ACTION SHAPING: POLICIES ABSORB WHAT THEY CAN EXPRESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward shaping has a theorem: a potential-based term can be removed without changing the optimal policy. The same practice on the action channel, an offset added in training and dropped at deployment, has no theorem. Nothing cancels an action offset, so the correction is kept at deployment or removed without a guarantee. We call it *action shaping* and state its principle. A trainable policy absorbs an offset its own output layer can reproduce exactly, which is what we mean by *express*; what is absorbed can be removed with the return intact. Its minimal instance is a zero-initialized linear head behind a learnable gate, added to an actor that trains through a learned action-value function, with no penalty or schedule. The gate rises and then falls on its own, for deterministic and stochastic actors alike, and on 20 tasks removing the head costs almost nothing. The condition is exact reproduction, not capacity: a nonlinear head with more parameters is not absorbed, and in a paired control, one linear path added to a nonlinear base head restores absorption. Exact reproduction gives the loss a flat direction that gradient noise drifts along, and the offset's amplitude indicates, before removal, what dropping the head will cost. Action shaping thus gains the counterpart of the shaping theorem, a condition for absorption, together with the mechanism behind it and a diagnostic that reads it. Policies absorb what they can express, and only that.

1 INTRODUCTION

Reinforcement learning routinely trains with structure it does not deploy: shaped rewards, privileged observations, exploration bonuses, expert corrections. For the reward channel there is a theorem that says which of these can be removed without changing the optimal policy, and it has guided reward shaping (Ng et al., 1999). A privileged critic can be removed for free by construction, since the deployed policy never computes it; the action channel has nothing comparable. An offset added to the action changes which states are visited, no telescoping sum cancels it, and a co-trained correction is kept at deployment or removed without a guarantee. We call this practice *action shaping* and ask the reward-shaping question of it: when can the offset be removed for free? We show that the answer is not an invariance but an optimization phenomenon (Figure 1).

A head added to a TD3 or SAC actor for training is *absorbed*: its scalar gate, initialized near zero, rises early and then, with no schedule, no penalty, and no weight decay, falls back toward zero, and removing the head afterwards costs almost nothing. The head is the minimal instance of action shaping, a second linear layer on the representation the actor already computes, so its absorbed state is known in closed form, an exact reference point for measuring what training does. With a one-hidden-layer network as the head, the gate does not return to zero, and removing the head at the end costs 0.44 to 0.78 of normalized return. With the linear head's gradient to the shared trunk cut, so that the representation is shaped by the base head alone, the gate returns to zero as before. Whether the head is absorbed is decided not by whether it sends gradient into the trunk but by what the base can do with its own weights.

The relevant sense of "can" is precise and central to this paper: a trainable trunk can approximate a nonlinear offset, so capacity does not separate the two heads. What separates them is that for the linear head there is a change of the base head's own weights, and of nothing else, that reproduces the offset exactly, and for the nonlinear head there is none. We call an offset *expressible* when such a change exists: a linear offset on the representation the base head reads is expressible, a nonlinear

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

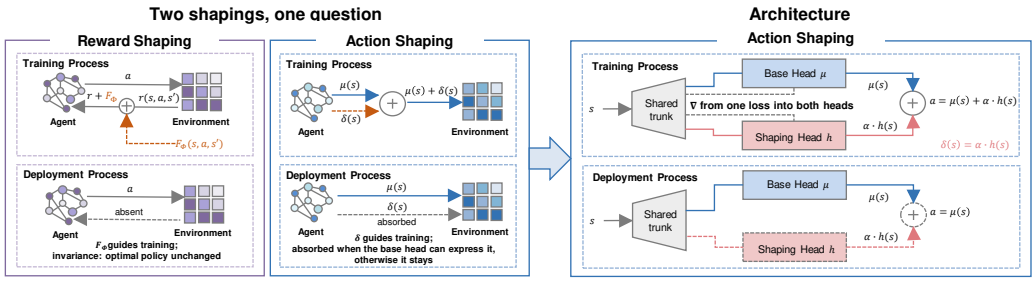


Figure 1: **A trainable policy absorbs a training-time action offset its own output layer can reproduce, and only that.** Reward shaping adds a term in training and a theorem says which terms leave the optimal policy unchanged; action shaping adds an offset $\delta = \alpha h$ to the action, and nothing cancels it. What removes it for free is absorption: when the base head μ can reproduce $\mu + \alpha h$ with its own weights, $W_\mu + \alpha W_h$, and is free to train, the gate α falls back toward zero and removing the head costs almost nothing. On a frozen base, or for a head that μ cannot reproduce, it is not absorbed. Right: the shaped actor in training and the base actor alone at deployment.

offset is not, and neither is a linear offset on a nonlinear base head. Under expressibility the shaped actor and a base-only actor with merged readout weights compute the same function, so the loss is constant along a line in parameter space that transfers the offset to the base, and we prove that, at a minimizer, the point on that line where the head contributes nothing is the flattest (§3). Gradient noise moves parameters along flat directions, and in the idealization of stochastic gradient descent with label noise we derive a closed-form prediction: the gate decays exponentially, and the head's weights decay too, but more slowly, by a factor of the head's size. The agents, trained with Adam and a learned critic, show that pattern. Without expressibility there is no line; without a trainable base there is no motion along it. Policies absorb what they can express, in this sense of the word.

The condition sorts existing practices. A training-time offset the base can express and is free to learn is absorbed; on a frozen base it stays, as in residual policy learning; one the base cannot express stays too, or collapses under strong noise, or is removed on a schedule, as in decaying action priors and attenuated residual policies. The offset's amplitude falls as the head is absorbed, so the training log alone tells what removal will cost. Across 20 environments, those whose amplitude ends below a tenth of its peak lose nothing in interquartile mean when the head is removed, and the others lose 0.03 of normalized return; within one run, removal is costly around the peak and small at each end.

We make three contributions. **Absorption requires expressibility and a trainable base.** A zero-initialized, scalar-gated linear head co-trained with a TD3 or SAC actor engages and is then absorbed; seven arms on a six-task suite show that absorption requires an expressible offset and a trainable base, does not require gradient flow from the head into the trunk, and, in a paired control, returns on a nonlinear base head as soon as one linear path makes the offset reproducible again (§4.4). **Expressibility creates a flat line.** A flat-line proposition shows that an expressible offset gives the loss a flat direction that leads to a closed gate. Under label-noise stochastic gradient descent the drift along it closes the gate exponentially while the head decays more slowly, by a factor of its size, as a minimal model confirms (§3, Appendix B). **The amplitude predicts the removal cost.** Its terminal residual does so across tasks and its trajectory within a run, without a task-specific threshold (§2.2, §4.3). A first-order improvement theorem also explains what moves the gate, which a state-value critic supplies only by sampling (§3.1). Anyone who drops at inference a training-time head or correction branch that acts on the output assumes that the base has absorbed it; this paper gives a condition under which that holds on the action channel, and a way to check it.

2 Action Shaping and the Amplitude Diagnostic

2.1 The Shaped Actor

Let $z(s) = \phi(s) \in \mathbb{R}^k$ be the representation computed by the actor's trunk (two hidden layers in continuous control), and let the base head read an action from it, $\mu(s) = W_\mu z(s) + b_\mu \in \mathbb{R}^m$. Our

instance of action shaping adds a second head on the same representation, $h(s) = W_h z(s) + b_h$, and a single learnable scalar *gate* α, and acts with

$$\pi_{\text{shaped}}(s) \;=\; f\big(\mu(s) + \alpha\, h(s)\big), \qquad \delta(s) := \alpha\, h(s), \tag{1}$$

where f is the host algorithm's output map (a $\tanh$ scaled to the action bounds for TD3, the squashed Gaussian mean for SAC, a softmax over logits for discrete actions). We call δ the *offset* and $\|\delta\|$ its *amplitude*. The *shaping head* is zero-initialized ($W_h = 0$, $b_h = 0$) and $\alpha \sim U(-\varepsilon, \varepsilon)$ with $\varepsilon = 0.01$, so the shaped actor coincides with the base actor at the start of training, while $\alpha \neq 0$ keeps the head's gradient nonzero; ε is the only added constant (sensitivity in Appendix L).

Training changes nothing in the host algorithm except the action it differentiates through: the actor loss of TD3, SAC, or PPO is evaluated at the shaped action, and its gradient updates ϕ, W_μ, W_h, and α jointly with the same optimizer. There is no penalty on α or δ, no decay schedule, and no weight decay: the amplitude evolves under the actor loss alone. At deployment the shaping head is dropped and the agent acts with $f(\mu(s))$ alone; the head adds one linear layer, under six thousand parameters in every continuous-control environment.

2.2 What we measure

We log two quantities throughout training: the amplitude $\|\delta_t\|$, the mean of $\|\delta(s)\|$ over a fixed batch of evaluation states at step t, and the *removal cost*, the return of the shaped actor minus the return of the base actor alone, both returns evaluated every 5K steps on the same environment seeds and normalized per environment (Appendix H).

The *terminal residual* summarizes an amplitude trajectory in one number,

$$r \;=\; \frac{\text{mean}_{t \in \text{last 20 evaluations}} \|\delta_t\|}{\max_t \|\delta_t\|}, \tag{2}$$

the fraction of its own peak that the offset retains at the end of training; r is a ratio of pre-activation amplitudes in continuous control; Appendix E gives the action-space version, the reading to use where the output map saturates. The residual r is computed per run, and an environment's residual is the mean of r over its seeds. A run is *non-engaged* when the peak amplitude never exceeds 10^{-3}, so that the offset never takes part in training; it is *completed* when $r \leq 0.10$, and *incomplete* otherwise. The threshold 0.10 is a fixed constant; the rank correlations of §4.3 do not use it and stay between 0.565 and 0.671 ($p < 0.01$) under all four residual conventions of Appendix H.

Reading the gate. The three states give a protocol for any training-time branch behind a learnable scalar: if the scalar returned to zero, the base alone computes what the shaped actor computed; if it rose and stayed, the branch still contributes and dropping it can cost return; if it never rose, the branch never took part. The gate and the offset it scales, not the norm of the head, whose weights decay far more slowly (§3.2), are the quantities to read.

2.3 Seven arms

We train the same actor in seven experimental arms that differ only in what the shaping head reads, what the base is allowed to learn, and what either head can compute.

- **Shared** (default): h reads $z(s)$; every parameter trains.
- **Detach**: h reads $\text{sg}(z(s))$; the forward pass is identical to Shared, but no gradient flows from the shaping head into the trunk.
- **Independent**: h reads its own trainable trunk $\phi'(s)$ of the same size; the two heads share no representation.
- **Frozen**: ϕ and W_μ are frozen at initialization; only W_h, b_h, and α train.
- **MLP head**: h is a one-hidden-layer network on $z(s)$ with 16, 64, or 256 hidden units and a zero-initialized output layer; everything else is as in Shared.
- **MLP base head**: μ is a one-hidden-layer network on $z(s)$ with 64 hidden units and h stays linear, so the offset is linear and the base head is not; a second version adds a zero-initialized linear map of $z(s)$ to the base head's output, which puts $\mu + \alpha h$ back inside the base head's own function class.

The first four arms vary how the base can change and whether gradients cross between the heads; the fifth varies what the offset can be and the last two what the base head can be. We say the offset is *expressible* when it lies in the function class of the base head on the shared representation, that is, when there exist base-head parameters that reproduce $\mu + \alpha h$ exactly for the current trunk. A linear head on a linear base head is expressible by construction, $\mu(s) + \alpha h(s) = (W_\mu + \alpha W_h)\, z(s) + (b_\mu + \alpha b_h)$; an MLP head is not, since no base-head parameters reproduce a nonlinear function of z, and neither is a linear head on an MLP base head, whose function class is not closed under adding an affine map, unless that base head carries a linear path of its own. Expressibility is relative to the base head: the trunk could in principle be retrained to make any offset linear in a new representation, and §4.4 reports whether training does so. Expressibility does not require the offset to be additive.

For the linear head the absorbed state is known in closed form, the base-only actor with weights $W_\mu + \alpha W_h$, and the offset δ is exactly that state's base head minus the current one; this reference point is what makes the measurement exact. The question is whether training reaches it by itself, so that the head can be dropped. The linear head is a probe of whether training absorbs a branch, and the arms of §4.4 show that training absorbs exactly the expressible ones on a trainable base.

3 Why the offset helps and why it should decay

3.1 First-order improvement and the critic

Let $G(\alpha)$ be the return of the shaped policy π_α as a function of α, with $\pi_0 = \pi_\mu$, and define the per-state functional

$$\Phi_h(s) \;:=\; \frac{d}{d\alpha}\, \mathbb{E}_{a\sim\pi_\alpha(\cdot|s)}\big[Q^{\pi_\mu}(s,a)\big]\Big|_{\alpha=0}, \tag{3}$$

which equals $\nabla_a Q^{\pi_\mu}(s, f(\mu))^\top J_f(\mu)\, h(s)$ for a deterministic continuous policy, J_f the Jacobian of the output map, and $\mathrm{Cov}_{\pi_\mu(\cdot|s)}\big(h(s), Q^{\pi_\mu}(s,\cdot)\big)$ for a softmax policy over discrete actions (Appendix A).

Theorem 1 (First-order improvement)**.** *Under the regularity conditions of Appendix A,* $G(\alpha) = G(0) + \frac{\alpha}{1-\gamma}\, \mathbb{E}_{s\sim d^{\pi_\mu}}[\Phi_h(s)] + O(\alpha^2)$.

The offset helps to first order exactly when it is aligned, on average, with the critic's action gradient; for a deterministic policy the first-order term is the deterministic policy gradient (Silver et al., 2014) taken with respect to α at $\alpha = 0$. Two consequences follow. First, the gradient of the actor loss with respect to α is the same inner product at the shaped action: a Q-critic supplies it as a pathwise gradient, whereas under PPO the actor loss carries it only through a score-function estimate over sampled actions (§4.1). Second, the first-order term vanishes as the base policy approaches optimality (Appendix A.1), so the gradient that raised α fades as the base improves; this explains why the amplitude stops rising, and §3.2 why it decreases.

3.2 Expressibility creates a flat line

Write the actor's parameters as $\theta = (\phi, W_\mu, b_\mu, W_h, b_h, \alpha)$ and let $\mathcal{L}(\theta)$ be the actor loss of the host algorithm evaluated at the shaped action. For a linear head, the shaped pre-activation is $(W_\mu + \alpha W_h)z + (b_\mu + \alpha b_h)$, so the loss depends on the head parameters only through the merged readout $\bar{W} = W_\mu + \alpha W_h$ and $\bar{b} = b_\mu + \alpha b_h$.

Proposition 2 (Flat line; proof in Appendix A.2)**.** *Let the head be linear. For every* θ *and every* $\lambda \in \mathbb{R}$*, the parameter* $\theta_\lambda = (\phi,\, W_\mu - \lambda W_h,\, b_\mu - \lambda b_h,\, W_h,\, b_h,\, \alpha + \lambda)$ *satisfies* $\mathcal{L}(\theta_\lambda) = \mathcal{L}(\theta)$*. Along this line the trace of the Hessian of* $\mathcal{L}$ *with respect to* (W_h, b_h) *equals* $(\alpha+\lambda)^2$ *times a quantity independent of* λ*, and at a minimizer of* $\mathcal{L}$ *it is minimized at* $\alpha + \lambda = 0$*.*

The line of Proposition 2 is the set of ways to split one and the same shaped actor between its base head and its shaping head. At $\alpha = 0$ the base head alone computes the shaped actor's output and the shaping head contributes nothing; at a minimizer, this split is the flattest. The loss is constant along the line, so a noiseless gradient flow does not move along it; what moves a parameter along a flat direction is gradient noise. Near a manifold of minimizers, stochastic gradient descent with label noise drifts along the manifold as a gradient flow of the trace of the Hessian (Blanc et al., 2020;

Damian et al., 2021; Li et al., 2022; Ziyin, 2024), the drift that in other settings carries it to sparse solutions (Pesme et al., 2021) and simpler invariant subnetworks (Kunin et al., 2021; Chen et al., 2023); on the line of Proposition 2 we obtain that trace in closed form.

