RETHINKING DEPTH SCALING IN REINFORCEMENT LEARNING THROUGH HORIZON GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling neural networks has driven major advances in supervised and self-supervised learning, yet the role of network depth in reinforcement learning remains unclear. Prior studies report limited gains from deeper networks, while a recent study finds substantial benefits only in self-supervised reinforcement learning. We reveal a missing piece of this puzzle: depth has a limited effect on in-distribution performance but substantially improves generalization to unseen goals. We study horizon generalization in locomotion, manipulation, and navigation tasks across self-supervised, off-policy, and on-policy reinforcement learning methods. In most settings, deeper policies maintain higher success rates as evaluation goals move farther beyond the training distribution, even when shallow and deep models perform similarly on training goals. Deeper networks also outperform wider, shallow networks with comparable parameter counts, showing that the gains are not due to model size alone. Mechanistic analyses further reveal depth-dependent differences in how policies represent and use relative goal information. Our results suggest that network depth primarily extends the range over which reinforcement learning policies generalize, and that standard in-distribution evaluations may underestimate its value.

1 INTRODUCTION

Out-of-distribution (OOD) generalization is essential for reinforcement learning (RL) agents deployed in real-world systems (Packer et al., 2018; Cobbe et al., 2019; Ghosh et al., 2021). In robotics, an agent may encounter initial configurations, dynamics, physical parameters, or external disturbances that differ from those observed during training (Peng et al., 2018; Pinto et al., 2017). Despite its practical importance, OOD generalization remains relatively underexplored in RL, where most evaluations focus on performance within the training distribution (Cobbe et al., 2019). RL agents often perform poorly under distribution shifts because they learn from limited and policy-dependent interaction data (Cobbe et al., 2019; Ghosh et al., 2021). Goal-conditioned RL offers a useful setting for systematically studying this problem, as generalization can be measured by varying the goals presented at evaluation time (Kaelbling, 1993; Schaul et al., 2015; Andrychowicz et al., 2017). In particular, horizon generalization tests whether an agent can reach goals that require longer trajectories, or different goal distances, than those seen during training (Myers et al., 2025). This ability is important for agents that must operate beyond a fixed training distribution (Dulac-Arnold et al., 2021).

Deep architectures have been a major driver of progress in supervised and self-supervised learning (He et al., 2016; Chen et al., 2020). Increasing model depth and scale has improved representation quality, robustness, and emergent capabilities in computer vision (Li et al., 2018; Jia et al., 2021) and language modeling (Kaplan et al., 2020; Wei et al., 2022; Schaeffer et al., 2023). However, previous studies found limited benefits from depth scaling in RL (Ota et al., 2024; Nauman et al., 2024; Lee et al., 2025). Recent work (Wang et al., 2025) reported that increasing network depth greatly improves performance and enables new goal-reaching capabilities in Contrastive Reinforcement Learning (CRL) (Eysenbach et al., 2022). However, it argued that this scaling law is limited to self-supervised RL and that conventional RL algorithms do not benefit from depth scaling.

In this study, we complement the existing scaling narrative in RL by showing that the main benefit of increasing network depth may not appear in standard RL evaluations, but instead in improved

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.

OOD generalization. This pattern shows an emerging capability: in-distribution (ID) performance changes slightly with depth, while generalization can emerge and then strengthen once the network is sufficiently deep. We further show that this effect occurs across several algorithms and is not limited to self-supervised RL. For OOD generalization, depth scaling is more effective than width scaling, and increasing computational depth through looped layers also improves this ability without increasing the number of parameters.

We investigate this hypothesis through a systematic study of horizon generalization (Myers et al., 2025) across locomotion, navigation, and manipulation tasks. We evaluate agents on goals beyond the training distribution and compare networks with different depths and widths under both ID and OOD conditions. Our experiments cover standard on-policy and off-policy algorithms in addition to CRL, allowing us to separate effects specific to self-supervised goal-conditioned learning from those that hold more broadly across RL algorithms. Starting from a standard four-layer network, we find that scaling provides limited gains in ID evaluation but substantially improves performance on longer-horizon and unseen goals. This improvement is more pronounced under depth scaling than width scaling, suggesting that parameter count alone cannot explain the effect. Finally, using mechanistic interpretability tools, we investigate how depth affects the representation and causal use of relative goal direction within the policy.

