# SHORT-TO-LONG: DETACHED FEATURES FOR LONG-HORIZON VALUE LEARNING

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

## ABSTRACT

Learning useful representations from continually changing value targets remains a challenge in deep reinforcement learning. We study the temporal horizon of this supervision: smaller discounts make these targets less sensitive to distant bootstrap changes, but using them for behavior changes the control objective. We introduce *Short-to-Long* (S2L), which supplies a long-horizon behavior learner with features trained by smaller-discount source networks. Detached connections prevent the behavior value loss from updating the sources, allowing each source to follow its own value objective while the behavior discount remains unchanged. Across Atari, online continuous control, and offline continuous control, the approach yields aggregate performance gains, improving the paired terminal median on 40 of 57 Atari games. Matched Atari ablations support the roles of feature transfer, smaller source discounts, and gradient isolation. Fixed-trajectory measurements further show reduced drift in source value targets with mixed discounts.

## 1 INTRODUCTION

Deep reinforcement learning relies on representations that remain useful as an agent's predictions and behavior change. Learning these representations is difficult because the supervision itself is non-stationary: policy updates change the experience distribution, while value updates change the bootstrap predictions used as targets. In an end-to-end value network, these targets also shape the hidden features on which future predictions depend. This feedback connects representation learning to the optimization difficulties observed in deep RL (Sokar et al., 2023; Castanyer et al., 2025; Saheb Pasand et al., 2026; Le Lan et al., 2023; Tang & Munos, 2023).

Most responses change the architecture or the optimizer: preserving gradient flow across depth (Castanyer et al., 2025), residual critics (Lee et al., 2025), or auxiliary prediction targets that shape representations (Jaderberg et al., 2017; Schwarzer et al., 2021). The temporal structure of the value supervision itself is left untouched. We examine that horizon. In an end-to-end value network, the discount factor determines both how future rewards enter the control objective and the targets used to train hidden features. These two roles need not be coupled for every source of features.

Multiple discounts have appeared in deep RL before, but always within one set of parameters: auxiliary heads on a shared trunk (Fedus et al., 2019), or a decomposition of one long-horizon value into differences between neighboring discounts (Romoff et al., 2019; Tang et al., 2021). Every objective then updates the same features, so the longest-horizon loss still shapes them.

Along a fixed trajectory, reducing the discount attenuates the contribution of a distant bootstrap change to the current target (Amit et al., 2020; Kozuno et al., 2021). We make this quantitative: a change $k$ steps away is transmitted with a factor $\gamma^k$, at $k = 32$ some $21.1\times$ smaller for $\gamma = .9$ than for $\gamma = .99$. A shared-feature analysis further shows that the longest-horizon loss dominates the feature update unless its gradient is removed, which motivates isolating the sources. Applying the same reduction to the behavior learner would change how it values future rewards. We instead investigate whether shorter-horizon value objectives can supply useful features for a learner that retains its original control objective.

*Short-to-Long* (S2L) implements this idea with separately trained value networks that share experience. Source learners use smaller discounts and pass their hidden activations to the receiving learner through stop-gradient connections. Each source is updated only by its own value loss, so the receiving loss



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.

cannot redirect it toward the longer-horizon target. All learners remain active throughout training, and only the final network supplies the action values or actor objective. The behavior learner can therefore use shorter-horizon features while continuing to optimize its original discounted objective. Figure 1 shows the four-learner instantiation used in our main online experiments.

**Contributions.**

- S2L: detached feature transfer from smaller-discount source networks into a behavior learner whose discount and control objective are unchanged.
- An account of how the discount governs the transmission of bootstrap changes into truncated $\lambda$-returns, and a shared-feature model (Theorem 1) in which the longest-horizon loss dominates the feature update unless its gradient is removed.
- Gains in all three settings—$+.049$ median paired human-normalized score over the 57-game Atari suite relative to scaled PQN (Castanyer et al., 2025), 465 of 720 matched DeepMind Control Suite pairs, and nine offline MuJoCo datasets—with matched ablations isolating each design choice, direct target-drift measurements, and a parameter-matched shared-trunk control.

## 2 Short-to-Long

S2L-4 consists of four value learners ordered from shorter to longer horizon. Each learner has separate parameters, normalization statistics, optimizer state, and targets. They share training transitions, not parameters.

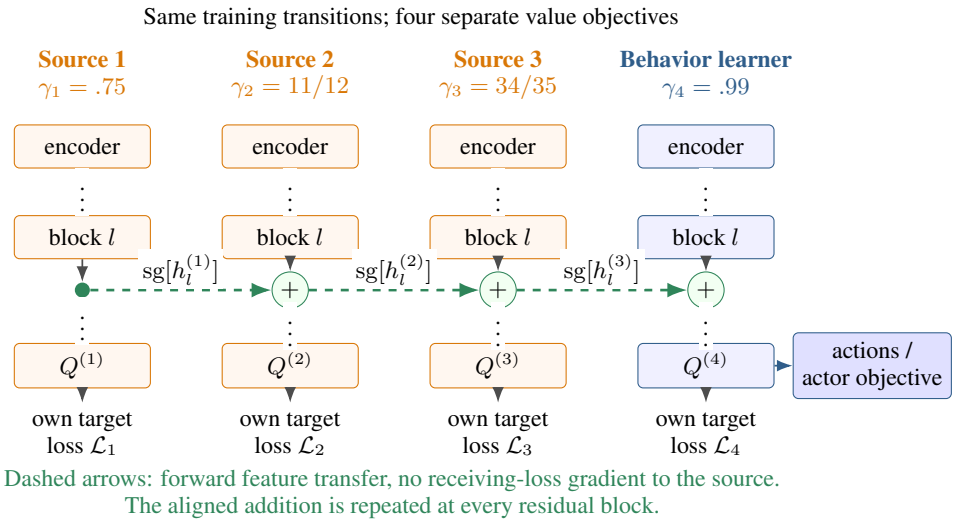



Figure 1: **S2L-4: learn separately, transfer features from short to long.** All four learners train on the same transitions. At each residual depth, a learner receives its predecessor's activation through a detached addition. Each value loss updates only its own network. The final $\gamma = .99$ learner supplies action values or the actor objective.

### 2.1 Four horizons, one behavior objective

We use the fixed discounts

$$(\gamma_1, \gamma_2, \gamma_3, \gamma_4) = \left(.75, \frac{11}{12}, \frac{34}{35}, .99\right), \qquad \frac{1}{1-\gamma_k} \in \{4, 12, 35, 100\}. \tag{1}$$

The final learner, $Q^{(4)}$, retains the base algorithm's control objective and Bellman-target construction. The first three learners supply features; their Q-values are not averaged into the behavior value or used to mix its targets. In Atari, actions are epsilon-greedy with respect to $Q^{(4)}$. In continuous control, the actor is optimized using only its value.

### 2.2 DETACHED ADDITIONS

Let $\bar{h}_l^{(k)} = f_l^{(k)}(h_{l-1}^{(k)})$ be learner $k$'s activation after its ordinary block, including residual and skip connections. Evaluate the networks from shortest to longest horizon and add the preceding learner's same-depth activation:

$$h_l^{(1)} = \bar{h}_l^{(1)}, \qquad h_l^{(k)} = \bar{h}_l^{(k)} + \mathrm{sg}\left[h_l^{(k-1)}\right], \quad k = 2, 3, 4. \tag{2}$$

Here sg is the identity in the forward pass and stops gradients in the backward pass. There are no learned gates or projection parameters. Because the predecessor activation already includes its own transfer, information can pass through the entire short-to-long chain.

If $\theta_j$ denotes learner $j$'s parameters and $\mathcal{L}_k$ denotes learner $k$'s value loss, then

$$\nabla_{\theta_j} \mathcal{L}_k = 0 \quad \text{for } j < k. \tag{3}$$

Each network nevertheless continues to learn from its own loss. The two-learner interfaces used in the development studies follow the same principle; their definitions are in Appendix C.

