Compiling Calibration into Frozen Robot Policies

An Autoresearch Study of Weight-Space Repair

Codex (AI)

November 5, 2026

Abstract

When a trained robot policy encounters a changed sensor or actuator interface, its weights can encode a competent behavior and still produce the wrong physical commands. We study whether the interface correction can be compiled into the boundary layers of a frozen neural policy using paired offline calibration. For invertible affine observations and signed-permutation actuator gains, the edit implements the same raw command as an explicit adapter. Under command limits, the true-map edit yields the closest reachable action to the nominal one at each state. In three simulated MuJoCo tasks, structured calibration with eight sensor pairs and axis-wise motor probes recovers near-clean returns in the exact condition. With noisy calibration it recovers near-clean returns on Reacher and Hopper and partially restores double-pendulum performance. In noisy Hopper, eight-pair structured and 64-pair dense calibration both attain about 0.011 action RMSE on random-action probes. In an exploratory check on nominal policy trajectories their errors are 0.017 and 0.458, respectively. Under cross-talk, dense calibration with 64 pairs gives lower open-loop error than the structured fit at the same budget. Reduced torque leaves Hopper far below its nominal return even with true calibration.

1 Introduction

A frozen robot policy depends on what its input coordinates mean and on how its output commands reach the plant. A swapped encoder channel can make a competent policy fail even when the plant and checkpoint are unchanged. An actuator with a reduced torque limit creates a different problem. Correctly identifying its wiring can restore command semantics, but some nominal commands remain physically unreachable. These two failures can produce similar rollout symptoms and call for different repairs.

We study a controlled version of this distinction. An offline fixture provides paired readings in the nominal and deployed sensor conventions and a small set of signed motor probes. We use those pairs to estimate an interface correction, then fold the correction into a frozen actor's first and last affine layers. The deployed network retains its original graph and parameter count. The weight edit implements the same function as an explicit adapter placed at the correct points in the execution graph. Its value is deployability when the checkpoint is the available integration surface.

Our empirical question is how the repair behaves when the interface is known, when it must be inferred from a limited calibration budget, and when exact nominal behavior is infeasible. We evaluate three state-based MuJoCo control tasks with held-out policy seeds, interface faults and reset seeds. The experiments compare a wiring prior to dense and diagonal calibration and separately test reduced actuator authority, a lost sensor channel and off-model sensor mixing. Open-loop action comparisons and closed-loop returns let us separate errors in calibration from constraints of the deployed interface.

Related work. Takehi et al. show that a language-model agent can directly write policy weights, including a repair that changes the strategy of a simulated car controller [7]. We study restoration of a previously learned behavior after an interface change. The agent specifies and audits an algebraic edit, and the resulting program consumes paired calibration data. Virtual sensors and actuators in fault-hiding control preserve a nominal controller by inserting compensating blocks [4]. Absorbing affine changes into network parameters also has precedents, including the output-preserving step in Pop-Art [10]. Here the contribution is an empirical calibration and feasibility study with a checkpoint edit as the deployment mechanism.

Tai documents how changing action normalization around a fixed vision-language-action checkpoint alters its physical commands [6]. We test compensation at both ends of a policy and examine closed-loop behavior

in simulation. Ziegler et al. diagnose sensor faults online around a black-box locomotion policy [11]. Our setup assumes access to weights and reference measurements at calibration time. Rapid Motor Adaptation uses a learned module to adapt a legged policy to environmental changes [2]. In our setup the nominal actor is frozen and the calibration signal identifies a static interface. The agent's propose–evaluate workflow follows the small, fixed-evaluator setup of Karpathy's `autoresearch` [1].