Proposition 3 (Drift along the line)**.** *Let the trunk be fixed, absorb the biases into z as a constant feature, let $\Sigma = \mathbb{E}[zz^\top]$, nonsingular, and $S = \operatorname{tr}\Sigma$, and let the loss be a squared error in an m-dimensional action. On the zero-loss manifold the trace of the Gauss–Newton Hessian depends on the head parameters only through $mS(1+\alpha^2) + \operatorname{tr}(W_h \Sigma W_h^\top)$. In the small-step limit of stochastic gradient descent with label noise, whose dynamics on the manifold is the projected gradient flow of that trace (Li et al., 2022, Corollary 5.2), the gate and the head evolve, to leading order in α, as*

$$\frac{d\alpha}{dt} = -\,c\,\frac{mS - \operatorname{tr}(W_h \Sigma W_h^\top)}{1 + \|W_h\|_F^2}\,\alpha, \qquad \frac{dW_h}{dt} = -\,c\,W_h \Sigma, \tag{4}$$

with one positive constant c set by the step size and the noise level; the gate decays exponentially whenever $\operatorname{tr}(W_h \Sigma W_h^\top) < mS$, which holds for every head with $\|W_h\|_2^2 < m$, and in every case the gate and the head both tend to zero.

The proof (Appendix A.3) evaluates the trace on the line and projects its gradient onto the tangent space; the denominator is the squared length of the tangent vector along the line, and the head's output power $\operatorname{tr}(W_h \Sigma W_h^\top)$ enters the numerator because that vector is not orthogonal to the head directions. The gate returns to zero in every case, so an expressible offset does not merely stop rising. The head decays per direction at the rate of one eigenvalue of Σ, whereas the gate decays at the rate of m times their sum minus the head's output power, divided by $1 + \|W_h\|_F^2$, so for a small head the ratio of the two rates is of the order of the number of weights in the head, mk. On the time scale on which the gate closes, the head is therefore nearly constant, and it decays only slowly afterwards. This is the two-process picture the data show: the gate reads functional absorption, and the head's remaining norm is not evidence that absorption is incomplete. Appendix B checks the proposition in its own idealization, stochastic gradient descent with label noise started on the line; every prediction of the proposition holds there: the gate decays exponentially, within a factor of 1.6 of the flow (the excess being the finite-step correction) and in proportion to the noise variance; the head decays only after the gate has closed and more than a hundred times slower; and a linear head on a one-hidden-layer base, which that base can approximate but not reproduce, does not decay at all. The agents, trained with Adam and a bootstrapped critic, show its pattern: the gate closes while most of the head's output remains (§4.2).

An inexpressible offset has no such line: if the base head cannot reproduce $\mu + \alpha h$ exactly for any choice of its parameters, then every minimizer of $\mathcal{L}$ that uses the offset needs $\alpha \neq 0$, and there is no flat direction through $\alpha = 0$ for the drift to follow. The condition is exact reproduction, not approximation, as the linear head on a one-hidden-layer base shows in the minimal model and on the agents alike (§4.4). A frozen base removes the line for a different reason: W_μ cannot move, so the split cannot change. Detaching the head's gradient from the trunk changes neither the loss nor the line, and §4.4 finds that it does not change whether α returns to zero.

4 Experiments

Setup. In continuous control we train the shaped variants AS-TD3, AS-SAC, and AS-PPO of TD3, SAC, and PPO (Fujimoto et al., 2018; Haarnoja et al., 2018; Schulman et al., 2017) for 1M steps with 5 seeds on each of 20 environments, ten from MuJoCo v5 and ten from the DeepMind Control Suite (Todorov et al., 2012; Tassa et al., 2018). Scores are min-max normalized per environment with fixed anchors and aggregated by the interquartile mean (IQM) with 95% stratified bootstrap intervals (Agarwal et al., 2021); terminal scores average the last 20 evaluations (Appendix H). Mechanism experiments use six environments, the *mechanism suite* (Hopper, HalfCheetah, Humanoid, finger/spin, walker/run, humanoid/walk), under TD3 and SAC with 5 seeds, 60 runs per arm (180 for the three-width MLP-head arm).

4.1 What the base retains

Removing the head costs almost nothing on a trainable base and much more on a frozen one. Over the 20 environments, the IQM of the per-run removal cost is $+0.006$ [0.002, 0.012] under

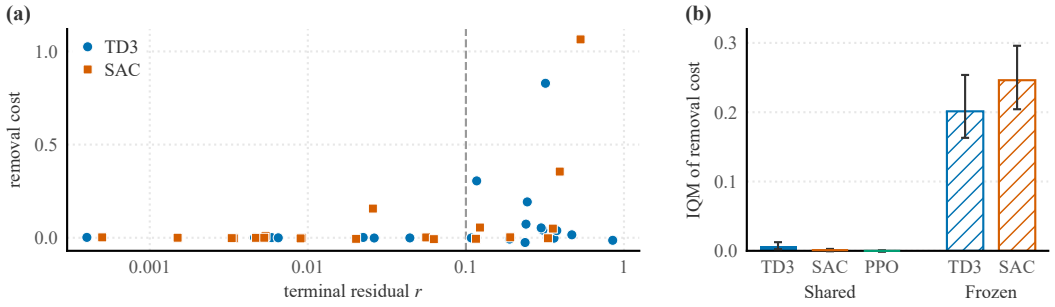


Figure 2: **The removal cost follows the residual, and is large on a frozen base.** (a) Removal cost in normalized return against the terminal residual r of §2.2, one point per environment and algorithm (20 environments under TD3 and under SAC, mean over 5 seeds); the dashed line marks the threshold $r = 0.10$. (b) IQM of the per-run removal cost with its 95% stratified bootstrap interval, for the shared head under TD3, SAC and PPO and for the frozen base under TD3 and SAC.

TD3 and $+0.001$ [-0.001, 0.003] under SAC, and the per-environment interval over TD3 and SAC contains zero in 15 of 20 environments (Figure 7); §4.3 shows that the terminal residual predicts where the cost is paid. With the trunk and base head frozen at initialization, the shaped actor scores 0.328 (TD3) and 0.371 (SAC), the base alone 0.011 and 0.009, and the IQM of the per-run removal cost is $+0.201$ and $+0.246$ (Figure 2b). Under PPO, whose critic is a state-value function, α stays within 0.05 of its initialization in 90 of the 100 runs, and removing the head costs -0.0001 [-0.0014, 0.0008]: the head is inert (§3.1).

4.2 The arc: rise, then a decline that depends on the critic

In the shared arm, in every environment of the mechanism suite, the amplitude $\|\delta_t\|$ traces an *arc*: it rises and then falls (Figure 3). Under TD3 the median amplitude in four of the six environments peaks within 150K steps and falls to between 0.5% and 25% of its peak by 1M; finger/spin ends at 40% and humanoid/walk at 35%. Under SAC all six peak within 100K steps and end at or below 11% of their peak. The fall is due to α: across the six environments the median decline of $|\alpha|$ from its peak is 98%, while $\|h\|$ ends between 20% and 94% of its own peak, above half in ten of the twelve environment-algorithm cells, so the head retains most of its norm and its gate closes (Appendix J). Over all 20 environments the head engages in every one of the 200 TD3 and SAC runs, and the arc completes (§2.2) within 1M steps in 7 under TD3 and 13 under SAC; the removal cost that remains comes from the others (§4.3), and in each of six unfinished TD3 arcs run three times as long the residual falls further (Appendix E).

The critic can decide whether the gate closes. On the four environments of the update-to-data experiment, the one TD3 gate that the twin critic leaves open at 1M, Hopper, closes under a ten-member ensemble critic (Chen et al., 2021), its terminal-to-peak ratio of $|\alpha|$ falling from 21% to 0.2%, and the SAC gates close with either critic (Appendix K).

4.3 Removal cost along the amplitude trajectory

The cost of removing the head follows the residual, not the task, as it should if absorption is what makes the head removable. We split the 20 environments by the fixed criterion $r \leq 0.10$: in the environments whose arc completed, removing the head costs $+0.0007$ [-0.0007, 0.0019] of normalized return under TD3 (7 environments) and -0.0003 [-0.0014, 0.0008] under SAC (13), nothing in either case (the IQM of the per-run cost, as in §4.1). In the 13 and 7 environments whose arc did not complete, it costs $+0.028$ [0.013, 0.053] and $+0.030$ [0.005, 0.075]. Without any threshold, the size of an environment's removal cost rises with its residual: the rank correlation is 0.61 under TD3 ($p = 0.004$, bootstrap interval [0.22, 0.80] over environments) and 0.64 under SAC ($p = 0.003$, [0.29, 0.85]) (Figure 2a).

The same dependence appears inside a single run. For every run of the mechanism suite we remove the head at four checkpoints set by a rule on the amplitude, and the cost is small during the

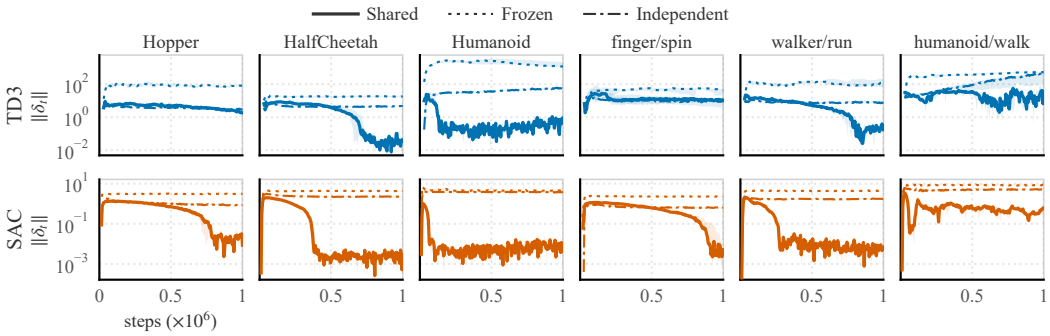


Figure 3: **The shared amplitude rises and falls, further under SAC, while the frozen and independent ones stay high.** $\|\delta_t\|$ per environment on the mechanism suite, median over 5 seeds with the interquartile range shaded, log scale shared within a row. Three of the five SAC frozen-base Humanoid runs diverge by 220K steps, so that line is the median over the rest.

Table 1: **Removal cost is largest at or just after the amplitude peak.** The head is removed at four points of each run, located by a rule that reads only the amplitude in action space (Appendix H), on the mechanism suite; mean with 95% bootstrap CI over 30 runs per algorithm.

Checkpoint	TD3	SAC
rising	$+0.015$ [0.001, 0.032]	-0.003 [-0.009, 0.002]
peak	$+0.090$ [0.051, 0.134]	$+0.045$ [0.009, 0.094]
post-peak	$+0.056$ [0.028, 0.091]	$+0.109$ [0.041, 0.193]
final	$+0.017$ [-0.005, 0.049]	-0.012 [-0.032, 0.002]

rise, largest at or just after the peak, and indistinguishable from zero at the end, under both algorithms (Table 1). Around its peak the offset therefore raises the shaped actor's return by up to $+0.090$ under TD3 and $+0.109$ under SAC, the improvement that §3.1 describes.

4.4 What absorption requires: seven arms

Section 3.2 predicts that the offset is absorbed when the base can express it and is free to change, and that cutting the gradient between the heads does not prevent it; we measure the terminal residual and the removal cost of every arm of §2.3, and each arm behaves as predicted (Figure 4).

Averaged over the twelve cells the shared arm's residual is 0.10 and removal costs $+0.003$. Detaching the head from the trunk, so that no gradient from the offset reaches the representation, leaves the residual at 0.06 and the cost at $+0.0004$: the base absorbs the offset without receiving gradient from it, because it trains on the same critic at the same shaped action. Giving the head its own trunk raises the residual to 0.55 (TD3) and 0.68 (SAC) and the cost to $+0.22$ and $+0.55$; the offset is still linear in a representation, but not in the representation the base reads, so the base cannot express it without first re-learning that representation. Freezing the base leaves the residual at 0.68 under TD3 and 0.77 under SAC, never below 0.57 in any environment, with the removal cost of §4.1.

Linearity, not width, determines the outcome. The MLP-head arm changes only what the head can compute. With a one-hidden-layer head of 16, 64, or 256 units, everything else unchanged, the residual stays between 0.58 and 0.68 under TD3 and between 0.74 and 0.83 under SAC, removal costs between $+0.44$ and $+0.78$, and none of the 36 (width, algorithm, environment) cells completes its arc, against eight of twelve for the linear head trained in the same sweep.

A head the base head can reproduce is absorbed, and one linear path determines whether it can. The last two arms move the nonlinearity to the base head and keep the shaping head linear. On a one-hidden-layer base head of 64 units the head engages in all twelve (environment, algorithm) cells and its arc completes in none: the residual is 0.60 under TD3 and 0.82 under SAC, and removing the head costs $+0.38$ and $+0.35$. Adding a zero-initialized linear path to that same base head, which restores the merge of §2.3 and changes nothing else, lowers the residual in all twelve paired

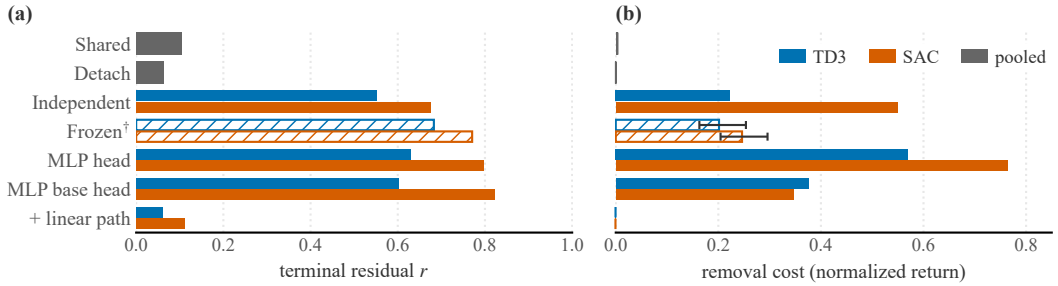


Figure 4: **Absorption requires expressibility and a trainable base.** Terminal residual r (a) and removal cost (b) for each arm of §2.3 on the mechanism suite, averaged over seeds and environments; the last row adds a zero-initialized linear path to the MLP base head. Shared and Detach are pooled over algorithms, and the MLP-head row is the mean over widths 16, 64, and 256, which agree within 0.11 in r. †: Frozen residual over the finite runs (27 of 30 under SAC), Frozen cost the IQM over the 20 environments of §4.1, with its 95% stratified bootstrap interval.

cells, its median from 0.79 to 0.012, and brings the cost to an IQM of -0.0002 [-0.0007, 0.0004] (Appendix C).

4.5 Discrete actions

We ask the same question for discrete actions, a second geometry: on 26 Atari games with a discrete SAC host (Zhou et al., 2024), the offset acts on the logits and the base is read out by dropping it. Under one protocol for all three policies, the base actor alone scores a human-normalized IQM of 0.183 [0.157, 0.216], the shaped actor 0.170, and the unshaped host 0.154; removing the head raises the IQM by 0.013 [0.003, 0.023] (Appendix M).

5 Related work

Reward shaping and the action channel. Potential-based shaping characterizes the reward modifications that leave the optimal policy unchanged and can be removed at deployment (Ng et al., 1999). The action channel has no such invariance, since an offset changes the state distribution, so the question has to be answered by the training dynamics; our answer is a condition for absorption and the mechanism behind it (long form in Appendix N).

Folding a branch into a layer. Structural re-parameterization (Ding et al., 2021; Hu et al., 2022; Wang et al., 2026a) and linear over-parameterization (Guo et al., 2020; Arora et al., 2018; Du et al., 2018) train a multi-branch block and fold it into one layer after training; our linear head admits the same merge (§2.3), which is our reference point for absorption. We show that training reaches it by itself, and only where an exact merge exists: a base head that can approximate the branch but not reproduce it keeps it, and one linear path that restores the merge restores absorption (§4.4). The condition thus predicts, for any co-trained branch, whether training will absorb it, and for a branch that admits no merge it predicts that the branch stays (Appendix G); the gate reads either outcome while training runs.