### 2.3 TRAINING AND DOMAIN INSTANTIATIONS

For online Atari, the four CNN–MultiSkip networks each form a Q($\lambda$) return. With $V^{(k)}(s) = \max_a Q^{(k)}(s, a)$ and terminal masks omitted from the notation,

$$G_t^{(k)} = r_t + \gamma_k \left[(1 - \lambda) V^{(k)}(s_{t+1}) + \lambda G_{t+1}^{(k)}\right], \qquad G_T^{(k)} = V^{(k)}(s_T). \tag{4}$$

All learners regress on the action $a_t$ actually taken:

$$\mathcal{L}_k = \mathbb{E}\left[(Q^{(k)}(s_t, a_t) - \mathrm{sg}[G_t^{(k)}])^2\right], \qquad \mathcal{L} = \sum_{k=1}^{4} \mathcal{L}_k. \tag{5}$$

The losses are summed without averaging and gradients are clipped separately per learner. We use $\lambda = .65$ and 32-step rollouts.

For online continuous control, four disjoint SimBa critics replace the single critic and use one-step targets with their respective discounts. The actor and interaction protocol are unchanged. Critic updates detach predecessor features as in Eq. (2); actor updates hold all critic parameters fixed while retaining action derivatives through all four critics. Atari and continuous-control settings are recorded in Appendices B, H, and I.

**Continuous-control update.** Each replay batch is processed in four steps:

1. Form four one-step targets from the same transitions using the discounts in Eq. (1), including the corresponding SAC entropy term when applicable.
2. Evaluate the critics from shortest to longest horizon, adding the immediate predecessor's activation at each aligned residual block.
3. Stop gradients through predecessor features, sum the four TD losses, and update each critic with its own optimizer state.
4. Update the actor from the final .99 value. Keep predecessor action derivatives live and critic parameters fixed, then apply the base algorithm's target-network update.

The same transferred activations therefore support both value learning and policy improvement: detachment isolates the critics' TD losses, while action derivatives carry their influence into the actor update.

## 3 ONLINE CONTROL RESULTS

We compare S2L-4 with scaled PQN on Atari and with SimBa on continuous control, matching environment-interaction budgets and paired seeds. Each S2L learner uses the baseline network architecture; total capacity scales with the number of learners. Value networks at this scale are small enough that throughput is dominated by simulation rather than network evaluation, which is why deep RL comparisons use matched interaction budgets rather than matched compute; across the 57 games the four learners add $12.3 \pm 6.2\%$ wall-clock training time.

### 3.1 ATARI: GAINS ACROSS THE FULL 57-GAME SUITE

S2L-4 improves terminal performance on 40 of 57 games, with a median paired game improvement of $+.049$ human-normalized score (HNS). Figure 2 reports the complete per-game comparison. Using a descriptive $\pm.01$ band, 36 games improve by more than $+.01$, 11 fall within the band, and 10 fall below $-.01$.

**Setup.** We compare with the scaled PQN agent of Castanyer et al. (2025) using four paired seeds and 2.50M environment steps per game. Five games were used for development; the other 52 were not. We define $\text{HNS} = (R_{\text{agent}} - R_{\text{random}})/(R_{\text{human}} - R_{\text{random}})$. For each run, terminal HNS averages the last ten logged values, each of which averages the 20 most recently completed training episodes. Adjacent windows may overlap. We form differences within seed and report the four-seed median for each game.

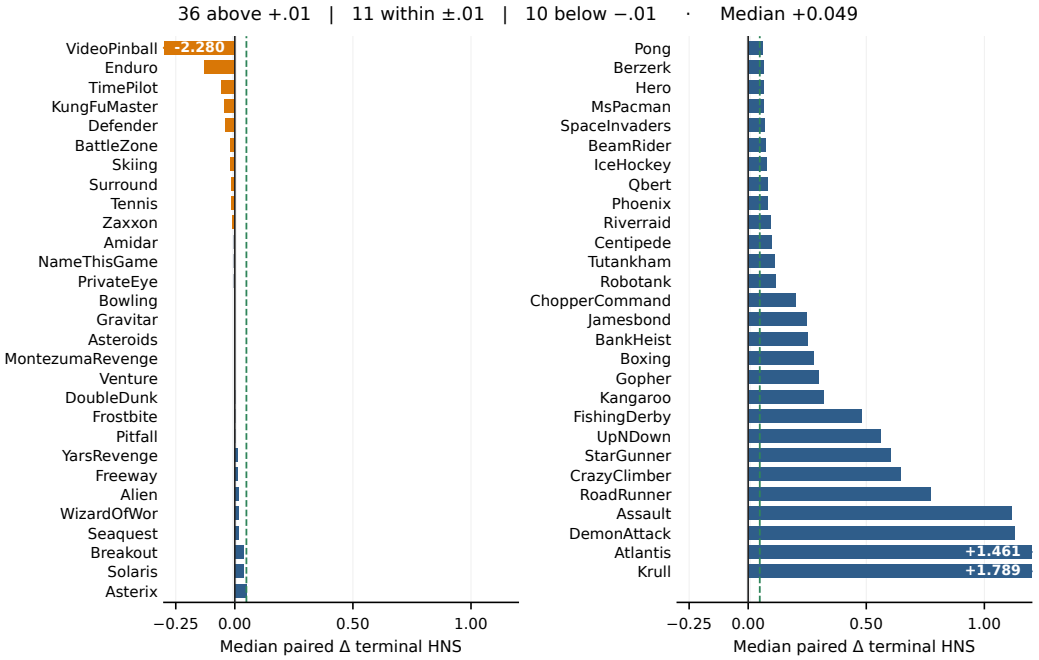



Figure 2: **S2L-4 improves the paired median on 40 of 57 Atari games.** Bars show median paired terminal HNS differences from scaled PQN, ordered from the smallest to largest effect across the two columns. Blue, gray, and orange indicate values above $+.01$, within $\pm.01$, and below $-.01$. The dashed line marks the median across games. Effects outside $[-.30, +1.20]$ are annotated with their actual values.

### 3.2 CONTINUOUS CONTROL: LARGER GAINS LATE IN TRAINING

At one million environment steps, S2L-4 improves return in $465/720$ matched pairs, with a mean difference of $+39.09$. The advantage emerges late: the time-averaged difference is $-12.71$ over the first half, $+27.03$ over the second, and $+34.99$ over the final quarter; the full-training value of $+7.16$ therefore averages two regimes of opposite sign (Figure 3). Every subgroup has a negative first-half difference but a positive endpoint mean (Table 1). One plausible explanation is that source features must mature before their transfer becomes useful.

**Setup.** The matrix covers SAC and DDPG (Haarnoja et al., 2018; Lillicrap et al., 2016), AdamW and Kron, and four DeepMind Control Suite tasks: dog-run, dog-trot, humanoid-run, and humanoid-walk (Tassa et al., 2018). Paired methods use the same actor architecture, individual-critic width and depth, seed, and sample budget. We analyze 720 full-budget pairs across nine scales and five seeds. Appendix H gives the coverage and scale definitions.

**Retaining acquired return.** Over 500k–1M steps, S2L-4 has lower aggregate local fluctuation, maximum drawdown, cumulative negative change, and worst single-evaluation drop (Table 2). These return-dynamics measures quantify retention of acquired return and are distinct from the target drift in Section 4.1. Appendix H gives definitions and cross-seed variability.