The main contributions of this work are as follows:

- We show that network depth mainly improves OOD goal generalization, while its effect on ID performance is often limited.
- Experiments across contrastive, off-policy, and on-policy RL show that the effect holds across algorithms and several control domains with distinct characteristics.
- We find that depth generalizes better than width at similar parameter counts, and that repeated computation improves generalization without adding parameters.
- Using mechanistic interpretability tools, we find that deeper actors encode relative goal direction more accurately, while causal interventions show that direction-relevant representations influence policy behavior.

2 Related Work

Scaling neural networks has driven major gains in supervised and self-supervised learning (He et al., 2016; Chen et al., 2020), but similar benefits have been less consistent in RL. Deep residual architectures have enabled the optimization of substantially deeper networks in vision, while increasing model scale has also produced strong gains in language and vision–language representation learning (He et al., 2016; Kaplan et al., 2020; Jia et al., 2021).

Early work on neural scaling in RL found that increasing network size alone does not reliably improve performance. Ota et al. (2021) showed that wider actor–critic networks often outperform deeper ones in continuous-control tasks. Their strongest results used shallow, wide networks, while deeper standard MLPs were harder to optimize. However, these depth comparisons did not use the residual and normalization methods later developed for stable scaling. The study also evaluated SAC (Haarnoja et al., 2018) and TD3 (Fujimoto et al., 2018) only on standard benchmark tasks, without testing generalization beyond the training distribution.

Later studies showed that larger RL networks can help when paired with suitable architectures and regularization. BRO improved performance by scaling a regularized critic, while SimBa used residual connections and normalization to make parameter scaling more stable. In both studies, the main gains came from critic capacity, especially width, while actor scaling was less reliable. SimBa also found width scaling more effective than depth scaling for the critic. These studies focused on performance within the training task distribution and examined only modest depth ranges.

Wang et al.Wang et al. (2025) challenged the view that depth provides limited benefits in RL by showing strong scaling effects in self-supervised contrastive RL. They scaled CRL actors and critics from a few layers to hundreds or more than one thousand layers and reported large gains across various control tasks. Based on comparisons with conventional RL algorithms, they argued that these benefits of depth are specific to self-supervised RL and do not extend to standard RL methods.

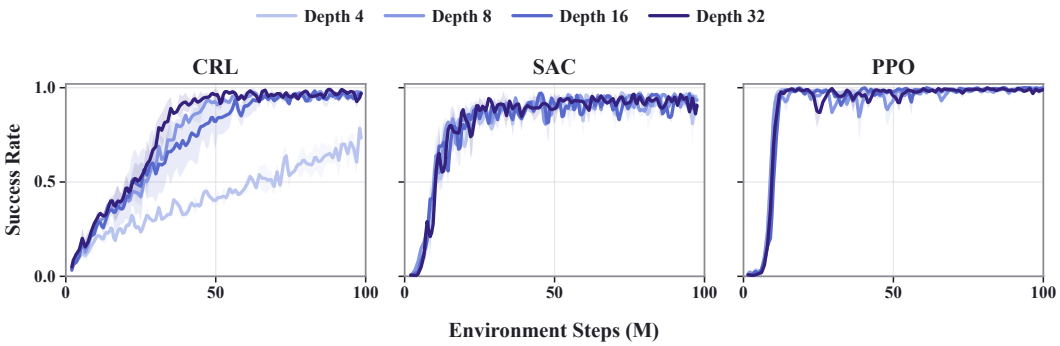


Figure 1: **In-distribution success rate during training.** CRL, SAC, and PPO are evaluated on Humanoid across four network depths: 4, 8, 16, and 32.

A separate line of work has studied generalization in RL under distribution shift (Packer et al., 2018; Cobbe et al., 2019; Ghosh et al., 2021). Goal-conditioned RL provides a natural setting for evaluating generalization to unseen goals (Schaul et al., 2015; Andrychowicz et al., 2017), while horizon generalization specifically examines whether policies can solve tasks requiring longer behavior than that encountered during training (Myers et al., 2025). However, how network depth affects generalization beyond the training distribution remains largely unexplored.

3 Revisiting What Depth Improves in RL

Existing work has established seemingly conflicting views of depth in reinforcement learning. Studies based on standard benchmarks have found limited gains in deeper networks, while recent results in contrastive RL suggest that substantial depth can produce qualitatively new capabilities. Yet these findings do not fully identify what depth contributes. Most prior evaluations remain within the training distribution, while strong evidence for deep scaling has primarily been shown in CRL. As a result, limited gains on standard tasks and large gains in very deep models may describe different aspects of the same phenomenon rather than opposing conclusions. This motivates a broader hypothesis: the main benefit of depth in RL may emerge not as higher performance on familiar tasks, but as stronger generalization beyond the conditions encountered during training.