Scalar-gated branches. In ReZero (Bachlechner et al., 2021) and LayerScale (Touvron et al., 2021) a residual branch is scaled by a learnable gate that starts at or near zero, and the gate grows and stays open. Gates into a frozen backbone close through a dead-gradient regime (Farazi et al., 2026), attention sublayers learn to mute themselves (Saikumar & Varghese, 2025), and sparsity penalties prune explicitly (Huang & Wang, 2018). Our α closes with no penalty and only after it has opened, and the three states of §2.2 distinguish these cases.

Training-only components. Privileged-sensor scaffolds train with a component and deploy without it (Hu et al., 2024). Residual distillation (Li et al., 2020), tapered normalization (Kanavalau et al., 2026), decaying action priors (Sood et al., 2023; 2025), and jump-start or attenuated residual

policies (Uchendu et al., 2023; Trumpp et al., 2026; Bolychev et al., 2026) remove one on a schedule a designer sets; in every case the schedule, not the base, removes it. Residual policy learning keeps the correction on a fixed base (Silver et al., 2018; Johannink et al., 2019; Wang et al., 2025; Ankile et al., 2025; Hirschowitz & Ramos, 2026), the regime of our frozen arm; under the condition of §3 no schedule is needed: an expressible branch on a trainable base drifts toward absorption without one.

Internalization and removability. Tsilivis et al. (2026) define internalization as absorbing an explicit procedure into weights of the same hypothesis class; their scaffold is removed by a curriculum, ours is absorbed, and we add the trainable-base condition, the flat-line mechanism, and an amplitude diagnostic. Our checkpoint interventions find within a run that a parametric branch's removability depends on the training trajectory, as concurrent work finds for an annealable prior (Yang et al., 2026).

6 Discussion

What removal means. Removal deploys the base actor as trained, with the head deleted; for the linear head the fold $W_\mu + \alpha W_h$ of §2.3 is the closed-form reference point of the absorbed state: the offset δ is the distance of the trained base head from it, and absorption is the finding that training closes the distance on its own, with no merge performed. Removal can also be free without absorption, for a head that never engaged (§2.2) or a redundant branch that collapses under strong noise (Appendix B); the reading of §2.2 covers those cases as well, and the condition speaks to absorption. Where no fold exists, for a nonlinear or multiplicative head (Appendices D and G) or a base that is frozen or belongs to another module, that reading is the only one.

Inexpressible, not slow. At three million steps, three times the budget, the width-64 MLP head stays unabsorbed, median residual 0.73 and removal cost $+0.91$ (Appendix D); for a linear offset the paired control of §4.4 separates exact reproduction from budget.

Batch size in the agents. In a batch-size sweep on the agents, which train with Adam and a bootstrapped critic, a batch of 64 completes the arc before one of 1024 in five of six pairs, the direction Proposition 3 predicts (Appendix F).

What the condition sorts. The condition reduces the question of when a co-trained branch can be removed for free to two questions about the agent: can the base head reproduce the branch exactly, and is the base trainable? An expressible offset on a trainable base decays under gradient noise and needs no schedule (§4.4). An offset on a fixed base stays, as in residual policy learning and our frozen arm. An offset the base head cannot reproduce has no flat line to a closed gate and stays as well, as in every inexpressible arm of §4.4 and in the same linear head composed multiplicatively, whose gate stays open in eleven of twelve suite cells (Appendix G). The recipe is short: add the head, watch its amplitude (in action space where the output map saturates, Appendix E), and drop the head once the amplitude has fallen below a tenth of its peak.

7 Conclusion

Anyone who adds a branch to the action channel during training and removes it at deployment now has a condition that predicts whether training absorbs it. We show that the condition is exact reproduction by a trainable base head, not capacity: a nonlinear head with more parameters is not absorbed, and in a paired control, one linear path added to a nonlinear base head restores absorption. We identify the mechanism that explains why no schedule is needed: expressibility gives the loss a flat direction that leads to a closed gate, and gradient noise moves the parameters along it. The gate reads either outcome while training runs: where the offset is absorbed it rises and then falls on its own, for deterministic and stochastic actors alike, and on 20 tasks removing the head costs almost nothing. The condition can be asked of any co-trained branch on the action channel, whether it comes from a controller, a prior, or a residual policy, and, through the softmax case of Theorem 1, of discrete actions as well as continuous ones: Propositions 2 and 3 give the mechanism for a linear head on a linear base head, and the arms of §4.4 and Appendix G show that the branches absorbed are exactly those the base head can reproduce. Policies absorb what they can express, and only that.

AI USE STATEMENT

We used generative AI tools to assist with drafting and editing the manuscript, with writing the analysis and experiment code, and, under the authors' direction, with the design and analysis of the experiments. They were not used to generate synthetic data or to produce or alter any experimental result. All AI-assisted material was reviewed and verified by the authors, who take full responsibility for this work.

ETHICS STATEMENT

This work studies an optimization phenomenon in reinforcement-learning agents on standard simulated benchmarks; it involves no human subjects or personal data, and we foresee no harm.

REPRODUCIBILITY STATEMENT

Every result in the paper is computed from the logged data of the runs it reports. The evaluation protocol, normalization anchors, terminal window, residual conventions, and bootstrap procedures are specified in Appendix H; per-environment results in Appendix I; the update-to-data, ε-sensitivity, and Atari protocols in Appendices K, L, and M; the minimal model in Appendix B; the proofs in Appendix A, and the hyperparameters in Appendix O. Code, configurations, the anchor files, and the logged evaluation data of every run will be released with the paper.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Lars Ankile, Zhenyu Jiang, Rocky Duan, Guanya Shi, Pieter Abbeel, and Anusha Nagabandi. Residual off-policy RL for finetuning behavior cloning policies. *arXiv preprint arXiv:2509.19301*, 2025.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, pp. 244–253, 2018.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an Ornstein-Uhlenbeck like process. In *Proceedings of the Thirty Third Conference on Learning Theory (COLT 2020)*, pp. 483–513, 2020.

Anton Bolychev, Georgiy Malaniya, Sinan Ibrahim, and Pavel Osinenko. An agency-transferring model-free policy enhancement technique. *arXiv preprint arXiv:2606.09825*, 2026.

Feng Chen, Daniel Kunin, Atsushi Yamamura, and Surya Ganguli. Stochastic collapse: How gradient noise attracts SGD dynamics towards simpler subnetworks. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double Q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021.

Alex Damian, Tengyu Ma, and Jason D. Lee. Label noise SGD provably prefers flat global minimizers. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pp. 27449–27461, 2021.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. RepVGG: Making VGG-style ConvNets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, 2021.

Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pp. 382–393, 2018.

Moshiur Farazi, Sameera Ramasinghe, Bekir Sait Ciftler, Mahbub Ahmed Turza, and Shafin Rahman. The gate always closes: On injecting auxiliary signals into frozen vision-language models. *arXiv preprint arXiv:2607.23335*, 2026.

Jinyuan Feng, Min Chen, Zhiqiang Pu, Tenghai Qiu, and Jianqiang Yi. Efficient multi-task reinforcement learning via task-specific action correction. *arXiv preprint arXiv:2404.05950*, 2024.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, pp. 1587–1596, 2018.

Shuxuan Guo, José M. Álvarez, and Mathieu Salzmann. ExpandNets: Linear over-parameterization to train compact convolutional networks. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, pp. 1861–1870, 2018.

Zhongyu He, Yuanfan Li, Fei Huang, Tianyu Chen, Siyuan Chen, Xingyang Li, Meng Hsuan Yu, Xiangrong Liu, Leyi Wei, Lu Pan, Ke Zeng, and Xunliang Cai. SIRI: Self-internalizing reinforcement learning with intrinsic skills for LLM agent training. *arXiv preprint arXiv:2606.02355*, 2026.

Ethan Hirschowitz and Fabio Ramos. Warp RL: Reshaping base policy distributions for dynamics adaptation. *arXiv preprint arXiv:2606.31043*, 2026.

Edward S. Hu, James Springer, Oleh Rybkin, and Dinesh Jayaraman. Privileged sensing scaffolds reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Mu Hu, Junyi Feng, Jiashen Hua, Baisheng Lai, Jianqiang Huang, Xiaojin Gong, and Xian-Sheng Hua. Online convolutional re-parameterization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, 2022.

Zehao Huang and Naiyan Wang. Data-driven sparse structure selection for deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV 2018)*, pp. 317–334, 2018.

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 *International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029, 2019. doi: 10.1109/ICRA.2019.8794127.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 267–274, 2002.

Andrei Kanavalau, Carmen Amo Alonso, and Sanjay Lall. Gated normalization removal and scale anchoring in pre-norm transformers. *arXiv preprint arXiv:2602.10408*, 2026.

Anssi Kanervisto, Christian Scheller, and Ville Hautamäki. Action space shaping in deep reinforcement learning. In *2020 IEEE Conference on Games (CoG 2020)*, pp. 479–486, 2020.

Abdallah Khemais. Exact network surgery: Functional invariance and gradient plasticity in reactive computational graphs. *arXiv preprint arXiv:2607.16568*, 2026.

Emek Barış Küçüktabak, Karankumar Patel, Zhaodong Yang, Jinda Cui, Kazuhiro Sasabuchi, and Jun Takamatsu. Real-world reinforcement learning with MPC scaffolding for dexterous manipulation. *arXiv preprint arXiv:2609.14878*, 2026.

Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel L. K. Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. In *International Conference on Learning Representations (ICLR 2021)*, 2021.

Guilin Li, Junlei Zhang, Yunhe Wang, Chuanjian Liu, Matthias Tan, Yunfeng Lin, Wei Zhang, Jiashi Feng, and Tong Zhang. Residual distillation: Towards portable deep neural networks without shortcuts. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss? A mathematical framework. In *International Conference on Learning Representations (ICLR 2022)*, 2022.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. SKILL0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026.

Guozheng Ma, Lu Li, Haoyu Wang, Zixuan Liu, Pierre-Luc Bacon, and Dacheng Tao. What makes value learning efficient in residual reinforcement learning? *arXiv preprint arXiv:2602.10539*, 2026.

Muhammad A. Muttaqien, Tomohiro Motoda, Ryo Hanai, and Yukiyasu Domae. ORPA: Online residual policy adaptation for robot manipulation control with human feedback. *arXiv preprint arXiv:2608.17323*, 2026.

Andrew Y Ng, Daishi Harada, and Stuart J Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, pp. 278–287, 1999.

Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of SGD for diagonal linear networks: a provable benefit of stochasticity. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pp. 29218–29230, 2021.

Tu-Hoa Pham, Giovanni De Magistris, Don Joven Agravante, Subhajit Chaudhury, Asim Munawar, and Ryuki Tachibana. Constrained exploration and recovery from experience shaping. *arXiv preprint arXiv:1809.08925*, 2018.

Dhananjay Saikumar and Blesson Varghese. Data-free pruning of self-attention layers in LLMs. *arXiv preprint arXiv:2512.20636*, 2025.

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

Yipeng Shi, Zhipeng Ma, Yue Wang, Qitai Tan, Yang Li, Peng Chen, and Zhengzhou Zhu. PATS: Policy-aware training scaffolding for agentic reinforcement learning. *arXiv preprint arXiv:2607.21419*, 2026.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning (ICML)*, pp. 387–395, 2014.

Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

Shivam Sood, Ge Sun, Peizhuo Li, and Guillaume Sartoretti. DecAP: Decaying action priors for accelerated imitation learning of torque-based legged locomotion policies. *arXiv preprint arXiv:2310.05714*, 2023.

Shivam Sood, Laukik Nakhwa, Sun Ge, Yuhong Cao, Jin Cheng, Fatemah Zargarbashi, Taerim Yoon, Sungjoon Choi, Stelian Coros, and Guillaume Sartoretti. APEX: Action priors enable efficient exploration for robust motion tracking on legged robots. *arXiv preprint arXiv:2505.10022*, 2025.

Xiaowei Sun, Jin Li, Yili Hong, Yikun Fu, and Yanghua Xiao. One policy, any budget: Internalizing budget-aware search via reinforcement learning. *arXiv preprint arXiv:2609.00813*, 2026.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

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 (IROS)*, pp. 5026–5033, 2012.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021)*, pp. 32–42, 2021.

Raphael Trumpp, Ehsan Javanmardi, Jin Nakazato, Manabu Tsukada, and Marco Caccamo. RaceMOP: Mapless online path planning for multi-agent autonomous racing using residual policy learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

Raphael Trumpp, Denis Hoornaert, Mirco Theile, and Marco Caccamo. Efficient real-world autonomous racing via attenuated residual policy optimization. *arXiv preprint arXiv:2603.12960*, 2026.

Nikolaos Tsilivis, Nirmit Joshi, Marko Medvedev, Julia Kempe, and Nati Srebro. Learning through internalization. *arXiv preprint arXiv:2606.20937*, 2026.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, Sergey Levine, and Karol Hausman. Jump-start reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, pp. 34556–34583, 2023.

Dong Wang, Wenwu Tang, Yun Cheng, and Olga Saukh. DLR: Zero-inference-cost latent residuals for low-rank pre-training. *arXiv preprint arXiv:2606.28932*, 2026a.

Jingchu Wang, Bingbing Xu, Yige Yuan, Dan Zhang, Bin Xie, Xiaoqian Sun, and Huawei Shen. R^2PO: Decoupling rollout and inference policies for LLM reasoning. *arXiv preprint arXiv:2601.11960*, 2026b.

Pengcheng Wang, Xinghao Zhu, Yuxin Chen, Chenfeng Xu, Masayoshi Tomizuka, and Chenran Li. Residual policy gradient: A reward view of KL-regularized objective. *arXiv preprint arXiv:2503.11019*, 2025.

Ruiming Wu, Shuang Li, Liding Zhang, Alois Knoll, and Zhaopeng Chen. Gated residual body-hand coordination for whole-body humanoid teleoperation. *arXiv preprint arXiv:2609.18763*, 2026.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi Fan, Guanya Shi, and Yuke Zhu. Self-improving vision-language-action models with data generation via residual RL. *arXiv preprint arXiv:2511.00091*, 2025.

Roy Xing, Seyoung Ree, and Brian Plancher. MPC-Injection: Biasing off-policy locomotion RL toward controller-induced behavior basins. *arXiv preprint arXiv:2606.26392*, 2026. Conference on Robot Learning (CoRL).

Zonglin Yang, Ziming Zhao, Wei Tang, Xunyu Jiang, Yihong Liu, Tailin Chen, Zifu Yu, and Jiayu Liu. Training trajectories determine circuit removability in annealable soft-prior transformers. *arXiv preprint arXiv:2609.10287*, 2026. Accepted at PRICAI 2026.

Haoran Ye, Yuxing Lu, Haonan Dong, Zhaochen Su, and Guojie Song. Harness-Zero: Harness distillation via agent-as-harness. *arXiv preprint arXiv:2609.24974*, 2026.

Dongjie Yu, Kun Lei, Zhennan Jiang, Jia Pan, and Huazhe Xu. Beyond action residuals: Real-world robot policy steering via bottleneck latent reinforcement learning. *arXiv preprint arXiv:2605.19919*, 2026.

Pengfei Zhang, Teng Sun, and Xianchao Xiu. Beyond noise steering: Dual-latent space reinforcement learning for generative robot policy. *arXiv preprint arXiv:2609.11270*, 2026.

Haibin Zhou, Tong Wei, Zichuan Lin, Junyou Li, Junliang Xing, Yuanchun Shi, Li Shen, Chao Yu, and Deheng Ye. Revisiting discrete soft actor-critic. *Transactions on Machine Learning Research (TMLR)*, 2024.

Liu Ziyin. Symmetry induces structure and constraint of learning. In *International Conference on Machine Learning (ICML)*, 2024.

A PROOFS

This appendix proves Theorem 1 for a deterministic continuous actor and for the softmax over discrete actions, the continuous case written for the identity output map, with the Jacobian factor of a general smooth map such as the scaled $\tanh$ of TD3 supplied at its end, and then proves the corollary used in §3.1. For a reparameterized Gaussian actor such as SAC's, $a = f(\mu(s) + \alpha h(s) + \sigma(s)\,\xi)$ with $\xi \sim \mathcal{N}(0, I)$ and $\sigma(s)$ bounded below by $\sigma_{\min} > 0$, the same four steps hold inside the expectation over ξ, with (A1) taken on the whole action set; the distribution shift is then bounded through the total-variation distance between the two Gaussians, at most $\alpha B_h/(2\sigma_{\min})$, without (A2), and the first-order term is $\mathbb{E}_\xi\left[\nabla_a Q^{\pi_\mu}(s,a)^\top J_f(\mu(s) + \sigma(s)\,\xi)\right] h(s)$, the derivative in equation 3.

