# DEEP Q-LEARNING WITHOUT A TARGET NETWORK

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A target network is the standard device for stabilizing deep Q-learning, at the price of storing a full copy of the online network and tuning its refresh period. We ask what a target actually needs to store and answer by lowering the storage level from a full copy, through partial copies, to none. A first-order analysis shows that a target decouples only the parameter directions it stores, and in deep networks negative feedback from a frozen copy is what keeps partial targets alive. Removing storage entirely exposes spurious solutions at which the loss vanishes while the values are wrong. A bounded multiplicative output scaling, which scales the network's own prediction by a factor slightly above one before regressing it onto the bootstrap, repairs this with a value-error bound. We call the resulting method Tempered Q-learning. In a stress test where the full copy diverges, it survives while storing nothing, matches or exceeds the copy's return once the copy's period is matched in environment time, and on ALE Breakout reaches the full copy's mean return at a fraction of the spread. The replacement is two lines of code, with no second network and no synchronization period to tune.

## 1 INTRODUCTION

Since Mnih et al. (2015), the standard recipe for stabilizing deep Q-learning has been to keep a frozen copy of the online network and bootstrap from it. The copy doubles parameter storage during training, which is not trivial once value networks get large, and it brings a synchronization period whose units, we will show, decide survival. It was therefore striking when Vincent et al. (2026) showed that freezing only the final linear layer, combined with an iterated-head correction, recovers most of the full target's benefit. What their paper does not say is *which* part of the copy matters, or why. We try to supply the explanation, and then push it as far as it goes, down to storing nothing at all.

Throughout, we work on what we call the range of storage levels: architectures that store some parameter subset $\omega$ for the bootstrap, from nothing (target-free TD), through per-layer scale-and-shift vectors (0.2 to 3% of parameters), through frozen last-$k$ layers, up to the full copy. The four contributions below answer one question in stages, what a target must store to stabilize bootstrapped learning, and the answer they reach is that nothing must be stored, since the separation between prediction and bootstrap that the copy provides can be built without one. We call the method that this question leaves standing Tempered Q-learning. What a target network has guarded since Mnih et al. (2015) is, in the end, one step of distance between prediction and bootstrap, and this paper shows that the distance can be built not from a second network but from a slight tempering of the bootstrap.

**Contributions.**

1. *A first-order theory of partial targets* (§3). We prove that a target decouples exactly the parameter directions it stores. At synchronization the vector field and the Jacobian along the unstored directions coincide with those of target-free TD, and on the classical counterexample this principle is what separates divergence from stability.
2. *The negative feedback* (§4). We identify why deep partial targets survive where the linear theory predicts failure. We prove that an online adapter's self-feedback against its frozen copy is negative without further assumptions, and verify that this is both the mechanism behind last-layer targets and the cause of their fragility at high update-to-data (UTD) ratio.



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.

3. *Degeneracy and repair at zero storage* (§5). We show that a predictor head without a target admits spurious solutions with vanishing loss and wrong values, identify the bound that excludes them, and propose the minimal form that satisfies it with no stored parameters, bounded multiplicative output scaling, which we call Tempered Q-learning.
4. *A stress test of Tempered Q-learning* (§6–7). We push the method to the setting where the full target copy diverges and transplant the stabilizing component of each recent target-reducing method into the same setting as a controlled comparison. Tempered Q-learning survives there (its constant member on every seed), matches or exceeds the copy once the copy's period is matched in environment time, and has no period to tune. The loss alone reverses a target-necessity verdict.

## 2 SETUP: TARGETS CLASSIFIED BY STORAGE LEVEL

We consider Q-learning that minimizes $\mathcal{L}(\theta) = \big(r+\gamma \max_{a'} \bar{Q}(s',a') - Q_\theta(s,a)\big)^2$ over transitions, with the bootstrap $\bar{Q}$ held constant under the stop-gradient operator $\mathrm{sg}[\cdot]$ (a semi-gradient update). The architectures below are listed from the most stored to the least; the first three differ only in $\bar{Q}$, and the last stores nothing and changes the online estimate instead:

- **Full target:** $\bar{Q} = Q_{\bar{\theta}}$ with $\bar{\theta} \leftarrow \theta$ every $T$ gradient steps.
- **Partial target:** only a subset $\omega$ of the parameters is copied and frozen ($\bar{\omega}$); the rest, $\theta_{\mathrm{sh}}$, is shared with the online network (a bar always marks a frozen copy). The stored subset is either per-layer scales and shifts $(g, b)$ in the style of FiLM (Perez et al., 2018), 0.2 to 3% of the parameters, or the last $k$ layers ($k$=1 is iS-QN (Vincent et al., 2026)).
- **Target-free:** $\bar{Q} = Q_\theta$ (the current parameters, with the gradient stopped through the bootstrap).
- **Tempered Q-learning (proposed):** target-free, plus a scaling head $h$ on the online output, $\hat{Q}(s,a) = \big[h(Q_\theta(s,\cdot))\big]_a$, trained toward $r + \gamma \max_{a'} \mathrm{sg}[Q_\theta(s',a')]$. In general $h$ may be any trainable predictor; the proposed method restricts it to the constant $1.1q$ or the bounded learned scaling (§5). Since multiplying the prediction by $c$ equals dividing the bootstrap by $c$, this is Q-learning with a tempered bootstrap. The greedy policy uses $\hat{Q}$.

### 2.1 TESTBEDS

Five testbeds, each isolating one mechanism. (i) The *linear two-state counterexample* (Tsitsiklis & Van Roy, 1997), the smallest case whose divergence is fully analyzable, where we prove why a copy stabilizes (Props. 2–3) and why the proposed scaling stabilizes without one (effective discount $\gamma/c$, §5). (ii) The *deep two-state counterexample*, the same MDP with a two-hidden-layer network and per-layer scale-and-shift, where depth overturns the linear analysis (§4), the mechanism of last-layer targets. (iii) Baird (1995)'s *star problem*, seven states collapsing onto one successor, the sharpest exposure of the spurious solutions of storage-free predictor heads (§5). (iv) A *chain MDP* with exact $Q^*$ (Appendix A), which checks that the proposed head learns correct values, not only stable ones. (v) A *deep Q-learning stress test*: CartPole, where a large value scale separates the constant from the learned scaling, and MinAtar, where UTD and loss (Huber vs. mean squared error, MSE) are pushed until the full copy dies (§6–7). Divergence is declared at $\max |Q| > 10^5$.

Four hypotheses derived from the theory were written down before the larger runs (Table A1, Appendix B); we report the two that held alongside the two that were refuted. Proofs not given in the text are in Appendix D.

## 3 FIRST-ORDER ANALYSIS: STABILIZATION CONFINED TO THE STORED DIRECTIONS

**Proposition 1** (Synchronization identity)**.** *At any synchronization instant ($\bar{\omega} = \omega$), the TD vector field of a shared-backbone partial target is identical to that of target-free TD.*

*Proof.* At synchronization the target and online functions coincide, so bootstrap values agree with the target-free ones, and semi-gradients (flowing only through the online branch) agree. □

One instant does not determine a flow; the separation is a property of the flow, not of an instant. We use the classical two-state instance (Tsitsiklis & Van Roy, 1997): states $s_1, s_2$ with features $\phi(s_1)=1, \phi(s_2)=2$, transitions $s_1 \to s_2$, $s_2 \to s_2$, zero reward, so $V^* \equiv 0$; online value $V(s) = g\,\theta\phi(s) + b$ with backbone $\theta$ and scale-and-shift $(g, b)$; the bootstrap value $\bar{V}(s) = \bar{g}\,\theta\phi(s) + \bar{b}$ uses the stored scale-and-shift $(\bar{g}, \bar{b})$ over the same live $\theta$. Expected updates with step size $\alpha$ sample the $s_1 \to s_2$ transition (uniform two-transition weighting shifts thresholds, not structure; Appendix D).

**Proposition 2** (Backbone block of the update Jacobian)**.** *At synchronization, the backbone-diagonal of the update Jacobian is $\partial\dot{\theta}/\partial\theta = \alpha g^2(2\gamma-1)$ for both target-free TD and the scale-and-shift target (expansive iff $2\gamma > 1$), versus $-\alpha g^2 < 0$ for the full target (contractive). The stabilization of a full target is precisely $\nabla_\theta(\text{bootstrap}) = 0$; a partial target zeroes that gradient only along $\omega$.*

The scale-and-shift channel is not negligible (its dynamics are of the same order in the step size), so the honest object is the coupled system:

**Proposition 3** (The saddle of the coupled linear dynamics)**.** *With $b = \bar{b} = 0$, $\delta = \theta(2\gamma\bar{g} - g)$ and updates $\dot{\theta} = \alpha\delta g$, $\dot{g} = \alpha\delta\theta$, the Jacobian at synchronization ($\bar{g}$=$g$) is*

$$J = \alpha \begin{pmatrix} g^2(2\gamma-1) & \theta g(2\gamma-2) \\ 2\theta g(2\gamma-1) & -\theta^2 \end{pmatrix}, \qquad \det J = \alpha^2 g^2 \theta^2\,(2\gamma-1)(3-4\gamma) \;<\; 0 \text{ for } \gamma > \tfrac{3}{4}. \quad (1)$$

*For $\gamma > 3/4$ this variational matrix (taken at a synchronization point, not at an equilibrium, since $\delta \neq 0$ there) has eigenvalues of opposite sign, a saddle; for $\gamma \in (\frac{1}{2}, \frac{3}{4}]$ the determinant is non-negative and the divergence is carried by the synchronization map alone. At every synchronization point the scale-and-shift target sits in target-free TD's expansive class, while the full target's flow between synchronizations is contractive; trajectory statements go through the synchronization map (Appendix D) and the $T$=20 numerics below.*

Deterministic trajectories ($\alpha$=0.1, $T$=20, $\gamma$=0.9) bear this out: target-free TD and the scale-and-shift target overflow within 20 and 80 steps, the same superexponential class, while the full target only reaches $|V(s_1)| = 9.6$ after 100 steps. In this depth-one instance a frozen last layer simply *is* the full target (no backbone lies beneath it), so the separation between the two families holds for a model without an intermediate representation; §4 is where depth changes it. In general, a target's stabilizing contraction is confined to the span of its stored directions.

## 4 Beyond first order: negative feedback from a frozen copy

The linear analysis of §3, and in particular the saddle of Prop. 3, predicts that a deep scale-and-shift target also diverges with target-free TD. It does not. On the deep two-state counterexample (two-hidden-layer MLP, $\gamma$=0.99, five seeds), target-free TD diverges every time within a hundred steps or so, while the scale-and-shift target is stable every time and converges to the true values. Three controls (five seeds each; Table A3, Appendix B) settle what matters: a trainable scale-and-shift whose bootstrap updates with the online parameters diverges 5/5; the scale-and-shift frozen on both sides (plain-backbone target-free) diverges 5/5; a trainable scale-and-shift against a frozen copy is stable 5/5 and converges to $V^*$. Neither trainability nor freezing alone suffices; what is left is the asymmetry itself.