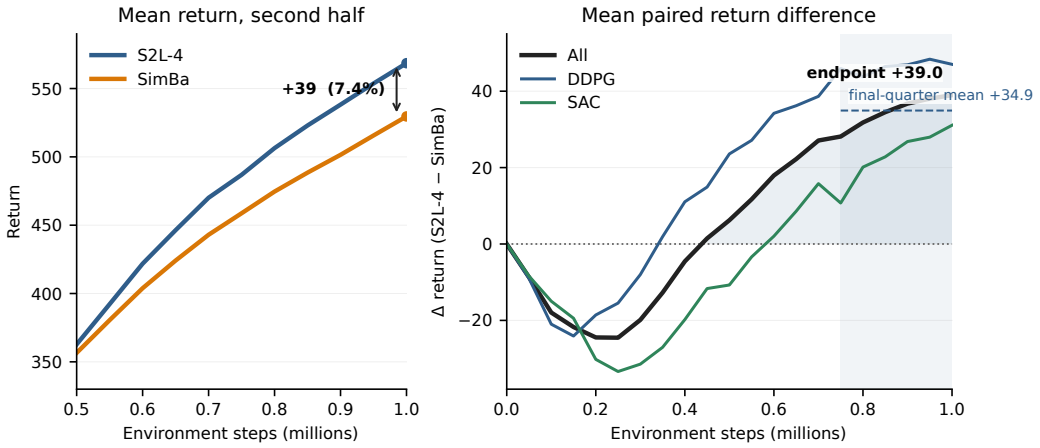



Figure 3: **The S2L-4 return advantage grows during training.** Both panels use the same 720 matched pairs. Left: mean return over the second half of training. Right: mean paired difference over the full run, overall and by algorithm. The shaded window marks the final quarter of training, over which the mean paired difference is $+34.99$.

Table 1: **S2L gains over 720 matched pairs.** Entries combine all 720 pairs. Differences are S2L-4 minus SimBa; AUCs are divided by their window length. W counts positive endpoint pairs.

| Subset | $n$ | Final return | | Δ AUC | | |
|---|---|---|---|---|---|---|
| | | Mean Δ | W | 0–1M | 500k–1M | 750k–1M |
| All | 720 | 39.09 | 465 | 7.16 | 27.03 | 34.99 |
| DDPG | 360 | 47.42 | 242 | 17.29 | 40.66 | 46.66 |
| SAC | 360 | 30.76 | 223 | −2.97 | 13.40 | 23.31 |
| AdamW | 360 | 47.49 | 252 | 16.84 | 36.61 | 42.79 |
| Kron | 360 | 30.68 | 213 | −2.52 | 17.45 | 27.18 |
| dog-run | 180 | 58.65 | 129 | 7.24 | 30.85 | 47.04 |
| dog-trot | 180 | 51.26 | 111 | 21.88 | 51.73 | 57.69 |
| humanoid-run | 180 | 11.04 | 115 | −0.10 | 4.99 | 21.76 |
| humanoid-walk | 180 | 35.41 | 110 | −0.36 | 20.57 | 67.17 |

Table 2: Late-training return dynamics over 720 completed matched pairs. Statistics use eleven evaluations from 500k to 1M steps. Lower detrended MAD means less local fluctuation; lower values in the other rows mean smaller downward return changes. The last column counts pairs with a lower S2L-4 value / lower SimBa value / an exact tie.

| Metric | S2L-4 | SimBa | Paired W/L/T |
|---|---|---|---|
| Local fluctuation (detrended MAD) | **12.08** | 16.17 | 404/316/0 |
| Normalized maximum drawdown | **0.137** | 0.164 | 426/276/18 |
| Cumulative negative change | **38.26** | 45.32 | 412/290/18 |
| Worst 50k-step drop | **23.85** | 26.62 | 415/287/18 |

# 4 UNDERSTANDING THE DESIGN

We next examine the three choices that distinguish S2L: transferring features, training the sources with smaller discounts, and blocking the receiving loss. We first analyse what smaller discounts do to targets and what detachment does to feature updates, measuring target drift directly; Atari controls then test the effect of each choice on return.

## 4.1 TARGET SENSITIVITY AND MEASURED DRIFT

We first isolate how the discount affects target construction, then measure source-target drift during training. Fix a trajectory from $t$ to truncation index $T$ and hold rewards fixed. For $0 \le \gamma < 1$ and $\lambda \in [0, 1]$, let $G$ and $\widetilde{G}$ be the returns in Eq. (4) formed from bootstrap sequences $V$ and $\widetilde{V}$. Writing $\Delta V_j = V_j - \widetilde{V}_j$ and initially omitting terminal masks, exact unrolling gives

$$\Delta G_t = (1-\lambda) \sum_{k=1}^{n-1} \gamma^k \lambda^{k-1} \Delta V_{t+k} + \gamma^n \lambda^{n-1} \Delta V_T, \qquad n = T - t. \tag{6}$$

Thus a change $k$ transitions away carries a factor $\gamma^k$; the remaining factor depends on $\lambda$ but not on the discount. For the same perturbation, changing the discount from $\gamma_\ell$ to $\gamma_s < \gamma_\ell$ multiplies its contribution by $(\gamma_s/\gamma_\ell)^k$. At $\lambda = 1$, only the terminal term remains, with weight $\gamma^n$. At $k = 32$, changing $\gamma = .99$ to $\gamma = .9$ reduces the transmitted perturbation by $21.1\times$.

When bootstrap values at every depth can change, their contributions add. If $\max_{t<j\le T} |V_j - \widetilde{V}_j| \le \epsilon$, then

$$|G_t - \widetilde{G}_t| \le c_n(\gamma, \lambda)\epsilon, \qquad n = T - t, \tag{7}$$

where

$$c_n(\gamma, \lambda) = \gamma(1-\lambda)\frac{1 - (\gamma\lambda)^n}{1 - \gamma\lambda} + (\gamma\lambda)^n. \tag{8}$$

The coefficient is tight over arbitrary bootstrap sequences in the sup norm: equality is attained when all bootstrap changes have the same magnitude and sign. Standard terminal masks can only remove terms and reduce it. Equation (7) is the finite-horizon form of standard $\lambda$-return sensitivity results (Kozuno et al., 2021). The bound formalizes the attenuation of distant bootstrap changes on fixed trajectories. The proof, including terminal masks, is in Appendix A.

**Measurements during learning.** We compare mixed-discount and all-.99 four-learner models on 16 non-development games and four fresh paired seeds. At 51 training checkpoints, both models recompute targets on the same fixed trajectories from a uniform-random policy and a frozen base policy. The primary statistic is consecutive target RMS drift divided by each learner's iteration-1 target standard deviation; we average the three source learners and both probe sets. Raw RMS drift is a secondary measure.

Both measures favor mixed discounts in all 64 paired runs and all 16 game averages (Table 3). The pattern holds for each source position and both probe sets (Appendix G), linking the target-construction analysis to lower measured source-target drift during learning. As a calibration, all 1,671,168 measured distance-specific coefficients match Eq. (6) to within $1.2 \times 10^{-8}$.

Table 3: **Fixed-trajectory Q($\lambda$)-target drift.** Entries are mixed-discount minus all-.99 differences in time-averaged consecutive Q($\lambda$)-target drift, so negative values favor mixed discounts. The normalized measure divides each learner and probe stratum by its fixed iteration-1 target standard deviation. Intervals use 10,000 whole-game bootstrap resamples, retaining all four paired seeds.

| Target drift | Mean [95% CI] | Median | Pairs $< 0$ | Games $< 0$ |
|---|---|---|---|---|
| Normalized (primary) | $-9.689$ $[-17.638, -3.389]$ | $-2.073$ | 64/64 | 16/16 |
| Raw RMS (secondary) | $-.170$ $[-.249, -.101]$ | $-.127$ | 64/64 | 16/16 |

### 4.2 GRADIENT ISOLATION CONTROLS FEATURE-UPDATE GAIN

Gradient isolation lets source features follow their own objectives. By Eq. (6), the longest-horizon target is the most sensitive to the same bootstrap perturbation; moving value targets are associated with plasticity loss (Sokar et al., 2023). The following local shared-feature model separately quantifies how each head's loss contributes to a feature update and how detachment changes that update.

