.06537.

[20] Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel van de Panne. DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics*, 36(4), 2017. doi: 10.1145/3072959.3073602. URL https://www.cs.ubc.ca/ van/papers/2017-TOG-deepLoco/index.html.

[21] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 2020. doi: 10.15607/RSS.2020.XVI.064. URL https://arxiv.org/abs/2004.00784.

[22] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 91–100. PMLR, 2022. URL https://arxiv.org/abs/2109.11978.

[23] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.010. URL https://arxiv.org/abs/1804.10332.

[24] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109. URL https://homes.cs.washington.edu/ todorov/papers/TodorovIROS12.pdf.

[25] Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems*, 2017. URL https://arxiv.org/abs/1702.02453.