The sign of the scale-and-shift channel's feedback flips between the two asymmetric cases: $\partial\delta/\partial g = (2\gamma - 1)\,\theta > 0$ when the bootstrap updates with the online parameters, versus $\partial\delta/\partial g = -\theta < 0$ against a frozen copy. When the scale-and-shift's functional leverage is high (deep networks), this negative feedback dominates the backbone expansion: the online adapter absorbs the TD gap faster than the backbone can grow. We call this the *negative feedback*.

The sign argument above concerns a single scale. To extend it to an arbitrary adapter $\omega$ and architecture, and to quantify how much of the backbone expansion the feedback cancels, we need a setting in which the coupled dynamics can be written as explicit matrices; the *tangent model* is that setting. At a synchronization point ($\omega = \bar{\omega}$) linearize the network outputs in $(\theta, \omega)$, with $J_\theta = \nabla_\theta F$, $J_\omega = \nabla_\omega F$, state weights $D$, and transition operator $P$. (The exact update field adds

residual-weighted curvature terms, since $\delta \neq 0$ at synchronization; Appendix D quantifies them.) The coupled tangent dynamics are

$$\begin{pmatrix} \dot{d\theta} \\ \dot{d\omega} \end{pmatrix} = \begin{pmatrix} A & -B \\ C & -G \end{pmatrix} \begin{pmatrix} d\theta \\ d\omega \end{pmatrix}, \quad \begin{aligned} A &= \alpha J_\theta^\top D(\gamma P - I) J_\theta, \quad B = \alpha J_\theta^\top D J_\omega, \\ C &= \alpha J_\omega^\top D(\gamma P - I) J_\theta, \quad G = \alpha J_\omega^\top D J_\omega. \end{aligned} \tag{2}$$

**Proposition 4** (Negative feedback from a frozen copy and its projected reduction)**.** *Assume the tangent model at a synchronization point (outputs linearized in $(\theta, \omega)$; exact-field corrections in Appendix D) and expected (batch) semi-gradient updates. (i)* Negative feedback. *The adapter's self-feedback block is $-G$, where $G = \alpha J_\omega^\top D J_\omega \succeq 0$ is a Gram matrix; the block is therefore negative semidefinite regardless of environment, discount, and architecture. This is exact in the tangent model, and exact outright whenever the online output $F$ is affine in $\omega$ (a last-layer head, a single scale). By contrast, when the bootstrap updates with the online parameters the block is $\alpha J_\omega^\top D(\gamma P - I) J_\omega$, which can be expansive. (ii)* Projected reduction. *Assume in addition that $\lambda_{\min}(G|_{\text{range}})$ is large relative to the backbone-scale blocks $\|A\|, \|B\|, \|C\|$ (time-scale separation, the finite-leverage proxy for the $\kappa \to \infty$ limit, Appendix D). Then the quasi-static optimum $d\omega^* = G^\dagger C\, d\theta$ ($G^\dagger$ the inverse on the range of $G$) can be substituted, and the backbone follows the effective dynamics*

$$\dot{d\theta} = \alpha\, K^\top (I - \Pi_\omega) W K\, d\theta, \qquad K = D^{1/2} J_\theta, \;\; W = D^{1/2}(\gamma P - I) D^{-1/2}, \tag{3}$$

*where $\Pi_\omega$ is the orthogonal projector onto $\mathrm{span}(D^{1/2} J_\omega)$ (the $D$-orthogonal projector onto $\mathrm{span}(J_\omega)$ in the original coordinates). In words, the feedback projects the TD instability out of the directions the adapter spans. Two extremes confirm this: $\Pi_\omega = 0$ recovers target-free TD exactly, and $\Pi_\omega = I$ freezes the backbone ($\dot{d\theta} = 0$), reaching the full target's stability by a route other than its contraction $-\alpha K^\top K$. The reduction is asymptotic in the separation limit and approximate at finite leverage; it holds on compact time intervals, and stability over $[0, \infty)$ additionally needs the unstable eigenspace of $WK$ to lie in that span, which is the leverage condition the experiments below test.*

Part (ii) is the dual of the storage principle: a target *decouples* the parameter directions it stores, and a fast adapter *projects out* the function directions it spans. Its validity conditions do real work, and they coincide: under the leverage scaling of Appendix D the exact field's curvature corrections to the adapter's self-block are $O(1/\kappa)$ against $G$, while the mixed blocks keep $O(1)$ corrections unless the adapter enters additively or the residual is small, which a fast adapter itself arranges by absorbing the TD gap (Appendix D). The scalar instance fails both at once (the corrections create Prop. 3's saddle; adapter and backbone move at the same rate); the deep instance satisfies both. Two validations (details and remaining gaps in Appendix D): on a two-state instance with a full-span adapter the divergence-to-stability transition lands in the predicted leverage window, and on Baird's star problem a frozen copy rescues an adapter seeded with the unstable mode while the same adapter updating with the online parameters accelerates divergence. The same experiment surfaced a second channel, span enlargement, which a two-factor design separates from the frozen copy.

This feedback reframes the last-layer target: the iS-QN online head (Vincent et al., 2026) trained against its frozen copy is precisely this asymmetry. It explains why last-layer targets close most of the gap and predicts their failure at high UTD, where backbone instability outruns one head's absorption. Both predictions hold on CartPole (Table A2): at UTD 8 the storage levels are monotone in return and divergence (last-layer target 3/5 diverged); at UTD 1 all partial targets are stable and the full target's staleness costs it the lead. A matched control sharpens the mechanism: at UTD 8 with the synchronization period restored in *environment* time ($T \times 8$), the last-layer target survives 5/5. The fragility is the environment-time period shrinking toward the target-free limit of Prop. 1 as UTD rises at fixed gradient-step $T$. *Span* gives the feedback its reach: adding per-layer frozen scale-and-shift copies (0.2% extra storage) to the last-layer target enlarges the span, and where the last-layer target alone dies 4/5 the combination survives 3/3 at full-copy return ($2.38 \pm 0.04$ vs. $2.42 \pm 0.03$).

The feedback is architecture-sensitive (full on the deep counterexample and Baird, partial on CartPole and MinAtar, as the leverage condition predicts; Appendix D) and is not staleness in disguise: delay without a trainable frozen copy does not stabilize (second control above).

# 5 REMOVING THE TARGET ENTIRELY: SPURIOUS SOLUTIONS AND A BOUNDED FIX

Push storage to zero: keep one network, place an arbitrary trainable predictor head $h$ on the online side, and bootstrap from the base network under stop-gradient. This section shows what goes wrong when $h$ is unrestricted and arrives from there at the bounded scaling of §2:

$$\hat{Q}(s,a) = \big[h\big(Q_\theta(s,\cdot)\big)\big]_a, \qquad \mathcal{L}(\theta,h) = \big(r + \gamma \max_{a'} \mathrm{sg}[Q_\theta(s',a')] - \hat{Q}(s,a)\big)^2. \tag{4}$$

At a stationary state where the loss (4) vanishes on the visited support, that is, a pair $(\theta, h)$ with $h \circ Q_\theta = \mathcal{B}Q_\theta$ ($\mathcal{B}$ the Bellman optimality operator), the base network supplies the *target* and the predictor the online estimate, the asymmetry that stabilizes representation learning (Chen & He, 2021; Grill et al., 2020). What helps in representation learning hurts here:

**Proposition 5** (Spurious zero-loss solutions)**.** *If $h$ can represent the map $Q_\theta(s,\cdot) \mapsto (\mathcal{B}Q_\theta)(s,\cdot)$ on the visited support, and $h^*$ denotes a head that realizes it, then $(\theta, h^*)$ is a zero-loss stationary point of the expected loss (of the sample loss when transitions are deterministic) for* arbitrary $\theta$.

*Proof.* The residual vanishes identically, hence so does every gradient. □

**Proposition 6** (Spurious solutions with nonzero residuals)**.** *Spurious solutions survive even when residuals cannot vanish: in the scalar instance with linear $h(x) = wx$, both parameter gradients are proportional to the same weighted residual sum, so stationary manifolds with $\theta$ free exist with individually nonzero TD errors (e.g., $w = 6\gamma/5$ on the two-transition instance).*

**Proposition 7** (Attraction to spurious solutions)**.** *On the single-transition instance the manifold $\{w = 2\gamma,\ \theta \text{ free}\}$ has transversal eigenvalue $-\alpha\theta^2 \le 0$: gradient descent finds it (numerics in Appendix D).*

On Baird's counterexample (Baird, 1995) (Table A4, Appendix B), the variant with a predictor head reaches TD loss $3 \times 10^{-30}$ while the value error freezes at 7.7; even a scalar affine predictor collapses to a constant, because all bootstrap targets share one successor state. *Identifiability, not capacity, is the switch.* On the exactly solvable chain MDP (Table A5, Appendix B), a width-256 predictor attains TD loss $2 \times 10^{-6}$ with $63\%$ greedy-policy accuracy (controls: $100\%$), and the capacity–error relation is non-monotone; the only soundness-restoring interventions are structural: an input bottleneck (rank-1) that makes the Bellman map unrepresentable, and the learned scalings below. Unlike finite-data Bellman-error pathologies (Fujimoto et al., 2022), this manifold exists with infinite data.

**The guarantee–stabilization trade-off.** The first attempt to exclude spurious solutions is to keep the head near the identity.

**Proposition 8** (Additive bound)**.** *Restrict the head to $h(q) = q + \varepsilon\,\psi(q)$, where $\psi$ is any function with $|\psi| \le 1$ and $\varepsilon > 0$ is the allowed perturbation. If the set of visited state–action pairs is closed under transitions (all pairs are visited in our testbeds), every zero-loss stationary point satisfies $\|Q - Q^*\|_\infty \le \varepsilon/(1-\gamma)$, where $\|\cdot\|_\infty$ is the maximum absolute value over states and actions.*

*Proof.* Zero loss gives $h(Q) = \mathcal{B}Q$ on the support, so $\|Q - \mathcal{B}Q\|_\infty \le \varepsilon$; conclude by the Bellman-residual bound. □

On the chain this is tight in practice (bound $1.0$, observed $0.24$, policy accuracy $1.00$). But the same $\varepsilon$ caps the feedback's absorption: on CartPole, where $Q^* \sim 10^2 \gg \varepsilon$, the additive bound diverges 5/5 at both UTD levels, while the *unbounded* affine predictor stabilizes (Table A2) yet carries no guarantee and collapses on Baird. The obvious repair is unsound:

**Proposition 9** (Divergence admitted by symmetric bounds)**.** *For $h(q) = q(1+\varepsilon s)$, $s \in [-1,1]$, the recursion $Q = \mathcal{B}Q/(1+\varepsilon s)$ with $s \approx -1$ has effective discount $\gamma/(1-\varepsilon)$, which exceeds $1$ for $\varepsilon > 1-\gamma$: the relative residual constraint is satisfiable along divergent trajectories. (Observed: a chain seed diverges to $5 \times 10^7$ while satisfying it.)*

The expansion-only form is the closure of the two failures above, which force a *scale-free* deviation bound (the additive bound's absorption is capped by $\varepsilon$ against values far larger) and *preservation of the discount bound* (the symmetric bound admits effective discount $\gamma/(1-\varepsilon) > 1$); within $h = q(1+\varepsilon s)$ this requires $s \geq 0$, and the expansion-only family is the minimal one satisfying both (an ablation adds a third requirement, a deviation channel responsive everywhere; Appendix C).