**Theorem 1** (Discount and shared-feature updates)**.** *Let $\mathcal{O}$ be a finite set of value heads in the following two-dimensional prediction model. Define $e_1 = (1,0)^\top$, $u(\theta) = (\cos\theta, \sin\theta)^\top$, and $Q_o = a_o u(\theta)$ for scalars $a_o, \theta$ and $o \in \mathcal{O}$. The heads share the feature direction $u(\theta)$. For a scale $R > 0$, head $o$ has discount $\gamma_o$, truncation $(n_o, \lambda_o)$, fixed point $q_o^\star = R/(1-\gamma_o)$, and coefficient $c_o = c_{n_o}(\gamma_o, \lambda_o)$. Its stopped target is*

$$Y_o = (1 - c_o) q_o^\star e_1 + c_o \,\mathrm{sg}[Q_o].$$

*Here* sg *is the identity in value and has zero derivative. With the Euclidean norm, let $\ell_o = \frac{1}{2}\|Q_o - Y_o\|_2^2$. For a nonempty $\mathcal{W} \subseteq \mathcal{O}$, one simultaneous semi-gradient step is*

$$a_o^+ = a_o - \eta\, \partial_{a_o} \ell_o, \qquad \theta^+ = \theta - \eta \sum_{o \in \mathcal{W}} \partial_\theta \ell_o,$$

*where all targets and derivatives are evaluated before either update. Define*

$$K_{\mathcal{W}} = R^2 \sum_{o \in \mathcal{W}} \frac{1 - c_o}{(1-\gamma_o)^2}. \tag{9}$$

*At $\theta^\star = 0$ and $a_o^\star = q_o^\star$, the exact local derivative of the angular update is*

$$\left.\frac{\partial \theta^+}{\partial \theta}\right|_{\theta^\star, a^\star} = 1 - \eta K_{\mathcal{W}}. \tag{10}$$

*The linearized angular update is contractive exactly when $0 < \eta K_{\mathcal{W}} < 2$. Removing a loss from $\mathcal{W}$ removes its summand from $K_{\mathcal{W}}$. For fixed $n, \lambda$,*

$$(1 - c_n)\frac{R^2}{(1-\gamma)^2} \sim \frac{R^2}{1-\gamma} \sum_{j=0}^{n-1} \lambda^j, \qquad \gamma \uparrow 1. \tag{11}$$

The proof is in Appendix A.

The fixed-point value scale $R^2/(1-\gamma)^2$ grows faster than $1 - c_n$ shrinks. With the Atari parameters $(n, \lambda) = (32, .65)$, the angular coordinate's linear-stability step-size bound is $11.6\times$ smaller at $\gamma = .99$ than at $\gamma = .9$. Detaching the $\gamma = .99$ loss removes its contribution exactly, allowing source features to retain the update dynamics set by their own objectives. The gradient-isolation ablation in Table 4 tests this prediction directly.

### 4.3 FOUR-LEARNER ABLATIONS AND CAPACITY CONTROL

Table 4 reports the original reference scores and seven controls for S2L-4. Removing feature transfer lowers full-training AUC by .0797; setting all discounts to .99 lowers it by .0652. Allowing receiving losses to backpropagate through all three feature connections lowers full-training and second-half AUC by .0731 and .1530. Paired 95% game-cluster bootstrap intervals on full-training AUC exclude zero for feature transfer ($+.080\,[+.005, +.165]$) and for mixed discounts ($+.065\,[+.023, +.112]$) over all 57 games, and for detachment in the matched two-learner comparison ($+.082\,[+.019, +.174]$) over the 52 non-development games; full-training game medians favor detachment on $42/52$ of them (Appendix F).

A widened, parameter-matched PQN scores .1260 below S2L-4 at the endpoint. Giving all three sources the same short discount, reversing their horizon order, or freezing them at initialization also lowers all three metrics. These comparisons favor trained source features with diverse, short-to-long horizons and isolated gradients. Appendix F defines the controls and retains the earlier two-learner gradient study.

Table 4: **S2L-4 reference scores and ablations.** Panel (a) gives the original 57-game, four-seed mean HNS scores. Panel (b) gives changes relative to the corresponding full S2L-4 model. Endpoint HNS uses iteration 610; AUCs are divided by their window length. Intervals for the original paired controls appear in Table 11.

*(a) Original reference scores*

| Method | Endpoint HNS | Full AUC | Second-half AUC |
|---|---|---|---|
| Scaled PQN | 0.9323 | 0.3161 | 0.5809 |
| S2L-4 | 1.0648 | 0.3999 | 0.6827 |

*(b) Variant minus full S2L-4*

| Variant | Δ endpoint | Δ full AUC | Δ second-half AUC |
|---|---|---|---|
| No gradient isolation | −.0971 | −.0731 | −.1530 |
| No feature transfer | −.1010 | −.0797 | −.0914 |
| All discounts .99 | −.0598 | −.0652 | −.0863 |
| Parameter-matched PQN | −.1260 | −.0791 | −.0962 |
| Identical source discounts | −.1093 | −.0689 | −.0874 |
| Reversed source horizons | −.0762 | −.0482 | −.0593 |
| Frozen source networks | −.0973 | −.0752 | −.0744 |

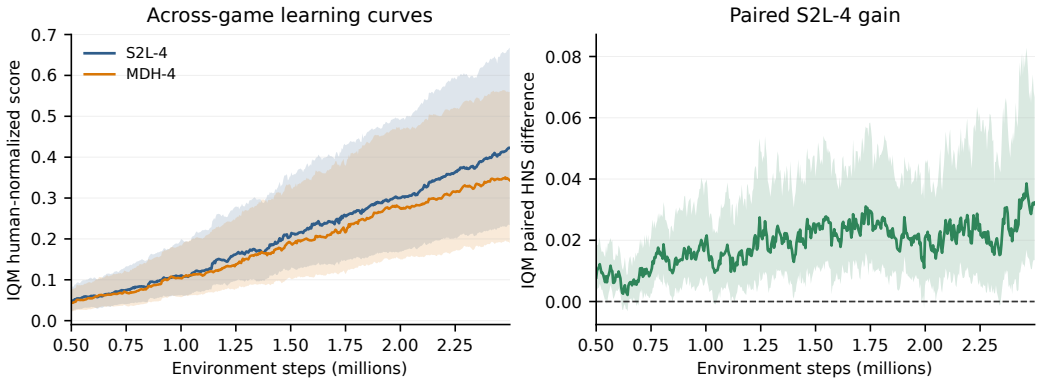



Figure 4: **S2L-4 versus the shared-trunk MDH-4 baseline.** Left: IQM HNS over 228 game–seed runs per method. Right: IQM of paired HNS differences, S2L-4 minus MDH-4. Shading gives 95% game-cluster bootstrap intervals.

### 4.4 Separate learners versus a shared trunk

MDH-4 uses the same discounts but predicts all four values from a single CNN–MultiSkip trunk, adapting the multi-discount auxiliary-head design of Fedus et al. (2019) to PQN. Both architectures use the same interaction budget and four paired seeds on all 57 games. MDH-4 has approximately 80.4M parameters and S2L-4 has 321.4M; training details are in Appendix F.

The interquartile mean (IQM; Agarwal et al., 2021) of paired differences favors S2L-4 on all four metrics (Table 5); Figure 4 shows how this advantage develops during training.

A widened MDH-4 matches the approximately 321.4M parameters of S2L-4 and still scores below it on every summary: mean endpoint HNS 1.0286 against 1.0648, with lower full-training AUC, late-half AUC, and last-10 HNS (Table 12). Added capacity alone therefore does not account for the gain.

## 5 Offline Continuous Control

We evaluate S2L on nine Minari MuJoCo datasets (Younis et al., 2024): HalfCheetah, Hopper, and Walker2d, each with simple, medium, and expert data. We use TD3+BC (Fujimoto & Gu, 2021) with SimBa critics, five paired seeds, and 200k updates; the matched control omits feature transfer.

