ON SCALING POLICY NETWORKS IN POLICY-GRADIENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling network capacity has produced reliable gains in supervised learning, but scaling policy networks in reinforcement learning remains far less predictable. We study how policy depth interacts with optimization and interaction budget using a simple on-policy policy-gradient learner with Monte Carlo reward-to-go and a learned value baseline, alongside supporting bootstrapped variants. Across continuous-control experiments, we find that optimization conditions do not remain comparable as depth changes. Performance is highly sensitive to learning rate, while the region associated with strong performance shifts systematically with policy depth. As a result, transferring a learning rate selected for a shallow policy can confound additional capacity with optimization mismatch. Selecting learning rates separately by depth substantially changes the apparent scaling curve and reduces the measured penalty for 32-layer policies by about two fifths in our primary experiment. Reversing the transfer direction can also change the apparent depth ordering. Correcting this mismatch does not make deeper policies uniformly better. At moderate interaction budgets, deeper policies generally remain worse than shallow ones, while longer training produces task-dependent gains rather than a universal crossover. Our results show that policy-network scaling is jointly shaped by architectural depth, optimization protocol, and interaction budget. Shallow-rate transfer and per-depth selection answer different scientific questions and can yield materially different scaling conclusions.

1 INTRODUCTION

Scaling curves are meaningful only relative to the training procedure used to produce them. In supervised and self-supervised learning, increasing model capacity yields predictable gains when data, optimization, and compute are allocated appropriately (Hestness et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022). Reinforcement learning (RL) has likewise benefited from larger networks when supported by appropriate architectural and optimization choices (Sinha et al., 2020; Lee et al., 2025a; Nauman et al., 2024; Schwarzer et al., 2023), and self-supervised goal-conditioned agents have scaled beyond a thousand layers (Wang et al., 2025). These successes leave a basic measurement question unresolved. *When policy depth changes, what should be held fixed to measure whether depth helps?*

A common scaling experiment changes network depth while keeping the training recipe fixed. This implicitly assumes that the recipe transfers across scale. If a learning rate selected for a shallow policy becomes increasingly mismatched as depth grows, the resulting curve mixes the effect of additional depth with optimization mismatch. Selecting a learning rate separately at each depth answers a different question, namely how each architecture performs under a specified optimization search. The two protocols can therefore produce different scaling curves from the same experimental runs.

Optimization sensitivity is not unique to RL. Initialization and residual design affect deep-network training (Glorot & Bengio, 2010; He et al., 2016a), hyperparameter transfer is an explicit goal of μP and its depthwise extensions (Yang et al., 2021; 2024), and tuning can materially change comparisons across learning domains (Melis et al., 2018; Lucic et al., 2018; Patterson et al., 2024). On-policy RL adds another coupling because every update changes the policy that collects subsequent experience (Igl et al., 2020). Optimization choices can therefore affect both parameter updates

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

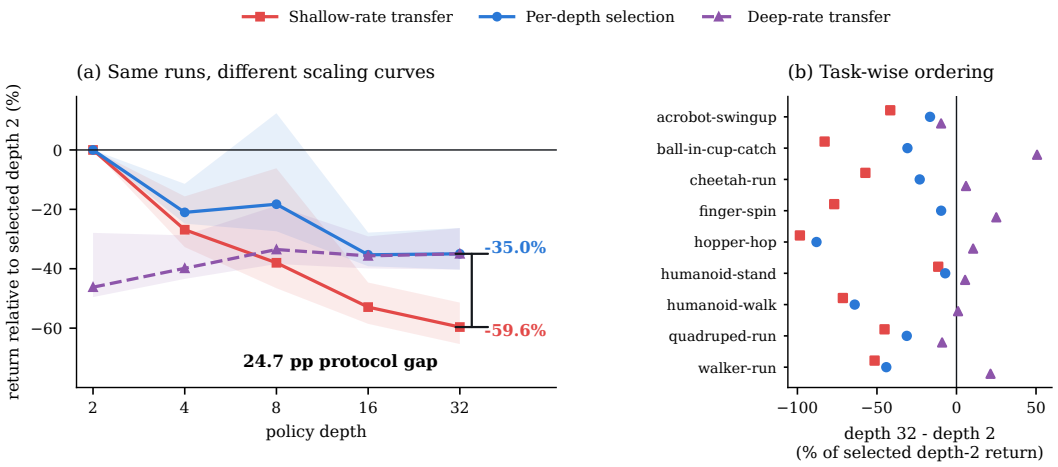


Figure 1: **The same runs produce different policy-depth scaling curves under different learning-rate protocols.** (a) Mean contrasts over nine DeepMind Control tasks at 20M interactions. Shallow-rate transfer uses the depth-2-selected rate, per-depth selection chooses a rate independently at each depth, and deep-rate transfer uses the depth-32-selected rate. All protocols use the same selected depth-2 reference. At depth 32, the deficit is 59.6% under shallow-rate transfer and 35.0% after per-depth selection, a 24.7 percentage-point (pp) gap. (b) Task-wise depth-32 versus depth-2 point contrasts. Deep-rate transfer changes the sign of the point estimate on seven of nine tasks. These signs are not significance classifications.

and future experience. We do not isolate this feedback as the causal source of depth-dependent sensitivity. Instead, we measure how strongly the comparison protocol changes the observed depth effect.

We study this question using a simple on-policy policy-gradient learner that collects a new trajectory batch for each actor update, together with width-64 residual policies. Our primary experiment contains nine DeepMind Control tasks, five policy depths, five common learning rates, and ten seeds at 20M interactions, yielding 2,250 controlled runs. We ask four questions. *Does additional policy depth improve performance? How much does optimization transfer across depth change the measured scaling curve? What does the on-policy setting add? What does this measurement perspective establish beyond methods designed to make larger networks trainable?*

Let $J(d, \eta)$ denote seed-mean evaluation return at policy depth d and actor learning rate η for one task and interaction budget. For candidate grid $\mathcal{G}$, define $\eta^\star(d) \in \arg\max_{\eta \in \mathcal{G}} J(d, \eta)$. We compare

$$J_{\text{fixed}}(d; d_0) = J\big(d, \eta^\star(d_0)\big), \tag{1}$$

$$J_{\text{tuned}}(d) = \max_{\eta \in \mathcal{G}} J(d, \eta). \tag{2}$$

The first asks whether a learning rate selected at anchor depth d_0 transfers across depth. The second asks how performance changes when each depth receives the same candidate search. Neither isolates architectural capacity from the rest of the learning system.

Across the primary grid, the selected learning rate decreases by about 0.20 decades per depth doubling, while nearby learning-rate mismatch can be costly. Shallow-rate transfer produces a mean depth-32 deficit of 59.6%, compared with 35.0% after separate rate selection. Reversing the transfer direction also changes the depth-32 versus depth-2 point-estimate ordering on seven of nine tasks. A fixed-recipe curve therefore reflects both policy depth and the transferability of the chosen optimization recipe.

Correcting this mismatch does not produce uniformly positive depth scaling. After separate rate selection, depths 16 and 32 still underperform depth 2 on eight of nine DeepMind Control tasks by paired interval comparisons at 20M interactions. With additional interaction, deeper policies become competitive or advantageous on some tasks such as cheetah-run, while other tasks retain

negative depth effects. We therefore do not claim a universal RL depth-scaling law. Policy-depth scaling is jointly shaped by architectural depth, optimization protocol, and interaction budget.

2 EXPERIMENTAL FRAMEWORK

We study policy depth in a controlled setting that separates architectural depth from the learning-rate protocol used to compare depths. The primary experiment varies actor depth while holding the learner, policy width, critic architecture, interaction budget, and learning-rate candidate set fixed. Supporting benchmarks and longer-budget experiments test whether the main observations extend beyond this grid.

Learner. Each iteration collects 32 complete episodes, corresponding to 32,000 transitions on the 1,000-step DeepMind Control tasks (Tassa et al., 2018). Every actor update therefore uses a newly collected on-policy batch. With discount $\gamma = 0.99$, we compute reward-to-go targets with a value bootstrap at time-limit truncation and no bootstrap at true termination (Pardo et al., 2018). Under these conventions, the advantage estimator corresponds to GAE with $\lambda = 1$ (Schulman et al., 2016). Values and targets are computed before either network is updated, and advantages are normalized over the batch.

The actor receives one full-batch Adam step (Kingma & Ba, 2015) minimizing the score-function objective (Williams, 1992; Sutton et al., 1999)

$$\mathcal{L}_\pi(\theta) = -\frac{1}{|\mathcal{B}|} \sum_{(s_t,a_t)\in\mathcal{B}} \log \pi_\theta(a_t \mid s_t)\, \widetilde{A}_t. \tag{3}$$

There are no trust-region constraints or clipped objectives (Schulman et al., 2015; 2017), importance ratios, entropy bonuses, repeated actor epochs, or gradient-norm clipping. After the actor update, the critic is fitted for five epochs with minibatches of 256 and learning rate 10^{-3}. Learning rates remain constant. We use this simple learner as an experimental instrument rather than an algorithmic contribution. Appendix A provides implementation details.

Architecture. Policies are width-64 residual MLPs (He et al., 2016b). Depth d counts affine layers inside the $d/2$ residual blocks, excluding the input stem and output head. Each block computes

$$h \leftarrow h + W_2\, \mathrm{SiLU}(W_1\, \mathrm{LN}(h) + b_1) + b_2. \tag{4}$$

The second affine map is zero-initialized, so each block initially implements the identity. No depth-dependent residual-branch scaling is applied. A final LayerNorm (Ba et al., 2016) precedes the output head. The policy is a diagonal Gaussian with learned state-independent log standard deviation. We vary policy depth over $d \in \{2, 4, 8, 16, 32\}$ while keeping the critic at depth 2. Increasing depth also increases parameter count and computation, so our primary comparisons control interaction budget rather than compute budget.