**Proposition 10** (Bounded output scaling)**.** *Let $h(q) = q\,(1+\varepsilon\,\sigma(r(q)))$ with $\sigma \in (0,1)$, so the multiplicative factor lies in $(1, 1+\varepsilon)$. Then every self-consistent solution of the effective recursion $Q = \mathcal{B}Q/(1+\varepsilon\sigma)$ is bounded, $\|Q\|_\infty \leq \|r\|_\infty/(1-\gamma)$ with $\|r\|_\infty = \sup_{s,a}|r(s,a)|$ (by domination, not contraction of a fixed map; Appendix D), and zero-loss stationary points on a transition-closed support satisfy $\|Q-Q^*\|_\infty \leq \varepsilon\|\mathcal{B}Q\|_\infty/(1-\gamma) \leq \varepsilon\|r\|_\infty/(1-\gamma)^2$; the deployed estimate $\hat{Q} = \mathcal{B}Q$ is within $\gamma$ times that bound of $Q^*$.*

This expansion-only family is the method we propose, Tempered Q-learning. Empirically the bounded scaling is the only zero-storage variant passing *both* diagnostics: chain calibration at policy accuracy 1.00 across seeds ($|Q-Q^*| = 0.21$) *and* CartPole stability with $0/10$ diverged. Against the unguaranteed affine predictor it trades return for the bound at UTD 8 (155.9 versus 195.1). Its zero-parameter member, fixed $h(q) = 1.1q$, already survives all four MinAtar stress tests of §7 20/20 at matching or better return (2.21 vs. 1.87 Breakout; 10.03 vs. 8.94 Space Invaders; 5/5 on Asterix, where the learned scaling keeps 2/5), and on ALE Breakout it reaches $14.1 \pm 0.4$ against the learned scaling's $8.8 \pm 1.8$ and the full copy's $13.7 \pm 7.1$: at small value scales the family's discount cap is the whole mechanism and nothing needs to be learned. Learning is what carries the family to large value scales: on CartPole, where $Q^* \sim 10^2$, the constant 1.1 diverges $10/10$ while the learned scaling is stable. The learned scaling's Asterix deaths are trajectory events outside the fixed-point bound; the constant has none, so a practitioner can start from the constant and add the learned residual only when the value scale demands it.

## 6 Stress test: the loss confound and the storage levels

Experiments are on MinAtar (Young & Tian, 2019) (150k–500k frames, Adam $2.5 \times 10^{-4}$ unless stated, UTD 8, synchronization every $10^3$ gradient steps); run counts are stated per result, and GPU nondeterminism makes seeded runs independent samples; Figure 1 collects the training curves referred to below.

**Sameness under the Huber loss.** Under the Huber loss (Figure 1a), all seven architectures in this range reach mean training return $3.0 \pm 0.15$; only the affine head collapses late.

**The controlled pair.** Flipping only the loss to MSE (Figure 1b) (same optimizer, learning rate, and UTD), target-free DQN diverges within 25k frames ($\max|Q| > 3 \times 10^5$) while the full target learns cleanly (final return $6.04 \pm 0.20$ over five runs, Table 1). This controlled pair changes only the loss in standard semi-gradient DQN (§8), and it shows that the choice of loss alone can reverse a target-necessity conclusion.

**Storage levels under MSE.** Under MSE at UTD 8 (Table 1), survival and return are monotone in stored depth. The last-layer target (the iS-QN $K{=}1$ architecture) diverges in four of five runs and its survivor trails every deeper run, which is why iterated heads were needed at high UTD. Two stored layers already match the full copy's return. The comparison varies stored *directions*, not memory (the two-layer level holds 93–99% of all parameters). The threshold moves with the game (five runs per level): on Space Invaders every level below the full copy diverges 5/5 and the copy survives two of five; on Seaquest and Asterix every level, the copy included, dies 5/5. Divergence counts are monotone in stored depth in every game; where the threshold sits depends on the game.

**Value bias.** In the calibration probe of Figure 1d (predicted minus realized return), the full copy overestimates by orders of magnitude, the affine head drifts upward through zero, and the learned scaling keeps a flat single-digit conservative bias in every run (Appendix B).

**Storage levels on ALE Breakout.** Five runs per level (Figure 1c) (Nature CNN, MSE with clipped rewards, UTD 2, 1.5M agent steps). Nothing diverges, and mean final-window returns are monotone

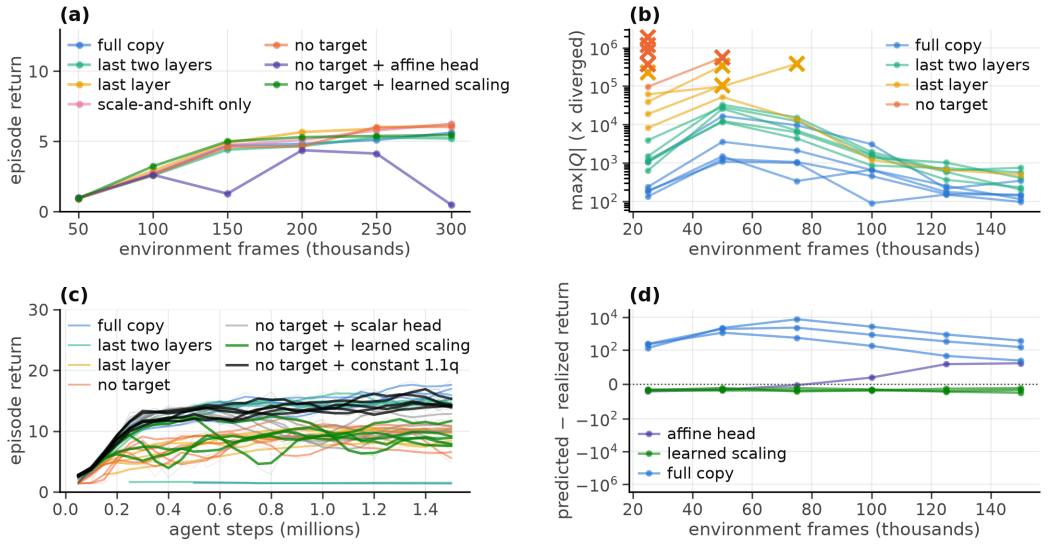



Figure 1: **Stress test across storage levels and value calibration.** **(a)** Loss-confound check: MinAtar Breakout, Huber loss, UTD 8, return of one run per level. **(b)** Stability by storage level under MSE: the same game with the MSE loss, $\max |Q|$ of five runs per level ($\times$ marks divergence); full curves in Figure A1. **(c)** Storage levels at scale: ALE Breakout, five runs per level, faint raw trajectories with smoothed curves. **(d)** Value-bias check: predicted minus realized return on MinAtar Breakout (symlog axis). Interpretation in §6.

Table 1: Storage levels in the MSE, UTD 8 stress test (MinAtar Breakout, five runs per row, 150k frames). **Top:** targets that store parameters. **Bottom:** Tempered Q-learning, which stores nothing (§5). Values are mean $\pm$ s.d. over surviving runs; the training return is averaged over training, the final return over the last window.

| storage level | survival (5 runs) | training return | final return |
|---|---|---|---|
| target-free | 5/5 diverged (25–50k frames) | n/a | n/a |
| last layer (iS-QN) | 4/5 diverged (25–75k frames) | 2.25 (survivor) | 5.53 (survivor) |
| last two layers | 5/5 survived | $2.41 \pm 0.04$ | $6.11 \pm 0.32$ |
| full copy | 5/5 survived | $2.42 \pm 0.03$ | $6.04 \pm 0.20$ |
| Tempered, constant $1.1q$ (proposed) | 5/5 survived | $2.21 \pm 0.05$ | $5.39 \pm 0.22$ |
| Tempered, learned (proposed) | 5/5 survived | $1.87 \pm 0.16$ | $4.02 \pm 0.60$ |

in stored depth: $8.5 \pm 2.7$ (no target), $9.3 \pm 1.0$ (last layer), $11.5 \pm 5.6$ (last two layers), $13.7 \pm 7.1$ (full copy). On four of five seeds the two-layer and full-copy targets learn far above everything else (13.5 to 18.3); on the fifth both plateau near 1.4 while the shallower levels learn. The learned scaling ($8.8 \pm 1.8$) tracks the last-layer band while storing nothing and learns on the seed that traps both deep targets. Two heads run afterwards finish the picture: HANQ's scalar predictor reaches $10.2 \pm 1.8$ but its value scale drifts upward and its returns decay after 500k steps, while the constant $1.1q$ reaches $14.1 \pm 0.4$, the full copy's mean ($13.7 \pm 7.1$) at a sixteenth of its spread, every run between 13.7 and 14.8; on the IQM the copy keeps a lead (16.3 vs. 14.0) only because that statistic discounts its trapped seed. This is one dense-reward game at a mild UTD where nothing diverges, a case study rather than a benchmark; storage buys the ceiling here, not the floor.

**Refresh period in environment steps.** At UTD 32, synchronizing every $10^3$ *gradient* steps is synchronizing every 31 *environment* steps, and the full target itself diverges; restoring the period in environment time restores clean learning (Appendix B). The same correction rescues the full copy in the four-game comparison (Table 2, second reference row): it then survives 18 of 20 runs, but returns less than the proposed constant scaling on Space Invaders (7.70 vs. 10.03) and about the same elsewhere. The heads sidestep the choice, since nothing is synchronized.