Performance difference lemma. The derivation rests on the following lemma (Kakade & Langford, 2002).

Lemma 4 (Performance difference, Kakade & Langford, 2002)**.** *For any two policies π and π' acting in the same MDP,*

$$J(\pi') - J(\pi) \;=\; \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^{\pi'}}\left[\mathbb{E}_{a\sim\pi'(\cdot|s)}\big[A^{\pi}(s,a)\big]\right], \tag{5}$$

where $d^{\pi'}$ is the discounted state visitation distribution under π' and $A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$ is the advantage under π.

Equation 5 is exact and holds without functional-form or smoothness assumptions on the policies. The proof telescopes the discounted return; see Kakade & Langford (2002).

Regularity conditions.

Assumption 5 (Continuous action space)**.**

- (A1) *$Q^{\pi_\mu}(s,a)$ is twice continuously differentiable in a in a neighborhood of $a = \mu(s)$ of a common radius $r_0 > 0$ for all s, with $\|\nabla_a^2 Q^{\pi_\mu}\| \le M_Q$ and $\|\nabla_a Q^{\pi_\mu}(s,\mu(s))\| \le G_Q$ for all s.*
- (A2) *The transition kernel satisfies $\mathrm{TV}\big(P(\cdot|s,a_1),\,P(\cdot|s,a_2)\big) \le L_P\|a_1 - a_2\|$ for all s, a_1, a_2.*
- (A3) *The shaping head output is bounded, $\|h(s)\| \le B_h$ for all s.*
- (A4) *Rewards are bounded, $|r(s,a)| \le R_{\max}$.*
- (A5) *The action gradient $\nabla_a Q^{\pi_\mu}(s,\mu(s))$ is continuous in the parameters of the base policy, for every s.*

Assumption 6 (Discrete action space)**.** *The action space is finite, $|\mathcal{A}| < \infty$, and (A3) and (A4) of Assumption 5 hold. The state-distribution shift is then bounded directly through the total-variation distance between policies, which is $O(\alpha)$ by the Lipschitz property of the softmax, without requiring smoothness of the transition kernel in the action argument. For the limit in Appendix A.1, $Q^{\pi_\mu}(s,a)$ is continuous in the parameters of the base policy, for every s and a.*

Derivation.

Proof of Theorem 1. **Continuous deterministic policies.**

Step 1: apply the performance difference lemma. With $\pi' = \pi_\alpha$, the shaped policy at gate value α, and $\pi = \pi_\mu$,

$$G(\alpha) - G(0) = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^{\pi_\alpha}}\big[A^{\pi_\mu}\big(s,\,\mu(s) + \alpha\,h(s)\big)\big]. \tag{6}$$

Step 2: separate the state-distribution shift. The difference $d^{\pi_\alpha} - d^{\pi_\mu}$ introduces a second-order correction:

$$\mathbb{E}_{d^{\pi_\alpha}}[\psi(s)] = \mathbb{E}_{d^{\pi_\mu}}[\psi(s)] + \underbrace{\mathbb{E}_{d^{\pi_\alpha} - d^{\pi_\mu}}[\psi(s)]}_{O(\alpha)}. \tag{7}$$

Under Assumption 5(A2) the total-variation distance satisfies $\mathrm{TV}(d^{\pi_\alpha}, d^{\pi_\mu}) = O\big(\gamma L_P \alpha B_h/(1-\gamma)\big)$, obtained by recursively bounding the per-step distribution shift with the Lipschitz condition on P. The integrand $A^{\pi_\mu}(s, \mu(s) + \alpha h(s))$ vanishes at $\alpha = 0$ and is $O(\alpha)$ by the Taylor expansion of Step 3, so the cross-term is $O(\alpha) \times O(\alpha) = O(\alpha^2)$:

$$\mathbb{E}_{d^{\pi_\alpha} - d^{\pi_\mu}}\big[A^{\pi_\mu}\big(s,\, \mu(s) + \alpha h(s)\big)\big] = O\left(\frac{\gamma L_P \alpha^2 B_h^2 G_Q}{1-\gamma}\right). \tag{8}$$

Step 3: Taylor-expand the advantage. The advantage at the base action is zero, $A^{\pi_\mu}(s, \mu(s)) = Q^{\pi_\mu}(s, \mu(s)) - V^{\pi_\mu}(s) = 0$. Under Assumption 5(A1) the first-order expansion around $a = \mu(s)$ gives

$$A^{\pi_\mu}\big(s,\, \mu(s) + \alpha h(s)\big) = \nabla_a Q^{\pi_\mu}(s, \mu(s))^\top \cdot \alpha\, h(s) + O(\alpha^2 M_Q B_h^2), \tag{9}$$

using $\nabla_a A^{\pi_\mu} = \nabla_a Q^{\pi_\mu}$, since $V^{\pi_\mu}(s)$ does not depend on a.

Step 4: combine. Substituting equation 8 and equation 9 into equation 6,

$$\begin{aligned} G(\alpha) - G(0) &= \frac{1}{1-\gamma}\, \mathbb{E}_{d^{\pi_\mu}}\big[\alpha\, \nabla_a Q^{\pi_\mu}(s, \mu(s))^\top h(s)\big] + O(\alpha^2) \\ &= \frac{\alpha}{1-\gamma}\, \mathbb{E}_{d^{\pi_\mu}}\big[\Phi_h(s)\big] + R(\alpha), \end{aligned} \tag{10}$$

where $\Phi_h(s) = \nabla_a Q^{\pi_\mu}(s, \mu(s))^\top h(s)$ is the continuous case of equation 3, and the remainder satisfies $R(\alpha) = O(\alpha^2)$ with leading constant $C_{\text{cont}} = \frac{1}{1-\gamma}\big(\frac{1}{2} M_Q B_h^2 + \frac{2\gamma L_P G_Q B_h^2}{1-\gamma}\big)$, the first term being the Taylor remainder of Step 3 and the second the state-distribution shift of Step 2.

Output maps other than the identity. TD3 acts with $a = f(\mu(s) + \alpha h(s))$ for f a $\tanh$ scaled to the action bounds (Equation 1), and the four steps go through for any output map that is C^2 with $\|J_f\| \le L_f$ and $\|\nabla^2 f\| \le M_f$, since $f(\mu + \alpha h) - f(\mu) = \alpha\, J_f(\mu)\, h + O(\alpha^2 M_f B_h^2)$. Step 2 holds with B_h replaced by $L_f B_h$; Step 3 expands Q^{π_μ} around $a = f(\mu(s))$, where (A1) is now taken, with h replaced by $J_f(\mu(s))\, h$ and the remainder constant $\frac{1}{2} M_Q B_h^2$ by $\frac{1}{2}\big(M_Q L_f^2 + M_f G_Q\big) B_h^2$; and the first-order term becomes $\Phi_h(s) = \nabla_a Q^{\pi_\mu}\big(s, f(\mu(s))\big)^\top J_f(\mu(s))\, h(s)$, which is the derivative in equation 3 by the chain rule. The identity map is the case $J_f = I$, $M_f = 0$; for the scaled $\tanh$, L_f is the action scale and M_f is $4/(3\sqrt{3}) \approx 0.77$ times it.

Softmax policies over discrete actions.

Step 1: apply the performance difference lemma. With $\pi_\alpha(a|s) = \mathrm{softmax}\big(\mu(s) + \alpha \cdot h(s)\big)_a$, where $\mu(s)$ is the vector of base logits,

$$G(\alpha) - G(0) = \frac{1}{1-\gamma}\, \mathbb{E}_{s \sim d^{\pi_\alpha}}\left[\sum_a \big(\pi_\alpha(a|s) - \pi_\mu(a|s)\big)\, Q^{\pi_\mu}(s, a)\right]. \tag{11}$$

Step 2: separate the state-distribution shift. Under Assumption 6, $\mathrm{TV}(\pi_\alpha(\cdot|s), \pi_\mu(\cdot|s)) = O(\alpha)$ by the Lipschitz property of the softmax, so $\mathrm{TV}(d^{\pi_\alpha}, d^{\pi_\mu}) = O(\gamma\alpha/(1-\gamma))$. The cross-term is again $O(\alpha^2)$.

Step 3: expand the policy difference. The softmax Jacobian $\partial\pi(a|s)/\partial\mu_b(s) = \pi(a|s)\big(\mathbf{1}\{a = b\} - \pi(b|s)\big)$ gives

$$\pi_\alpha(a|s) - \pi_\mu(a|s) = \alpha \cdot \pi_\mu(a|s)\, \big(h_a(s) - \mathbb{E}_{\pi_\mu}[h(s)]\big) + O(\alpha^2), \tag{12}$$

and substituting into the inner sum,

$$\begin{aligned} \sum_a \big(\pi_\alpha(a|s) - \pi_\mu(a|s)\big)\, Q^{\pi_\mu}(s, a) &= \alpha \sum_a \pi_\mu(a|s)\, \big(h_a - \mathbb{E}_{\pi_\mu}[h]\big)\, Q^{\pi_\mu}(s, a) + O(\alpha^2) \\ &= \alpha\, \mathrm{Cov}_{\pi_\mu(\cdot|s)}\big(h(s),\, Q^{\pi_\mu}(s, \cdot)\big) + O(\alpha^2). \end{aligned} \tag{13}$$

Step 4: combine.

$$G(\alpha) - G(0) = \frac{\alpha}{1-\gamma}\, \mathbb{E}_{d^{\pi_\mu}}\big[\Phi_h(s)\big] + R(\alpha), \tag{14}$$

where $\Phi_h(s) = \mathrm{Cov}_{\pi_\mu(\cdot|s)}\big(h(s),\, Q^{\pi_\mu}(s, \cdot)\big)$ is the softmax case of equation 3 and $|R(\alpha)| \le C_{\text{disc}} \cdot \alpha^2$ with $C_{\text{disc}} = B_h^2 R_{\max}/(1-\gamma)^3$. □

A.1 SELF-DIMINISHING COROLLARY

Corollary 7 (Self-diminishing property)**.** *If the base policy π_μ is optimal, then $\Phi_h(s) = 0$ for all shaping heads h and all states s, for deterministic and softmax policies; for a reparameterized Gaussian actor the gradient of the full actor loss with respect to h vanishes at the soft optimum.*

Proof. Continuous deterministic. At optimality the first-order necessary condition on the pre-activation gives

$$J_f(\mu(s))^\top \nabla_a Q^{\pi_\mu}(s, f(\mu(s))) = 0,$$

so $\Phi_h(s) = \nabla_a Q^{\pi_\mu}(s, f(\mu(s)))^\top J_f(\mu(s))\, h(s) = 0$; with the identity map this reads $\nabla_a Q^{\pi_\mu}(s, \mu(s)) = 0$ and $\Phi_h(s) = \nabla_a Q^{\pi_\mu}(s, \mu(s))^\top h(s) = 0$.

Discrete. A softmax policy has full support, so it is optimal at s only if every action attains $\max_a Q^{\pi_\mu}(s, a)$, and then $Q^{\pi_\mu}(s, \cdot)$ is constant and the covariance vanishes. As the base policy approaches optimality its mass concentrates on $\mathcal{A}_s^* = \arg\max_a Q^{\pi_\mu}(s, a)$, and writing the covariance with $\max_b Q^{\pi_\mu}(s, b)$ subtracted from $Q^{\pi_\mu}(s, \cdot)$ gives $|\Phi_h(s)| \le 4B_h R_{\max}\, \pi_\mu(\mathcal{A} \backslash \mathcal{A}_s^* \mid s)/(1-\gamma) \to 0$, under (A3) and (A4).

SAC (entropy-regularized). At the soft optimum the stationarity condition for the maximum-entropy objective gives $\nabla_\mu \big(\mathbb{E}[Q^{\pi_\mu}] + \beta\, \mathcal{H}(\pi)\big) = 0$. The Q-part of this gradient, projected onto h, is precisely $\Phi_h^{\mathrm{SAC}}(s)$, and the entropy part is $\beta\, (\nabla_\mu \mathcal{H})^\top h$. At soft optimality these two terms cancel, so the total gradient signal reaching h from the SAC actor loss vanishes. While $\Phi_h^{\mathrm{SAC}}(s)$ alone need not be exactly zero, since it equals $-\beta(\nabla_\mu \mathcal{H})^\top h$, the training dynamics drive h through the full actor loss, and the gradient of that full loss with respect to h vanishes at the soft optimum. □

The corollary extends from an optimal base to one that approaches optimality under Assumption 5(A5), taken at $a = f(\mu(s))$ for an output map other than the identity, and its discrete counterpart in Assumption 6. For a fixed head h and state s, $\Phi_h(s)$ is then continuous in the parameters of the base policy, since J_f is continuous, $\mu(s)$ is continuous in those parameters, and the softmax covariance is continuous in the logits and in $Q^{\pi_\mu}(s, \cdot)$; hence $\Phi_h(s) \to 0$ as the parameters converge to those of an optimal base policy. Under SAC the statement applies to the gradient of the full actor loss with respect to h, as in the SAC case of the proof, and not to Φ_h^{SAC} alone.

The corollary shows that the per-state functional is not an independent signal but a measure of the base policy's sub-optimality. As the base policy improves through training, Φ_h diminishes, weakening the gradient signal to both h and α.

A.2 FLAT LINE

Proof of Proposition 2. The merged readout $(\bar{W}, \bar{b})$ is unchanged along the line, and the trunk ϕ is unchanged, so the shaped action, hence the loss, is unchanged. Differentiating $\mathcal{L}$ twice with respect to W_h through $\bar{W} = W_\mu + \alpha W_h$ gives $\nabla^2_{W_h} \mathcal{L} = \alpha^2\, \nabla^2_{\bar{W}} \mathcal{L}$, whose trace is α^2 times a quantity that depends on θ only through $(\phi, \bar{W}, \bar{b})$, all constant along the line; at a minimizer of $\mathcal{L}$ the Hessian with respect to $(\bar{W}, \bar{b})$ is positive semidefinite, because any change of $\bar{W}$ can be made through W_μ alone, so that quantity is nonnegative and the trace is least at $\alpha + \lambda = 0$. □

A.3 DRIFT ALONG THE LINE

Setting. The trunk is fixed and its features are $z(s) \in \mathbb{R}^k$ with the constant feature appended, so that biases are columns of the weight matrices; $\Sigma = \mathbb{E}[zz^\top]$, assumed nonsingular, and $S = \operatorname{tr} \Sigma$, expectations over a fixed set of training states. The parameters are $\theta = (W_\mu, W_h, \alpha)$ with $W_\mu, W_h \in \mathbb{R}^{m \times k}$, the model is $f_\theta(s) = (W_\mu + \alpha W_h)\, z(s)$, and the loss is $\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}\|f_\theta(s) - y(s)\|^2$ with the target perturbed by independent noise at every step of stochastic gradient descent. Let Γ be the zero-loss manifold, assumed non-empty, and let $\bar{W} = W_\mu + \alpha W_h$; on Γ the Hessian of $\mathcal{L}$ equals the Gauss–Newton matrix $\mathbb{E}[J^\top J]$ with $J = \partial f_\theta / \partial \theta$. Li et al. (2022), Corollary 5.2, give the limiting dynamics of label-noise stochastic gradient descent on Γ, as the step size $\eta \to 0$ over $\Theta(\eta^{-2})$ steps, as the projected gradient flow $\dot{\theta} = -\, c\, P_\theta \nabla R(\theta)$, where $R(\theta) = \operatorname{tr} \mathbb{E}[J^\top J]$, P_θ is the projection onto

the tangent space of Γ, and $c > 0$ collects the step size and the noise variance. The tangent space is the null space of the Hessian, and the normal space its range, so P_θ is the orthogonal projection.