Primary controlled grid. Our main experiment uses nine DeepMind Control tasks at 20M interactions. Every task is evaluated at five policy depths, five learning rates $\mathcal{G} = \{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$, and ten seeds, giving $9 \times 5 \times 5 \times 10 = 2{,}250$ runs. This complete grid forms the basis of the main comparisons in Sections 3–5.

Additional 10^{-2} runs are available for a subset of task-depth cells. Diagnostic analyses may use these expanded candidates, while aggregate protocol curves, bootstrap comparisons, and held-out checks use the common five-rate grid so that all tasks and depths receive identical candidate support. Throughout, "selected" or "tuned" means the best observed configuration within the tested candidate set, not an unconstrained optimum. Appendix B documents the expanded coverage.

Comparison protocols. For each task, let $\eta^\star(d)$ denote the learning rate selected at policy depth d. Table 1 summarizes the three comparison protocols.

These protocols use the same underlying runs. Only the learning-rate selection rule changes. Fixed-rate transfer measures architectural differences together with optimization-transfer cost, while per-depth selection measures performance under the same specified search opportunity. Neither isolates architectural capacity from the rest of the learning system.

Evaluation and uncertainty. Each DMC evaluation averages three deterministic mean-action episodes with frozen observation-normalization statistics. Evaluations occur at update boundaries

Table 1: Comparison protocols. Rates are selected separately for each task and "selected" means within the tested candidate set.

Protocol	Rate at depth d	Scientific question
Shallow-rate transfer	$\eta^\star(2)$	Does a shallow-policy recipe transfer across depth?
Per-depth selection	$\eta^\star(d)$	How does performance change under the same candidate search?
Deep-rate transfer	$\eta^\star(32)$	What changes when transfer direction is reversed?

after crossing each 25,000-step interval. A run's reported return averages evaluations over the final 10% of reached training steps. We retain completed, nonduplicate runs that reach at least 95% of the target interaction budget. Learning rates maximize seed-mean return separately for each task and depth.

For task e and protocol p, we report

$$\Delta_{p,e}(d) = 100\frac{J_{p,e}(d) - J_{\text{tuned},e}(2)}{|J_{\text{tuned},e}(2)|}. \tag{5}$$

The selected depth-2 return provides a common reference across protocols. Aggregate results average these percentages equally across tasks. All nine primary DMC reference returns are positive, though percentage normalization can give greater influence to tasks with small reference returns.

Task-wise uncertainty uses paired percentile bootstrap intervals (Efron, 1979) over shared seed identifiers, with learning rates reselected within each replicate. Candidates and intervals require at least five runs and five shared seeds, respectively. We use 20,000 replicates for primary task-wise analyses and 5,000 for interaction-budget comparisons. Intervals are not adjusted for multiple comparisons. Split-seed checks address selection optimism and are reported in Appendix D.

Supporting experiments. Gymnasium provides a second benchmark (Towers et al., 2025), including seven MuJoCo v5 environments (Todorov et al., 2012) and two classic-control environments. Its learning-rate windows and seed support vary across depth, so we treat it as corroborating evidence rather than pooling it with DMC. Appendix G reports its coverage and results. Longer-budget continuations and supporting alternative configurations use restricted or different candidate sets and are analyzed separately. Appendix A gives their implementation details.

The complete primary grid first lets us ask whether the learning-rate response changes systematically with policy depth.

3 Depth changes the learning-rate response

A fixed learning rate can distort a depth comparison only if two conditions hold. The preferred learning rate must change with depth, and performance must deteriorate when that rate is mismatched. We observe both in the complete DMC grid (Figure 2). Together, these effects create depth-dependent transfer costs when a rate selected at one depth is reused at another. We characterize this empirical response but do not identify why the preferred rate changes with depth.

Selected rates decrease with depth. Across the primary grid, regressing $\log_{10}\eta^\star$ on $\log_2 d$ gives a slope of -0.201 decades per depth doubling, with a task-bootstrap 95% interval of $[-0.264, -0.144]$ (Figure 2, left). Ball-in-cup-catch selects 3×10^{-3} at depth 2 and 10^{-4} at depth 32, while cheetah-run shifts from 10^{-3} to 3×10^{-4}. A regression with task fixed effects gives the same slope.

We interpret this trend descriptively. The selected rate depends on the parameterization, learner, batch size, and candidate grid. Similar directional trends appear in supporting configurations reported in the appendix. The primary experiment keeps batch size fixed, and known coupling between learning rate and batch size (Smith et al., 2018; Shallue et al., 2019) motivates testing whether the trend persists when both vary.

Rate mismatch can be costly. For each task-depth cell, we compare the selected rate with neighboring candidates. When both one-step directions are available, we use the worse-performing neighbor. At a boundary, we use the available direction. Moving one grid step, approximately half a decade,

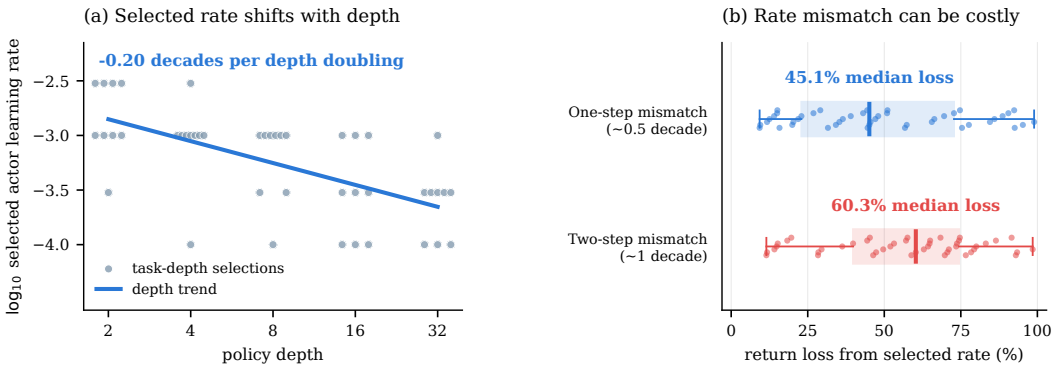


Figure 2: **Learning-rate response changes with policy depth in the primary DMC grid.** (a) The selected actor learning rate shifts toward smaller values with depth. Regressing $\log_{10} \eta^\star$ on $\log_2 d$ gives a slope of -0.201 decades per depth doubling, with a task-bootstrap 95% interval of $[-0.264, -0.144]$. (b) Return loss after moving one grid step, approximately half a decade, or two steps, approximately one decade, from the selected rate in the worse available direction. Median losses are 45.1% and 60.3%. Boxes show interquartile ranges and whiskers show the full observed range. These statistics describe the tested parameterization and candidate grid rather than a universal scaling law.

reduces return by a median of 45.1%, with losses above 25% in 73% of the 45 cells (Figure 2, right). Moving two steps gives a median loss of 60.3%.

These values measure sensitivity rather than the expected cost of an arbitrary perturbation. Pooling both available one-step directions gives a smaller median loss of 24.2%. Additional sensitivity results are reported in the appendix.

Together, these findings show why transferring one learning rate across depths is not a neutral choice. The high-performing region shifts with depth, while performance can fall substantially away from that region. A fixed-rate scaling curve therefore reflects both policy depth and optimization-transfer cost.

4 Optimization transfer changes the apparent scaling curve

Section 3 established that the learning rate associated with strong performance shifts with depth and that mismatched rates can be costly. We now ask how much this matters for the scaling comparison itself. The effect is substantial. Applying the learning rate selected for the shallow policy makes the apparent penalty of depth considerably larger than when rates are selected separately at each depth.

Under shallow-rate transfer, the mean return contrasts at depths 4, 8, 16, and 32 are -26.9%, -38.0%, -52.9%, and -59.6%, respectively. Selecting a learning rate separately at each depth changes these contrasts to -21.0%, -18.3%, -35.3%, and -35.0% (Figure 1). At depth 32, the difference between the two protocols is 24.7 percentage points, with a 95% bootstrap interval of $[19.2, 31.8]$. Equivalently, the shallow-transfer deficit is 1.71 times as large.

These curves are reconstructed from the same tasks, architectures, interaction budgets, candidate rates, and underlying runs. Only the learning-rate allocation rule changes. The measured scaling curve therefore depends on whether the experiment asks how well a shallow optimization recipe transfers or how well each depth performs under the same candidate search. The separately selected curve is also no longer monotonic with depth. This is important because the qualitative shape of the scaling curve changes even before asking whether deeper policies ultimately outperform shallower ones.

The protocol effect varies substantially across tasks. On cheetah-run, depth 2 selects 10^{-3} and attains return 473.2. Depth 32 reaches approximately 202 when forced to use that rate, but 363.8 at its own selected rate of 3×10^{-4}. Its deficit therefore changes from 57.3% to 23.1%. On finger-spin, the deficit falls from 76.9% to 9.6%. On hopper-hop, it remains large, changing only from 98.4%

to 88.0%. Optimization transfer therefore explains a large fraction of the apparent depth penalty on some tasks but little of it on others. Full task-wise comparisons and alternative scoring rules are reported in Appendix D.