Table 2: Controlled comparison in the MSE, UTD 8 stress test (four MinAtar games, five runs per cell). Each row adds one method's stabilizing component to the target-free level; the reference rows are the full target copy at the usual period ($10^3$ gradient steps, 125 environment steps at UTD 8) and at the period matched in environment time ($T{=}8{\times}10^3$). Return is the mean over surviving runs; the superscript $d/5$ marks $d$ divergences out of five and "all died" five of five. Bold: best in the column among rows with no deaths. Storage is the fraction of parameters frozen for the bootstrap; all rows run under replay. Bias: the Breakout probe (Figure 1d), nearer zero better; the value-error guarantee is the bound of §5. Middle block: the proposed method; bottom block: compared components (the scalar predictor is HANQ's). IQM and confidence intervals in Table A7.

| stabilizing component | storage | Breakout ↑ | Space Inv. ↑ | Seaquest ↑ | Asterix ↑ | surv./20 | bias ($\to 0$) | value-error guarantee |
|---|---|---|---|---|---|---|---|---|
| *full copy, usual period (reference)* | 100% | $2.42 \pm 0.03$ | $7.26^{3/5}$ | all died | all died | 7 | $+22$ to $+367$ | none |
| *full copy, period in environment time* | 100% | $2.28 \pm 0.03$ | $7.70 \pm 0.18$ | $0.24 \pm 0.02$ | $0.56^{2/5}$ | 18 | n/a | none |
| Tempered, constant $1.1q$ (proposed) | 0 | $2.21 \pm 0.05$ | $\mathbf{10.03}$ | $\mathbf{0.27}$ | 0.58 | **20** | n/a | $\boldsymbol{0.1\|\mathcal{B}Q\|/(1-\gamma)}$ |
| Tempered, learned (proposed) | 0 | $1.87 \pm 0.16$ | 8.94 | 0.26 | $0.56^{3/5}$ | 17 | $-2.1$ to $-4.6$ | $\boldsymbol{\varepsilon\|\mathcal{B}Q\|/(1-\gamma)}$ |
| scalar predictor (Snyder & Wei, 2025) | 0 (1 online) | $2.46 \pm 0.11$ | 9.63 | $\mathbf{0.27}$ | $\mathbf{0.62}$ | **20** | $\mathbf{+0.6}$ **to** $\mathbf{+3.1}$ | none (Prop. 5) |
| mellowmax $\beta{=}1$ (Kim et al., 2019) | 0 | $\mathbf{2.83 \pm 0.08}$ | $11.32^{1/5}$ | $0.26^{1/5}$ | all died | 13 | $-3.6$ to $-4.5$ | $\gamma \log \lvert A\rvert/\beta(1-\gamma)$ |
| iterated heads $K{=}2$ (Vincent et al., 2026) | last layer | $2.31 \pm 0.03$ | 7.31 | all died | all died | 10 | 85 to 150 | none |
| min(copy, online) (Hendawy et al., 2026) | 100% | $2.42 \pm 0.04$ | $7.10^{2/5}$ | all died | all died | 8 | n/a | none |
| LN $+ \ell_2$ (Gallici et al., 2025) | 0 | $2.50 \pm 0.09$ | $6.82^{4/5}$ | all died | all died | 6 | $+21$ to $+66$ | none |
| LN (Gallici et al., 2025) | 0 | $2.45 \pm 0.12$ | all died | all died | all died | 5 | $+22$ to $+39$ | none |

## 7 Discussion: interchangeability of the stabilizers

Read together, TD stability is purchased from a set of partly interchangeable stabilizers: storage, loss clipping, normalization (Bhatt et al., 2024), smoothing, learning-rate headroom (Appendix B), span, and online adapters (§4–5).

Table 2 puts numbers on this: each recent target-reducing method's stabilizing component, transplanted alone into the stress test, everything else fixed. On Breakout every transplant rescues target-free TD and mellowmax leads (2.83 vs. 2.42), though its knob is sensitive ($\beta{=}3$ and $\beta{=}10$ both diverge). On Space Invaders (full copy: three of five runs lost) normalization collapses while every zero-storage head and the iterated heads survive 5/5 above the full copy's survivors; on Seaquest and Asterix, where no storage level survives at the gradient-step period (§6), the scalar predictor and the constant scaling stay alive on every seed (the learned scaling keeps 2/5 on Asterix) while mellowmax's knife-edge widens from one lost run to five. Smoothing buys return inside the full copy's stability window; the online adapters buy the window itself. Every method that keeps a stored copy keeps its deaths, the full copy 7/20, the copy with the minimum rule of Hendawy et al. (2026) 8/20, iterated heads over a stored last layer 10/20. The constant scaling, storing nothing and synchronizing nothing, survives 20/20, matches iterated heads on Breakout (2.21 vs. 2.31), leads on Space Invaders (10.03 vs. 7.31), and on ALE Breakout reaches $14.1 \pm 0.4$ where the last-layer target reaches $9.3 \pm 1.0$. These rows transplant components under replay, not full methods: PQN's own recipe removes the replay buffer, where its LayerNorm does hold (§6), and CrossQ's joint-batch normalization (Bhatt et al., 2024) was not transplanted, so the normalization rows speak to per-sample LayerNorm only. Within that protocol the output scaling heads (HANQ's scalar predictor and this paper's bounded scalings) are the currency that transfers. Their mechanism is not the negative feedback of Prop. 4, which needs a frozen copy, but the discount cap of Prop. 10: a factor above one on the prediction side shortens the effective horizon of the bootstrap. For the constant this is exact: under MSE and Adam its update equals plain target-free TD at discount 0.9 with rewards divided by 1.1, and that run reproduces it on all four games (20/20 alive, returns within the constant's intervals, Appendix B). The constant is a shorter horizon applied without a copy or a period; the learned scaling, whose factor varies with state and action, is not reducible this way.

Read the way iS-QN reports its own results, as return relative to the copy with its period matched in environment time (100; diverged runs scored as zero) averaged over the four games: no target 0, last layer 0, iterated heads 49, the copy at its usual period 35, learned scaling 90, constant scaling 125, HANQ's scalar predictor 131 (per game and for ALE Breakout in Table A6).

**Practical benefits.** The target network can be removed from deep Q-learning and replaced by a scaling of the network's output, with less memory, fewer divergences, and no new tuning. The parameters stored for the bootstrap are zero (the full copy stores 132,854 on MinAtar and 1,686,180

on the Nature CNN), training memory on the Nature CNN (parameters plus two Adam moments) drops from 25.7 to 19.3 MiB, a 25% saving, and the extra online parameters are 0 for the constant and 838 for the learned scaling (580 on the Nature CNN) (Appendix C); Algorithm 1 shows the two-line change. The last-layer target of iS-QN keeps a frozen head, a chain of $K$ online heads, a rolling synchronization, and the choice of $K$, whereas Tempered Q-learning keeps one network and one multiplication.

**Algorithm 1** Tempered Q-learning, deep Q-learning without a target network. An output scaling $h_\phi$ (the constant $h(q) = 1.1\,q$ or the bounded $h(q) = q\,(1 + \varepsilon\,\sigma(r_\phi(q)))$) replaces the frozen copy. Only lines 3 and 4 differ from DQN; $h_\phi$ trains with the same optimizer, and with no copy there is no synchronization period. Under Double DQN a uniform positive scaling preserves the greedy action, so the update is unchanged (per-action scaling not benchmarked).

1: **for** each gradient step **do**
2: sample $(s, a, r, s', d)$ from the replay buffer
3: $y \leftarrow r + (1 - d)\,\gamma \max_{a'} Q_\theta(s', a')$ ▷ online network, no gradient, no target copy
4: $\mathcal{L} \leftarrow \big([h_\phi(Q_\theta(s))]_a - y\big)^2$; update $(\theta, \phi)$ by one optimizer step on $\mathcal{L}$
5: **end for**
6: act with $\arg\max_a [h_\phi(Q_\theta(s))]_a$

# 8 Related work and limitations

## 8.1 Related work

Fellows et al. (2023) analyze why a *full* target network stabilizes TD through Jacobian conditioning. Their analysis concerns partiality in time; this paper concerns partiality in parameters, which directions are stored. Piché et al. (2023) give a pointwise synchronization identity for the full target, and their functional regularizer keeps a lagged copy at the full-storage end of the storage levels. Props. 1–2 extend it to the unstored directions.

Vincent et al. (2026) propose the frozen-last-layer architecture and show its effect empirically; the negative feedback of §4 adds the dynamical mechanism and predicts its fragility at high UTD (Tables 1 and A2), the setting where storage matters (Voelcker et al., 2025). Related lines change what the target *is* (Vincent et al., 2025a;b;c; Hendawy et al., 2026), never which directions are stored.

Gallici et al. (2025) remove the parameter copy and the replay buffer together and cache Q($\lambda$) targets once per rollout, a bootstrap pinned to data rather than to parameters, which lies outside the storage levels. Snyder & Wei (2025) replace the target with a learned single-scalar predictor in offline RL; Prop. 5 explains why that predictor's capacity must be low, and the bounded scaling adds a value-error guarantee. Tang et al. (2023) analyze the same asymmetry in latent self-prediction (Tian et al., 2021; Schwarzer et al., 2021; 2023); §5 shows that for value bootstrapping it must be bounded.

The effect of the loss on stability is treated by Patterson et al. (2023); Young & Tian (2019); Obando-Ceron & Castro (2021); van Hasselt et al. (2018); §6 isolates it as a controlled pair in standard semi-gradient DQN that changes only the loss. Other stabilizers, of which §7 compares those interchangeable with storage, include a target network with ridge regularization (Zhang et al., 2021), targets learned in function space (Asadi et al., 2024), regularized off-policy fixed points (Kolter, 2011), double estimation (van Hasselt et al., 2016), averaged targets (Anschel et al., 2017), and policy churn (Schaul et al., 2022).

## 8.2 Limitations

The projected reduction of Prop. 4(ii) is asymptotic under the tangent model and time-scale separation, and approximate at finite leverage. The guarantee of the bounded scaling (Prop. 10) constrains fixed points, not trajectories; the learned scaling's three divergences out of five runs on Asterix are that gap. Distilled low-capacity targets (Fu et al., 2025; Vincent et al., 2025c) were not compared. The full-scale validation is one game, ALE Breakout at UTD 2, where target-free TD itself does not diverge.

### AI USE STATEMENT

We used generative AI tools for three purposes. First, to aid and polish the writing, improving the clarity, grammar, and concision of text written by the authors. Second, for retrieval and discovery, that is, to survey and identify related work and to format references; every cited paper and named method was read and checked by the authors against its source publication. Third, to assist in proving mathematical claims, by checking derivations and suggesting proof steps for propositions formulated by the authors; every proof was re-derived and verified by the authors.

### ETHICS STATEMENT

This work studies the training dynamics of value-based reinforcement learning on standard simulated benchmarks (MinAtar, the Arcade Learning Environment, CartPole, and small analytic counterexamples). It involves no human subjects, no personal data, and no application-specific deployment. The proposed replacement of the target network reduces the memory required to train a Q-network; we are not aware of a use that this enables which the standard target network does not. All experiments ran on two consumer GPUs; the total compute is reported in the supplementary material.

### REPRODUCIBILITY STATEMENT

All diagnostics are deterministic or seeded and ran on two consumer GPUs (counterexamples in minutes, MinAtar pilots overnight). The supplementary material contains the code, organized as a small library in which every method in this range differs only in one bootstrap function, together with per-experiment scripts, a command-to-artifact reproduction table, and the complete training log behind the figures; the figures regenerate from the shipped log without GPU time.