The trace. The blocks of J are $\partial f/\partial W_\mu = z^\top \otimes I_m$, $\partial f/\partial W_h = \alpha\, z^\top \otimes I_m$, and $\partial f/\partial \alpha = W_h z$. Hence

$$R(\theta) = mS + \alpha^2 mS + \mathbb{E}\|W_h z\|^2 = mS\,(1+\alpha^2) + \mathrm{tr}(W_h \Sigma W_h^\top), \tag{15}$$

which does not depend on W_μ; with a trainable trunk the additional terms depend on the head only through $\bar{W}$ and are constant along the line.

The tangent space and the projection. A displacement $(dW_\mu, dW_h, d\alpha)$ is tangent to Γ when it leaves $\bar{W}$ unchanged, $dW_\mu + \alpha\, dW_h + d\alpha\, W_h = 0$. Two families span it: the line of Proposition 2, $u = (-W_h, 0, 1)$, of squared length $1 + \|W_h\|_F^2$, and the head directions $v_V = (-\alpha V, V, 0)$ for $V \in \mathbb{R}^{m\times k}$. The gradient of R has components $\partial R/\partial W_\mu = 0$, $\partial R/\partial W_h = 2W_h\Sigma$, and $\partial R/\partial \alpha = 2mS\alpha$. The two families are not orthogonal, $\langle u, v_V\rangle = \alpha\langle W_h, V\rangle$, and because the head gradient is of order one this coupling contributes to the gate's equation at the same order as the gate gradient itself. Orthogonalizing u against the head directions gives $u_\perp = \big(-W_h/(1+\alpha^2),\, -\alpha W_h/(1+\alpha^2),\, 1\big)$ with $\|u_\perp\|^2 = (1 + \alpha^2 + \|W_h\|_F^2)/(1+\alpha^2)$, and the α component of the exact projection is $\langle \nabla R, u_\perp\rangle/\|u_\perp\|^2 = 2\alpha\,\big(mS(1+\alpha^2) - \mathrm{tr}(W_h\Sigma W_h^\top)\big)/(1+\alpha^2+\|W_h\|_F^2)$; solving the normal-space equation $(1+\alpha^2)M + \langle W_h, M\rangle W_h = 2\alpha(W_h\Sigma + mS\,W_h)$ for the normal component gives the same value, and the head component $\big(2W_h\Sigma - \alpha\,(P_\theta \nabla R)_\alpha W_h\big)/(1+\alpha^2)$, which is the full head gradient $2W_h\Sigma$ to leading order in α. Reading off the α and W_h components of $-\,c\,P_\theta \nabla R$ gives, with the factor 2 absorbed into c, the exact gate equation and the head equation to leading order in α,

$$\dot\alpha = -\,c\,\frac{mS(1+\alpha^2) - \mathrm{tr}(W_h\Sigma W_h^\top)}{1+\alpha^2+\|W_h\|_F^2}\,\alpha, \qquad \dot W_h = -\,c\,W_h\Sigma, \qquad \dot W_\mu = -\,\dot\alpha\,W_h - \alpha\,\dot W_h, \tag{16}$$

the last equation being the statement that the flow stays on Γ. Dropping the α^2 terms of the first equation gives Proposition 3.

Remarks. The gate decays at the rate of $mS - \mathrm{tr}(W_h\Sigma W_h^\top)$, the summed curvature of all mk head weights less the head's output power, divided by the squared length of the line's tangent vector; the rate is positive whenever $\|W_h\|_2^2 < m$, since $\mathrm{tr}(W_h\Sigma W_h^\top) \le \|W_h\|_2^2 S$, and when the output power exceeds mS the gate first rises and then decays: the trace restricted to Γ is a strictly convex function of (α, W_h) whose only critical point is $(0,0)$, and the projected flow is the gradient flow of that coercive trace on Γ, so from every starting point it converges to $\alpha = 0$ and $W_h = 0$. The head decays per eigen-direction of Σ at the rate of one eigenvalue. Since S is the sum of the k eigenvalues, the ratio of the two rates is $\big(mk - \mathrm{tr}(W_h\Sigma W_h^\top)/\bar\lambda\big)$ times the ratio of the mean eigenvalue $\bar\lambda$ to the eigenvalue of the direction in question, divided by $1 + \|W_h\|_F^2$; for a small head it is of the order of the number of weights in the head. The mechanism behind the head's slow decay is visible in the projection: the noise on α has variance proportional to $\mathbb{E}\|W_h z\|^2$ and does not vanish at $\alpha = 0$; a kick of size ζ in α moves $\bar W$ by ζW_h, and the restoring gradient on W_h is then $\zeta\,(\zeta W_h\Sigma)$, whose sign does not depend on the sign of ζ. The coupling $\alpha\langle W_h, V\rangle$ between the two tangent families enters the gate's equation at first order in α, which is the output-power term above, and the head's equation at order α^2; while the gate is of order one the latter is comparable to the head term when $\alpha^2(mS - \mathrm{tr}(W_h\Sigma W_h^\top))/(1+\|W_h\|_F^2)$ is comparable to the eigenvalues of Σ, it enters with the opposite sign, and it dominates until the gate has closed. The head's norm is therefore stationary or slightly growing while the gate decays, and shrinks only afterwards; the minimal model measures $+2.1\times10^{-9}$ per step for $\ln\|W_h\|_F$ at $|\alpha| \approx 0.2$ against $+1.9\times10^{-9}$ predicted with the coupling kept (Appendix B). An inexpressible head that the targets need has no zero-loss point with $\alpha = 0$, so the flow has no such point to reach; a frozen base removes W_μ from the coordinates, so the zero-loss set is $\{\alpha W_h = D\}$ for the fixed offset D that the base needs, which contains no point with $\alpha = 0$; along it the trace over the trained parameters, $\alpha^2 mS + \mathrm{tr}(D\Sigma D^\top)/\alpha^2$, is minimized at $\alpha^4 = \mathrm{tr}(D\Sigma D^\top)/(mS)$, where the gate stays.

B MINIMAL MODEL OF ABSORPTION

Model. States are $s \sim \mathcal{N}(0, I_{16})$. A trainable trunk ϕ (two hidden layers of 64 units, ReLU) computes $z = \phi(s)$; the base head is linear, $\mu = W_\mu z + b_\mu \in \mathbb{R}^4$; the shaping head is a zero-initialized linear map of z; the gate is initialized as $\alpha \sim U(-0.01, 0.01)$; the action is $a = \mu + \alpha h$ with no squashing, so that the merge identity of §2.3 is exact. The critic is exact and fixed, $Q(s, a) = -\frac{1}{2}\|a - g(s)\|^2$, where g is a fixed random two-layer tanh network with 32 hidden units and outputs scaled to unit root mean square, and the loss is $\mathcal{L} = \frac{1}{2}\mathbb{E}\|a - g(s)\|^2$, minimized over every trainable parameter by Adam with learning rate 3×10^{-4}, the optimizer of the agents. The seven configurations of Table 2 are the first five arms of §2.3 plus two MLP-base controls, an MLP head and a linear head on a one-hidden-layer base head of 32 units; the table's last row repeats the shared arm under plain SGD. Gradient noise is the controlled variable: N0 uses one fixed batch of 4096 states, so the gradient is exact and training is deterministic (two runs of the same seed agree bitwise); N1 draws a fresh batch of 64 states at every step; N2 adds Gaussian noise of standard deviation 0.3 to the target $g(s)$ at every step, a stand-in for a noisy critic. Every configuration runs for 100K steps with 3 seeds; $\|\delta\|$, $\|h\|$, and the removal cost $\mathcal{L}(\mu) - \mathcal{L}(\mu + \alpha h)$ are logged every 500 steps on a fixed evaluation set of 2048 states; the residual r uses the last 5% of steps over the peak, with completion at $r \leq 0.10$ as in the paper.

Table 2: **Minimal model.** Terminal residual r (median over 3 seeds), number of seeds completing ($r \leq 0.10$), and the fraction of its peak norm that the head retains at the end ($\|h\|_{\text{end}}/\|h\|_{\text{peak}}$, median), for each configuration under three noise levels; Adam in every row but the last, which uses plain SGD (learning rate 10^{-2}).

	N0: full batch			N1: minibatch			N2: minibatch + target noise		
Arm	r	done	h kept	r	done	h kept	r	done	h kept
Shared, linear head	0.997	0/3	0.84	0.695	0/3	0.96	0.0014	3/3	0.14
Detach	0.996	0/3	0.85	0.687	0/3	0.98	0.0012	3/3	0.13
Independent trunk	0.999	0/3	0.66	0.996	0/3	0.80	0.979	0/3	0.99
Frozen base	1.000	0/3	0.64	0.993	0/3	0.51	0.993	0/3	0.51
MLP head, linear base	0.995	0/3	0.60	0.777	0/3	0.87	0.086	2/3	0.13
Linear head, MLP base	0.999	0/3	0.91	0.875	0/3	0.99	0.983	0/3	0.99
MLP head, MLP base	0.996	0/3	0.84	0.866	0/3	0.70	0.120	1/3	0.22
Shared, linear head (SGD)	non-engaged at every noise level: peak amplitude below 6×10^{-4}								

What the model shows. Without gradient noise nothing returns (Figure 5): under N0 every Adam configuration ends with $r \geq 0.995$, the shared head included, whose α rises throughout the run, to 0.43 at 100K. With minibatch noise the shared head turns over: α peaks near 0.28 in the first half of the run and declines from there, to 0.20 at 100K, with $r = 0.70$ still falling at the budget and the head retaining 96% of its norm. With target noise added it returns within the budget, $r = 0.0014$ and completion at about 58K steps in every seed, and there the head's norm falls as well, to 14% of its peak. The drift therefore acts on α first, as the rate ratio of §3.2 predicts, and reaches W_h once the noise is strong; the trained agents, whose $|\alpha|$ falls by 98% while $\|h\|$ keeps 20% to 94% (§4.2), sit between the two noisy regimes. Detaching the head from the trunk changes nothing at any noise level (differences in r below 0.01). The independent-trunk and frozen-base heads never return ($r \geq 0.97$ and ≥ 0.99), and removing the head from the frozen base costs 1.65 in loss against at most 0.03 for the shared head.

What decides absorption is exact expressibility. The linear head on the one-hidden-layer base is the decisive control: the base can approximate a linear function of z but cannot reproduce $\mu + \alpha h$ exactly, and its head keeps its amplitude at every noise level (r of 0.88 to 1.00, head norm retained at 91% to 99%), even under the target noise that closes the linear-on-linear gate in every seed. The MLP head on the MLP base behaves the same way at N1 ($r = 0.87$) and only partly returns at N2 ($r = 0.12$, one seed of three); a sum of two one-hidden-layer networks of the same width is not a network of that width, so no exact line exists there either. The MLP head on the linear base is not absorbed under minibatch noise ($r = 0.78$) but does return under target noise ($r = 0.086$, two seeds

of three) with its norm reduced to 13% of peak and a removal cost of 0.0004: the branch collapses while the base learns the task on its own, the stochastic collapse of a redundant branch (Chen et al., 2023) rather than absorption along a flat line. The diagnostic of §4.3 holds in that case too: the residual is small and the removal free, though the mechanism differs; the agents' MLP heads do not collapse at three times the budget (Appendix D).

Predictions and the optimizer. The no-noise, detach, independent-trunk, and frozen-base predictions hold. The shared head turns over under minibatch noise and its gate closes under target noise, as reported above, and the two MLP-base controls are not absorbed; they isolate exact expressibility as the condition, which the agent test of Appendix C confirms. Under plain SGD at learning rate 10^{-2} the head never engages: the peak amplitude stays below 6×10^{-4} against 0.15 to 0.62 under Adam, while the base alone drives the loss from 2.0 to 0.01. The zero-initialized head sits at a point where the gradient with respect to W_h carries a factor α and the gradient with respect to α a factor $\|h\|$, both near zero; Adam's per-parameter normalization turns those small but consistent gradients into steps of the size of the learning rate, and plain SGD does not.

Drift check. To test Proposition 3 directly we start on the line. A linear-head actor is trained by Adam without noise on 512 fixed states until the training loss is below 10^{-6}, which is reachable because the model has more parameters than constraints, leaving $\alpha \approx -0.21$ and $\|W_h\|_F \approx 3.4$; from there training continues with plain stochastic gradient descent on the full batch, label noise of standard deviation σ added to the target at every step, the trunk frozen so that Σ is fixed, three seeds. At step size 10^{-2} over 200K steps the measured rate of $\ln|\alpha|$ is -8.2×10^{-8} per step at $\sigma = 0.3$ and -9.9×10^{-9} at $\sigma = 0.1$, against -7.5×10^{-8} and -8.3×10^{-9} from the projected flow of Appendix A.3, output-power term included, evaluated along the trajectory, ratios of 1.09 and 1.19, and the two rates differ by a factor of 8.25 for a ninefold change in σ^2. Over this budget the gate moves by at most 2%, which fixes its initial rate; the head's norm does not shrink but grows slightly, at $+2.1 \times 10^{-9}$ per step against $+1.9 \times 10^{-9}$ predicted with the coupling of Appendix A.3 kept. With $\sigma = 0$ nothing moves, and the MLP head, the frozen base, and the linear head on an MLP base keep at least 96% of $|\alpha|$ under the same noise, the last of them growing by 25%.

A run from the same starting points raises the step size to 0.05 and the budget to 3M steps so that the drift outruns the gate's own fluctuations. There the gate decays exponentially: $\ln|\alpha|$ is linear in the step with $R^2 = 0.996$ at $\sigma = 0.3$ and 0.990 at $\sigma = 0.45$, reaching $|\alpha| < 0.02$ after 0.80M and 0.36M steps, at rates of -2.8×10^{-6} and -6.2×10^{-6} per step against -1.8×10^{-6} and -4.1×10^{-6} from the flow (ratios 1.59 and 1.57; at step size 0.03 the ratio falls to 1.26, so the excess is the finite-step correction: a discrete-time correction to the flow, which enlarges the stationary variance of each Hessian mode by $1/(1 - \eta\lambda_i/2)$, reproduces the measured ratio of each of the nine runs to within 1%), and the two rates differ by a factor of 2.24 for a σ^2 ratio of 2.25. Once the gate has closed the head begins to shrink, at -1.8×10^{-8} and -2.7×10^{-8} per step (R^2 of 0.99 and 0.97), 1.03 and 0.97 times the rate of the flow; the gate therefore decays 150 and 228 times faster than the head against predicted ratios of 101 and 146, and the head still holds 96% and 92% of its norm at 3M steps. Under the same step size and noise the MLP head keeps 65% to 70% of its gate with no exponential form ($R^2 \leq 0.92$), and the linear head on an MLP base grows its gate by a factor of 3.5 and stays there.

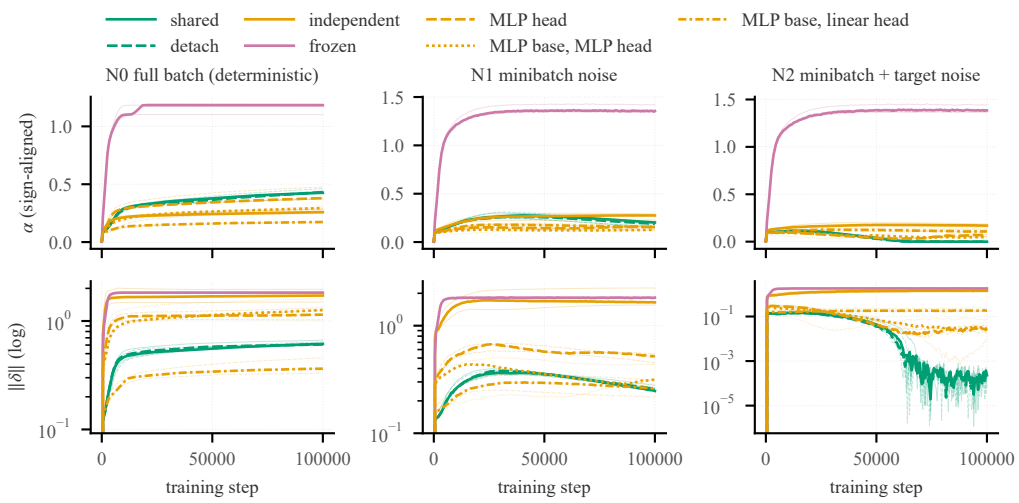


Figure 5: **Without gradient noise nothing returns.** Minimal model under Adam. Sign-aligned α (top) and $\|\delta\|$ on a log scale (bottom) over 100K steps under the three noise levels; color by whether the base head can reproduce the offset, green where it can, orange where it cannot, pink where the base is frozen, with the line style separating arms inside a color; thin lines are seeds and thick lines the median. Under N0 nothing returns; under N1 the shared and detached heads turn over; under N2 they return to zero, the MLP head on the linear base collapses, and the linear head on the MLP base stays.