Shallow-rate transfer is not a neutral fixed-recipe baseline. By construction, the shallow policy is evaluated at the rate selected for it, while deeper policies may incur transfer mismatch. Reversing the transfer direction makes this asymmetry visible. When the learning rate is instead selected at depth 32 and transferred to shallower policies, depth 32 has a higher point-estimate return than depth 2 on seven of nine tasks (Figure 1). These are ordering signs rather than seven statistically resolved wins.

Using the same selected depth-2 denominator as elsewhere, policies trained with the depth-32-selected rate average -46.2% at depth 2 and -35.0% at depth 32. Under per-depth selection, depth 32 remains below depth 2 on all nine tasks. Changing the anchor therefore changes which architecture bears the optimization mismatch and can change the apparent ordering even though the underlying experiment corpus is unchanged.

We summarize this asymmetry with the within-grid mismatch

$$M_e(d; d_0) = J_{\text{tuned},e}(d) - J_{\text{fixed},e}(d; d_0), \tag{6}$$

which measures the return lost at depth d when it uses the learning rate selected at anchor depth d_0 rather than its own selected rate. Under a shallow anchor, $M_e(2; 2) = 0$ in-sample while deeper policies may have nonzero mismatch. Under a deep anchor, $M_e(32; 32) = 0$ and the asymmetry reverses. A fixed-recipe depth comparison therefore combines the depth difference itself with unequal optimization-transfer costs. The shallow- and deep-anchor experiments answer different transfer questions rather than providing interchangeable estimates of an architecture-only depth effect.

The protocol gap is not explained solely by selecting and evaluating rates on the same seeds. When learning-rate selection and evaluation are performed on disjoint subsets of the ten seeds, the depth-32 shallow-transfer versus per-depth-selection gap remains substantial. These checks address selection optimism (Cawley & Talbot, 2010), although they reuse the same seed pool and are not independent replications. Appendix D reports the detailed split results.

The same broad protocol dependence also appears outside the complete DMC grid. On the five main Gymnasium tasks, the depth-32 deficit changes from 61.1% under shallow-rate transfer to 31.3% after per-depth selection. Candidate windows and seed support vary with depth, so we treat these results as corroborating rather than matched estimates. Appendix G provides the full analysis.

Taken together, these results show that learning-rate transfer accounts for a substantial portion of the apparent penalty of policy depth, but not all of it. A fixed-recipe scaling curve measures architecture together with the transferability of the optimization recipe used to construct it. Per-depth selection removes that particular asymmetry, but it does not imply that deeper policies are advantageous.

5 Does additional policy depth improve performance?

Sections 3–4 showed that shallow-rate transfer can substantially exaggerate the apparent penalty of depth. We now ask whether deeper policies actually perform better once learning rates are selected separately by depth.

At 20M interactions, generally no. After per-depth selection, depths 16 and 32 each have paired intervals entirely below depth 2 on eight of nine DMC tasks. Depth 16 has one resolved improvement, while depth 32 has none. Intermediate depths are more mixed. Depth 4 improves on one task and underperforms on two, while depth 8 improves on one and underperforms on three. Thus, correcting learning-rate transfer reduces the average depth penalty but does not remove it. Optimization mismatch explains part of the unfavorable depth scaling at 20M, not all of it.

Gymnasium shows the same broad pattern under its available search windows. Deeper policies generally remain below depth 2, with few resolved improvements. Because its learning-rate coverage is incomplete and depth dependent, we treat this as corroborating evidence and report the full analysis in Appendix G. A natural possibility is that deeper policies require more interaction before their additional capacity becomes useful. We therefore examine $\lambda = 1$ continuations that retain the fixed depth-2 critic. Cheetah-run, quadruped-run, and walker-run contain all five depths through 150M

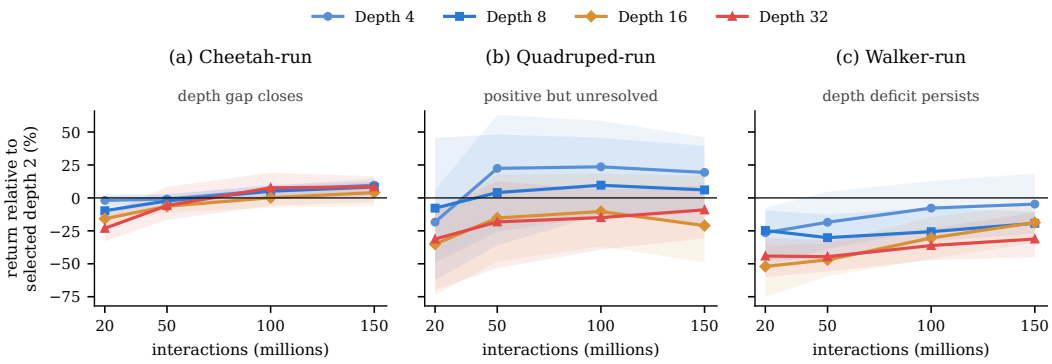


Figure 3: **Policy-depth contrasts across interaction budgets.** Each panel reports return relative to the selected depth-2 policy at the same budget for the $\lambda = 1$ policy-depth continuations. Cheetah-run shows a closing depth gap and later positive contrasts, quadruped-run develops positive but unresolved estimates, and walker-run retains negative depth contrasts. Later budgets retain only two inherited learning-rate candidates per depth. Shaded bands show 95% paired-seed bootstrap intervals with rate reselection. Connected points share training histories, so comparisons across budgets are not independent experiments.

interactions (Figure 3). Later budgets select among only two inherited learning-rate candidates per task and depth, so these experiments use a restricted search rather than a fresh full sweep.

The three tasks show qualitatively different outcomes. On cheetah-run, the depth deficit closes as the interaction budget grows. The depth-32 contrast changes from -23.1% at 20M to -5.8% at 50M and $+7.7\%$ at 100M, although the 100M interval includes zero. By 150M, depths 4 and 8 exceed depth 2 by 9.7% and 8.3%, respectively, with both intervals above zero. Depths 16 and 32 also have positive point estimates of $+4.0\%$ and $+8.4\%$, but their intervals remain unresolved.

Quadruped-run provides an intermediate case. Depth 4 has positive point estimates of 22.4%, 23.6%, and 19.3% at 50M, 100M, and 150M, respectively, but all corresponding intervals include zero. Walker-run shows the opposite pattern. Every evaluated deeper-policy contrast remains negative across the observed budgets.

The learning curves provide an additional view of these outcomes in Appendix H. They show learning speed, seed variability, and late-training behavior for the available configurations. A favorable final-window contrast does not necessarily imply that a deeper policy learned faster throughout training, and a transient peak need not persist to the final evaluation window. We therefore interpret these results as comparisons of sustained late-training performance at each interaction budget rather than as general claims of sample-efficiency dominance.

These results rule out a simple interaction-budget crossover in which deeper policies eventually become uniformly advantageous. Additional interaction can materially change the depth comparison, but the outcome depends on the task. The evidence therefore supports neither a universal "depth hurts" conclusion nor a universal "depth eventually wins" conclusion. Related work on warm-starting and continual learning similarly cautions against assuming that more training always helps (Ash & Adams, 2020; Dohare et al., 2024).

Supporting long-budget experiments show the same heterogeneity. In Gymnasium, Ant develops resolved gains at larger budgets, whereas HalfCheetah does not show a comparable depth-32 advantage. A separate 300M joint actor–critic experiment also produces mixed outcomes across tasks. Because that experiment also changes critic depth, learner configuration, and simulator implementation, we treat it as corroborating evidence rather than a matched policy-depth or interaction-budget comparison. Additional details are provided in Appendix F.

Overall, the scaling picture is conditional on optimization and interaction budget. At 20M interactions, deeper policies generally remain worse even after separate learning-rate selection. With additional interaction, some tasks exhibit depth advantages while others do not. Policy depth therefore does not have a single favorable or unfavorable scaling trend independent of training budget.

6 Related work

Our work connects four strands of research on hyperparameter transfer across scale, methods that enable larger RL networks, scaling-law and resource-allocation studies, and methodology for comparing tuned learning systems.

Kernel-limit and lazy-training analyses distinguish regimes of nearly fixed features from nonlinear feature learning (Jacot et al., 2018; Chizat et al., 2019; Yang & Hu, 2021). The maximal update parameterization (μP) enables hyperparameter transfer from smaller to wider models (Yang et al., 2021), while Depth-μP extends this question along the depth axis (Yang et al., 2024). CompleteP further distinguishes hyperparameter transferability from preserving nonlinear feature learning as depth grows (Dey et al., 2025), and Lee et al. (2026) study these issues when scaling continuous-control agents. These approaches ask how parameterization or training design can preserve desirable optimization behavior across scale. Our question is complementary. Under one fixed residual parameterization, we ask how much imperfect optimization transfer changes the depth comparison itself.

A separate line of work develops learning systems in which larger RL networks become effective. Spectral normalization supports deeper actor–critic networks (Bjorck et al., 2021); D2RL uses dense connections to improve deep policies on manipulation and locomotion tasks (Sinha et al., 2020); and SimBa combines residual blocks, observation standardization, and layer normalization to improve parameter scaling across RL algorithms (Lee et al., 2025a). SimbaV2 further studies normalization at larger model and compute scales (Lee et al., 2025b), while Palenicek et al. (2025) examine scaling the update-to-data ratio in off-policy control. In self-supervised goal-conditioned RL, Wang et al. (2025) demonstrate benefits from networks as deep as 1,024 layers. Related work also identifies primacy bias, dormant neurons, and plasticity loss as obstacles to continued learning (Nikishin et al., 2022; Sokar et al., 2023; Lyle et al., 2023), while BRO, BBF, and BRC study high-capacity value functions in continuous control, Atari, and multi-task learning (Nauman et al., 2024; Schwarzer et al., 2023; Nauman et al., 2025). These studies establish that architecture, normalization, regularization, and learning-system design can make larger networks useful. We instead hold the primary learning system fixed and ask how the protocol used to compare depths changes the measured benefit of scaling. Our results therefore complement, rather than contradict, reported gains from methods designed explicitly for larger networks.