## REFERENCES

O. Anschel, N. Baram, N. Shimkin. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning. *ICML*, 2017.

K. Asadi, Y. Liu, S. Sabach, M. Yin, R. Fakoor. Learning the target network in function space. *ICML*, 2024.

L. Baird. Residual algorithms: Reinforcement learning with function approximation. *ICML*, 1995.

A. Bhatt, D. Palenicek, B. Belousov, M. Argus, A. Amiranashvili, T. Brox, J. Peters. CrossQ: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. *ICLR*, 2024.

X. Chen, K. He. Exploring simple Siamese representation learning. *CVPR*, 2021.

M. Fellows, M. Smith, S. Whiteson. Why target networks stabilise temporal difference methods. *ICML*, 2023.

P. Fu, O. Rybkin, Z. Zhou, M. Nauman, P. Abbeel, S. Levine, A. Kumar. Compute-optimal scaling for value-based deep RL. *NeurIPS*, 2025.

S. Fujimoto, D. Meger, D. Precup, O. Nachum, S. S. Gu. Why should I trust you, Bellman? The Bellman error is a poor replacement for value error. *ICML*, 2022.

M. Gallici, M. Fellows, B. Ellis, B. Pou, I. Masmitja, J. N. Foerster, M. Martin. Simplifying deep temporal difference learning. *ICLR*, 2025.

J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, M. Valko. Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS*, 2020.

A. Hendawy, H. Metternich, T. Vincent, M. Kallel, J. Peters, C. D'Eramo. Use the online network if you can: Towards fast and stable reinforcement learning. *ICLR*, 2026.

H. K. Khalil. *Nonlinear Systems*. Prentice Hall, third edition, 2002.

S. Kim, K. Asadi, M. Littman, G. Konidaris. DeepMellow: Removing the need for a target network in deep Q-learning. *IJCAI*, 2019.

J. Z. Kolter. The fixed points of off-policy TD. *NeurIPS*, 2011.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

J. S. Obando-Ceron, P. S. Castro. Revisiting Rainbow: Promoting more insightful and inclusive deep reinforcement learning research. *ICML*, 2021.

A. Patterson, V. Liao, M. White. Robust losses for learning value functions. *IEEE TPAMI*, 45(5):6157–6167, 2023.

E. Perez, F. Strub, H. de Vries, V. Dumoulin, A. Courville. FiLM: Visual reasoning with a general conditioning layer. *AAAI*, 2018.

A. Piché, V. Thomas, J. Marino, R. Pardinas, G. M. Marconi, C. Pal, M. E. Khan. Bridging the gap between target networks and functional regularization. *TMLR*, 2023.

T. Schaul, A. Barreto, J. Quan, G. Ostrovski. The phenomenon of policy churn. *NeurIPS*, 2022.

M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, P. Bachman. Data-efficient reinforcement learning with self-predictive representations. *ICLR*, 2021.

M. Schwarzer, J. S. Obando-Ceron, A. Courville, M. G. Bellemare, R. Agarwal, P. S. Castro. Bigger, better, faster: Human-level Atari with human-level efficiency. *ICML*, 2023.

B. Snyder, C.-Y. Wei. HANQ: Hypergradients, asymmetry, and normalization for fast and stable deep Q-learning. *RLJ/RLC*, 2025.

Y. Tang, Z. D. Guo, P. H. Richemond, B. A. Pires, Y. Chandak, R. Munos, M. Rowland, M. G. Azar, C. Le Lan, C. Lyle, A. György, S. Thakoor, W. Dabney, B. Piot, D. Calandriello, M. Valko. Understanding self-predictive learning for reinforcement learning. *ICML*, 2023.

Y. Tian, X. Chen, S. Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. *ICML*, 2021.

J. Tsitsiklis, B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE TAC*, 1997.

H. van Hasselt, A. Guez, D. Silver. Deep reinforcement learning with double Q-learning. *AAAI*, 2016.

H. van Hasselt, Y. Doron, F. Strub, M. Hessel, N. Sonnerat, J. Modayil. Deep reinforcement learning and the deadly triad. *arXiv:1812.02648*, 2018.

T. Vincent, D. Palenicek, B. Belousov, J. Peters, C. D'Eramo. Iterated Q-Network: Beyond one-step Bellman updates in deep reinforcement learning. *TMLR*, 2025a.

T. Vincent, F. Wahren, J. Peters, B. Belousov, C. D'Eramo. Adaptive Q-Network: On-the-fly target selection for deep reinforcement learning. *ICLR*, 2025b.

T. Vincent, T. Faust, Y. Tripathi, J. Peters, C. D'Eramo. Eau De Q-Network: Adaptive distillation of neural networks in deep reinforcement learning. *Reinforcement Learning Journal (RLC)*, 2025c.

T. Vincent, Y. Tripathi, T. Faust, A. Akgül, Y. Oren, M. Kandemir, J. Peters, C. D'Eramo. Bridging the performance-gap between target-free and target-based reinforcement learning. *ICLR*, 2026.

C. A. Voelcker, M. Hussing, E. Eaton, A. Farahmand, I. Gilitschenski. MAD-TD: Model-augmented data stabilizes high update ratio RL. *ICLR*, 2025.

K. Young, T. Tian. MinAtar: An Atari-inspired testbed for thorough and reproducible reinforcement learning experiments. *arXiv:1903.03176*, 2019.

S. Zhang, H. Yao, S. Whiteson. Breaking the deadly triad with a target network. *ICML*, 2021.

## A TESTBED DETAILS

Each setting isolates one mechanism. (i) *Linear two-state counterexample* (Tsitsiklis & Van Roy, 1997): states $s_1, s_2$ with features 1 and 2, transitions $s_1 \to s_2$, $s_2 \to s_2$, zero reward, $V^* \equiv 0$; fully analyzable divergence, used for Props. 2–3 and the predictor fixed-point instances. (ii) *Deep version*: the same MDP with a two-hidden-layer (width 32) MLP and per-layer feature scale-and-shift, expected updates; the smallest setting where depth can overturn the linear analysis (§4). (iii) *Baird's star problem* (Baird, 1995): seven states, all transitions to state 7 under the target policy; because every bootstrap target shares one successor, it is the sharpest test of spurious predictor solutions. (iv) *Chain MDP*: deterministic left/right actions, reward 1 at the terminal right end, $\gamma{=}0.9$; $Q^*$ computable exactly and all state–action pairs covered in full batch, so value error and greedy-policy accuracy are exact. (v) *CartPole* (MLP, no normalization, Adam $10^{-3}$) and *MinAtar* (standard convolutional network, Adam $2.5 \times 10^{-4}$, replay $10^5$, batch 32) with UTD and loss as setting axes. Freezing a layer includes its scale-and-shift: a live scale over a frozen layer reopens a multiplicative feedback loop and destroys the two-layer variant.

## B COUNTEREXAMPLE, CARTPOLE, AND CHAIN TABLES; FULL TRAINING CURVES

Figure A1 and Tables A1–A5 collect the hypothesis table and the complete diagnostics referenced from the main text. Synchronization-scaling details (§6): at UTD 32 with $T{=}10^3$ gradient steps the full target compounds to $\max|Q| > 10^7$ within 25k frames; restoring the period in environment time ($T{=}8{\times}10^3$) restores clean learning (final return 6.07, single runs). Learning-rate control (§7): lowering the learning rate rescues target-free MSE (alive at $10^{-4}$ and $5 \times 10^{-5}$), but at every tested rate it stays below the full target at the standard rate (mean returns $2.43 > 2.32 > 2.24$): storage buys roughly $2.5\times$ learning-rate headroom, and the headroom converts to return (single runs). Calibration probe under Huber (§6, predicted initial value versus realized discounted return): the full target ends at $+22$ to $+367$ after transient peaks in the thousands, the affine predictor drifts through zero to $+15$, and the learned scaling holds a single-digit conservative bias in every run ($-2.1$ to $-4.6$).

Table A1: Hypotheses written down before the corresponding experiments (an internal record, not an external preregistration) and their outcomes.

| # | hypothesis | outcome |
|---|---|---|
| 1 | scale-and-shift target diverges with target-free TD | **linear: confirmed; deep: refuted** ⇒ §4 |
| 2 | predictor head reaches low TD loss with wrong values | confirmed (Baird machine zero; chain) |
| 3 | spurious predictor solutions monotone in capacity | **refuted as monotone** ⇒ identifiability (§5) |
| 4 | stability monotone in stored depth $k$ | confirmed (CartPole UTD 8; MinAtar MSE, 19/20 runs) |

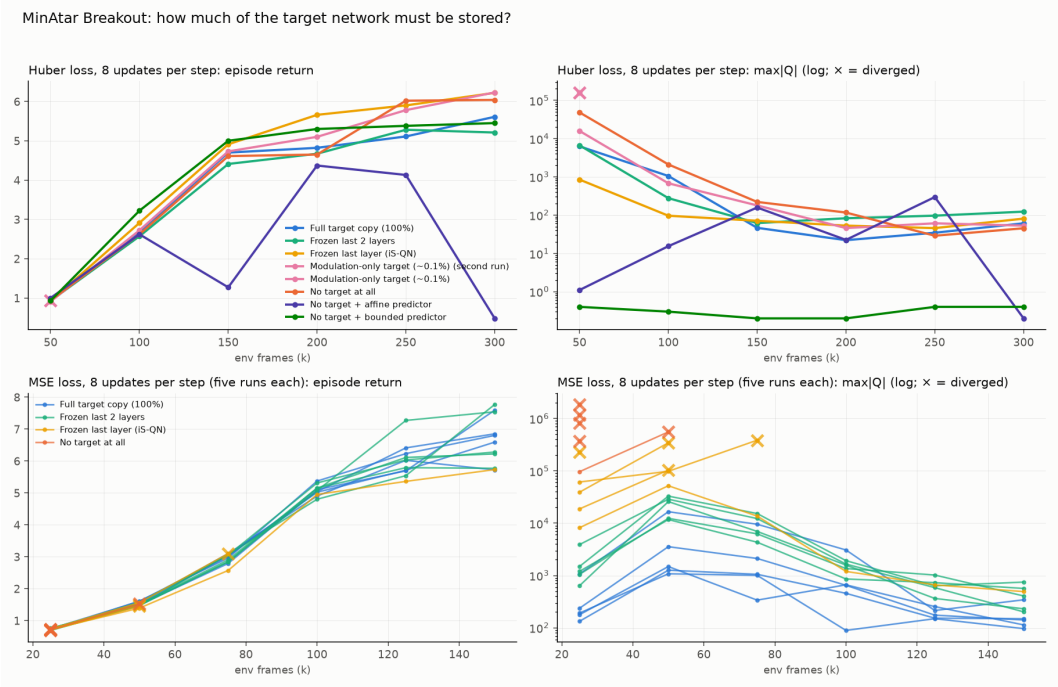



Figure A1: Complete four-panel version of Figure 1a,b: returns and $\max|Q|$ under both losses, including the bounded predictor's small value scale (green, top right) and the per-seed MSE return curves (bottom left).