C A NONLINEAR BASE HEAD, WITH AND WITHOUT A LINEAR PATH

The minimal model separates exact reproduction from approximation with its decisive control, the linear head on the MLP base, which keeps its amplitude at every noise level (Figure 5). This appendix runs that separation on the agents, on the mechanism suite and the protocol of §4: 6 environments, TD3 and SAC, 5 seeds, one million steps, 120 runs. In both arms the base head is a one-hidden-layer network of 64 units on the shared representation and the shaping head stays linear, so the offset is linear and the base head is not; in the second arm a zero-initialized linear map of z is added to the base head's output. The two arms are the same function at initialization, seed by seed, and differ only in whether the base head's own parameters can reproduce $\mu + \alpha h$ exactly. With the linear path they can, by $W_{\text{skip}} \mapsto W_{\text{skip}} + \alpha W_h$ and $b_{\text{skip}} \mapsto b_{\text{skip}} + \alpha b_h$; without it they cannot, since the function class of a one-hidden-layer network is not closed under adding an affine map.

Outcome. Every cell of both arms engaged, so the two arms are comparable. Without the linear path the median residual over the twelve cells is 0.786 and the arc completes in none of them. With it the median residual is 0.012 and the arc completes in nine cells. The residual is lower with the path than without it in twelve of twelve paired cells. Removal costs an interquartile mean of $+0.296$ [0.240, 0.360] without the path and -0.0002 [-0.0007, 0.0004] with it. Table 3 gives every cell. The means over the six environments quoted in §4.4 are the column means of Table 3: a residual of 0.60 under TD3 and 0.82 under SAC without the path, and a removal cost of $+0.38$ and $+0.35$ without the path.

Table 3: **A linear path lowers the residual in every cell and the removal cost to within 0.004 of zero.** Terminal residual r and removal cost per (environment, algorithm) cell, each the mean over 5 seeds; *base* is the one-hidden-layer base head and *+path* the same base head with a zero-initialized linear map added to its output. Cost is the normalized return of the shaped actor minus the base actor alone, as in §2.2; a value that rounds to zero is printed without a sign.

	Terminal residual r				Removal cost			
	TD3		SAC		TD3		SAC	
Environment	base	+path	base	+path	base	+path	base	+path
Hopper	0.476	0.005	0.902	0.183	+0.786	−0.004	+0.559	−0.001
HalfCheetah	0.765	0.006	0.954	0.008	+0.367	+0.003	+0.401	+0.002
Humanoid	0.915	0.081	0.765	0.016	+0.947	−0.002	+0.572	−0.001
finger/spin	0.173	0.003	0.556	0.351	+0.012	0.000	+0.004	−0.001
walker/run	0.452	0.003	0.956	0.008	+0.137	+0.002	+0.323	0.000
humanoid/walk	0.829	0.266	0.806	0.097	+0.007	−0.001	+0.218	−0.002

D THE MLP SHAPING HEAD AT THREE MILLION STEPS

Section 4.4 reports that a one-hidden-layer shaping head is not absorbed within one million steps at any of three widths; the alternative reading is that the trunk could reshape itself until the offset is linear in the representation, and one million steps may not be enough for it. This appendix runs the width-64 head three times as long on the mechanism suite: 6 environments, TD3 and SAC, 3 seeds, three million steps, 36 runs, everything else as in §4. A reproduction check ties these runs to §4.4: their residual read at one million steps, averaged per algorithm, must fall within 0.15 of the width-64 values reported there, and it does.

Outcome. At one million steps the mean residual over the six environments is 0.59 under TD3 and 0.84 under SAC, against 0.68 and 0.82, the width-64 values of the width sweep of §4.4, whose widths 16, 64 and 256 give 0.62, 0.68 and 0.58 under TD3 and 0.83, 0.82 and 0.74 under SAC. At three million steps the median residual over the twelve (environment, algorithm) cells is 0.73 and the arc completes in none. The median change in residual from one million to three million steps, within the same runs, is a drop of 0.02; in four of the six TD3 cells the residual rises. Removal costs an interquartile mean of $+0.91$ [0.86, 0.95] of normalized return. Table 4 gives every cell. Three times the budget leaves the nonlinear head unabsorbed in every cell; with Appendix C, which separates the same two readings for a linear offset, the budget reading is closed on both sides.

Table 4: **The MLP shaping head stays unabsorbed at three million steps.** Residual r of the same runs read at one million and at three million steps, and removal cost at three million, per (environment, algorithm) cell for the width-64 head, each the mean over 3 seeds. Cost is the normalized return of the shaped actor minus the base actor alone, as in §2.2.

	TD3			SAC		
Environment	r 1M	r 3M	cost 3M	r 1M	r 3M	cost 3M
Hopper	0.597	0.735	+0.820	0.798	0.721	+0.706
HalfCheetah	0.700	0.829	+1.221	0.858	0.782	+1.337
Humanoid	0.851	0.848	+0.790	0.952	0.916	+1.177
finger/spin	0.382	0.467	+0.702	0.783	0.672	+1.092
walker/run	0.382	0.437	+0.976	0.798	0.609	+0.948
humanoid/walk	0.644	0.385	+0.001	0.856	0.863	+0.008

E THREE MILLION STEPS FOR THE UNFINISHED TD3 ARCS

Under TD3 the arc completes within one million steps in seven of twenty environments (§4.3). Whether the other thirteen are short of budget cannot be read from a single budget, so this appendix runs six of them three times as long: manipulator/bring_ball, acrobot/swingup, finger/spin,

humanoid/walk, InvertedDoublePendulum and InvertedPendulum, the three highest residuals among the thirteen at one million steps and three from the middle of that ranking, 5 seeds, three million steps, 30 runs, every setting as in §4. Because the exploration and warm-up schedules are absolute step counts, a three-million-step run is a one-million-step run continued, and each run carries its own reading at one million steps as a paired control.

Outcome. Every environment engaged in five of five seeds, and the residual at three million steps is below the residual of the same runs at one million steps in six of six environments: 0.827 to 0.732, 0.488 to 0.215, 0.321 to 0.190, 0.206 to 0.176, 0.490 to 0.056 and 0.294 to 0.225. On InvertedDoublePendulum the ratio falls below the completion threshold while a large offset remains in action space, and removing the head there costs $+0.49$ of normalized return. Table 5 gives every environment.

Table 5: **Three times the budget lowers the residual r in each of six unfinished TD3 arcs.** Residual r and removal cost of the same runs read at one million and at three million steps, and the run's peak amplitude $\|\delta\|$ (pre-activation, §2.2), each the mean over 5 seeds; cost is the normalized return of the shaped actor minus the base actor alone.

	Residual r		Removal cost		Peak
Environment	1M	3M	1M	3M	$\|\delta\|$
manipulator/bring_ball	0.827	0.732	+0.935	+0.056	20395
acrobot/swingup	0.488	0.215	+0.033	−0.038	22.1
finger/spin	0.321	0.190	+0.155	+0.009	160
humanoid/walk	0.206	0.176	+0.045	+0.004	285
InvertedDoublePendulum	0.490	0.056	+0.950	+0.489	21.1
InvertedPendulum	0.294	0.225	+0.376	+0.487	5.7

The offset in action space. The residual is a ratio to the run's own peak amplitude, taken before the squashing; the cost of removal depends on the offset that remains in action space. On the two pendulum tasks the peak is 21 and 5.7 pre-activation units on a one-dimensional action, so a residual of 0.056 on InvertedDoublePendulum still leaves a mean offset of 0.85 before the squashing and of 0.18 in an action bounded by one after it, and the base actor without it keeps half the normalized return; on InvertedPendulum a residual of 0.225 still leaves a mean offset of 1.2 before the squashing, and removal costs $+0.49$ there too. On humanoid/walk the mismatch runs the other way: a pre-activation residual of 0.18 sits on a saturated squashing, the offset applied in action space is 7% of its peak, and removal costs $+0.004$. Read in action space, as the mean over states of $\|f(\mu(s)+\delta(s))-f(\mu(s))\|$, the residual tracks the six removal costs with a rank correlation of 0.71. The proposition of §3.2 concerns the pre-activation line and the residual is measured there, which is where to read whether the gate has closed; the cost of removal is paid in action space, and where the output map saturates at the peak, predicting that cost requires reading the offset there.

F Batch size and the closing time of the gate

Proposition 3 describes the return of the gate as a drift under gradient noise at a rate proportional to the noise level, and Appendix B confirms the proportionality in the idealization. On the agents the noise level is set by the training batch size B, since the variance of a minibatch gradient scales as $1/B$, so the proposition predicts that the gate closes sooner at smaller B. This appendix varies B and nothing else: the shared arm under TD3 and SAC on HalfCheetah, Humanoid and walker/run, the three environments of the mechanism suite whose arc completes within one million steps in at least four of five seeds under both critics at the default $B = 256$, run again at $B \in \{64, 256, 1024\}$ with 5 seeds for one million steps, 90 runs, every other hyperparameter as in §4; the batch of evaluation states behind the amplitude is fixed and does not vary with B.

Measures. Per run, the closing time T is the number of steps from the peak of the amplitude to the first evaluation at which the trailing mean over 20 evaluations is at or below 10% of the peak, censored at one million steps. A cell is one (environment, critic, B) and takes the median over seeds; a pair is one (environment, critic) across the three batch sizes.

Outcome. Every cell engaged in five of five seeds. The arc completes earlier at $B = 64$ than at $B = 1024$ in five of six pairs; the exception is walker/run under TD3, 535K against 500K steps. Table 6 gives every cell and Figure 6 the median curves.

Table 6: **The gate across batch sizes.** Closing time T from the peak of the amplitude (K steps), median over 5 seeds; definition in the text.

Environment	Critic	B	T
HalfCheetah	TD3	64	520
		256	490
		1024	705
HalfCheetah	SAC	64	330
		256	355
		1024	440
Humanoid	TD3	64	195
		256	120
		1024	205
Humanoid	SAC	64	110
		256	110
		1024	135
walker/run	TD3	64	535
		256	730
		1024	500
walker/run	SAC	64	135
		256	270
		1024	345

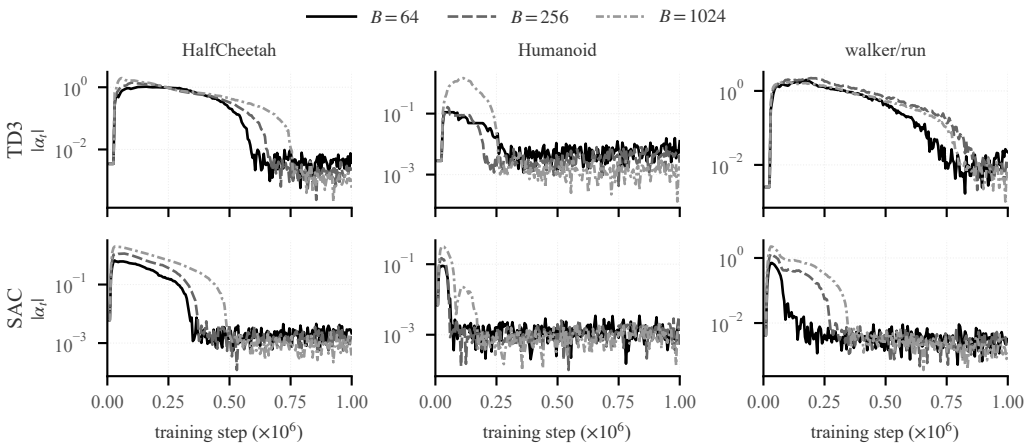


Figure 6: **The gate rises and falls at every batch size.** $|\alpha_t|$ during training, median over 5 seeds, log scale; rows labelled TD3 and SAC; black $B = 64$, grey dashed $B = 256$, light grey dash-dot $B = 1024$. The closing time T of Table 6 is read from the amplitude rather than from these curves.

G THE SAME HEAD COMPOSED MULTIPLICATIVELY

Among the seven arms of §2.3, an offset linear in the representation the base reads is inexpressible only where the base head is changed (Appendix C); here the base head is that of the shared arm and the linear head enters the action multiplicatively, so that the offset is inexpressible while the head stays linear. The gain arm acts with $f\big(\mu(s) \odot (1 + \alpha\, g(s))\big)$, $g(s) = W_g z(s)$ with W_g zero-initialized and no bias, so that the shaped actor coincides with the base actor at initialization exactly as in the shared arm and the head has the same parameter count less one bias vector. The composite carries the

term $\alpha\,(W_\mu z) \odot (W_g z)$, quadratic in z wherever $\alpha W_g \neq 0$ and the base head reads z in a coordinate the gain reads, so no base-head parameters reproduce it: the offset is inexpressible although the head is linear, and the condition of §2.3 predicts that the gate stays open. The offset is $\delta = \alpha\,\mu \odot g$, taken before f, and the amplitude, residual and removal cost are those of §2.2; one more quantity is logged, the state-dependence index $\mathrm{sdi} = \mathbb{E}_s\|g(s) - \bar{g}\|/\mathbb{E}_s\|g(s)\|$ over the evaluation batch, which is near zero only if the head has degenerated into a constant gain, the one case in which the composite becomes expressible again. The arm runs on the mechanism suite, six environments under TD3 and SAC with seeds 0 to 2, 36 runs of one million steps, against a shared-arm control of the same 12 cells (36 runs) trained alongside it on the same hardware with the same seeds.

Outcome. A cell is one (environment, algorithm); the gain arm is summarized by its seed median and the control by its seed maximum, so that a cell counts only if the gain arm exceeds every control seed. Every cell engages, with the peak amplitude above 10^{-3} in at least two seeds (12 of 12 cells); the gain residual exceeds the control's (11 of 12) and the completion threshold 0.10 (11 of 12); and the gain removal cost exceeds the control's by more than 0.1 in four cells (HalfCheetah and walker/run under TD3 and under SAC). Table 7 gives every cell.

Table 7: **A linear head composed multiplicatively stays.** Terminal residual r, removal cost and state-dependence index of the gain arm against the paired shared arm on the mechanism suite; gain cells are the median over three seeds (the index its mean), control cells the seed maximum; cost in normalized return at 1M steps; a value that rounds to zero is printed without a sign.

	Terminal residual r				Removal cost				sdi	
	TD3		SAC		TD3		SAC		TD3	SAC
Environment	gain	shared	gain	shared	gain	shared	gain	shared	gain	gain
Hopper	0.368	0.226	0.837	0.026	+0.018	+0.011	−0.030	+0.001	1.09	0.58
HalfCheetah	0.532	0.007	0.949	0.002	+0.651	+0.002	+0.298	+0.002	0.94	0.36
Humanoid	0.819	0.020	0.926	0.013	+0.005	+0.007	−0.014	+0.011	0.67	0.09
finger/spin	0.211	0.420	0.779	0.324	+0.099	+0.232	+0.010	+0.014	0.74	0.59
walker/run	0.099	0.010	0.947	0.005	+0.584	+0.003	+0.148	+0.001	1.02	0.30
humanoid/walk	0.704	0.216	0.881	0.237	+0.008	0.000	−0.001	+0.043	0.63	0.34

Reading. In eleven of the twelve cells the gate does not close: the gain residual is above 0.70 in eight cells and above the control's seed maximum in eleven, and the index stays between 0.30 and 1.09 in every cell but Humanoid under SAC (0.09), so the head kept a state-dependent gain rather than collapsing to a constant one. The one cell at the threshold, walker/run under TD3, keeps ten times the control's residual (0.099 against 0.010). Removing the gain head costs $+0.65$ and $+0.58$ of normalized return on HalfCheetah and walker/run under TD3 and $+0.30$ and $+0.15$ under SAC, the four cells in which the gain head carries a large part of the shaped action; in the other cells the gate stays open at a small amplitude and removal moves the return by at most 0.03, since the offset there moves the action little (Appendix E), except on finger/spin under TD3, $+0.10$. The one cell in which the gain residual is below the control's, finger/spin under TD3, is one where the shared arc itself does not complete within the budget (seed maximum $r = 0.42$).