Scaling studies in RL ask another complementary question. Hilton et al. (2023) relate model size, interactions, and compute through intrinsic performance, while Cheng et al. (2026) study the allocation of sampling compute in on-policy LLM RL. We do not fit compute-optimal scaling laws or study LLM post-training. Instead, within a fixed interaction budget, we reconstruct policy-depth comparisons under different learning-rate allocation rules using the same underlying runs.

Depth is only one axis of model scaling. Prior work varies width, parameter count, update-to-data ratio, sampling, and compute (Lee et al., 2025a; Palenicek et al., 2025; Hilton et al., 2023; Cheng et al., 2026). Our controlled design instead fixes policy width and interaction budget while varying depth. This isolates the measurement question studied here, but it does not imply that depth is the preferred or compute-optimal way to scale an RL agent. Because increasing depth also increases parameter count and per-update computation, our results should be interpreted as policy-depth comparisons under a specified resource allocation rather than as general laws of model-size scaling.

Finally, our study builds on work showing that empirical conclusions can depend strongly on implementation and tuning. Implementation details can dominate nominal differences between policy-gradient objectives (Engstrom et al., 2020; Andrychowicz et al., 2020), while reproducibility and evaluation studies emphasize seed variability, benchmark aggregation, baseline construction, and hyperparameter selection (Henderson et al., 2018; Jordan et al., 2020; Agarwal et al., 2021; Patterson et al., 2024). Similar effects appear outside RL. Extensive tuning changes comparisons of language-model architectures (Melis et al., 2018), and GAN rankings depend strongly on search effort and random restarts (Lucic et al., 2018). Random and Bayesian search provide alternative ways to allocate optimization effort (Bergstra & Bengio, 2012; Snoek et al., 2012). Our contribution is to connect this evaluation perspective to policy-network scaling. Even with tasks, architectures, interaction budgets, and underlying runs held fixed, changing only how learning rates are selected

and transferred across depth can substantially change the measured scaling curve and sometimes its apparent ordering.

7 Implications and limitations

What does a policy-depth scaling curve measure? The central implication of our results is that a policy-depth scaling curve is conditional on how optimization effort is allocated across depth. A fixed-recipe comparison asks whether an optimization setting selected at one depth transfers to another, whereas per-depth selection asks how each depth performs under a specified search opportunity. Neither quantity isolates architecture from the learning system. Our experiments show that this distinction matters in practice. Changing only the learning-rate allocation rule substantially changes the magnitude and sometimes the apparent ordering of the depth effect. The relevant question is therefore not whether a fixed or tuned curve is the "correct" scaling curve, but which scientific question the comparison is intended to answer.

What does the on-policy setting add? Optimization dependence is not unique to reinforcement learning. Supervised learning also depends on hyperparameter tuning and on how data and compute are allocated. On-policy RL nevertheless introduces an additional coupling because every policy update changes the policy that generates subsequent experience. An optimization choice can therefore affect both parameter updates and the distribution of future training experience. Our experiments motivate treating this interaction carefully, but they do not isolate it as the causal source of the observed depth-dependent learning-rate response. A matched fixed-data or off-policy comparison would be needed to separate this feedback from other sources of optimization sensitivity.

What should scaling studies report? Our results suggest reporting transferred-recipe and separately selected curves together when feasible. The comparison should also specify the anchor depth, candidate search space, selection and evaluation seeds, interaction budget, and whether interactions or compute are controlled. A fixed recipe is informative when transfer itself is the object of study. Per-depth selection is more informative when the question concerns performance attainable under a specified optimization budget. Reporting both makes explicit how much an observed scaling trend depends on the transferability of the training recipe.

What are the limits of this study? Learning rate is the principal optimization variable swept systematically, while critic capacity, batch size, and other optimizer settings are held fixed or not jointly optimized. We therefore interpret the remaining depth deficits as conditional on this search setup and report the candidate ranges explicitly. Future work should vary these settings jointly, especially given known learning-rate and batch-size interactions (McCandlish et al., 2018). Our conclusions are also specific to one width-64 residual parameterization, low-dimensional continuous-control benchmarks, and a simple on-policy learner. Because increasing depth also increases parameter count and computation, we compare policies at matched interaction budgets rather than claim compute-optimal scaling.

8 Conclusion

We studied how optimization protocol affects conclusions about policy-network depth scaling in on-policy reinforcement learning. The learning rate associated with strong performance shifts systematically with policy depth, and performance can degrade substantially when that rate is mismatched. Consequently, transferring the learning rate selected for a shallow policy makes deeper policies appear considerably worse than selecting rates separately at each depth. Reversing the transfer direction further shows that anchor choice changes which architecture bears the optimization mismatch and can alter the apparent ordering. Correcting this transfer mismatch does not, however, make deeper policies uniformly advantageous. At 20M interactions, large depths generally remain worse than depth 2 after separate rate selection, while longer training produces task-dependent gains rather than a universal crossover. The broader lesson is that policy depth does not scale well or poorly in isolation. Policy-depth scaling should be interpreted jointly with optimization protocol and interaction budget.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript editing, LaTeX organization, document formatting, and language polishing. We have not used generative AI tools to generate experimental results or to make final scientific judgments, and the remaining disclosure categories are not applicable to this work. We reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The experiments concern simulated continuous-control benchmarks. Their immediate contribution is methodological, clarifying how optimization and evaluation choices affect empirical scaling claims. These simulation results do not establish the reliability or safety of deploying learned policies in physical systems.

REPRODUCIBILITY STATEMENT

Section 2 describes the learner, architecture, primary experimental grid, evaluation procedure, and comparison protocols. Appendix A provides implementation and training details, Appendix B documents experimental coverage and run inclusion criteria, and Appendix C specifies the evaluation and statistical procedures. Additional details supporting the main optimization-transfer, learning-rate-response, and interaction-budget results are provided in Appendices D, E, and F. Together, these sections document the experimental settings and analysis procedures used to obtain the results reported in the paper.

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*, 2021. URL https://arxiv.org/abs/2108.13264.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020. URL https://arxiv.org/abs/2006.05990.

Jordan Ash and Ryan Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3884–3894, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/288cd2567953f06e460a33951f55daaf-Abstract.html.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL https://arxiv.org/abs/1607.06450.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. ReZero is all you need: fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1352–1361. PMLR, 2021. URL https://proceedings.mlr.press/v161/bachlechner21a.html.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012. URL https://jmlr.org/papers/v13/bergstra12a.html.

Johan Bjorck, Carla P. Gomes, and Kilian Q. Weinberger. Towards deeper deep reinforcement learning with spectral normalization. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2106.01151.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010. URL https://jmlr.org/papers/v11/cawley10a.html.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2101.05982.

Zhoujun Cheng, Yutao Xie, Yuxiao Qu, Amrith Setlur, Shibo Hao, Varad Pimpalkhute, Tongtong Liang, Feng Yao, Zhengzhong Liu, Eric Xing, Virginia Smith, Ruslan Salakhutdinov, Zhiting Hu, Taylor Killian, and Aviral Kumar. IsoCompute Playbook: Optimally scaling sampling compute for LLM RL. *arXiv preprint arXiv:2603.12151*, 2026. URL https://arxiv.org/abs/2603.12151.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/ae614c557843b1df326cb29c57225459-Abstract.html.

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: CompleteP enables compute-efficient deep transformers. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.01618.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024. doi: 10.1038/s41586-024-07711-7. URL https://www.nature.com/articles/s41586-024-07711-7.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2005.12729.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256. PMLR, 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016a. URL https://arxiv.org/abs/1603.05027.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016b. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1709.06560.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017. URL https://arxiv.org/abs/1712.00409.

Jacob Hilton, Jie Tang, and John Schulman. Scaling laws for single-agent reinforcement learning. *arXiv preprint arXiv:2301.13442*, 2023. URL https://arxiv.org/abs/2301.13442.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. URL https://arxiv.org/abs/2203.15556.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. *arXiv preprint arXiv:2006.05826*, 2020. URL https://arxiv.org/abs/2006.05826.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Scott Jordan, Yash Chandak, Daniel Cohen, Mengxue Zhang, and Philip Thomas. Evaluating the performance of reinforcement learning algorithms. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4962–4973. PMLR, 2020. URL https://proceedings.mlr.press/v119/jordan20a.html.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Adam Lee, M. Ganesh Kumar, Blake Bordelon, and Cengiz Pehlevan. CompleteP for RL: Maintaining feature learning when scaling deep reinforcement learning. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/66419.

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 *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2410.09754.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. In *International Conference on Machine Learning*, 2025b. URL https://arxiv.org/abs/2502.15280.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? a large-scale study. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1711.10337.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23190–23211. PMLR, 2023. URL https://proceedings.mlr.press/v202/lyle23b.html.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018. URL https://arxiv.org/abs/1812.06162.

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1707.05589.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937. PMLR, 2016. URL https://proceedings.mlr.press/v48/mniha16.html.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample-efficient continuous control. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.16158.