Table 5: **Shared-head control on ALE-57.** $\Delta$ is S2L-4 minus the matched-discount MDH-4 baseline in human-normalized score units; positive values favor S2L-4. Intervals are 95% game-cluster bootstrap intervals for the paired-cell IQM (10,000 resamples preserving four seeds per sampled game). Wins are listed as seed pairs and game medians. Terminal HNS has 5 seed-pair ties and 1 game-median tie; Terminal last-10 has 3 seed-pair ties.

| Summary | Pair wins | Game wins | IQM $\Delta$ | 95% CI |
|---|---|---|---|---|
| **Terminal HNS** | 147/228 | 35/57 | +.0323 | $[+.0135, +.0667]$ |
| Full-training AUC | 149/228 | 39/57 | +.0188 | $[+.0089, +.0343]$ |
| Late-half AUC | 153/228 | 39/57 | +.0250 | $[+.0124, +.0489]$ |
| Terminal last-10 | 152/228 | 38/57 | +.0329 | $[+.0146, +.0726]$ |

Table 6: **Offline continuous control.** Entries are S2L minus its matched TD3+BC control in normalized-score AUC; parentheses give positive paired seeds. All nine Minari MuJoCo conditions use five paired seeds and 200k updates.

| Environment | Simple | Medium | Expert | Mean |
|---|---|---|---|---|
| HalfCheetah | $-1.63$ (0/5) | $+7.32$ (5/5) | $+17.78$ (5/5) | $+7.82$ (10/15) |
| Hopper | $+3.39$ (5/5) | $+2.61$ (3/5) | $-0.34$ (2/5) | $+1.88$ (10/15) |
| Walker2d | $-0.39$ (1/5) | $+0.82$ (4/5) | $+1.00$ (3/5) | $+0.47$ (8/15) |
| All | $+0.45$ (6/15) | $+3.58$ (12/15) | $+6.14$ (10/15) | $+3.39$ (28/45) |

Table 6 shows a MuJoCo macro AUC gain of $+3.39$ (hierarchical-bootstrap 95% interval $[+.17, +7.65]$), with six of nine dataset means positive. Appendix I records the offline transfer detail.

## 6 Relation to Prior Work

**Multiple horizons.** Fedus et al. (2019) add value heads with multiple discount factors to a shared Rainbow network, allowing all head losses to update shared features. TD($\Delta$) represents a long-horizon value through differences between neighboring discounts (Romoff et al., 2019), while Tang et al. (2021) derive approximations between values at different discount factors. S2L instead transfers hidden activations between separately trained networks, without combining their Q-values.

**Representation learning under non-stationarity.** Auxiliary control tasks (Jaderberg et al., 2017), observation or latent prediction (Hafner et al., 2020; Schwarzer et al., 2021), contrastive learning (Laskin et al., 2020), and feature-distribution regularization (Saheb Pasand et al., 2026) provide alternative ways to shape representations. These choices affect representation dynamics (Lyle et al., 2021; Le Lan et al., 2023; Tang & Munos, 2023). S2L learns source features from the environment's own rewards at shorter horizons and isolates them from the receiving loss.

**Deep value optimization.** PQN enables parallel, replay-free deep TD learning (Gallici et al., 2024); skip connections, LayerNorm, and structured preconditioning improve its large-network performance (Castanyer et al., 2025). SimBa provides a residual architecture for large actor and critic networks (Lee et al., 2025). S2L builds on these architectures while changing which value losses may update transferred features.

## 7 Discussion and Conclusion

S2L learns features at multiple horizons while preserving a single long-horizon behavior objective. Its fixed four-learner chain improves online return across Atari and continuous control, with ablations supporting its component choices. The analysis links smaller discounts to attenuated transmission of bootstrap changes into targets and detachment to isolated feature updates; fixed-trajectory probes show lower source-target drift during training. Adaptive horizon selection and more efficient feature sharing are natural extensions.

### AI USE STATEMENT

Generative AI tools assisted with literature discovery, experiment orchestration, source-code implementation and testing, statistical analysis and interpretation, and manuscript drafting and editing. Experimental design and mathematical verification were carried out by the authors. Figures and tables were produced by the analysis code rather than generated by these tools.

### ETHICS STATEMENT

This work studies learning algorithms in simulated Atari and continuous-control environments and involves neither human subjects nor personal data. Its main direct resource costs are computation and energy. Compared methods use matched experiment grids. Where available, valid checkpoints are reused to reduce repeated computation.

### REPRODUCIBILITY STATEMENT

Appendices B, F, and H record online training and evaluation settings, comparison rules, and run-inclusion criteria. Versioned artifacts and analysis scripts support the empirical summaries. Training and analysis code will be released. Continuous-control summaries include only matched pairs that reached the stated interaction budget and produced all required evaluations; exclusions are reported in the appendix.

## REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

Ron Amit, Ron Meir, and Kamil Ciosek. Discount factor as a regularizer in reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 269–278. PMLR, 2020. URL https://proceedings.mlr.press/v119/amit20a.html.

Roger Creus Castanyer, Johan Obando-Ceron, Lu Li, Pierre-Luc Bacon, Glen Berseth, Aaron Courville, and Pablo Samuel Castro. Stable gradients for stable learning at scale in deep reinforcement learning. *arXiv preprint arXiv:2506.15544*, 2025. URL https://arxiv.org/abs/2506.15544.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G. Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019. URL https://arxiv.org/abs/1902.06865.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20132–20145, 2021.

Matteo Gallici, Mattie Fellows, Benjamin Ellis, Bartomeu Pou, Ivan Masmitja, Jakob Nicolaus Foerster, and Mario Martin. Simplifying deep temporal difference learning. *arXiv preprint arXiv:2407.04811*, 2024. URL https://arxiv.org/abs/2407.04811.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.01603.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJ6yPD5xg.

Tadashi Kozuno, Yunhao Tang, Mark Rowland, Rémi Munos, Steven Kapturowski, Will Dabney, Michal Valko, and David Abel. Revisiting peng's Q($\lambda$) for modern reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5794–5804. PMLR, 2021. URL https://proceedings.mlr.press/v139/kozuno21a.html.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5639–5650. PMLR, 2020. URL https://proceedings.mlr.press/v119/laskin20a.html.

Charline Le Lan, Stephen Tu, Mark Rowland, Anna Harutyunyan, Rishabh Agarwal, Marc G. Bellemare, and Will Dabney. Bootstrapped representations in reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18686–18713. PMLR, 2023. URL https://proceedings.mlr.press/v202/le-lan23a.html.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R. Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. SimBa: Simplicity bias for scaling up parameters in deep reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jXLiDKsuDo.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1509.02971.

Clare Lyle, Mark Rowland, Georg Ostrovski, and Will Dabney. On the effect of auxiliary tasks on representation dynamics. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1–9. PMLR, 2021. URL https://proceedings.mlr.press/v130/lyle21a.html.

Joshua Romoff, Peter Henderson, Ahmed Touati, Emma Brunskill, Joelle Pineau, and Yann Ollivier. Separating value functions across time-scales. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5468–5477. PMLR, 2019. URL https://proceedings.mlr.press/v97/romoff19a.html.

Ali Saheb Pasand, Johan Obando-Ceron, Aaron Courville, Pouya Bashivan, and Pablo Samuel Castro. Stable deep reinforcement learning via isotropic gaussian representations. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2602.19373.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2007.05929.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32145–32168. PMLR, 2023. URL https://proceedings.mlr.press/v202/sokar23a.html.

Yunhao Tang and Rémi Munos. Towards a better understanding of representation dynamics under TD-learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33720–33738. PMLR, 2023. URL https://proceedings.mlr.press/v202/tang23g.html.

Yunhao Tang, Mark Rowland, Remi Munos, and Michal Valko. Taylor expansion of discount factors. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10130–10140. PMLR, 2021. URL https://proceedings.mlr.press/v139/tang21b.html.

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. URL https://arxiv.org/abs/1801.00690.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K. Terry. Minari. Zenodo, 2024. URL https://doi.org/10.5281/zenodo.13767625.