**Consistency versus a lottery (Breakout, Huber, 500k frames).** Across five runs the unbounded affine predictor (scale and shift, extending Snyder & Wei, 2025) is a lottery: one run collapses

Table A2: CartPole (no normalization, Adam $10^{-3}$, 20k environment steps, 5 seeds): mean evaluation return / diverged seeds. This protocol is divergent for target-free TD at both UTD levels (a plain MLP without scale-and-shift also diverges 5/5 at both); UTD controls *how much* storage is needed. At UTD 1 the full copy trails several partial methods: with no instability to suppress, its staleness is pure cost (§4).

| method (storage) | UTD 8 | UTD 1 |
|---|---|---|
| No target at all (0) | 9.1 / 5 | 9.3 / 5 |
| Scale-and-shift target (2.9%) | 80.3 / 1 | 163.1 / 0 |
| No target + affine predictor (2 scalars) | 195.1 / 0 | 134.1 / 0 |
| No target + learned bounded scaling (0) | 155.9 / 0 | 159.0 / 0 |
| Frozen last layer (iS-QN) | 125.0 / 3 | 60.8 / 0 |
| Frozen last two layers | 132.4 / 0 | 160.4 / 0 |
| Full target copy (100%) | 253.0 / 0 | 123.6 / 0 |

Table A3: Controls of §4 on the deep two-state counterexample (two-hidden-layer MLP, per-layer scale-and-shift, $\gamma$=0.99, five seeds). The online scale-and-shift is trainable or frozen and the bootstrap scale-and-shift is co-moving, frozen, or a frozen copy; divergence at $\max|V| > 10^5$.

| method | online scale-and-shift | bootstrap scale-and-shift | diverged | outcome |
|---|---|---|---|---|
| target-free TD | trainable | updates with the online parameters | 5/5 | diverges within about a hundred steps |
| frozen on both sides (plain-backbone target-free) | frozen | frozen (the same) | 5/5 | diverges |
| scale-and-shift target | trainable | frozen copy | 0/5 | stable, converges to $V^*$ |

to $\max|Q| \approx 0.2$ (§5's spurious solution sets realized online), one finishes degraded, three are excellent; mean return $2.98 \pm 0.99$. The learned scaling lands at the same mean with a third of the spread ($2.99 \pm 0.28$), nothing stored, zero failures: the guarantee buys the tail. The calibration probe above (predicted initial value versus realized discounted return) shows that no method is calibrated here, and that the learned scaling holds the smallest bias by an order of magnitude ($-2.1$ to $-4.6$ in every run, a stable underestimate, the benign direction (van Hasselt et al., 2016)).

**Robust aggregates.** Table A7 reports interquartile means (for five seeds, the mean of the middle three) with 95% percentile-bootstrap confidence intervals ($10^4$ resamples over seeds; survivors only; no interval where two or fewer runs survive) for every headline comparison. No conclusion changes. On the Atari storage levels the IQM *raises* the deeper levels (full copy $13.7 \to 16.3$, two layers $11.5 \to 13.8$) by discounting the takeoff-trap seed, and the last-layer and bounded scaling intervals overlap throughout, so the within-noise reading of §6 is unaffected under the robust statistic. The constant head's interval $[13.8, 14.6]$ lies above every level's interval except the full copy's upper half.

## C Architectures and hyperparameters

**Chain.** Value network: one-hot state (8) $\to$ 64 $\to$ 64 $\to |A|$=2, ReLU; full-batch expected updates over all non-terminal state–action pairs; Adam $10^{-3}$; 20k steps; $\gamma$=0.9.

**CartPole.** MLP: observation (4) $\to$ 128 $\to$ 128 $\to |A|$=2, ReLU. Scale-and-shift variants add a per-hidden-layer feature-wise scale and shift (initialized to identity) after each hidden linear layer. Adam $10^{-3}$; replay $5 \times 10^4$; batch 64; $\gamma$=0.99; $\varepsilon$-greedy $1 \to 0.05$ over 5k steps; synchronization every 500 gradient steps; 20k environment steps; Huber loss; no normalization layers.

**MinAtar.** Convolutional network: 16 filters 3×3 $\to$ flatten $\to$ 1024-to-128 linear ($16{\cdot}8{\cdot}8 \to 128$) $\to$ head ($128 \to |A|$), ReLU; scale-and-shift variants add feature-wise scale/shift after the convolution (16 channels) and after the hidden linear layer (128 units). Frozen-last-$k$ targets freeze the head ($k$=1) or the hidden linear layer and head ($k$=2), always together with the scale-and-shift applied to the frozen layers. Adam $2.5 \times 10^{-4}$; replay $10^5$; batch 32; $\gamma$=0.99; $\varepsilon$: $1 \to 0.1$ over 100k; synchronization every $10^3$ gradient steps; sticky actions 0.1; divergence declared at $\max|Q| > 10^5$.

Table A4: Baird's star problem ($\gamma$=0.99, $V^*\equiv 0$, expected updates, 5000 steps). The target-free field diverges in 8 steps; both partial targets stabilize it; the predictor variants reach machine-zero TD loss while the value error freezes. The scale-and-shift rescue is quantitatively consistent with Prop. 4: at Baird's classic initialization the separation ratio $\lambda_{\min}(G)/\lambda_{\max}(A)$ is 1.4, versus 0.16 in the scalar two-state instance where the same scale-and-shift target diverges. Both instances are linear; the ratio, not linearity, discriminates.

| method | outcome | final TD loss | final value error |
|---|---|---|---|
| No target | diverged at step 8 | n/a | n/a |
| Scale-and-shift target | stable | $2.0 \times 10^{-7}$ | 7.18 (decreasing) |
| Full target ($T$=100) | stable | $2.3 \times 10^{-7}$ | 10.19 (decreasing) |
| No target + affine predictor | zero loss, wrong values | $3.5 \times 10^{-30}$ | **7.68 (frozen)** |
| No target + MLP-16 predictor | zero loss, wrong values | $2.2 \times 10^{-29}$ | **6.00 (frozen)** |

Table A5: Chain MDP with exact $Q^*$ ($\gamma$=0.9, 20k full-batch Adam steps, 5 seeds). TD loss can be tiny while values are wrong (Prop. 5); capacity does not order the failure; structural constraints (bottleneck, learned scalings) restore soundness. The symmetric bound confirms Prop. 9: one seed diverges to $5 \times 10^7$ while satisfying the relative residual.

| online estimate | TD loss | mean $\lvert Q - Q^*\rvert$ | policy acc. |
|---|---|---|---|
| $Q$ itself, no target | $6.8 \times 10^{-5}$ | 0.005 | 1.00 |
| $Q$ itself, full target | $4.2 \times 10^{-12}$ | 0.001 | 1.00 |
| affine $h(Q)$ | $1.9 \times 10^{-4}$ | 3.13 | 0.60 |
| MLP-4 $h(Q)$ | $4.0 \times 10^{-1}$ | 84.9 | 0.40 |
| MLP-64 $h(Q)$ | $3.8 \times 10^{-5}$ | 2.44 | 0.43 |
| MLP-256 $h(Q)$ | $2.0 \times 10^{-6}$ | 0.60 | 0.63 |
| unlearned scaling $q{+}\varepsilon r(q)$ | $1.6 \times 10^{-3}$ | 4.96 | 0.03 |
| bottleneck (rank-1 input) | $6.0 \times 10^{-2}$ | 0.48 | 0.94 |
| additive bound $q{+}\varepsilon \tanh r(q)$ | $2.6 \times 10^{-4}$ | 0.24 | **1.00** |
| symmetric bound $q(1{+}\varepsilon \tanh r(q))$ | $3.1 \times 10^{13}$ | $10^7$ (1 seed) | 0.34 |
| bounded scaling $q(1{+}\varepsilon\sigma(r(q)))$ | $3.4 \times 10^{-5}$ | 0.21 | **1.00** |

**Atari.** ALE Breakout-v5, sticky actions 0.25, standard preprocessing (frame skip 4, grayscale, 84×84, four-frame stack, rewards clipped to $[-1, 1]$). Nature architecture: conv 32@8×8 stride 4 $\to$ conv 64@4×4 stride 2 $\to$ conv 64@3×3 stride 1 $\to$ linear 3136 $\to$ 512 $\to$ head 512 $\to$ $|A|$, ReLU. Frozen-last-$k$: $k$=1 head, $k$=2 linear+head. Adam $10^{-4}$; replay $5 \times 10^4$ (uint8); batch 32; $\gamma$=0.99; $\varepsilon$: 1 $\to$ 0.1 over 250k; synchronization every $4 \times 10^3$ gradient steps; UTD 2; MSE loss; 1.5M agent steps.

**Stored parameters by method.** What each storage level actually stores, per architecture (percent of all parameters): full copy 100% everywhere; frozen last two layers 99.3% (MinAtar), 95.4% (Atari), 93.6% (CartPole), since the first fully-connected layer dominates the parameter count; frozen last layer 0.58% / 0.12% / 1.4%; scale-and-shift 0.22% / – / 2.9%; predictor heads store no target at all and train 1–838 extra online parameters. Freezing two layers is thus a statement about *which directions* are decoupled, not a memory saving; the memory-relevant levels are the last layer, the scale-and-shift, and the predictors.