Michal Nauman, Marek Cygan, Carmelo Sferrazza, Aviral Kumar, and Pieter Abbeel. Bigger, regularized, categorical: High-capacity value functions are efficient multi-task learners. *arXiv preprint arXiv:2505.23150*, 2025. URL https://arxiv.org/abs/2505.23150.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16828–16847. PMLR, 2022. URL https://proceedings.mlr.press/v162/nikishin22a.html.

Daniel Palenicek, Florian Vogt, Joe Watson, and Jan Peters. Scaling off-policy reinforcement learning with batch and weight normalization. *arXiv preprint arXiv:2502.07523*, 2025. URL https://arxiv.org/abs/2502.07523.

Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. Time limits in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4045–4054. PMLR, 2018. URL https://proceedings.mlr.press/v80/pardo18a.html.

Andrew Patterson, Samuel Neumann, Martha White, and Adam White. Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318):1–63, 2024. URL https://jmlr.org/papers/v25/23-0183.html.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. URL https://arxiv.org/abs/1312.6120.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level Atari with human-level efficiency. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 30365–30380. PMLR, 2023. URL https://proceedings.mlr.press/v202/schwarzer23a.html.

Christopher J. Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl. Measuring the effects of data parallelism on neural network training. *Journal of Machine Learning Research*, 20(112):1–49, 2019. URL https://jmlr.org/papers/v20/18-789.html.

Samarth Sinha, Homanga Bharadhwaj, Aravind Srinivas, and Animesh Garg. D2RL: Deep dense architectures in reinforcement learning. *arXiv preprint arXiv:2010.09163*, 2020. URL https://arxiv.org/abs/2010.09163.

Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1711.00489.

Jasper Snoek, Hugo Larochelle, and Ryan Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html.

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.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, 1999. URL https://proceedings.neurips.cc/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.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.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/iros.2012.6386109. URL https://doi.org/10.1109/iros.2012.6386109.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2407.17032.

Kevin Wang, Ishaan Javali, Michał Bortkiewicz, Tomasz Trzciński, and Benjamin Eysenbach. 1000 layer networks for self-supervised RL: Scaling depth can enable new goal-reaching capabilities. *arXiv preprint arXiv:2503.14858*, 2025. URL https://arxiv.org/abs/2503.14858.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. doi: 10.1007/BF00992696. URL https://link.springer.com/article/10.1007/BF00992696.

Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17084–17097, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/8df7c2e3c3c3be098ef7b382bd2c37ba-Abstract.html.

Greg Yang and Edward J. Hu. Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2011.14522.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor Programs VI: Feature learning in infinite depth neural networks. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f2ce452a4a7352feb7a9fb885267ff24-Abstract-Conference.html.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1901.09321.

Table 2: Training and evaluation settings for the primary policy-depth experiment and the supporting joint actor–critic depth experiment. The latter changes several factors jointly and is not a matched interaction-budget extension of the primary experiment.

Setting	DMC/Gym policy depth, 20M	Joint actor–critic depth, 300M
Architecture / width	Residual MLP / 64	Residual MLP / 64
Actor depths	2, 4, 8, 16, 32	2, 8, 32
Critic depth	2	Same as actor
Advantage estimator	GAE $\lambda = 1$	GAE $\lambda = 0.95$
Environments	8	1,024
Batch	32 complete episodes	32-step segments per env.
Actor updates per batch	1 full-batch Adam step	1 full-batch Adam step
Critic fitting	5 epochs, minibatches of 256	5 full-batch steps
Deterministic evaluation	3 episodes per point	1,024 episodes per point
Evaluation interval	25,000 interactions	500 updates ($\approx$ 16.4M interactions)
Training seeds	DMC: 10; Gymnasium: 1–10	10 per configuration

A Implementation and reproducibility details

The primary policy-depth experiments use a PyTorch implementation with eight environments collecting 32 complete episodes per actor update (Table 2). On the 1,000-step DeepMind Control tasks, this corresponds to 32,000 transitions per update (Table 3). The policy receives one full-batch update, followed by five critic epochs with minibatches of 256. Both Adam optimizers use $\epsilon = 10^{-5}$. The actor learning rate is swept, while the critic learning rate is fixed at 10^{-3}. Discounting uses $\gamma = 0.99$. The primary estimator is reward-to-go minus a learned value baseline, equivalently GAE with $\lambda = 1$ under the terminal and time-limit conventions described in Section 2.

A running observation normalizer is shared across environments, and normalized observations are clipped to $[-10, 10]$. Evaluation freezes the normalization statistics. Rewards are not normalized. Gaussian actions sampled during training are clipped to the environment bounds for execution, while their unclipped values are retained for log-probability computation. The policy mean is not squashed. The log standard deviation is a learned state-independent parameter. Deterministic evaluation executes the mean action with the environment action bounds applied.

Hidden affine maps use orthogonal initialization (Saxe et al., 2013) with gain $\sqrt{2}$, except for the second affine map in each residual block, which is initialized to zero. Biases are zero. The actor output head uses gain 0.01, while the critic head uses gain 1. Actor and critic parameters are disjoint. A depth-32 policy has 16 residual blocks, plus an input stem and output head, giving 34 affine layers in total. Fixed width therefore does not imply fixed parameter count.

Initialization can affect signal and gradient propagation at large depth. Fixup rescales residual-network initialization, while ReZero uses zero-initialized residual gates (Zhang et al., 2019; Bachlechner et al., 2021). Our zero-initialized second affine maps make the residual blocks initially implement the identity, but we implement neither complete recipe and do not assume their transfer properties.

The primary learner differs from actor–critic systems that reuse collected experience more aggressively. Asynchronous actor–critic uses parallel actor-learners (Mnih et al., 2016), SAC uses an off-policy maximum-entropy objective (Haarnoja et al., 2018), TD3 uses paired critics and delayed actor updates (Fujimoto et al., 2018), and REDQ combines critic ensembles with multiple updates per collected transition (Chen et al., 2021). These methods are cited to define the scope of the learner rather than as evaluated baselines.

The supporting joint actor–critic depth experiment uses a JAX implementation with MuJoCo Playground and the MJWarp backend. Rollouts use 1,024 environments and 32-step segments, giving 32,768 transitions per rollout. Value estimates bootstrap at segment cuts. The critic receives five full-batch updates and has the same depth as the actor. Actor depth, critic depth, rollout collection, advantage estimation, critic updates, and simulator implementation all differ from the primary

Table 3: Coverage of the experimental configurations used directly or as supporting evidence in the paper. Joint depth varies actor and critic depth together. Other DMC rows vary policy depth with a fixed depth-2 critic.

Configuration	Tasks	Cells	Runs	Rates/cell	Seeds
DMC $\lambda = 1$, 20M	9	45	2,330	5–6	10
Gym $\lambda = 1$, 20M	9	41	1,513	3–7	5–10
DMC $\lambda = .95$, 20M	5	20	300	3	5
Gym $\lambda = .95$, 20M	6	24	645	3–4	1–10
DMC $\lambda = 1$, 50M	5	25	500	2	10
DMC $\lambda = 1$, 100M	5	20	332	1–2	1–10
DMC $\lambda = 1$, 150M	3	15	300	2	10
DMC $\lambda = .95$, 100M	5	15	410	1–4	10
Joint depth $\lambda = .95$, 300M	4	12	610	3–6	10

experiment. Comparisons between these configurations therefore do not isolate policy depth or interaction budget.

B Experimental coverage and run inclusion

The primary DeepMind Control grid contains nine tasks

acrobot-swingup, ball-in-cup-catch, cheetah-run, finger-spin,
hopper-hop, humanoid-stand, humanoid-walk, quadruped-run, walker-run.

Every task has policy depths $2, 4, 8, 16$, and 32, five common learning rates,

$$\mathcal{G} = \{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\},$$

and ten seeds. The complete primary grid therefore contains

$$9 \times 5 \times 5 \times 10 = 2{,}250$$

training runs.

Additional 10^{-2} candidates are available for depth 2 on ball-in-cup-catch, cheetah-run, finger-spin, quadruped-run, and walker-run. Further 10^{-2} candidates are available for hopper-hop at depths 2 and 4 and humanoid-walk at depth 2. These expanded candidates are used only for diagnostic analyses where appropriate. Aggregate protocol curves, bootstrap comparisons, and held-out checks use the common five-rate grid so that every task and depth receives identical candidate support.

We include deterministic-evaluation runs that reach at least 95% of their target interaction budget, satisfy the intended experimental configuration, and are not marked as duplicates. Standard policy-depth experiments use the residual architecture, width 64, critic depth 2, state-independent log standard deviation, and 32 complete episodes per actor update. The primary score averages evaluation points in the final 10% of reached training steps rather than selecting the best checkpoint.

The Gymnasium $\lambda = 1$ grid contains Ant, HalfCheetah, Hopper, Humanoid, HumanoidStandup, MountainCarContinuous, Pendulum, Swimmer, and Walker2d. Depth coverage is incomplete for some tasks, and some configurations extend to depth 64. The DMC $\lambda = 0.95$ experiments contain cheetah-run, hopper-hop, humanoid-walk, quadruped-run, and walker-run at the available policy depths.

The $\lambda = 1$ continuation experiments contain ball-in-cup-catch and finger-spin at 50M in addition to cheetah-run, quadruped-run, and walker-run. The latter three contain all five policy depths through 150M. Later interaction budgets retain only two learning-rate candidates per task and depth. They therefore represent restricted-search continuations rather than fresh five-rate sweeps.

C Evaluation and statistical procedures

For a training run with evaluation returns $\{r_i\}$ at interaction counts $\{t_i\}$, the primary run-level score averages evaluation points within the final 10% of the interactions reached by that run. This metric measures sustained late-training performance rather than the best observed checkpoint.