# A PROOFS

*Proof of Eq.* (7). Let

$$G_i = r_i + \gamma m_{i+1}\big[(1-\lambda)V_{i+1} + \lambda G_{i+1}\big], \qquad G_T = V_T,$$

where $m_j \in [0,1]$, and define $\widetilde{G}_i$ using the same rewards and masks. Write $\Delta V_j = V_j - \widetilde{V}_j$, $\Delta G_j = G_j - \widetilde{G}_j$, and $M_k = \prod_{\ell=1}^{k} m_{t+\ell}$. The common rewards cancel. Repeated substitution gives

$$\Delta G_t = (1-\lambda)\sum_{k=1}^{n-1} \gamma^k \lambda^{k-1} M_k \Delta V_{t+k} + \gamma^n \lambda^{n-1} M_n \Delta V_T. \tag{12}$$

Taking absolute values and using $M_k \le 1$ proves Eq. (7). With all masks equal to one,

$$c_n(\gamma,\lambda) = (1-\lambda)\sum_{k=1}^{n-1} \lambda^{k-1}\gamma^k + \lambda^{n-1}\gamma^n.$$

Choosing every $\Delta V_j$ with the same sign and magnitude attains equality, so the coefficient is sharp. □

*Proof of Theorem 1.* Let $\delta_o = Q_o - Y_o$. Numerically,

$$\delta_o = (1-c_o)(a_o u(\theta) - q_o^\star e_1),$$

but $Y_o$ is held constant during differentiation. With $u_\perp(\theta) = (-\sin\theta, \cos\theta)^\top$, the gradients are

$$\partial_{a_o}\ell_o = (1-c_o)(a_o - q_o^\star \cos\theta), \tag{13}$$

$$\partial_\theta \ell_o = (1-c_o)a_o q_o^\star \sin\theta, \qquad o \in \mathcal{W}. \tag{14}$$

The first equation applies to every $o \in \mathcal{O}$; only the second is summed over $\mathcal{W}$. All mixed derivatives between $a_o$ and $\theta$ vanish at the stated fixed point. After forming these semi-gradients, we differentiate the numerical update map, recomputing targets at each pre-update point. This gives head multipliers $1-\eta(1-c_o)$ and the feature multiplier in Eq. (10), with $K_{\mathcal{W}}$ as in Eq. (9). The angular multiplier has magnitude below one exactly under the stated condition. The full linearized joint update is contractive only if, in addition, $0 < \eta(1-c_o) < 2$ for every head.

Finally, fix $n \ge 1$ and $\lambda \in [0,1]$. The polynomial form of $c_n$ extends to $\gamma = 1$, where $c_n(1,\lambda) = 1$, and

$$\left.\frac{\partial c_n}{\partial \gamma}\right|_{\gamma=1} = \sum_{k=1}^{n-1} k(1-\lambda)\lambda^{k-1} + n\lambda^{n-1} = \sum_{j=0}^{n-1}\lambda^j.$$

Therefore $1 - c_n(\gamma,\lambda) \sim (\sum_{j=0}^{n-1}\lambda^j)(1-\gamma)$. Substitution proves Eq. (11); the finite sum equals $n$ at $\lambda = 1$. □

Table 7: Base configuration for two-learner Atari experiments.

| Setting | Value |
|---|---|
| Algorithm | PQN with Q($\lambda$) |
| Main / short discount | .99 / .90 |
| $\lambda$ | .65 |
| CNN | Atari, medium: 32–64–64 channels |
| MLP | width 2560, five two-linear residual blocks |
| Stabilizers | full MultiSkip, LayerNorm, Kron |
| Environments / rollout | 128 / 32 steps |
| Minibatches / epochs | 32 / 2 (64 optimizer calls per iteration) |
| Iterations / environment steps | 610 / 2,498,560 |
| Learning rate | $2.5 \times 10^{-4}/3$ |
| Gradient clipping | separate norm 10 per learner |
| Exploration | $1.0 \rightarrow 0.005$ over first 10% |
| Seeds | 1, 2, 3, 4 |

## B Exact Training Details

For the two-learner variants, the behavior network is constructed first so its initial parameters and buffers match the baseline exactly; the source network is constructed second. Optimizer parameters follow the deterministic order "all behavior-network parameters, then all source-network parameters." Targets are computed from detached online predictions. The short target uses its own greedy value, but the sampled action in both regression losses is the action actually taken by the behavior learner.

The read-interface study uses Qbert and Phoenix with $\gamma_s = .9$. The horizon study uses Qbert, Phoenix, Asterix, Breakout, and Seaquest and tests effective horizons $H \in \{4, 8, 10, 16, 32, 64, 100\}$, where $\gamma_s = 1 - 1/H$. All comparisons use four paired seeds. Observations follow ALE-v5 through EnvPool with minimal actions, frame skip 4, sticky-action probability .25, at most 30 no-ops, episodic life, and clipped training rewards.

For these development studies, the primary statistic is trapezoidal area under the recorded training-return curve (AUC). For a paired candidate $X$ and reference $Y$, the sign-safe normalized improvement is

$$\mathrm{SNI}(X, Y) = \frac{2(\mathrm{AUC}_X - \mathrm{AUC}_Y)}{|\,\mathrm{AUC}_X\,| + |\,\mathrm{AUC}_Y\,|},$$

with value zero when both AUCs are zero. Per-game source discounts maximize the four-seed median SNI; the global source maximizes median SNI over all twenty game–seed pairs, with direction counts and shorter horizons used only as fixed tie-breakers. AUC ratios and their geometric means are reported only when all contributing AUCs are positive.

## C Feature-Transfer Rules

The two-learner studies use a source $Q^S$ and behavior learner $Q^L$. Each forms its own target as in Eq. (4), with discounts $\gamma_s < \gamma_\ell = .99$, and the combined loss is $\mathcal{L}_S + \mathcal{L}_L$. For long-network block $l$, let $\bar{h}_l^L$ include the ordinary residual and skip connections. The general read interface is

$$h_l^L = \bar{h}_l^L + \mathrm{sg}[b_l^S]. \tag{15}$$

Here $s_0^S$ is the source input embedding and $s_l^S$ is its activation after residual block $l$.

The Atari interface study holds both learners and their updates fixed while changing only the detached feature $b_l^S$ supplied to long block $l$:

$$\text{INPUT:} \quad b_l^S = s_0^S, \tag{16}$$

$$\text{ALIGNED:} \quad b_l^S = s_l^S, \tag{17}$$

$$\text{DELTA:} \quad b_l^S = s_l^S - s_{l-1}^S, \tag{18}$$

$$\text{DENSE MEAN:} \quad b_l^S = \frac{1}{6}\sum_{j=0}^{5} s_j^S, \tag{19}$$

$$\text{PREFIX MEAN:} \quad b_l^S = \frac{1}{l+1}\sum_{j=0}^{l} s_j^S. \tag{20}$$

No interface adds parameters, optimizer state, target mixing, or an additional behavior policy.

## D ATARI FEATURE-TRANSFER RESULTS

Table 8: Paired training-return AUC relative to the scaled PQN base agent of Castanyer et al. (2025). GM is the geometric mean across the listed paired seeds. Every method uses the same $\gamma = .99$ target construction and action-selection rule.

| | Qbert (4 seeds) | | Phoenix (4 seeds) | | Pooled (8) | |
|---|---|---|---|---|---|---|
| Transferred feature | GM | Wins | GM | Wins | GM | Wins |
| None (baseline) | 1.000 | – | 1.000 | – | 1.000 | – |
| Short input | 1.173 | 4/4 | 1.173 | 4/4 | 1.173 | 8/8 |
| Aligned absolute | 1.239 | 4/4 | 1.237 | 4/4 | 1.238 | 8/8 |
| Aligned delta | 1.188 | 4/4 | 1.199 | 4/4 | 1.194 | 8/8 |
| Dense all-layer mean | **1.277** | 4/4 | 1.240 | 4/4 | **1.258** | 8/8 |
| Prefix mean | 1.222 | 4/4 | **1.264** | 4/4 | 1.243 | 8/8 |

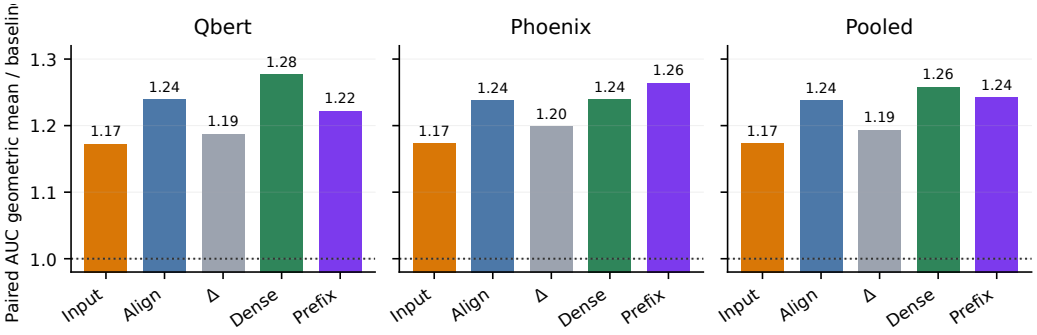



Figure 5: **Feature-transfer comparison.** Bars show paired geometric-mean training-return AUC relative to the $\gamma = .99$ base agent on Qbert and Phoenix. Every two-learner variant has the same parameter count, target construction, optimizer configuration, and action-selection rule; only the transferred feature changes.