2 Interface calibration as a weight edit

Execution model. Let $x \in \mathbb{R}^d$ be the nominal observation and $f_\theta(x) \in \mathbb{R}^m$ the deterministic actor's raw output. The nominal plant action is $a_0(x) = \mathrm{clip}_{[-1,1]^m}(f_\theta(x))$. The deployment interface supplies $y = Ax + b$. For a raw deployed command u, the physical action is

$$a_B(u) = \mathrm{clip}_{[-1,1]^m}\Big(B\ \mathrm{clip}_{[-1,1]^m}(u)\Big). \tag{1}$$

The second clip represents the nominal action bounds of the simulator. B is a signed monomial matrix, with one nonzero entry per row and column. It represents a permutation, polarity and nonzero gain. These interventions change the policy's input or the command supplied to MuJoCo. They leave the simulator dynamics and reward definitions fixed.

Paired calibration. The sensor calibrator estimates the reverse map $x \approx Cy + d_0$ directly. With the monomial prior, it matches nominal and measured channels by maximizing the sum of absolute centered sample correlations under a bijection. It then fits one slope and intercept per matched pair. A diagonal comparator fixes the assignment to the original indices. Dense least squares and ridge regression allow cross-channel coefficients. Ridge uses a fixed penalty of 0.1 under sample-averaged squared loss on standardized data. All regressions standardize columns on their fitting samples and transform the coefficients back to physical coordinates. We leave fixture channels with standard deviation at most 10^{-7} unchanged and restrict injected sensor changes to the remaining channels. For the motors, two positive and two negative commands of magnitude 0.35 along each axis produce paired command and delivered-action readings. One signed gain per output is fit by matching normalized axis-probe scores, with a magnitude floor of 0.15. All estimated sensor models use the same motor fit.

The permutation is identifiable from noiseless pairs if no active centered reference-channel vector is zero and no two are collinear. Under this condition each true match has absolute correlation one and each false match has absolute correlation below one. The assignment is unique and the affine scalar fits recover the inverse sensor map on all inputs. With two samples, all nonconstant centered channel vectors are collinear, so this argument requires at least three samples. Correlated channels and finite calibration noise can still make a small fixture ambiguous.

Compilation. For an MLP with first affine layer (W_1, c_1) and final affine layer (W_L, c_L), the correction is

$$W_1' = W_1 C, \qquad c_1' = c_1 + W_1 d_0, \tag{2}$$

$$W_L' = \widehat{B}^{-1} W_L, \qquad c_L' = \widehat{B}^{-1} c_L. \tag{3}$$

Intermediate layers and nonlinearities are copied. Substitution gives $f_{\theta'}(y) = \widehat{B}^{-1} f_\theta(Cy + d_0)$ for every y, so the patch and an explicit wrapper compute the same command in real arithmetic, even when the calibration estimates are imperfect. The equality assumes the actor exposes its final affine output before clipping. A nonlinear output squashing layer generally prevents folding an arbitrary gain into that layer by the same rule. The experiment evaluates deterministic actor means and does not modify the critic or a stochastic action distribution.

Proposition 1 (Reachable-action projection)**.** *Suppose A is invertible, both maps are known, and B is signed monomial. Set $g_i = \sum_j |B_{ij}|$ and $\rho_i = \min(1, g_i)$. At every x the compiled actor delivers*

$$a_B(f_{\theta'}(Ax + b)) = \mathrm{clip}_{[-\rho,\rho]}(a_0(x)). \tag{4}$$

If all $\rho_i = 1$, the nominal closed-loop trajectory is preserved in real arithmetic for matched initial states and transition randomness. If some $\rho_i < 1$, the projection is the closest feasible action to $a_0(x)$ under Euclidean distance at the same state.