For task e and policy depth d, the selected learning rate is

$$\eta_e^\star(d) \in \arg\max_{\eta \in \mathcal{G}} \frac{1}{N} \sum_{s=1}^{N} J_{e,s}(d, \eta). \tag{7}$$

Throughout the paper, "selected" or "tuned" refers to the best observed candidate in the tested set and not an unconstrained optimum.

The normalized depth contrast is

$$\Delta_{p,e}(d) = 100 \frac{J_{p,e}(d) - J_{\text{tuned},e}(2)}{|J_{\text{tuned},e}(2)|}, \tag{8}$$

where p denotes the comparison protocol. The selected depth-2 return remains the common denominator when the transfer anchor changes. All nine primary DMC reference returns are positive. Aggregate results average task-normalized percentages equally across the nine tasks.

Task-wise uncertainty uses paired percentile bootstrap intervals (Efron, 1979). For each comparison, we use the intersection of seed identifiers available across the candidate rates in the two compared cells. Seed identifiers are sampled with replacement while maintaining their pairing across depths and learning rates. Each bootstrap replicate recomputes candidate means, reselects the learning rate separately for the compared depths, and evaluates the return difference. The 2.5th and 97.5th percentiles form the reported interval.

Candidates require at least five available runs. Task-wise intervals additionally require at least five shared seed identifiers. When fewer than five shared seeds are available, we report a descriptive point estimate without an interval or significance classification. An interval containing zero is treated as unresolved and not as evidence of equivalence. Primary task-wise analyses use 20,000 bootstrap replicates. Interaction-budget analyses use 5,000 replicates. Task-wise intervals are not adjusted for multiple comparisons.

For the aggregate DMC protocol comparison, ten seed identifiers are resampled independently within each task while maintaining pairing across depths and learning rates. Each replicate reselects the relevant anchor and per-depth rates and recomputes the selected depth-2 denominator. Selection is restricted to the five-rate common grid, so every replicate retains the same nine tasks. The depth-32 per-depth-selection minus shallow-transfer gap is 24.7 percentage points, with percentile interval $[19.2, 31.8]$.

Bootstrap reselection reflects variability in the selection procedure but does not eliminate selection optimism. Appendix D therefore includes analyses in which rate selection and evaluation use disjoint subsets of the ten seeds.

The learning-rate slope interval in Section 3 is obtained by resampling tasks rather than seeds. This quantifies variation of the selected-rate trend across the benchmark tasks and is distinct from the seed-bootstrap uncertainty used for performance contrasts.

Learning-curve bands in Appendix H show pointwise standard errors across seeds at a rate selected using the final-window metric. They are descriptive and are neither 95% confidence intervals nor selection-adjusted uncertainty bands.

D Robustness of the optimization-transfer result

Table 4 reports the task-wise depth-32 values underlying the primary protocol comparison. Shallow-rate transfer evaluates depth 32 using the learning rate selected at depth 2. Per-depth selection evaluates depth 32 using its own selected candidate. Both percentage contrasts use the selected depth-2 return as the common reference.

The protocol gap persists when learning-rate selection and evaluation use disjoint subsets of the seeds. These checks address selection optimism (Cawley & Talbot, 2010), although the subsets reuse the same ten-seed pool and are therefore robustness analyses rather than independent replications (Table 5).

Alternative definitions of late performance change the magnitude of the depth-32 comparison without removing the protocol gap.

Table 4: Depth-32 protocol comparison on the primary DMC grid. Shallow transfer uses the depth-2-selected rate, while per-depth selection uses the rate selected at depth 32.

Task	$\eta^\star(2)$	$\eta^\star(32)$	$J_{\text{tuned}}(2)$	Shallow (%)	Selected (%)
acrobot-swingup	$3 \cdot 10^{-4}$	10^{-4}	44.1	-41.6	-16.6
ball-in-cup-catch	$3 \cdot 10^{-3}$	10^{-4}	625.2	-82.9	-30.8
cheetah-run	10^{-3}	$3 \cdot 10^{-4}$	473.2	-57.3	-23.1
finger-spin	10^{-3}	10^{-4}	785.7	-76.9	-9.6
hopper-hop	$3 \cdot 10^{-3}$	$3 \cdot 10^{-4}$	68.0	-98.4	-88.0
humanoid-stand	10^{-3}	$3 \cdot 10^{-4}$	9.1	-11.5	-7.1
humanoid-walk	$3 \cdot 10^{-3}$	$3 \cdot 10^{-4}$	7.4	-71.4	-64.0
quadruped-run	$3 \cdot 10^{-3}$	10^{-3}	346.8	-45.2	-31.3
walker-run	10^{-3}	$3 \cdot 10^{-4}$	270.6	-51.5	-44.1
Unweighted task mean				-59.6	-35.0

Table 5: Selection and evaluation on disjoint seed subsets. Entries are depth-32 percentage contrasts relative to depth 2 using rates chosen on the selection subset and evaluated on the reporting subset.

Selection	Reporting	Shallow (%)	Selected (%)	Gap (points)
1–5	6–10	-60.6	-31.5	29.1
6–10	1–5	-60.1	-34.0	26.2
Odd	Even	-59.0	-31.9	27.1
Even	Odd	-52.0	-26.0	26.0
All ten	All ten	-59.6	-35.0	24.7

Deep-rate transfer provides a complementary test of the asymmetry introduced by the anchor. With the depth-32-selected learning rate transferred to shallower policies, depth 32 has a higher point-estimate return than depth 2 on seven of nine tasks in-sample. When rate selection and evaluation are separated, the corresponding positive counts are five, five, six, and five tasks across the four splits (Table 6). These counts describe point-estimate ordering rather than statistically resolved wins.

E Learning-rate response details

Figure 4 shows two representative task-level response surfaces underlying the aggregate trend in Figure 2. In both tasks, the high-return region shifts toward smaller learning rates as policy depth increases.

For the primary selected-rate trend, we regress $\log_{10} \eta^\star$ on $\log_2 d$ over the 45 task–depth selections. The resulting slope is -0.201 decades per depth doubling. A task bootstrap with 5,000 replicates gives a 95% interval of $[-0.264, -0.144]$. Adding task fixed effects gives the same main-grid slope. We treat this as a descriptive trend under the tested parameterization, batch size, learner, and candidate grid rather than as a universal scaling exponent.

Learning-rate sensitivity is computed relative to the selected candidate in each task–depth cell. A one-step mismatch corresponds to one adjacent grid point, approximately half a decade in learning rate. When both adjacent directions are available, the main statistic uses the worse-performing neighbor. At a boundary, the available direction is used. This statistic gives a median loss of 45.1%, and 73% of the 45 cells lose more than 25% of return. Pooling both available one-step directions gives a smaller median loss of 24.2%.

A two-step mismatch corresponds to approximately one decade in learning rate and gives a median loss of 60.3%. These values quantify local sensitivity within the tested grid and should not be interpreted as the expected loss under a random learning-rate perturbation.

The selected-rate trend also appears directionally in supporting configurations, although their candidate windows differ. The available Gymnasium $\lambda = 1$ grid gives a pooled slope of -0.113, with task-bootstrap interval $[-0.166, -0.070]$. The available DMC $\lambda = 0.95$ grid gives a slope of

Table 6: Sensitivity of the depth-32 protocol comparison to the run-level evaluation metric. Rate selection is repeated separately for each metric.

Run-level metric	Shallow transfer (%)	Per-depth selection (%)
Final 10% of reached steps	−59.6	−35.0
Last evaluation point	−62.9	−35.6
Last five evaluation points	−61.0	−35.9
Whole-curve mean	−54.5	−39.5
Best checkpoint	−37.2	−24.8

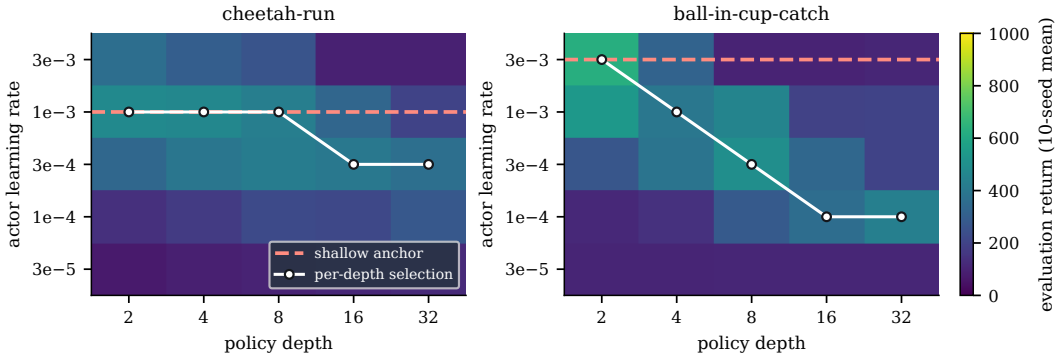


Figure 4: Depth–learning-rate response on two illustrative DMC tasks at 20M interactions. Colors show ten-seed mean return on the common five-rate grid. White markers identify the selected rate at each depth, while the dashed line transfers the depth-2-selected rate. The high-return region shifts toward smaller learning rates as depth increases.

-0.178. These values are corroborating descriptions rather than matched replications of the primary experiment.