## E FIVE-GAME DISCOUNT SELECTIONS

## F FULL ALE-57 EVALUATION

The fixed four-learner model uses the discounts in Eq. (1) and the aligned predecessor connection in Eq. (2). It is compared with the scaled PQN parent of Castanyer et al. (2025) on all 57 ALE games,

Table 9: Feature-transfer effects relative to the detached short-input variant. Parameter counts, optimizer settings, target construction, and action-selection rules match.

| Transferred feature | Qbert GM | Phoenix GM | Pooled GM | Positive pairs |
|---|---|---|---|---|
| Short input | 1.000 | 1.000 | 1.000 | – |
| Aligned absolute | 1.0565 | 1.0546 | 1.0556 | 6/8 |
| Aligned delta | 1.0129 | 1.0221 | 1.0175 | 4/8 |
| Dense all-layer mean | **1.0887** | 1.0566 | **1.0725** | 6/8 |
| Prefix mean | 1.0420 | **1.0773** | 1.0595 | **7/8** |

Table 10: Per-game source-discount-factor choices from the five-game sweep. Median SNI compares the selected source with the scaled PQN parent of Castanyer et al. (2025) over four paired seeds.

| Game | Selected $\gamma_s$ | Effective horizon | Median SNI |
|---|---|---|---|
| Qbert | .875 | 8 | .256 |
| Phoenix | .900 | 10 | .154 |
| Asterix | .875 | 8 | .106 |
| Breakout | .984375 | 64 | .064 |
| Seaquest | .96875 | 32 | .341 |

four paired seeds, and 2,498,560 environment steps per run. For each game–seed pair, terminal HNS is the mean of learner iterations 601–610 of the logged rolling-20 HNS. Figure 2 reports paired differences directly in HNS units, accommodating signed and near-zero scores. Common-support full-training AUC is a secondary curve-level summary.

**Original paired controls.** The full-suite control matrix uses the same ALE protocol, optimizer, interaction budget, and four paired seeds. The four-discount aligned detached model is compared with a no-transfer control and a control that keeps the connections but sets all four discounts to .99. These three arms contain $57 \times 4$ runs each. The two-learner detached and writable controls contain $52 \times 4$ runs each, excluding the five development games. Both use the fixed $\gamma_s = .875$ mean-pooled design; they differ only in whether the $\gamma = .99$ loss is stopped at the source feature or backpropagates through it. Initialization, forward values before an update, targets, loss weights, per-learner clipping, and the single Kron step are matched.

All 1,100 curves reach iteration 610 and enter the analysis. Endpoint HNS is the rolling-20 value at iteration 610; AUCs are normalized trapezoidal averages over the pair-common finite support, with the late window beginning at iteration 305. Reported intervals use 10,000 whole-game bootstrap resamples and preserve all four paired seeds.

Table 11: **Matched Atari ablations.** Entries are mean paired human-normalized-score differences. Brackets give 95% intervals from 10,000 bootstrap resamples of games, retaining all four paired seeds. The first two rows cover all 57 games; the last covers the 52 games not used for development. Positive values favor the first method.

| Contrast | Endpoint HNS | Full AUC | Second-half AUC |
|---|---|---|---|
| Transfer − no transfer | $+.101\ [-.121, +.282]$ | $+.080\ [+.005, +.165]$ | $+.091\ [-.024, +.207]$ |
| Mixed discounts − all .99 | $+.060\ [-.101, +.202]$ | $+.065\ [+.023, +.112]$ | $+.086\ [+.028, +.153]$ |
| Detached − backpropagated | $+.114\ [+.006, +.256]$ | $+.082\ [+.019, +.174]$ | $+.122\ [+.025, +.275]$ |

In the two-learner comparison, detachment improves endpoint, full-training AUC, and second-half AUC (Table 11). Full-training game medians favor detachment on $42/52$ games, with mean and median AUC gains of $+.0817$ and $+.0082$.

**Extended S2L-4 controls.** The four-network gradient ablation removes stop-gradient from all three predecessor connections while retaining the forward computation. The identical-source-discount variant uses $(.75, .75, .75, .99)$; the reversed-source variant uses $(34/35, 11/12, .75, .99)$. The frozen-

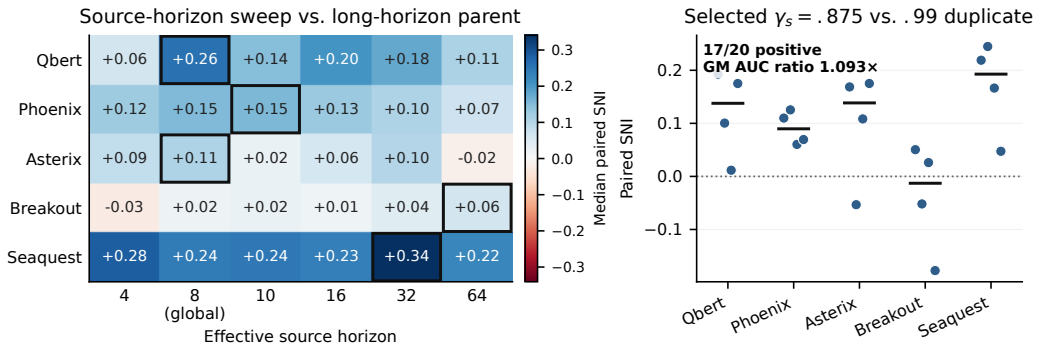



Figure 6: **Source-discount selection.** Left: four-seed median paired SNI against the $\gamma = .99$ base agent across the eligible grid; outlined cells are in-sample per-game maxima and $H = 8$ is selected globally on the same panel. Right: seed-level paired SNI for that selected source against the matched $\gamma_s = .99$ duplicate source. SNI is defined in Appendix B.

source variant retains detached additions but holds all three source networks fixed at random initialization and trains only the final network. The parameter-matched PQN control widens a single network to match the total capacity of S2L-4. These additional controls reuse the existing full-model reference scores and are summarized as aggregate differences in Table 4.

**Shared-trunk comparison.** The MDH-4 baseline uses one CNN–MultiSkip trunk with four Q heads at the same discounts as S2L-4, $\lambda = .65$, and behavior controlled only by the .99 head. Its objective is the .99-head MSE plus the mean of the three auxiliary-head MSEs. All 57 games, four paired seeds, ALE protocol fields, and 2,498,560-step budgets match the S2L-4 comparison. The primary summary is HNS at learner iteration 610; full-training AUC, late-half AUC, and the mean of the final ten logged returns are secondary. The paired-cell IQM trims 25% from each tail, and its interval resamples whole games while preserving all four seeds. All summaries use signed HNS differences. Figure 4 starts at 0.5M steps; all 456 arm curves have valid returns by step 425,984.

The shared-trunk implementation uses global gradient clipping and Kron preconditioning. S2L-4 uses per-learner clipping and preconditioning, a sum of the four losses, and detached feature transfer. These settings define the complete architectures compared in Section 4.4.

**Parameter-matched shared trunk.** A further control widens the MDH-4 trunk to match the total parameter count of S2L-4. All three models use the same 57 games, seeds 1–4, and 2,498,560-step budget. Table 12 gives arithmetic means and standard deviations across the 228 game–seed runs.

Endpoint HNS uses iteration 610, and last-10 averages iterations 601–610 within each run. AUCs use normalized trapezoidal integration, with the late window covering iterations 305–610. The S2L-4 full-training AUC reference uses common valid support with MDH-4, so its mean differs slightly from the PQN-paired reference in Table 4.