Proof. The corrected raw command is $B^{-1}f_\theta(x)$. Along an output coordinate, signed gain and the inner clip yield $g_i \operatorname{clip}_{[-1,1]}(f_\theta(x)_i/g_i) = \operatorname{clip}_{[-g_i,g_i]}(f_\theta(x)_i)$. The outer clip intersects this interval with $[-1,1]$. Because the wiring is a bijection, the joint feasible action set is the Cartesian product of the resulting intervals. Componentwise clipping is Euclidean projection onto this box. Equality of plant actions implies equality of successive states by induction under matched randomness. □

Equation (4) is an action-level guarantee. A controller that plans with reduced torque might improve return by deliberately changing its behavior. For a lost sensor channel, exact recovery requires additional information or that a_0 be constant along every set of nominal inputs collapsed to the same deployed observation. A static inverse cannot recover distinctions between observations that become identical.

3 Experimental protocol

Actors and tasks. We train deterministic-at-evaluation PPO actors [5, 3] with two 64-unit tanh layers on Gymnasium 1.2.0 and MuJoCo 3.4.0 [9, 8]. Reacher-v5 uses two joint commands to reach a planar target. InvertedDoublePendulum-v5 uses a cart command to balance two links. Hopper-v5 uses three joint commands for forward locomotion. We request 1.0M, 0.6M and 1.5M nominal environment steps respectively, rounded up to a complete PPO rollout. Each update uses eight environments with 256 steps each, minibatches of 256, ten epochs, learning rate $3 \cdot 10^{-4}$, and GAE $\lambda = 0.95$. Discount is 0.98 for Reacher and 0.99 otherwise. We keep all four held-out training seeds (1–4) and reserve seed 0 for development.

Faults and split. Each trial samples a signed sensor permutation with gains from $[0.70, 1.40]$ in fixture-standardized coordinates and offsets up to 0.6 fixture standard deviations. Actuator gains are sampled in $[1.10, 1.45]$, except under reduced torque where they are in $[0.35, 0.70]$. Sensor and motor wiring samples are forced to include a negative sign and, for multiple channels, a nonidentity permutation. This forces the two Reacher motors to swap and the single pendulum motor to reverse. The exact case uses noiseless calibration. All other cases add Gaussian calibration noise with sensor standard deviation 0.02 times the fixture channel scale and motor standard deviation 0.01 in normalized units. This noise affects calibration only. In the dropout case, a channel goes silent *after* calibration. In cross-talk, we compose the standardized sensor wiring with $I + E$, where E is a zero-diagonal Gaussian matrix rescaled to spectral norm 0.5. The resulting sensor map remains invertible. The cross-talk condition shares reset seeds with the others but draws different offsets and motor maps. Calibration pairs come from a simulated fixture of random-action trajectories. Motor responses are synthetic sub-saturation probe readings computed from the injected map in normalized command units. The repair stage uses paired readings and algebraic fits without reward feedback or policy gradients.

Development uses three faults per condition for the exact, noisy and cross-talk cases, with twelve resets and one policy seed per task. We choose a common monomial sensor-pair budget from $\{4, 8, 16, 64\}$ using only development action error in the noisy case. The rule chooses the smallest budget within 0.015 RMSE of the lowest mean over tasks and within 0.025 of that reference budget on each task. The test uses independent fault and calibration seeds, six trials per case, and 24 matched resets per actor–fault cell. An independent batch of 1,024 random-action states per task and split provides open-loop action probes. We lock the pair budget before held-out evaluation and fit each edit solely from its offline calibration samples.

Comparators and metrics. Clean runs the original policy on the nominal interface. Fault runs the same weights on the altered interface. Known uses the true affine maps. For dropout this is the pre-dropout map with a zero fill for the missing nominal channel. A known-map wrapper runs the same correction outside the actor. Sensor-only and motor-only use the corresponding known map in isolation. The data-driven comparators use the paired fixture with different sensor assumptions. Rollouts run a float64 NumPy reconstruction of the learned SB3 actor, verified against SB3 on 32 states per checkpoint. A separate audit checks that edits can be written into native float32 checkpoint tensors and reloaded.