F Interaction-budget details

The main interaction-budget comparison uses the $\lambda = 1$ policy-depth continuations for cheetah-run, quadruped-run, and walker-run. All three tasks contain depths $2, 4, 8, 16$, and 32 through 150M interactions. Later budgets retain only two inherited learning-rate candidates per task and depth. These comparisons therefore characterize the available continuation search rather than a fresh five-rate optimization at every budget.

The runs at different budgets share training histories. Consequently, points connected across interaction budgets are not independent experiments. The reported intervals test the depth contrast within a budget and do not test whether the contrast itself changes significantly between budgets.

Figure 5 gives a complementary view of the interaction-budget and optimization-protocol effects on cheetah-run. The 20M and 100M panels use policy-depth experiments with a fixed depth-2 critic and $\lambda = 1$. At 100M, all five policy depths now have both the shallow-rate-transfer and per-depth-selected comparisons. The later-budget selection is restricted to two inherited learning-rate candidates per depth. The 300M panel uses a separate joint actor–critic configuration with $\lambda = 0.95$ and should not be interpreted as a continuation of the first two panels.

On cheetah-run, the depth-32 estimate moves from -23.1% at 20M to -5.8% at 50M and $+7.7\%$ at 100M. The 100M interval includes zero. At 150M, depths 4 and 8 exceed depth 2 by 9.7% and 8.3%, with both intervals above zero. Depths 16 and 32 have positive point estimates of $+4.0\%$ and $+8.4\%$, but their intervals remain unresolved.

Quadruped-run develops positive depth-4 estimates of 22.4%, 23.6%, and 19.3% at 50M, 100M, and 150M, but the corresponding intervals include zero. Walker-run retains negative deeper-policy point estimates across the evaluated interaction budgets.

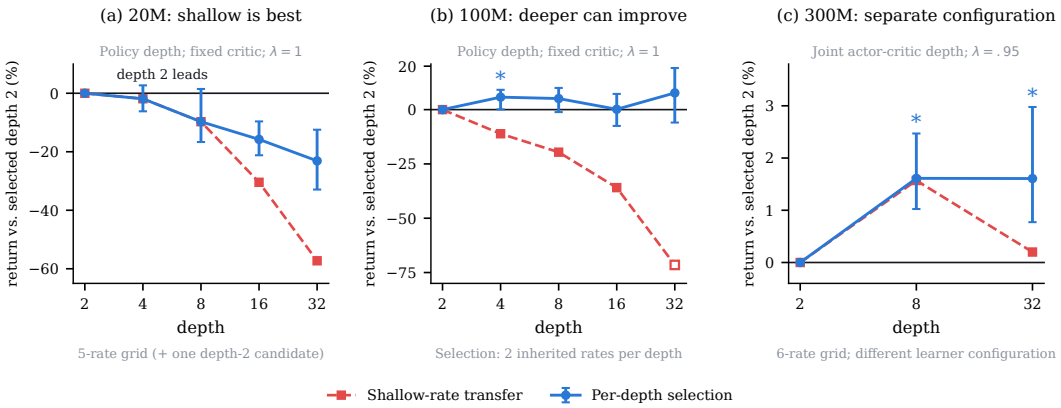


Figure 5: **Interaction budget and learning-rate protocol on an illustrative cheetah task.** At 20M interactions, depth 2 has the highest selected return. At 100M, the complete policy-depth comparison shows that per-depth selection can make deeper policies competitive or advantageous, while shallow-rate transfer produces much larger deficits. Selection at 100M is restricted to two inherited learning-rate candidates per depth. The 300M panel is a separate joint actor–critic depth experiment with $\lambda = 0.95$ and a different learner configuration. Error bars show paired 95% bootstrap intervals.

Table 7: Gymnasium $\lambda = 1$, 20M protocol comparison on the five main tasks. Gap is per-depth selection minus shallow-rate transfer before rounding.

Depth	Shallow transfer (%)	Per-depth selection (%)	Gap (points)
4	−6.9	−1.1	5.8
8	−30.5	−11.6	18.9
16	−49.7	−30.8	18.9
32	−61.1	−31.3	29.8

Ball-in-cup-catch and finger-spin also have 50M continuations. Their later coverage is less complete, so they are not used in the main interaction-budget figure. These continuations reinforce that the long-budget experiments use a restricted subset of tasks and learning-rate candidates.

These results support the conclusion that additional interaction can change the policy-depth comparison without establishing a universal crossover to favorable depth scaling.

G Gymnasium corroboration

The Gymnasium experiments use the same primary policy-depth learner and standard configuration filter as the DMC experiment (Towers et al., 2025). The $\lambda = 1$ grid contains Ant-v5, HalfCheetah-v5, Hopper-v5, Humanoid-v5, HumanoidStandup-v5, Swimmer-v5, Walker2d-v5, Pendulum-v1, and MountainCarContinuous-v0. The continuous-control environments use MuJoCo (Todorov et al., 2012).

Unlike the primary DMC grid, the available Gymnasium learning-rate windows vary with depth. The $\lambda = 1$, 20M experiments contain 41 task–depth cells, with five or ten seeds and between three and seven candidate rates per cell (Table 7). Fourteen of the 41 cells select a boundary candidate. We therefore interpret these results as corroborating within-window comparisons rather than as matched estimates of fully covered optima.

The main protocol summary uses Ant, HalfCheetah, Hopper, Humanoid, and Walker2d, which have the required depth coverage.

The protocol dependence therefore extends beyond DMC, although incomplete and depth-dependent candidate windows prevent a direct magnitude comparison between benchmarks. The task-wise depth comparisons also show the same broad pattern as DMC. Larger depths are generally unfa-

Table 8: Selected Gymnasium $\lambda = 0.95$ depth contrasts against depth 2 at the same interaction budget. The 20M values select within the available candidate windows. The 100M and 150M values use inherited learning rates. Asterisks indicate paired intervals excluding zero.

	Depth 8			Depth 32		
Task	20M	100M	150M	20M	100M	150M
Ant	-16.7^*	$+12.2^*$	$+23.7^*$	-21.9^*	$+1.8$	$+10.0^*$
HalfCheetah	$+3.7$	-18.3	-27.5	-8.2	-17.9^*	$+0.1$
Hopper	-29.4^*	-39.0^*	-21.6^*	-9.8^*	-36.5^*	-30.8^*
Humanoid	-20.7	-21.3^*	-21.5^*	-7.7	-16.7^*	-16.7^*
Walker2d	$+1.9$	-8.7^*	-11.2^*	-13.3	-14.6^*	-15.5^*

vorable after separate within-window selection, with relatively few resolved improvements. The complete DMC grid remains the basis of the primary inferential claims.

The longer-budget Gymnasium experiments provide an additional check on the interaction-budget conclusion (Table 8). These runs use $\lambda = 0.95$ and inherited learning rates at 100M and 150M, so they should be interpreted as fixed-configuration continuations rather than fresh searches at each budget.

Ant develops positive resolved contrasts at the longer budgets, while the other tasks do not show the same pattern. HalfCheetah depth 32 is approximately unchanged from depth 2 at 150M. These results support the main conclusion that additional interaction can reveal depth advantages on some tasks without producing a universal crossover.

H Evaluation learning curves

The following figures show the available evaluation trajectories for the principal experimental configurations. Each curve uses the learning rate selected by the final-window score for its task and depth. For display, each seed's evaluations are interpolated to a common set of points up to the shortest run endpoint. Lines show seed means and shading shows one pointwise standard error. The curves reuse the seeds used for learning-rate selection and therefore do not account for uncertainty in selecting the learning rate.

Figure 6 shows the complete primary DMC grid. These are the most important learning curves for interpreting the main results because they show all nine primary environments at 20M interactions.

Figure 7 shows the supporting joint actor–critic depth experiment at 300M interactions. Actor and critic depth vary together, so these curves should not be interpreted as a matched extension of the primary policy-depth experiment.

At 300M, the selected joint-depth results remain heterogeneous across tasks. CheetahRun has positive resolved differences at depths 8 and 32. HopperHop has positive but unresolved differences. HumanoidWalk remains negative, while WalkerRun has mixed unresolved outcomes. These results are consistent with the task-dependent long-budget behavior emphasized in the main paper.

Figure 8 shows the available DMC policy-depth learning curves with $\lambda = 0.95$ at 100M interactions. The tasks are cheetah-run, hopper-hop, humanoid-walk, quadruped-run, and walker-run.

These learning curves complement the final-window comparisons by showing learning speed, seed variability, and late-training behavior. A positive late-training return does not by itself establish a sample-efficiency advantage throughout training, and a transient peak need not translate into sustained final-window performance.