**Predictor heads.** Affine: $h(q) = wq + b$ with scalar $w, b$ shared across actions. Bounded scalings: residual network $r$ is $|A| \to 64 \to |A|$ with ReLU; $\varepsilon = 0.2$ throughout (an $\varepsilon$ sweep does not robustly buy return: $2.08 \pm 0.56$ at $\varepsilon$=0.5 and $1.58 \pm 0.19$ at $\varepsilon$=1.0 versus $1.87 \pm 0.16$ at $\varepsilon$=0.2, three runs each on the Breakout stress test; larger deviation comparisons trade the bounded scaling's consistency for variance). The bounded scaling's stationary-point bound $\varepsilon \|\mathcal{B}Q\|_\infty/(1-\gamma)$ is conditional and can be numerically loose near $\gamma \to 1$ (a one-state example at $\gamma$=0.99, $\varepsilon$=0.2 satisfies it with $\hat{Q}$ an order of magnitude below $Q^*$); calibration is therefore always also measured directly, via the bias probes. The bounded scaling uses a sigmoid, the additive bound $\tanh$, the ablation variants $\tanh \circ \mathrm{relu}$ and $\exp \circ \mathrm{sigmoid}$: the smooth strictly positive factor $q\, e^{\varepsilon\sigma(r(q))}$ also recovers the true values on the

Table A6: Return as a percentage of the environment-time-matched full target (100), interquartile mean with diverged runs scored as zero, MSE and UTD 8; the last column is ALE Breakout final return relative to the full copy. Last layer and iterated heads are the iS-QN architectures with $K$=1 and $K$=2.

| method (stored) | Breakout | Space Inv. | Seaquest | Asterix | mean | ALE |
|---|---|---|---|---|---|---|
| no target (0) | 0 | 0 | 0 | 0 | 0 | 62 |
| last layer, iS-QN $K$=1 | 0 | 0 | 0 | 0 | 0 | 68 |
| iterated heads, iS-QN $K$=2 | 102 | 95 | 0 | 0 | 49 | n/a |
| full copy, usual period (100%) | 107 | 31 | 0 | 0 | 35 | 100 |
| learned scaling (0) | 82 | 117 | 108 | 53 | 90 | 64 |
| constant scaling $1.1q$ (0) | 97 | 131 | 112 | 161 | 125 | 103 |
| scalar predictor (0) | 108 | 126 | 112 | 178 | 131 | 74 |

chain and survives the Breakout stress test (2/2), while the dead-zone variant $\tanh \circ \mathrm{relu}$ fails in one of two runs, so the deviation channel must stay responsive everywhere. The input bottleneck is a rank-1 linear projection $|A| \to 1 \to |A|$ with the ReLU at the one-dimensional bottleneck. Initialization: the affine and scalar heads start at the identity ($w$=1, $b$=0); the residual network $r$ uses PyTorch's default initialization, so the bounded scaling's factor at step 0 is $1 + \varepsilon\sigma(r(q))$ with $r(q)$ small and mildly action-dependent. Predictor parameters are trained with the same optimizer and learning rate as the value network.

## D PROOF DETAILS

**Prop. 2 (backbone block of the update Jacobian).** In the two-state instance with online value $g\theta\phi(s) + b$ and stored scale-and-shift $(\bar{g}, \bar{b})$, sampling the $s_1 \to s_2$ transition, the expected TD error is $\delta = \gamma(2\bar{g}\theta + \bar{b}) - (g\theta + b)$, and the semi-gradient update on the backbone is $\dot{\theta} = \alpha\delta\, g$ (the factor 2 from the squared loss is absorbed into the step size $\alpha$) (the online branch's gradient with respect to $\theta$ is $g\phi$). Weighting convention: all $2\gamma - 1$ coefficients below are for this single-transition sampling; uniform state weighting of both transitions replaces the threshold factor by $6\gamma - 5$ (the same feature-weighted sum $\delta_1 + 2\delta_2$ as in Prop. 6) and changes no sign structure. For target-free TD the bootstrap uses the live $(g, b)$, so $\partial\delta/\partial\theta = (2\gamma - 1)g$ and $\partial\dot{\theta}/\partial\theta = \alpha g^2(2\gamma - 1)$. For the scale-and-shift target the bootstrap uses the frozen $(\bar{g}, \bar{b})$ but the live $\theta$, so $\partial\delta/\partial\theta = 2\gamma\bar{g} - g$, which equals $(2\gamma - 1)g$ at synchronization ($\bar{g} = g$): the same expansive coefficient. For the full target the bootstrap is constant in $\theta$, so $\partial\delta/\partial\theta = -g$ and the block is $-\alpha g^2$.

**Prop. 3 (coupled saddle).** With $b$ suppressed, $\delta = \theta(2\gamma\bar{g} - g)$, $\dot{\theta} = \alpha\delta g$, $\dot{g} = \alpha\delta\theta$. At synchronization, $\partial\dot{\theta}/\partial\theta = \alpha g^2(2\gamma - 1)$, $\partial\dot{\theta}/\partial g = \alpha\theta g(2\gamma - 2)$, $\partial\dot{g}/\partial\theta = 2\alpha\theta g(2\gamma - 1)$, $\partial\dot{g}/\partial g = -\alpha\theta^2$, giving $\det J = \alpha^2 g^2\theta^2\big[-(2\gamma - 1) - 2(2\gamma - 1)(2\gamma - 2)\big] = \alpha^2 g^2\theta^2(2\gamma - 1)(3 - 4\gamma)$, negative for $\gamma > 3/4$. The determinant is also negative for $\gamma < \frac{1}{2}$; there is no contradiction with the observed convergence in that setting, because the linearization point is not an equilibrium. The flow's equilibria are the set $\{\theta = 0\}$ (where $\delta = 0$ for every $g$), and the Jacobian there has eigenvalues $\alpha g^2(2\gamma - 1)$ and 0: linearly stable exactly when $2\gamma < 1$. The saddle statement characterizes the instantaneous coupled flow at synchronization points away from equilibrium. Trajectory statements need the synchronization map: the frozen inner flow started from $\theta = g$ converges to $g = 2\gamma\bar{g}$ (it does not diverge within one loop), so in the slow-synchronization limit each synchronization multiplies the scale by $2\gamma$ — geometric divergence exactly when $2\gamma > 1$ — while the $T$=20 numerics of §3 show finite-$T$ scale-and-shift trajectories additionally tracking target-free TD's faster per-step class through the saddle direction.

**Prop. 4 (blocks and reduction).** Perturbing at a synchronization point, the online and bootstrap functions differ only through $\omega$: $du = J_\theta d\theta + J_\omega d\omega$ and $dv = J_\theta d\theta$ (the stored $\bar{\omega}$ is constant). Hence $d\delta = (\gamma P - I)J_\theta\, d\theta - J_\omega\, d\omega$, and pre-multiplying by $\alpha J_\theta^\top D$ and $\alpha J_\omega^\top D$ gives the blocks of Eq. (2); $-G = -\alpha J_\omega^\top D J_\omega$ is a negated Gram matrix, hence negative semidefinite. For (ii), setting $\dot{d\omega} = 0$ gives $d\omega^* = G^\dagger C\, d\theta$ with $G^\dagger$ the pseudoinverse (the inverse on the range of $G$; directions in $\ker J_\omega$ do not act on the function); substituting, $A - BG^\dagger C = \alpha J_\theta^\top D(\gamma P - I)J_\theta -$

Table A7: IQM and 95% bootstrap confidence intervals for Tables 1 and 2 and the Atari storage levels (d = diverged runs out of five). Mean, IQM, and CI are over survivors; $\text{IQM}_0$ is the unconditional statistic over all five runs with diverged runs scored at the return floor of zero, so survivor conditioning cannot flatter high-failure arms (it equals IQM where nothing diverged). Where fewer than three runs survive, the IQM column is the survivor mean. Under $\text{IQM}_0$ the full copy's Space Invaders entry falls from 7.26 to 2.39 while the zero-death adapter rows are unchanged.

| setting | arm | d | mean ± std | IQM | 95% CI | $\text{IQM}_0$ |
|---|---|---|---|---|---|---|
| Breakout storage levels | last two layers | 0 | $2.41 \pm 0.04$ | 2.42 | [2.36, 2.44] | 2.42 |
| | full copy | 0 | $2.42 \pm 0.03$ | 2.43 | [2.39, 2.45] | 2.43 |
| | full copy, period matched | 0 | $2.28 \pm 0.03$ | 2.27 | [2.25, 2.31] | 2.27 |
| Breakout comparison | mellowmax $\beta$=1 | 0 | $2.83 \pm 0.08$ | 2.83 | [2.75, 2.91] | 2.83 |
| | LayerNorm + $\ell_2$ | 0 | $2.50 \pm 0.09$ | 2.48 | [2.44, 2.60] | 2.48 |
| | single-scalar predictor | 0 | $2.46 \pm 0.11$ | 2.46 | [2.36, 2.58] | 2.46 |
| | LayerNorm | 0 | $2.45 \pm 0.12$ | 2.44 | [2.34, 2.58] | 2.44 |
| | constant scaling | 0 | $2.21 \pm 0.05$ | 2.21 | [2.16, 2.27] | 2.21 |
| | iterated heads $K$=2 | 0 | $2.31 \pm 0.03$ | 2.32 | [2.29, 2.34] | 2.32 |
| | bounded scaling | 0 | $1.87 \pm 0.16$ | 1.86 | [1.72, 2.05] | 1.86 |
| Space Invaders comparison | mellowmax $\beta$=1 | 1 | $11.32 \pm 0.12$ | 11.32 | [11.22, 11.44] | 11.29 |
| | constant scaling | 0 | $10.03 \pm 0.16$ | 10.04 | [9.87, 10.20] | 10.04 |
| | single-scalar predictor | 0 | $9.63 \pm 0.14$ | 9.65 | [9.47, 9.76] | 9.65 |
| | bounded scaling | 0 | $8.94 \pm 1.24$ | 8.99 | [7.58, 10.18] | 8.99 |
| | iterated heads $K$=2 | 0 | $7.31 \pm 0.33$ | 7.33 | [6.96, 7.64] | 7.33 |
| | full copy | 3 | $7.26 \pm 0.13$ | 7.26 | n/a | 2.39 |
| | full copy, period matched | 0 | $7.70 \pm 0.18$ | 7.68 | [7.53, 7.90] | 7.68 |
| Seaquest comparison | single-scalar predictor | 0 | $0.27 \pm 0.03$ | 0.27 | [0.24, 0.30] | 0.27 |
| | constant scaling | 0 | $0.27 \pm 0.02$ | 0.27 | [0.25, 0.30] | 0.27 |
| | bounded scaling | 0 | $0.26 \pm 0.02$ | 0.26 | [0.24, 0.28] | 0.26 |
| | mellowmax $\beta$=1 | 1 | $0.26 \pm 0.04$ | 0.24 | [0.23, 0.31] | 0.24 |
| | full copy, period matched | 0 | $0.24 \pm 0.02$ | 0.24 | [0.21, 0.26] | 0.24 |
| Asterix comparison | single-scalar predictor | 0 | $0.62 \pm 0.06$ | 0.64 | [0.56, 0.67] | 0.64 |
| | constant scaling | 0 | $0.58 \pm 0.01$ | 0.58 | [0.56, 0.59] | 0.58 |
| | bounded scaling | 3 | $0.56 \pm 0.01$ | 0.56 | n/a | 0.19 |
| | full copy, period matched | 2 | $0.56 \pm 0.03$ | 0.56 | [0.53, 0.58] | 0.36 |
| | min of copy and online, Space Inv. | 2 | $7.10 \pm 0.31$ | 7.10 | [6.90, 7.46] | 4.61 |
| Lower-discount equivalent (no target, $\gamma$=0.9, $r/1.1$) | Breakout | 0 | $2.27 \pm 0.03$ | 2.26 | [2.24, 2.30] | 2.26 |
| | Space Inv. | 0 | $9.90 \pm 0.31$ | 10.00 | [9.57, 10.12] | 10.00 |
| | Seaquest | 0 | $0.24 \pm 0.04$ | 0.24 | [0.20, 0.28] | 0.24 |
| | Asterix | 0 | $0.59 \pm 0.03$ | 0.59 | [0.56, 0.62] | 0.59 |
| Atari storage levels | no target | 0 | $8.52 \pm 2.66$ | 8.88 | [5.44, 10.86] | 8.88 |
| | last layer | 0 | $9.34 \pm 1.00$ | 9.27 | [8.36, 10.45] | 9.27 |
| | last two layers | 0 | $11.50 \pm 5.64$ | 13.83 | [5.46, 14.53] | 13.83 |
| | full copy | 0 | $13.69 \pm 7.08$ | 16.27 | [5.69, 18.15] | 16.27 |
| | learned scaling (zero storage) | 0 | $8.81 \pm 1.85$ | 8.66 | [6.93, 10.73] | 8.66 |
| | scalar predictor (zero storage) | 0 | $10.19 \pm 1.78$ | 10.47 | [8.28, 11.80] | 10.47 |
| | constant $1.1q$ (zero storage) | 0 | $14.13 \pm 0.43$ | 14.04 | [13.76, 14.62] | 14.04 |

$\alpha J_\theta^\top D J_\omega (J_\omega^\top D J_\omega)^{-1} J_\omega^\top D(\gamma P - I) J_\theta = \alpha K^\top (I - \Pi_\omega) W K$ after inserting $D^{1/2} D^{-1/2}$ and collecting $K = D^{1/2} J_\theta$, $W = D^{1/2}(\gamma P - I) D^{-1/2}$, $\Pi_\omega = D^{1/2} J_\omega (J_\omega^\top D J_\omega)^{-1} J_\omega^\top D^{1/2}$.