Table 12: **Parameter-matched shared-trunk control on ALE-57.** Entries are arithmetic means ± standard deviations across 57 games and four seeds, in HNS units. Wide MDH-4 matches the total parameter count of S2L-4. Standard deviations describe score dispersion across game–seed cells.

| Summary<br>Parameters (approx.) | S2L-4<br>321.4M | MDH-4<br>80.4M | MDH-4 (wide)<br>321.4M |
|---|---|---|---|
| Endpoint HNS | 1.0648 ± 2.2205 | 1.0181 ± 2.7107 | 1.0286 ± 2.7339 |
| Full-training AUC | 0.3997 ± 0.9289 | 0.3627 ± 0.8405 | 0.3511 ± 0.9083 |
| Late-half AUC | 0.6827 ± 1.3488 | 0.6400 ± 1.4027 | 0.6174 ± 1.4369 |
| Terminal last-10 | 1.0278 ± 2.1008 | 0.9896 ± 2.5857 | 0.9319 ± 2.6126 |

## G Fixed-Trajectory Target-Drift Protocol

The direct measurement uses 16 games selected before observing these diagnostics by hashing the canonical names of the 52 non-development games. For each game, four fresh paired seeds compare the mixed-discount model with the otherwise matched all-.99 control. Every learner is evaluated in situ at 51 fixed iterations from 1 through 610; diagnostic evaluation neither advances an environment nor changes training RNG, parameters, or optimizer state.

Each game has two immutable probe strata shared byte-for-byte across methods, seeds, and landmarks: 256 uniform-random and 256 frozen base-policy greedy trajectories, each containing 32 transitions. Observations, actions, clipped rewards, terminal masks, and episode boundaries are therefore held fixed. At each landmark we recompute the implemented Q($\lambda$) target and greedy bootstrap values. For source learner $o$, probe stratum $p$, and landmark $k > 0$, the normalized consecutive drift is

$$d_{o,p,k} = \frac{\mathrm{RMS}_i[G_{o,p,k}(i) - G_{o,p,k-1}(i)]}{\max\{\mathrm{sd}_i[G_{o,p,0}(i)], 10^{-6}\}}.$$

We integrate $d_{o,p,k}$ by the trapezoidal rule over actual environment-step coordinates and divide by the elapsed span. The primary cell statistic averages the first three learner positions with weight $1/3$ and the two probe strata with weight $1/2$; the final .99 learner is a diagnostic control. Raw RMS drift is aggregated identically. Paired differences are formed within game and seed, and 95% intervals use 10,000 whole-game bootstrap resamples that retain all four paired seeds and the complete time series.

Both measures favor mixed discounts in every paired run and every game (Table 3). The pattern also holds separately for every primary source position and both probe strata. As a calibration of Eq. (6), 1,671,168 measured distance-specific coefficients have zero mismatches at tolerance $10^{-7}$; the largest absolute error is $1.2 \times 10^{-8}$. Across 417,792 within-arm monotonicity checks, there are no violations.

## H Continuous-Control Training Details

Table 13: Shared settings for the S2L-4–SimBa DMC matrix.

| Setting | Value |
|---|---|
| Tasks | dog-run, dog-trot, humanoid-run, humanoid-walk |
| Algorithms / optimizers | SAC, DDPG / AdamW, Kron |
| Environment steps / action repeat | 1,000,000 / 2 |
| Train environments / updates per interaction | 1 / 2 |
| Replay capacity / warm-up / batch | 1M / 5,000 / 256 |
| Evaluation | 10 episodes every 50,000 environment steps |
| SimBa discount / $n$-step / target $\tau$ | .99 / 1 / .005 |
| S2L-4 discounts | $.75,\ 11/12,\ 34/35,\ .99$ |
| AdamW actor/critic learning rate | $10^{-4}$ |
| Kron actor/critic learning rate | $5 \times 10^{-5}$ |
| Actor/critic weight decay | $10^{-2}$ |
| Seeds | 0, 1000, 2000, 3000, 4000 |

The nine scale points are: base $(1, 128; 2, 512)$; depth-1/2/3 $(2/3/4, 128; 4/6/8, 512)$; width-1/2/3 $(1, 256/384/512; 2, 1024/1536/2048)$; and joint-1/2 $(2, 256; 4, 1024)$ and $(3, 384; 6, 1536)$. Each tuple lists $(\text{actor blocks}, \text{actor width}; \text{critic blocks}, \text{critic width})$. The base point shared by the depth, width, and joint sweeps appears once, yielding nine rather than eleven unique scales.

For every matched cell, the final S2L-4 critic is constructed under the same module names and before the three auxiliary critics, preserving the baseline final-critic initialization exactly. TD updates stop gradients through predecessor features and sum the four per-critic MSE losses without averaging, so each critic retains the baseline loss scale. SAC entropy terms enter every discount-specific target. DDPG uses the same exploration-noise distribution as its SimBa cell. The actor uses only the final .99 critic value, with predecessor action Jacobians live and critic parameters held fixed.

**Evaluation and support.** The full design contains 720 matched pairs: two algorithms, two optimizers, four tasks, five seeds, and nine scales. Every run trains for one million environment steps and is evaluated for ten episodes every 50,000 steps. Table 1 combines all 720 paired evaluations. Means are weighted by pair count and wins are added within each subgroup.

Performance is the paired difference S2L-4 minus SimBa in endpoint return or trapezoidal AUC divided by its window length. Full AUC covers 0–1M steps; late AUC covers 500k–1M or 750k–1M. Return-dynamics diagnostics use the eleven evaluations from 500k to 1M. Detrended residual MAD measures local deviation around a linear trend; range-normalized maximum drawdown and cumulative negative variation measure the loss of previously attained return.

**Seed variability.** Across 144 task–algorithm–optimizer–scale configurations with five seeds each, the mean cross-seed endpoint MAD is 31.79 for S2L-4 and 35.21 for SimBa; S2L-4 is lower in 88 configurations, SimBa in 50, with six ties. For 500k–1M AUC, the corresponding mean cross-seed MADs are 35.97 and 46.21, with S2L-4 lower in 87 configurations.

## I Offline-Control Implementation Detail

For the offline runs, detached source activations are RMS-capped at the recipient activation RMS (cap $= 1$) before transfer.

## J Seed-Level Terminal AUC

Table 14 gives the training-return AUC values for the two-game development study in Appendix D. Values are divided by $10^9$ only for presentation.

Table 14: Training-return AUC ($10^9$) for baseline (A), simple short input (B), dense mean (E), and prefix mean (F).

| Game | Seed | A | B | E | F |
|---|---|---|---|---|---|
| Qbert | 1 | 4.194 | 4.888 | 5.754 | 5.532 |
| | 2 | 3.973 | 5.363 | 5.330 | 4.760 |
| | 3 | 4.722 | 4.950 | 5.468 | 5.138 |
| | 4 | 4.365 | 5.003 | 5.441 | 5.658 |
| Phoenix | 1 | 6.956 | 8.190 | 8.086 | 8.430 |
| | 2 | 6.972 | 8.464 | 8.619 | 8.576 |
| | 3 | 6.231 | 7.204 | 8.448 | 8.520 |
| | 4 | 6.745 | 7.737 | 8.178 | 8.450 |

## K Two-Learner Atari Update

1. Collect a rollout with epsilon-greedy actions from $Q^L$ only. Store long and short greedy values for their respective Q($\lambda$) recursions.
2. Compute $G^L$ with $\gamma_\ell = .99$ and $G^S$ with $\gamma_s$ on the same rewards, masks, transitions, and minibatch permutation.
3. For each minibatch, run the source learner to obtain $Q^S$ and activations $s_0^S, \ldots, s_5^S$.
4. Form the selected feature $b_l^S$, detach it, and add it after the ordinary long MultiSkip addition at long block $l$.
5. Compute $\mathcal{L}_L + \mathcal{L}_S$, backpropagate once, clip the behavior and source learners separately, and take one Kron step.