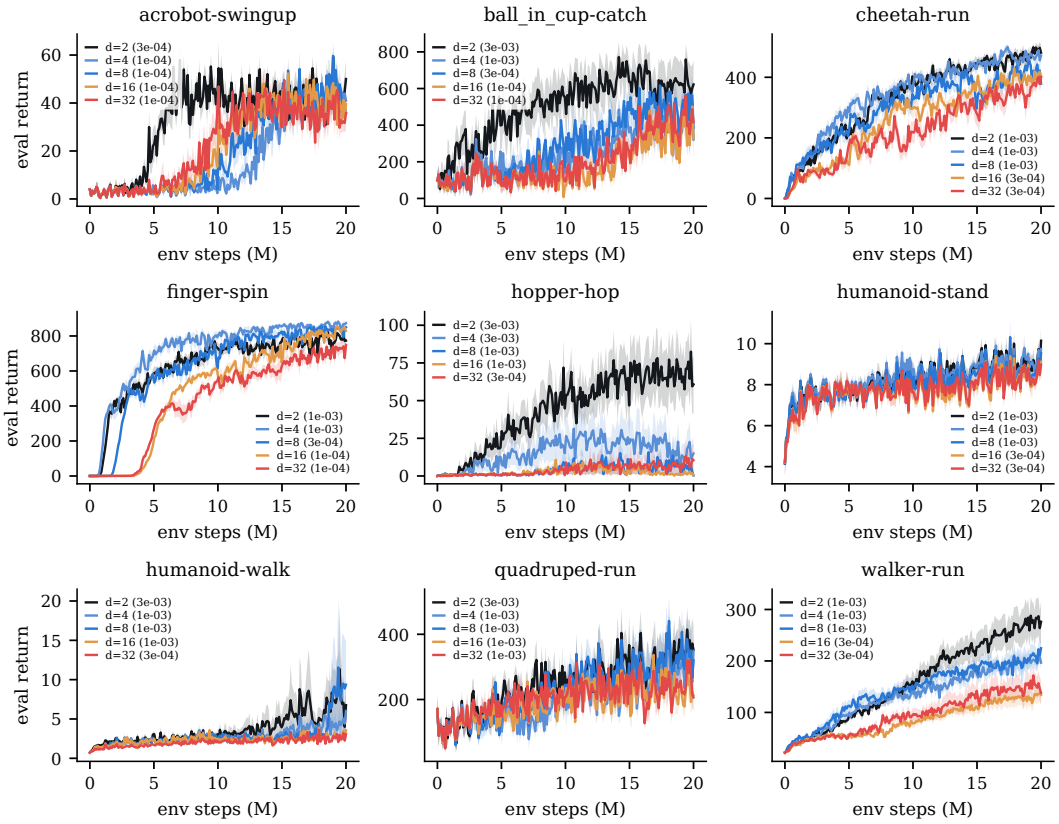


Figure 6: **Primary DMC policy-depth learning curves.** The nine tasks are acrobot-swingup, ball-in-cup-catch, cheetah-run, finger-spin, hopper-hop, humanoid-stand, humanoid-walk, quadruped-run, and walker-run. The experiment uses $\lambda = 1$, 20M interactions, and ten seeds per selected configuration. The learner and depth-2 critic are fixed while actor depth varies. Each depth is displayed at the learning rate selected using the final-window metric.

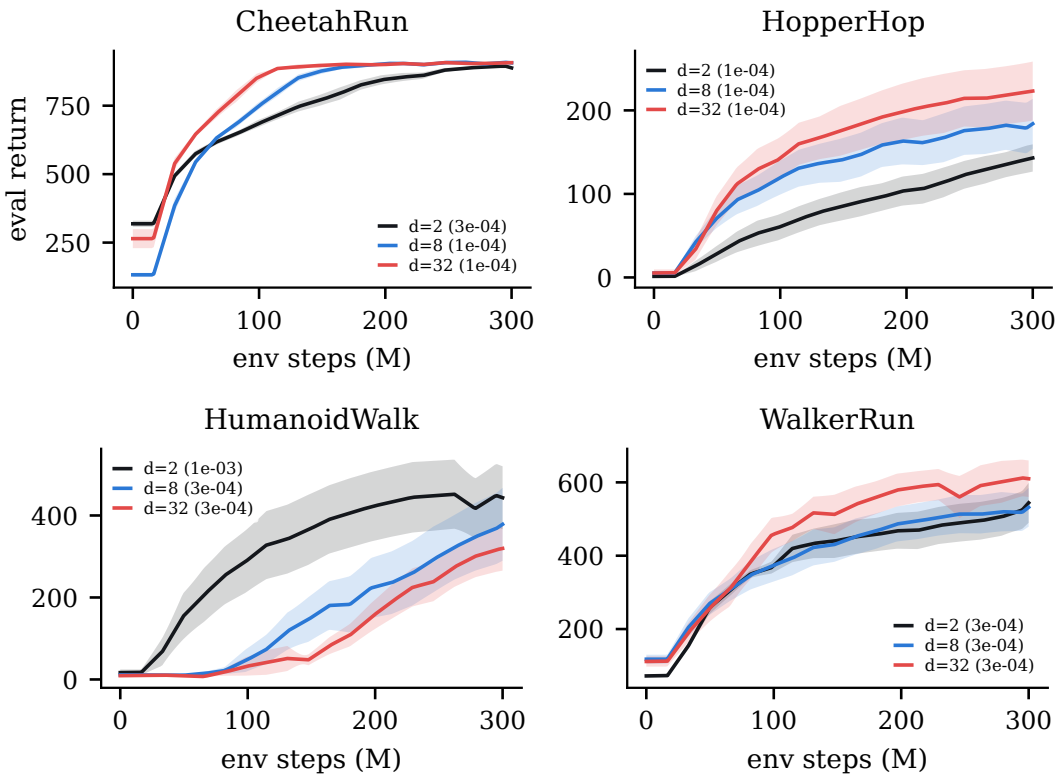


Figure 7: **Joint actor–critic depth learning curves at 300M interactions.** The four tasks are CheetahRun, HopperHop, HumanoidWalk, and WalkerRun. The experiment uses $\lambda = 0.95$, ten seeds, and actor and critic depths $2, 8$, and 32. Actor and critic depth scale together. The simulator and training settings differ from the primary policy-depth experiment, so comparisons should be made within this configuration.

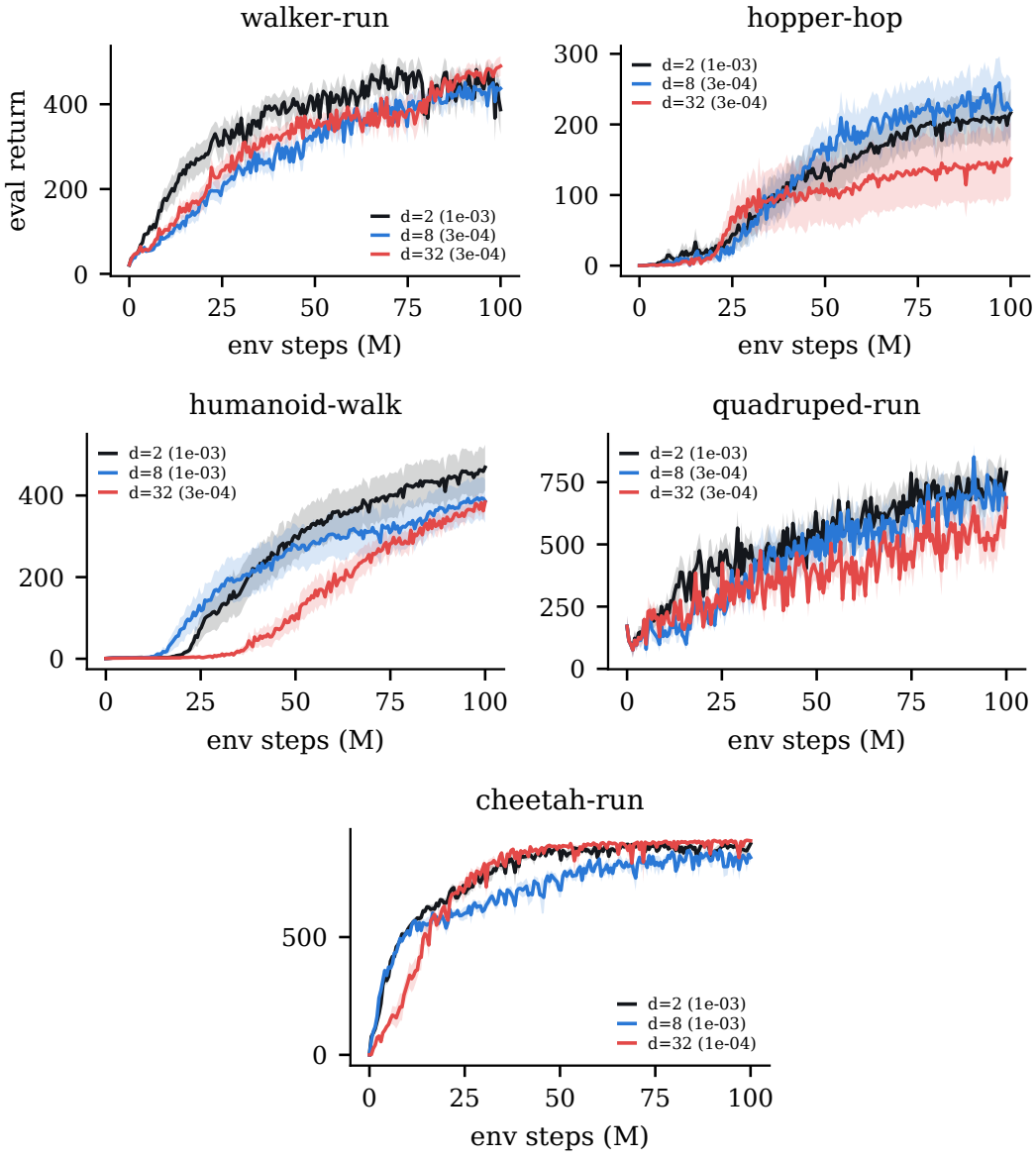


Figure 8: **DMC policy-depth learning curves with $\lambda = 0.95$ at 100M interactions.** The five tasks are cheetah-run, hopper-hop, humanoid-walk, quadruped-run, and walker-run. Each curve uses ten seeds. Cheetah-run has one tested learning rate per depth, while the other four tasks select among three or four available rates per depth. Candidate coverage remains incomplete for some task–depth cells, so these trajectories are supporting evidence rather than fully tuned long-budget comparisons.