We study this possibility in RL using the goal-conditioned setting, where the evaluation distribution can be changed directly by varying the goals given to the policy. Horizon generalization provides a direct OOD test by asking whether policies trained on nearby goals can reach more distant goals that require longer behavior (Myers et al., 2025). We use this setting as a proxy for systematically studying a specific form of OOD generalization in RL. It helps separate success on familiar goals from the ability to extend learned behavior beyond the training range. To determine whether the role of depth is specific to self-supervised learning, we consider several widely used forms of RL, including contrastive, standard off-policy, and on-policy methods. This broader view allows us to distinguish an algorithm-specific scaling effect from a more general role of depth across RL algorithms.

Our central hypothesis is that depth primarily expands the range over which a learned policy generalizes, rather than simply improving its performance on the training distribution. Under this view, differences between shallow and deep networks may remain small on familiar goals but become increasingly clear as goals move farther from the training support. This effect may appear to follow from greater expressive power or overparameterization, since deeper networks also contain more parameters. However, depth changes not only model capacity but also the structure of computation by applying a longer sequence of transformations to state and goal information. We therefore expect deeper policies to generalize better than shallow, wide policies with similar parameter counts. We further examine the role of expanded computation by constructing agents with repeated network blocks that share parameters. Overall, our perspective positions OOD generalization, rather than standard benchmark performance alone, as a central measure of what depth contributes to RL.

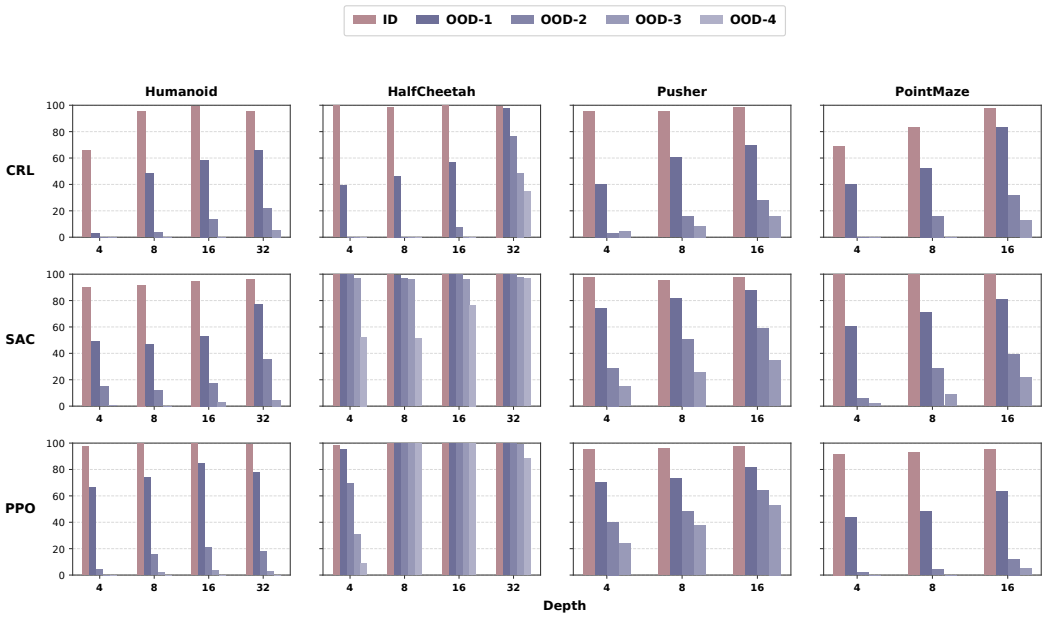


Figure 2: **Effect of depth scaling on out-of-distribution goal generalization.** Success rates are shown for ID and progressively more distant OOD goal regions across environments and algorithms.

4 Experiments

4.1 Experimental Setup

Environments. We evaluate depth scaling across four goal-conditioned continuous-control environments spanning distinct task domains: Humanoid and HalfCheetah for locomotion, Pusher for object manipulation, and PointMaze for navigation. Together, these environments provide diverse state and action spaces and goal geometries for studying whether the generalization benefits of depth extend across different control problems.

Goal Distributions. Across all environments, we construct progressively harder OOD evaluations by shifting the goal distribution beyond its training support while keeping all other evaluation conditions fixed. Specifically, the environment dynamics, initial-state distribution, observation and action spaces, episode horizon, and success criterion remain unchanged. The form of the distribution shift is adapted to the geometry of each environment.