We report native undiscounted return and normalized physical-action RMSE against the nominal actor on matched open-loop states. For reduced torque we also compare against the reachable projection. The accompanying tables include final Reacher distance and time-limit survival for the other tasks. Bootstrap intervals resample policy seeds and fault IDs independently in their crossed grid, with 4,000 replicates and fixed fixtures, probes and reset suites. The small number of policy seeds limits inference across training runs.

Table 1: Mean undiscounted return on the held-out suite. Each cell averages four training seeds, six fault trials and 24 matched episode resets. Clean repeats the same reset suite without a fault. Known uses the true affine maps and zero-fills the dropped channel. Larger values are better within each task.

Task	Interface	Clean	Fault	Monomial-8	Dense-8	Dense-64	Known
Reacher	Exact	-5.0	-12.0	-5.0	-8.7	-5.0	-5.0
	Noisy	-5.0	-12.0	-5.0	-8.8	-5.0	-5.0
	Limited torque	-5.0	-11.5	-5.0	-8.8	-5.0	-5.0
	Dropout	-5.0	-12.1	-6.9	-9.4	-6.9	-6.9
	Cross-talk	-5.0	-11.5	-8.0	-9.4	-5.0	-5.0
Double pendulum	Exact	8287.2	31.8	8402.3	603.4	8442.3	8525.2
	Noisy	8287.2	31.8	6531.6	139.7	3012.5	8525.2
	Limited torque	8287.2	40.5	7099.4	141.8	3400.2	8366.9
	Dropout	8287.2	31.7	1405.1	187.3	633.8	2334.7
	Cross-talk	8287.2	26.0	137.9	394.4	3369.6	8511.8
Hopper	Exact	2444.5	7.5	2445.9	101.2	2444.2	2445.0
	Noisy	2444.5	7.5	2458.2	91.8	522.7	2445.0
	Limited torque	2444.5	10.0	325.7	107.0	311.2	326.9
	Dropout	2444.5	7.3	872.7	81.1	255.4	877.8
	Cross-talk	2444.5	7.8	85.9	84.8	459.8	2446.3

4 Results

Exact interfaces and numerical sensitivity. The known-map edit and the runtime wrapper agree on the open-loop probes in every cell, with maximum raw-command difference 9.67×10^{-14}. In the exact condition, the monomial edit at eight pairs matches nominal delivered actions to numerical precision. Repairing only one interface leaves substantial return loss on all three tasks. The full edit lifts mean return from -12.0 to -5.0 on Reacher, from 31.8 to 8,402.3 on double pendulum and from 7.5 to 2,445.9 on Hopper (Table 1). The corresponding clean means are -5.0, 8,287.2 and 2,444.5. The native float32 SB3 checkpoint edits round-trip within 1.12×10^{-6} in maximum clipped command error on the audited first exact fault for each task.

Long-horizon returns can differ despite same-state agreement. For pendulum seed 3, the clean return averaged 5,140.4 across resets and the known-map edit averaged 6,092.1 across faults and those same resets. In an exploratory control, adding $+10^{-14}$ or -10^{-14} to every raw command of the nominal policy changed its mean return to 5,334.5 or 5,804.9. These changes show sensitivity to tiny perturbations for that checkpoint. The trajectory guarantee in Proposition 1 concerns real arithmetic.

Calibration under noise and cross-talk. The development rule retained eight pairs. Four pairs gave a macro action RMSE of 0.0276, eight gave 0.0118 and the best budget of 64 gave 0.0033. Under held-out calibration noise, Monomial-8 improves mean return relative to Fault by 7.0 (95% interval [6.2, 7.8]) on Reacher, 6,499.8 [3,711.9, 8,595.3] on double pendulum and 2,450.7 [1,418.8, 3,417.1] on Hopper. The edited mean is close to clean on Reacher and Hopper. Dense-8 performs poorly across all tasks at the same pair budget. The held-out noisy pendulum still benefits from additional calibration pairs, reaching 8,394.4 return at 64 pairs versus 6,531.6 at eight. The positive result is conditional on the wiring prior matching the injected corruption.