H Evaluation protocol and residual conventions

Evaluation. The twenty-environment configurations are trained for 1M environment steps with 5 seeds on each of the 20 continuous-control environments, ten from MuJoCo v5 and ten from the DeepMind Control Suite. Evaluation is paired and runs every 5,000 steps: the same checkpoint is rolled out twice on the same environment seeds, once with the shaping head active and once with it removed, so the removal cost of §2.2 is a difference between two returns measured at the same point of the same run rather than between two runs. A terminal score is the mean of the last $W = 20$ evaluations, that is, the last 100K steps of training. The same window is the numerator of the terminal residual in equation 2. The four checkpoints of Table 1 are fixed per run from the trajectory of its offset in action space alone, the mean over the evaluation states of $\|f(\mu(s) + \delta(s)) - f(\mu(s))\|$: *rising* is the first evaluation at which it reaches half of its peak, *peak* is the maximum, *post-peak* is the

first later evaluation at which it is back at or below half of the peak (midway between the peak and the end of training when there is none), and *final* is the end of training; the saved checkpoint nearest each is rolled out for 10 deterministic episodes with and without the head, and the interval is a 95% percentile bootstrap over the 30 runs of each algorithm on the mechanism suite.

Normalization. Per-environment scores are min-max normalized as $(\text{score} - \text{random})/(\text{best} - \text{random})$. The random anchor is the mean return of a uniform random policy (5 seeds, 100 episodes); the best anchor is, per environment, the maximum over evaluated methods of the seed-mean terminal return (last 20 evaluations), so a single run can score above 1; the anchors are fixed per environment and shared by every figure and table. Min-max normalization with two reference points per task follows the rliable protocol (Agarwal et al., 2021), and since every method shares the same anchors, normalization preserves the sign and order of every within-environment difference.

Aggregation. Aggregate scores are the interquartile mean (IQM) with 95% stratified bootstrap confidence intervals (Agarwal et al., 2021) over 50,000 replicates. Where a removal cost is reported as an IQM, it is the IQM of per-run paired differences, not a difference of two IQMs.

Residual conventions. We form the terminal residual per run by Equation 2, as the mean of the last 20 evaluations of $\|\delta_t\|$ over that run's own peak, and take an environment's residual to be the mean of r over its seeds. Three other conventions are defensible. Table 8 recomputes under each of them the two quantities that depend on the residual: the number of environments counted as completed at the fixed threshold $r \leq 0.10$, and the rank correlation between an environment's residual and the size of its removal cost. The completed count is 7, 8, 8 and 10 under TD3 and 13, 13, 14 and 13 under SAC across the four conventions, and the correlation stays between 0.565 and 0.671 with $p < 0.01$ in every case, so the correlation reported in §4.3 does not depend on the convention.

Table 8: **Residual conventions.** Each row is one way of forming an environment's terminal residual from the per-run amplitude trajectories. *Completed*: environments with $r \leq 0.10$, out of 20. *Spearman*: rank correlation between an environment's residual and the magnitude of its removal cost, with its p-value. The first row is the convention used in the paper, defined in equation 2; it counts no more environments as completed than any other row, and its correlation is the one reported in §4.3. The equal TD3 and SAC values in the second row are an exact coincidence: the two rankings differ, but their sums of squared rank differences are both 522.

	TD3		SAC	
Convention	completed	Spearman (p)	completed	Spearman (p)
Per-run r, mean over seeds	7	0.614 (0.0040)	13	0.636 (0.0026)
Seed-mean trajectory, last 20 evaluations over peak	8	0.608 (0.0045)	13	0.608 (0.0045)
Seed-mean trajectory, last point over peak	8	0.565 (0.0094)	14	0.571 (0.0085)
Per-run r, median over seeds	10	0.626 (0.0032)	13	0.671 (0.0012)

I PER-ENVIRONMENT RESULTS

Table 9 lists the terminal IQM of the shaped actors and the frozen arms in every environment, and Figure 7 the per-environment removal cost with its interval, pooled over the shaped TD3 and SAC actors (17 of 20 intervals contain zero for SAC alone, 15 for TD3 alone).

Table 9: **Per-environment terminal IQM of the shaped actors** (20 environments, 5 seeds, shaped evaluation, min-max normalized); $\pm$ is the half-width of the 95% stratified bootstrap CI. AS-TD3-Fr and AS-SAC-Fr are the frozen arms.

Environment	AS-TD3	AS-TD3-Fr	AS-SAC	AS-SAC-Fr	AS-PPO
InvertedPendulum	1.009 ± 0.006	0.912 ± 0.109	1.004 ± 0.024	1.000 ± 0.015	1.018 ± 0.030
InvertedDoublePendulum	1.001 ± 0.012	0.980 ± 0.032	1.005 ± 0.022	0.036 ± 0.003	0.763 ± 0.127
Reacher	1.004 ± 0.005	0.950 ± 0.007	1.020 ± 0.004	0.588 ± 0.183	0.969 ± 0.008
Swimmer	0.805 ± 0.122	0.594 ± 0.031	0.909 ± 0.144	0.649 ± 0.055	0.873 ± 0.397
Hopper	0.971 ± 0.097	0.050 ± 0.008	1.017 ± 0.139	0.961 ± 0.121	1.057 ± 0.214
HalfCheetah	0.995 ± 0.128	0.461 ± 0.053	1.023 ± 0.020	0.425 ± 0.073	0.568 ± 0.059
Walker2d	0.889 ± 0.100	0.083 ± 0.182	0.899 ± 0.045	0.310 ± 0.158	0.963 ± 0.347
Ant	1.034 ± 0.203	0.702 ± 0.181	0.780 ± 0.115	0.013 ± 0.001	0.353 ± 0.054
Humanoid	0.941 ± 0.014	0.089 ± 0.037	0.964 ± 0.127	-0.010 ± 0.000	0.192 ± 0.322
Pusher	0.984 ± 0.015	0.977 ± 0.012	1.019 ± 0.005	0.900 ± 0.013	0.901 ± 0.017
acrobot/swingup	0.127 ± 0.095	0.039 ± 0.087	0.284 ± 0.118	0.043 ± 0.071	1.101 ± 0.371
cartpole/swingup	0.992 ± 0.036	0.870 ± 0.305	1.006 ± 0.013	0.992 ± 0.030	0.964 ± 0.025
cheetah/run	1.058 ± 0.145	0.319 ± 0.265	0.942 ± 0.079	0.578 ± 0.065	0.618 ± 0.225
hopper/hop	0.857 ± 0.831	0.011 ± 0.007	0.837 ± 0.329	0.017 ± 0.180	0.019 ± 0.002
walker/run	0.895 ± 0.157	0.029 ± 0.056	1.037 ± 0.087	0.182 ± 0.034	0.103 ± 0.050
humanoid/walk	0.018 ± 0.440	0.007 ± 0.001	0.027 ± 0.374	0.007 ± 0.001	0.022 ± 0.001
swimmer/swimmer15	1.074 ± 0.275	0.151 ± 1.008	0.835 ± 0.439	0.502 ± 0.254	0.429 ± 0.181
finger/spin	1.104 ± 0.167	0.011 ± 0.230	0.924 ± 0.220	0.547 ± 0.155	0.018 ± 0.001
reacher/hard	0.998 ± 0.016	0.395 ± 0.161	1.004 ± 0.019	0.507 ± 0.261	0.504 ± 0.063
manipulator/bring_ball	-0.086 ± 0.268	-0.053 ± 0.262	0.110 ± 1.883	-0.132 ± 0.292	0.143 ± 0.673

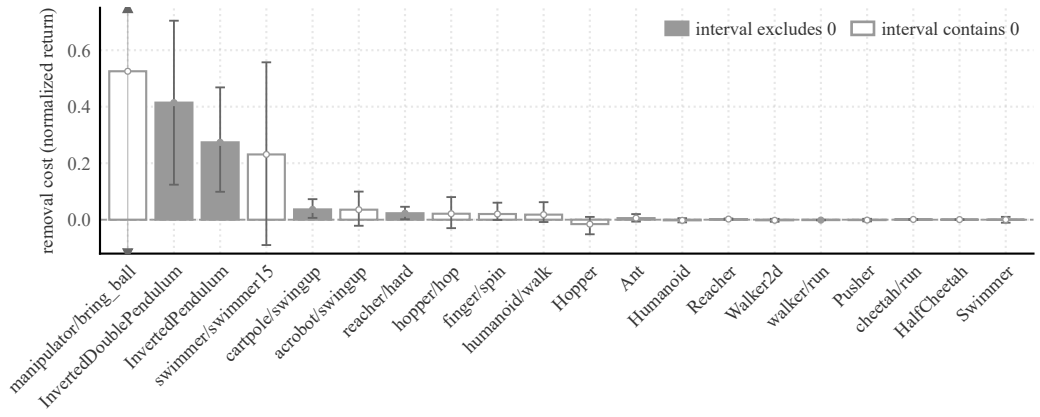


Figure 7: **The removal-cost interval contains zero in 15 of the 20 environments.** Per-environment removal cost (shaped minus base-only): the mean over the pooled runs with its 95% bootstrap CI. The shaped TD3 and SAC actors pooled, 5 seeds each, min-max normalized, sorted by magnitude. The interval of manipulator/bring_ball, $[-0.20, +1.58]$, is clipped to the axis.

J AMPLITUDE DECOMPOSITION

The amplitude factorizes as $\|\delta\| = |\alpha| \cdot \|h\|$, and Figure 8 plots the two factors separately for the six environments of the mechanism suite under the default configuration. Across the twelve (environment, algorithm) cells the median decline of $|\alpha|$ from its own peak is 98.14%. The head's output norm does not return to zero: the terminal $\|h\|$ lies between 20% and 94% of its own peak, between 20% and 88% under TD3 and between 51% and 94% under SAC, and above half of its peak in ten of the twelve cells, the two below half being Hopper (0.20) and walker/run (0.43) under TD3. The two factors are therefore not symmetric, the gate closing almost completely in ten of the twelve cells while the head keeps most of its scale, which is the asymmetry described in §3.2.

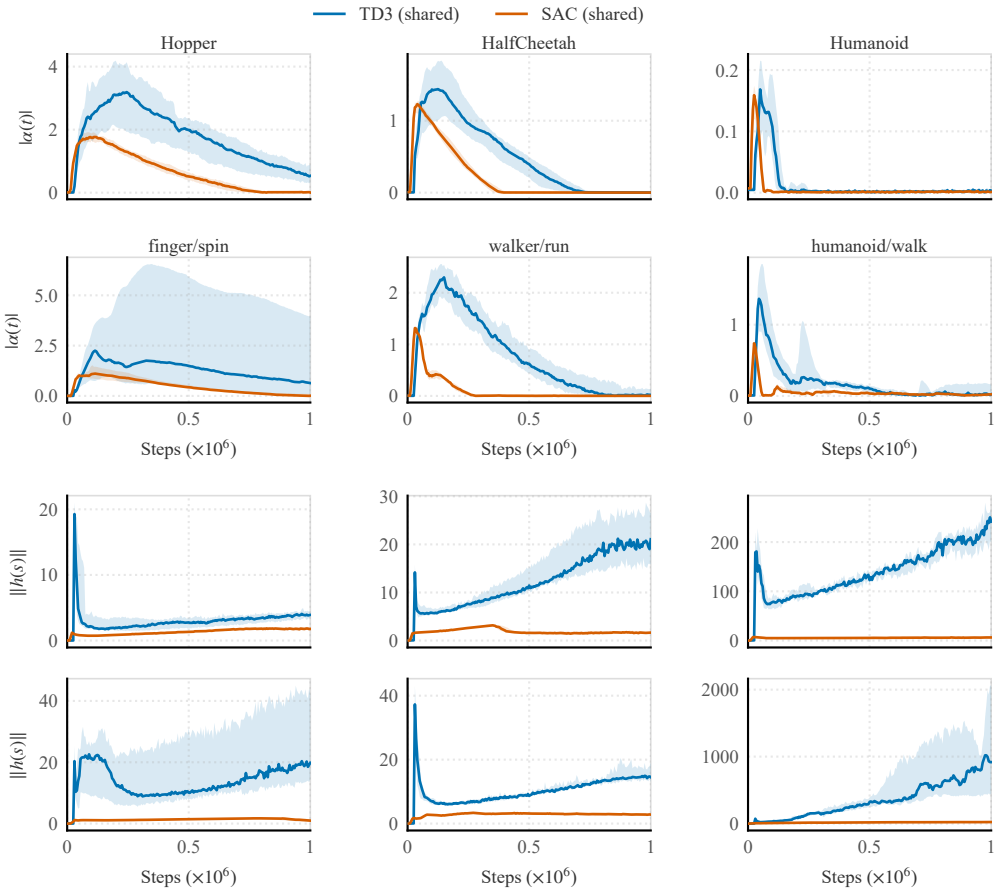


Figure 8: **Amplitude decomposition: the fall of the amplitude comes from the gate** (6 environments, 5 seeds, default configuration). Upper block: $|\alpha|$ over training. Lower block: $\|h(s)\|$ over training. TD3 in blue, SAC in orange; lines are the interquartile mean over seeds (median of 2,000 bootstrap resamples) and shading is the 95% stratified bootstrap CI.

K UPDATE-TO-DATA RATIO

Section 4.2 compares the twin critic with an ensemble critic (Chen et al., 2021) on the four environments of this experiment, which varies the update-to-data ratio inside the ensemble critic, with $N = 10$ networks and $M = 2$ sampled per update. Table 10 gives the per-environment and aggregate IQM for all four configurations at each ratio; within the batch shared by SAC-REDQ and its shaped variant, the shaped actor's aggregate IQM is above its host's at every ratio, by 0.029, 0.038 and 0.064.

The gate under the two critics. Table 11 compares, on these four environments, the terminal-to-peak ratio of $|\alpha|$ under the twin critic of the main runs and under the ensemble critic at an

Table 10: **Update-to-data scaling, per-environment IQM** (4 environments, 5 seeds, shaped evaluation). Min-max normalized; ± half-width of the 95% stratified bootstrap CI. Last row of each group: aggregate IQM. All four configurations use the same ensemble critic ($N = 10$, $M = 2$). The TD3-REDQ column is a separate run batch; the other three columns share one batch.

Environment	TD3-REDQ	AS-TD3-REDQ	SAC-REDQ	AS-SAC-REDQ
UTD = 1				
Hopper	0.983 ± 0.087	1.007 ± 0.083	0.900 ± 0.234	0.827 ± 0.146
HalfCheetah	0.962 ± 0.099	0.994 ± 0.039	1.058 ± 0.044	1.090 ± 0.035
Humanoid	0.963 ± 0.022	0.962 ± 0.021	1.017 ± 0.033	1.048 ± 0.064
walker/run	0.911 ± 0.178	0.814 ± 0.152	0.690 ± 0.215	0.928 ± 0.165
Aggregate	0.963 ± 0.037	0.962 ± 0.020	0.970 ± 0.058	0.999 ± 0.052
UTD = 5				
Hopper	0.975 ± 0.048	1.003 ± 0.163	0.842 ± 0.203	0.814 ± 0.205
HalfCheetah	1.067 ± 0.056	1.072 ± 0.097	1.188 ± 0.124	1.219 ± 0.073
Humanoid	0.946 ± 0.026	0.952 ± 0.068	1.027 ± 0.037	1.054 ± 0.076
walker/run	0.913 ± 0.127	0.911 ± 0.142	0.865 ± 0.034	0.872 ± 0.072
Aggregate	0.976 ± 0.025	0.978 ± 0.052	0.966 ± 0.036	1.004 ± 0.051
UTD = 20				
Hopper	0.856 ± 0.254	1.049 ± 0.128	0.639 ± 0.153	0.733 ± 0.242
HalfCheetah	1.095 ± 0.079	1.040 ± 0.219	1.114 ± 0.172	1.202 ± 0.188
Humanoid	0.945 ± 0.327	0.980 ± 0.022	1.010 ± 0.020	1.054 ± 0.046
walker/run	0.957 ± 0.074	0.927 ± 0.171	0.900 ± 0.126	0.952 ± 0.040
Aggregate	0.974 ± 0.052	0.995 ± 0.062	0.933 ± 0.067	0.997 ± 0.033

update-to-data ratio of 1, with 5 seeds each. Under TD3 the ensemble critic brings the one cell in which the twin critic leaves the ratio above 0.10 at 1M steps, Hopper, from 0.2112 to 0.0022, and the mean ratio over the four environments from 0.0637 to 0.0113; under SAC every ratio is at most 0.0102 with either critic.