**Exact field versus tangent model.** Differentiating the exact update $\dot\omega = \alpha J_\omega(\theta, \omega)^\top D\delta$ gives $\partial\dot\omega/\partial\omega = -\alpha J_\omega^\top D J_\omega + \alpha \sum_i d_i \delta_i \nabla^2_{\omega\omega} F_i$, and analogous residual-weighted curvature terms enter the mixed blocks; Eq. (2) is the tangent model that drops them. They vanish identically when $F$ is affine in the differentiated parameters (the self-block for a last-layer head or single scale), and under $J_\omega \mapsto \kappa J_\omega$ the self-block correction scales as $O(\kappa)$ against $G = O(\kappa^2)$, so the boundary layer and the Gram negativity are unaffected as $\kappa \to \infty$. The mixed-block corrections $\Delta B = -\alpha \sum_i d_i \delta_i \nabla^2_{\theta\omega} F_i$ and $\Delta C = \alpha \sum_i d_i \delta_i \nabla^2_{\omega\theta} F_i$ are different: when $\theta$ and $\omega$ interact multiplicatively

(as in feature scale-and-shift) they scale as $O(\kappa)$ like $B$ and $C$ themselves, so the Schur complement $BG^{\dagger}C$ keeps an $O(1)$ correction as $\kappa \to \infty$. The reduction of Prop. 4(ii) therefore converges to the projected tangent operator exactly when the adapter enters additively ($\nabla^2_{\theta\omega} F = 0$, as for a last-layer head) or when the residual $\delta$ is small, which the fast adapter produces by absorbing the TD gap; otherwise it holds up to that $O(1)$ term. They are *not* negligible at $\kappa \approx 1$: in the scalar instance of Prop. 3 ($\alpha$=1, $\gamma$=0.9, $\theta$=$g$=1) the tangent blocks give $\left(\begin{smallmatrix} 0.8 & -1 \\ 0.8 & -1 \end{smallmatrix}\right)$ with eigenvalues $0, -0.2$, while the exact Jacobian is $\left(\begin{smallmatrix} 0.8 & -0.2 \\ 1.6 & -1 \end{smallmatrix}\right)$ with eigenvalues $0.6, -0.8$: the corrections ($-B + \alpha\delta\, \partial_g J_\theta = -1 + 0.8$ and $C + \alpha\delta\, \partial_\theta J_\omega = 0.8 + 0.8$, with $\delta = 0.8$ at the point) create the saddle. The instance that defeats the feedback is exactly the instance where the tangent model breaks.

**The explicit $\kappa$-scaling limit for Prop. 4(ii).** Scale the adapter's leverage as $J_\omega \mapsto \kappa J_\omega$, so $B = \kappa B_1$, $C = \kappa C_1$, $G = \kappa^2 G_1$ with unit-scale blocks $B_1, C_1, G_1$ fixed; both $\Pi_\omega$ (a projection onto a span, invariant under scaling) and the Schur complement $A - BG^{-1}C = A - B_1 G_1^{-1} C_1$ are $\kappa$-invariant. In the rescaled fast variable $u = \kappa\, d\omega$ the system reads $\dot{d\theta} = A\, d\theta - B_1 u$, $\dot{u} = \kappa^2 (C_1\, d\theta - G_1 u)$: a standard singularly perturbed pair with $\varepsilon = \kappa^{-2}$. The boundary-layer system $du/d\tau = C_1\, d\theta - G_1 u$ (frozen $d\theta$, fast time $\tau = \kappa^2 t$) is exponentially stable on $\mathrm{range}(G_1)$, since $G_1$ is a Gram matrix and directions in $\ker J_\omega$ act trivially on the function and are quotiented out; its quasi-static root is $u^* = G_1^{-1} C_1\, d\theta$, the slow manifold. Tikhonov's theorem (Khalil, 2002) then gives convergence of the slow dynamics to the reduction of Prop. 4(ii) on compact time intervals as $\kappa \to \infty$. The time-scale separation assumption of Prop. 4(ii) is the finite-$\kappa$ proxy for this separation, and the ratio $\lambda_{\min}(G|_{\mathrm{range}})/\lambda_{\max}(A)$ is what our leverage diagnostic measures.

**Prop. 6 (gradient-parallel spurious solutions).** With $h(x) = wx$ and value $w\theta\phi(s)$, on the two-transition instance the residuals are $\delta_1 = 2\gamma\theta - w\theta$ and $\delta_2 = 2\gamma\theta - 2w\theta$, and the expected updates are $\dot{\theta} = \alpha w(\delta_1 + 2\delta_2)$ and $\dot{w} = \alpha\theta(\delta_1 + 2\delta_2)$: both proportional to the same scalar $\delta_1 + 2\delta_2 = \theta(6\gamma - 5w)$, which vanishes at $w = 6\gamma/5$ for any $\theta$, with $\delta_1, \delta_2 \neq 0$ there.

**Prop. 7 (attraction).** On the single-transition instance, $\delta = (2\gamma - w)\theta$, $\dot{\theta} = \alpha\delta w$, $\dot{w} = \alpha\delta\theta$. On the manifold $\{w = 2\gamma\}$, $\delta = 0$; the Jacobian there is $\left(\begin{smallmatrix} 0 & -\alpha w\theta \\ 0 & -\alpha\theta^2 \end{smallmatrix}\right)$ with eigenvalues 0 (along the manifold) and $-\alpha\theta^2 \leq 0$ (transversal). Numerically ($\gamma$=0.9): from $(\theta, w) = (1, 1)$, $w \to 1.800$, TD error $\to 4 \times 10^{-16}$, $\hat{V}(s_1) = 3.24$ against $V^* = 0$; a perturbed start returns to the same manifold elsewhere ($\hat{V} = 6.91$).

**Prop. 10 (bounded scaling).** A zero-loss stationary point satisfies $Q(s,a)(1+\varepsilon\sigma_{sa}) = (\mathcal{B}Q)(s,a)$ with $\sigma_{sa} \in (0,1)$, i.e., $Q = M\mathcal{B}Q$ with $M = \mathrm{diag}\big((1+\varepsilon\sigma_{sa})^{-1}\big)$, $\|M\|_\infty < 1$. Since $\|M\|_\infty < 1$, any self-consistent solution obeys $\|Q\|_\infty \leq \|\mathcal{B}Q\|_\infty \leq \|r\|_\infty + \gamma\|Q\|_\infty$ with $\|r\|_\infty = \sup_{s,a} |r(s,a)|$ (the absolute bound is needed because rewards may be negative), hence $\|Q\|_\infty \leq \|r\|_\infty/(1-\gamma)$: no divergent self-consistent solution exists (in contrast to the two-sided factor $(1-\varepsilon)^{-1}$ of Prop. 9; note $M$ varies with $Q$, so the bound is by domination rather than by contraction of a fixed map). For the error bound, $\|Q - Q^*\|_\infty \leq \|M\mathcal{B}Q - \mathcal{B}Q\|_\infty + \|\mathcal{B}Q - \mathcal{B}Q^*\|_\infty \leq \varepsilon\|\mathcal{B}Q\|_\infty + \gamma\|Q - Q^*\|_\infty$, giving $\|Q - Q^*\|_\infty \leq \varepsilon\|\mathcal{B}Q\|_\infty/(1-\gamma)$; with $\|\mathcal{B}Q\|_\infty \leq \|r\|_\infty/(1-\gamma)$ this is the closed form $\varepsilon\|r\|_\infty/(1-\gamma)^2$. The agent acts on $\hat{Q} = h(Q) = \mathcal{B}Q$ at such a point, and $\|\mathcal{B}Q - \mathcal{B}Q^*\|_\infty \leq \gamma\|Q - Q^*\|_\infty$, so the deployed estimate inherits the bound with an extra factor $\gamma$.

**Negative-feedback validation details (§4).** (a) On a two-state instance with a full-span adapter and off-policy weighting, the divergence-to-stability transition lands in the predicted leverage window ($\kappa^* \approx 0.97$; observed 0.5–1.0). (b) On Baird's star problem, seeding the adapter's span with the *unstable mode*: a frozen copy rescues it at high leverage, while the *same adapter* updating with the online parameters accelerates divergence monotonically (death at 5,940 steps at $\kappa$=0 versus 88 at $\kappa$=30). The alignment experiment also surfaced a second channel, *span enlargement*: high-leverage per-state features stabilize Baird regardless of frozen copy or alignment (they drive the dynamics toward the tabular block $D(\gamma P - I)$, Hurwitz by an M-matrix argument, not a property of arbitrary full-rank features; Baird's divergence is span-sensitive), and the two-factor design separates the channels: on the aligned mode, span enlargement cannot compensate the positive feedback of an adapter updating with the online parameters. Remaining gaps, stated plainly: the $\kappa$-limit is

asymptotic (compact time intervals), the control ($\max$) version is piecewise, and the threshold constant is order-of-magnitude on Baird (predicted $1.3$, observed 3–10). The time-scale condition also discriminates *within* linear instances: the separation ratio $\lambda_{\min}(G)/\lambda_{\max}(A)$ is $0.16$ in the scalar counterexample versus $1.4$ on Baird (Table A4), so the same scale-and-shift target diverges in one and rescues the other. Architecture sensitivity follows from the leverage condition, since $\partial F/\partial\omega$ shrinks over convolutional features. In the two-state instance of §3, when $2\gamma < 1$ everything converges: the equilibria $\{\theta{=}0\}$ are linearly stable exactly there.