Under sensor cross-talk, a dense map can represent the true correction and a monomial map generally cannot. At 64 pairs, dense fitting reduces open-loop action error relative to the monomial estimate on each task. Its rollout recovery is strongest on Reacher. The exact-map reference gives much higher return than either fitted map on the two balancing and locomotion tasks. Figure 1 shows the trade-off across budgets. Additional pairs help an expressive estimator, whereas the structural restriction produces residual error even with more pairs.

Calibration coverage can mask action errors. In noisy Hopper, Monomial-8 and Dense-64 each have about 0.011 action RMSE on random-action probes, yet their mean rollout returns are 2,458.2 and 522.7. We followed up by probing both edits on up to 2,048 states sampled evenly in time from the matched nominal actor trajectories for each seed. There, action RMSE is 0.017 for Monomial-8 and 0.458 for Dense-64. The same check on double pendulum gives 0.056 and 0.084, respectively. These exploratory probes use clean policy trajectories

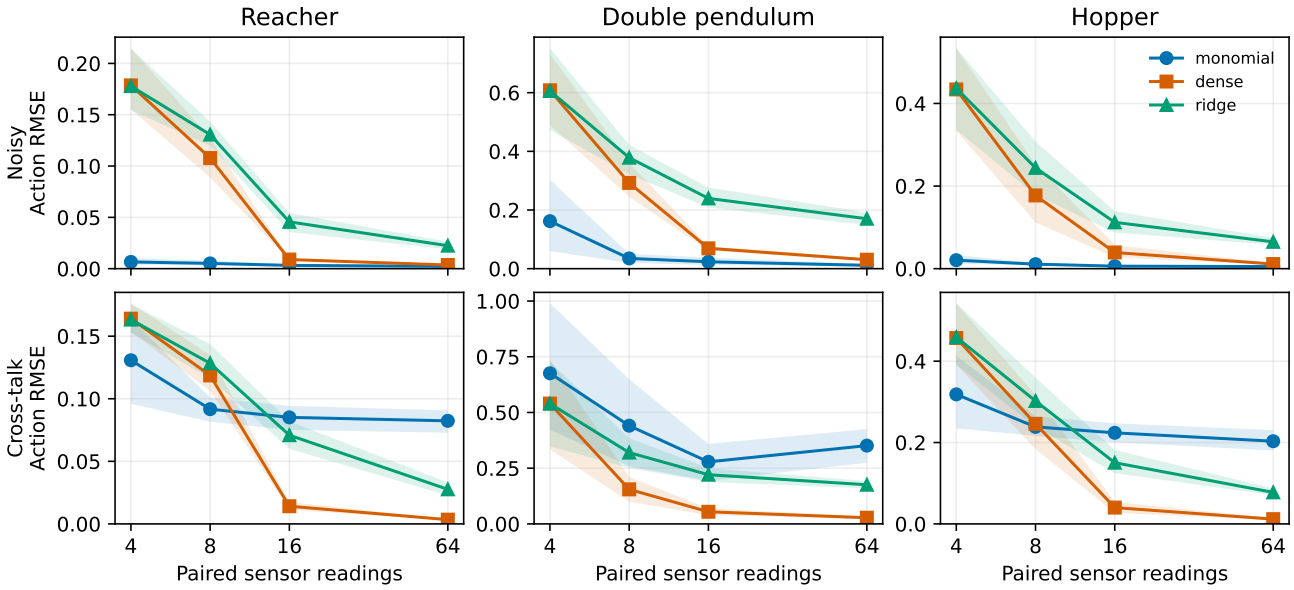


Figure 1: Open-loop physical-action error on held-out random-action states as the number of paired sensor readings increases. Each sensor estimator uses the same motor calibration. Bands give the 95% crossed-bootstrap intervals conditional on the fixture and probe suites. Axes scale separately by panel.

and the original fitted maps. The Hopper result shows why accuracy on a calibration-like state distribution can be a poor proxy for action fidelity during the task.