Table 11: **Terminal-to-peak ratio of $|\alpha|$ under the twin and the ensemble critic** (4 environments, 5 seeds, shaped actors). Per environment, the mean over seeds of $|\alpha_t|$ is averaged over the last 20 evaluations and divided by its maximum over training; *twin* is the critic of the main runs and *ensemble* the critic of Table 10 ($N = 10$, $M = 2$) at an update-to-data ratio of 1. The ensemble critic brings the one TD3 ratio that the twin critic leaves above 0.10, Hopper, to 0.0022.

	TD3		SAC	
Environment	twin	ensemble	twin	ensemble
Hopper	0.2112	0.0022	0.0087	0.0044
HalfCheetah	0.0016	0.0087	0.0015	0.0019
Humanoid	0.0185	0.0098	0.0102	0.0076
walker/run	0.0235	0.0246	0.0019	0.0038
Mean	0.0637	0.0113	0.0056	0.0044

L SENSITIVITY TO ε

The initial gate $\alpha \sim U(-\varepsilon, \varepsilon)$ is the only constant the method adds; since the head starts at zero, it sets not the offset at initialization, which is zero for every ε, but the learning signal the head receives and the scale at which the head's output first enters the action. At $\varepsilon = 0$ the offset is identically zero and stays there: the gradient of the actor loss with respect to (W_h, b_h) carries a factor α (§3.2), so at $\alpha = 0$ the shaping head receives no learning signal. Khemais (2026) shows that a zero gate over a zero-initialized output projection is an exact saddle point that gradient descent cannot escape,

which is why the gate is initialized away from zero. Figure 9 compares $\varepsilon \in \{0.01, 0.1\}$ on four environments under both algorithms. A tenfold larger ε raises the terminal return in four of the eight panels and lowers it in four, by at most 11.4% (last 20 evaluations, mean over seeds), so within this range the one constant the method adds needs no tuning.

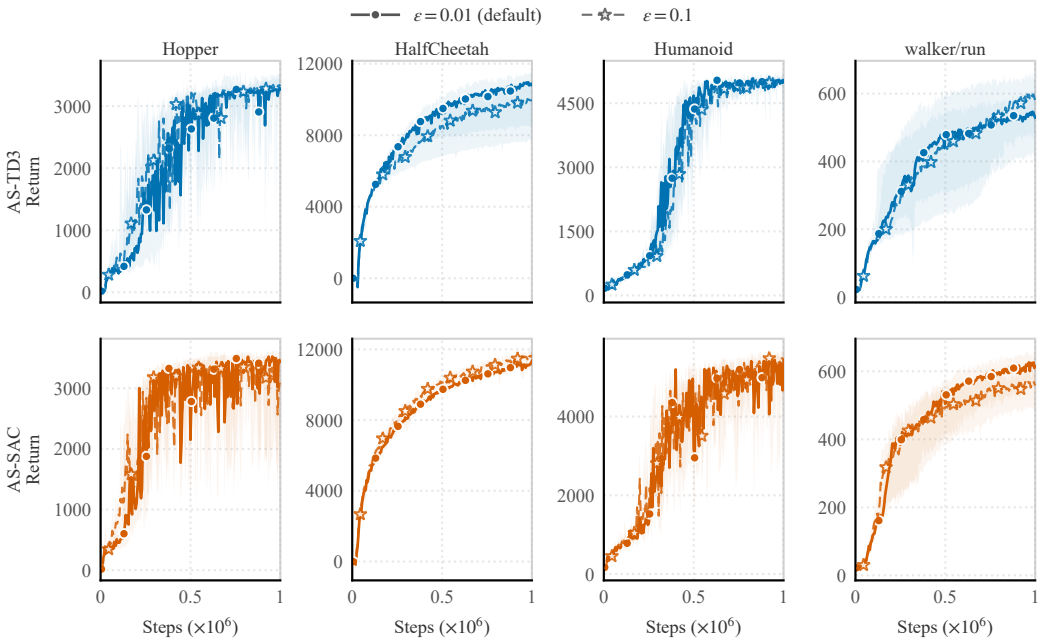


Figure 9: **The one constant the method adds needs no tuning across a tenfold range of** ε (4 environments, 5 seeds, shaped evaluation, raw return). Rows: shaped TD3 (top) and shaped SAC (bottom). $\varepsilon = 0.01$ (solid, circles): the value used everywhere else in the paper. $\varepsilon = 0.1$ (dashed, stars): a larger initial gate. Lines are the interquartile mean over seeds (median of 2,000 bootstrap resamples) and shading is the 95% stratified bootstrap CI.

M ATARI

The three interquartile means of §4.5, 0.183 [0.157, 0.216] for the base actor alone, 0.170 for the shaped actor and 0.154 for the unshaped host, and the difference of 0.013 [0.003, 0.023] by which the base actor alone exceeds the shaped actor (a difference of interquartile means, resampled with matched seeds), all come from one evaluation pass: at the checkpoint saved at 1M steps, each policy acts deterministically (argmax) for 10 episodes on matched seeds, and the interquartile mean is taken over the 130 seed-game pairs, with a game-stratified, seed-matched bootstrap for the intervals.

Table 12 gives the amplitude ladder of the seven games in which the gate engages, reconstructed from a median of 14 saved checkpoints per run: three complete the arc within 1M steps (Seaquest, Alien, Boxing; r from 0.016 to 0.057) and four are still active at 1M (BattleZone, ChopperCommand, Pong, Kangaroo; r from 0.34 to 1.0). In the other 19 games, Amidar, Assault, Asterix, BankHeist, Breakout, CrazyClimber, DemonAttack, Freeway, Frostbite, Gopher, Hero, Jamesbond, Krull, KungFuMaster, MsPacman, PrivateEye, Qbert, RoadRunner, UpNDown, the peak of $\|\delta_t\|$ never reaches 10^{-3} (median peak 3.3×10^{-5}), so no arc exists there to complete.

Table 12: **Atari amplitude ladder for the seven games in which the gate engages.** Seed-averaged $\|\delta_t\|$, the expected probability-space L2 distance between the shaped and the base policy on a fixed buffer of 512 frames per run, at the saved checkpoint nearest each share of the 1M-step budget (5 seeds per game); cells below 10^{-4} print as <0.0001. r is the terminal amplitude divided by the peak of the full ladder of saved checkpoints (median 14 per run), computed before rounding; the arc is completed when $r \leq 0.10$.

	$\|\delta_t\|$ at the checkpoint nearest a share of the 1M-step budget							
Game	5%	10%	25%	50%	75%	100%	r	Arc
Seaquest	0.0003	0.0035	0.0007	0.0002	<0.0001	0.0001	0.016	completed
Alien	0.0002	0.0019	<0.0001	<0.0001	<0.0001	0.0001	0.030	completed
Boxing	0.0001	0.0011	<0.0001	<0.0001	<0.0001	0.0001	0.057	completed
BattleZone	0.0003	0.0006	0.0149	0.0254	0.0206	0.0086	0.34	active
ChopperCommand	0.0002	0.0018	0.0477	0.0349	0.0328	0.0291	0.49	active
Pong	0.0005	0.0008	0.0176	0.0334	0.0882	0.0838	0.95	active
Kangaroo	0.0002	0.0014	0.0042	0.0010	0.0020	0.0099	1.00	active

N EXTENDED RELATED WORK

This appendix gives the longer form of §5.

Reward shaping and the action channel. Potential-based shaping characterizes the reward modifications that leave the optimal policy unchanged and can therefore be removed at deployment (Ng et al., 1999); the characterization is independent of the learner. The action channel has no invariance of that kind, since an offset on the action changes the state distribution, so the same question has to be answered by the training dynamics rather than by the MDP. Our answer is a condition for absorption and the mechanism behind it. Guidance can also enter through the replay buffer: injecting transitions from a model-predictive controller steers TD3 and SAC toward a designer-preferred gait while the reward stays unchanged (Xing et al., 2026); that guidance lives in the data, so nothing remains in the policy to remove. The phrase action space shaping has been used for reducing a discrete action space before training (Kanervisto et al., 2020), and reshaping the action space during training has meant learning limits on the allowed actions (Pham et al., 2018); both are unrelated to the offset studied here.

Folding a training-time branch into a layer. Structural re-parameterization trains a multi-branch block and folds it algebraically into a single layer for inference (Ding et al., 2021; Hu et al., 2022; Wang et al., 2026a), and linear over-parameterization does the same for stacked linear layers (Guo et al., 2020; Arora et al., 2018; Du et al., 2018); the merge is performed by hand after training. Our linear head admits the same merge (§2.3); we ask whether training itself arrives at the merged solution, and find that it does only under the conditions of §4.4.

Learnable residual scales. In ReZero (Bachlechner et al., 2021) and LayerScale (Touvron et al., 2021) a residual branch is scaled by a learnable gate that starts at or near zero; the gate grows and stays because the branch is a permanent part of the model, whereas on an actor the same gate rises and then returns to zero, and the paper is about why.

Train with, deploy without. Removing a training-time component at deployment is established for the observation channel by privileged-sensor scaffolds (Hu et al., 2024). Residual distillation trains a plain network alongside its shortcut-equipped twin and removes the shortcuts by a schedule (Li et al., 2020); gated normalization is tapered out by a schedule (Kanavalau et al., 2026); action priors added to the sampled action decay by a schedule (Sood et al., 2023; 2025); jump-start and attenuated residual methods phase out a base or guide policy by a schedule (Uchendu et al., 2023; Trumpp et al., 2026; Bolychev et al., 2026), as does model-predictive control used as scaffolding for real-world dexterous manipulation (Küçüktabak et al., 2026). The same design has appeared for language agents, where skill, critique, or budget scaffolds are supplied during reinforcement learning and withdrawn by a curriculum before deployment (Lu et al., 2026; He et al., 2026; Shi et al., 2026; Sun et al.,

2026), and where an agent harness used as guidance during training is distilled into the weights by fine-tuning and removed at deployment (Ye et al., 2026). In every case a designer decides when the component leaves. The condition of §3 says where a schedule is needed and where it is not: an expressible branch on a trainable base drifts toward absorption on its own.

Residual and correction policies. Residual policy learning adds a learned correction to a fixed or slowly adapting base (Silver et al., 2018; Johannink et al., 2019; Trumpp et al., 2024; Feng et al., 2024; Wang et al., 2025; Xiao et al., 2025; Wang et al., 2026b; Ankile et al., 2025; Yu et al., 2026; Hirschowitz & Ramos, 2026; Muttaqien et al., 2026; Zhang et al., 2026; Wu et al., 2026; Ma et al., 2026); the correction is meant to stay and the base is usually frozen, the regime our frozen arm reproduces, in which the offset stays.

Gates that close and branches that fall silent. Learnable gates on auxiliary pathways into a frozen backbone tend to close at zero through a dead-gradient regime (Farazi et al., 2026); deep attention sublayers learn to mute themselves during pre-training (Saikumar & Varghese, 2025); sparsity penalties prune branches on purpose (Huang & Wang, 2018). Our α closes with no penalty and only after it has opened, and the three states of §2.2 keep those cases apart.

Internalization and removability. Tsilivis et al. (2026) define internalization as a network absorbing an explicit procedure into weights of the same hypothesis class; their chain-of-thought scaffold is removed by a curriculum, ours is absorbed by training, and we add the trainable-base condition, the flat-line mechanism, and a diagnostic read from the amplitude alone. Concurrent work on annealable soft priors finds that removability depends on the training trajectory (Yang et al., 2026), as our checkpoint interventions find within a run for a parametric branch.

Why a redundant parameter drifts to zero. Near a manifold of minimizers, gradient noise moves stochastic gradient descent toward flatter points (Blanc et al., 2020; Damian et al., 2021; Li et al., 2022), toward sparse solutions (Pesme et al., 2021), and onto simpler invariant subnetworks (Kunin et al., 2021; Chen et al., 2023); Section 3 shows that an expressible offset creates such a manifold, flat in α with its flattest point at $\alpha = 0$, and computes the drift on it.

O HYPERPARAMETERS

The twenty-environment runs of §4 use the settings listed here; the other experiment families deviate only as Table 14 states (Table 15 gives the Atari settings), apart from the head and base variants of §2.3 and Appendices C and G, described where they are used.

Table 13: **Hyperparameters for continuous control.** One value per algorithm unless a row names the experiments it applies to.

	TD3	SAC	PPO
Actor and critic hidden layers	2×256, ReLU		
Actor output	$\tanh$	$\tanh$-squashed Gaussian	Gaussian, state-independent $\log \sigma$
Optimizer	Adam		
Learning rate	3×10^{-4} (actor and critic)	actor, critic, temperature 3×10^{-4}	3×10^{-4}, linearly annealed
Discount γ	0.99		
Target update	Polyak $\tau = 0.005$, actor and targets every critic step	Polyak $\tau = 0.005$	n/a
Batch size	256	256	rollout 2048, minibatch 64, 10 epochs
Exploration	$\mathcal{N}(0, 0.1^2)$ action noise	temperature tuned to entropy $-\lvert\mathcal{A}\rvert$	entropy coef. 0
Target smoothing	noise 0.2, clipped at 0.5	n/a	n/a
Other	n/a	$\log \sigma \in [-20, 2]$	clip 0.2, GAE $\lambda = 0.95$, value coef. 0.5, grad. norm 0.5
Updates per environment step	1	1	n/a
Replay buffer	10^6	10^6	n/a
Random-action warm-up	25,000 steps	10,000 steps	n/a
Training steps	10^6; 3×10^6 in Appendices D and E		
Evaluation	every 5,000 steps, 10 episodes, deterministic policy		
Seeds	5; 3 in Appendices D and G		
Shaping head			
Head h	linear map of the actor's last hidden layer (256 units), weights and bias initialized to zero		
MLP head	one hidden layer (ReLU), output layer initialized to zero; hidden units 16, 64, 256 (§4.4); 64 (Appendix D)		
Gate α	one learnable scalar, $\alpha_0 \sim U(-\varepsilon, \varepsilon)$, $\varepsilon = 0.01$; trained by the actor optimizer		
Action	$a = f\big(\mu(s) + \alpha\, h(s)\big)$, f the host's output map ($\tanh$ and scaling for TD3 and SAC)		

Table 14: **Hyperparameters that differ between experiments.** Every other setting is as in Table 13.

Experiment	Setting	Value
Batch size (Appendix F)	batch size	64, 256, 1024
Update-to-data ratio (Appendix K)	updates per environment step	1, 5, 20; 10 critics, minimum over 2
Gate initialization (Appendix L)	ε	0.01, 0.1

Table 15: **Hyperparameters for Atari** (discrete SAC).

Setting	Value
Preprocessing	frame skip 4, 84×84 grayscale, 4 stacked frames, up to 30 no-ops
Episode end	life loss in training, game over in evaluation
Network	convolutions $32\times8\times8/4$, $64\times4\times4/2$, $64\times3\times3/1$, fully connected 512
Optimizer	Adam with numerical constant 10^{-4}; policy, Q and temperature learning rates 3×10^{-4}
Discount γ	0.99
Batch size	64
Replay buffer	10^5
Learning starts	20,000 steps
Update frequency	every 4 steps
Target network	hard update every 8,000 steps
Temperature	automatically tuned to $0.89\times$ the maximum entropy
Training steps	10^6
Evaluation	10 deterministic episodes per seed at the 1M-step checkpoint (Appendix M)
Seeds	5 per game
Shaping head	linear map of the shared network's output to the logits, weights and bias initialized to zero; gate as in Table 13