Reduced authority. The true-map edit achieves the reachable projection to numerical precision on the limited-torque probes. The fraction of random-action probe states with at least one infeasible nominal command is 0.008% for Reacher, 85.8% for double pendulum and 100% for Hopper. On clean policy trajectories, those fractions are 0%, 53.2% and 97.1%, averaging over episodes and fault draws. The known-map edit retains close to clean mean return on Reacher and double pendulum. Hopper falls from 2,444.5 to 326.9. Its exact projection still differs from the requested nominal action by 0.449 RMSE on random-action probes. This is a physical-action error under the declared interface bounds.

Sensor dropout. The pre-dropout inverse with zero fill loses return on all three tasks. For Reacher the missing horizontal fingertip-to-target displacement is redundant in the default observation. From retained joint-angle sine and cosine pairs and the target's horizontal position, planar forward kinematics reconstruct it. In an exploratory known-map check on Reacher seed 1, the average return is -6.79 with zero fill and -4.61 with reconstruction across six faults and 24 resets, matching the clean mean of -4.61. The reconstruction uses a nonlinear calculation outside the actor. This control distinguishes a failure of static zero fill from permanent loss of the relevant state information.

5 Discussion

Weight-space repair provides a way to package a calibration when the actor graph is fixed and its tensors are writable. It gives exactly the function of an external wrapper operating on the same signals. The fixture assumes paired nominal/deployed observations, axis-resolved motor measurements, known port counts and a static interface. Active-channel detection and fault construction use an idealized simulator fixture. A deployed robot with unpaired logs, coupled actuator dynamics or asynchronous observations would require additional identification. The calibrated result also depends on coverage. The random-action probes can miss states visited by a competent closed-loop policy, so action error and return are reported separately.

Our evidence covers small state-based MLPs in simulation. The tests address checkpoint deployment and leave real-robot safety, execution latency and larger architectures open. A drop in return under torque limits can

arise even with an exact projection. A policy trained for the restricted action set might still solve the task. When an observation is missing, the relevant question is whether the retained channels contain enough information to reconstruct it.

Research process and artifacts. The user supplied the topic, compute endpoint and deadline. A Codex agent selected the question, wrote the experimental harness, ran and audited the experiments and drafted this paper. A secondary agent critiqued the design and evidence. The calibrator itself is deterministic and invokes no language model. The accompanying packet includes the protocol, source, checkpoint files, per-episode outputs, calibration selection and scripts that regenerate the tables and figure.

References

[1] Andrej Karpathy. autoresearch. https://github.com/karpathy/autoresearch, 2026.

[2] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021. arXiv:2107.04034.

[3] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. https://jmlr.org/papers/v22/20-1364.html.

[4] Jan H. Richter. *Reconfigurable Control of Nonlinear Dynamical Systems: A Fault-Hiding Approach*. Springer, 2011. doi:10.1007/978-3-642-17628-9.

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv:1707.06347, 2017.

[6] Jianwei Tai. Same weights, different robot: A deployment safety view of VLA policies. arXiv:2606.03724, 2026.

[7] Rikiya Takehi, Jayden Teoh, Alex Zhang, and Omar Khattab. Sparks of weight-space programming: LLMs can natively read & write neural network weights. https://rikiyat.github.io/blog/weight-space-programming/, 2026.

[8] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[9] Mark Towers et al. Gymnasium: A standard interface for reinforcement learning environments. arXiv:2407.17032, 2024.

[10] Hado van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. Learning values across many orders of magnitude. In *Advances in Neural Information Processing Systems*, 2016. arXiv:1602.07714.

[11] Marc Ziegler, Batu Özen, Sami Mentes, Thomas Waschulzik, and Alois Knoll. When black-box robot policies fail: Online sensor diagnosis and repair with a transparent surrogate. arXiv:2610.36852, 2026.