In Humanoid, training goals are sampled within a bounded radial region around the agent, while successive OOD levels place goals at progressively larger radial distances, preserving the target-direction distribution and target height. In HalfCheetah, goals lie along the forward axis, and successive OOD levels move them progressively farther beyond the range encountered during training. In Pusher, training goals are sampled from an inner rectangular region of the workspace. We construct a sequence of nested, progressively larger regions, with each OOD level corresponding to the newly introduced area outside the preceding region. In PointMaze, the initial location is fixed, and successive OOD levels contain goals with progressively longer shortest-path distances than those encountered during training. Exact goal ranges and spatial definitions are provided in Appendix B.1.

Algorithms and Architectures. We evaluate CRL (Eysenbach et al., 2022), Soft Actor-Critic (SAC) (Haarnoja et al., 2018), and Proximal Policy Optimization (PPO) (Schulman et al., 2017). CRL provides a direct comparison with the depth-scaling results of Wang et al. (2025), while SAC and PPO allow us to test whether these effects extend beyond contrastive RL.

Following Wang et al. (2025), all networks are residual multilayer perceptrons (MLPs) with LayerNorm and Swish activations. The hidden width is 256, and we evaluate

$$d \in \{4, 8, 16, 32\}.$$

Each residual block contains four Dense layers, so a depth-d network contains $d/4$ residual blocks and d Dense layers in its residual core. Input projections and output heads are excluded from the reported depth. We scale the actor and all corresponding critic or value networks jointly; thus, a depth-d configuration uses depth-d networks throughout.

Evaluation Metric. Our primary metric is the episode success rate (SR). An episode is considered successful if the agent reaches the goal-specific success region at least once within the evaluation horizon:

$$\text{Success}(\tau, g) = \mathbb{I}\left[\min_{0 \le t \le H} d_g(s_t, g) \le \epsilon_g\right],$$

where $d_g(s_t, g)$ denotes the environment-specific distance between the state and goal, ϵ_g is the corresponding success threshold, and H is the evaluation horizon.

4.2 In-Distribution Evaluation

We first evaluate policies on goals drawn from the training distribution, following the experimental protocol of Wang et al. (2025). This provides an in-distribution baseline and tests whether the previously reported depth-scaling effect is reproduced in our setting. Figure 1 reports the Humanoid results, with complete in-distribution learning curves for the remaining environments provided in Appendix A.1.

For CRL, increasing depth leads to faster learning, with deeper networks reaching high success rates substantially earlier than the depth-4 model. In contrast, SAC and PPO show considerably less sensitivity to depth: their learning curves largely overlap across the tested depths and converge to similarly high success rates. Overall, depth has a more pronounced effect on in-distribution learning for CRL, while its effect is comparatively limited for SAC and PPO. These in-distribution results are consistent with the findings of Wang et al. Wang et al. (2025), reproducing the same qualitative depth-scaling pattern. A complementary spatial coverage analysis in Appendix A.2 further shows that the shallow CRL policy visits a substantially smaller portion of the training-goal space, whereas SAC and PPO exhibit broad coverage even at shallow depths.

4.3 Depth Improves Generalization to Unseen Goals

Figure 2 reveals a pattern that is largely hidden by in-distribution evaluation. On ID goals, many configurations already achieve high or saturated success, leaving relatively little separation between depths. As evaluation moves beyond the training support, however, the difference becomes increasingly visible: shallow policies fail earlier, whereas deeper policies maintain higher success over progressively larger goal-distribution shifts. Depth therefore changes not only the level of performance, but also how far learned behavior extrapolates beyond the training range.

This contrast is particularly striking in HalfCheetah. Under CRL, all depths perform similarly on ID goals, yet $d = 32$ exhibits a qualitative jump in generalization, retaining substantial success through OOD-4 while shallower models collapse much earlier. SAC exhibits an analogous separation despite near-saturated ID performance, and PPO likewise achieves higher success beyond the training range as depth increases. Humanoid reinforces the same pattern: even when ID performance is comparable across depths, intermediate and deeper policies retain substantially higher success after moving beyond the training support. Beyond locomotion, in both Pusher and PointMaze, success in the more distant OOD regions increases systematically with depth across CRL, SAC, and PPO.

Although the ordering is not perfectly monotonic in every individual setting, the overall failure boundary moves outward with depth. Importantly, this behavior appears in self-supervised CRL, off-policy SAC, and on-policy PPO, and across locomotion, manipulation, and navigation. These results support our central hypothesis: the primary benefit of depth is expanding the range over which learned goal-directed behavior continues to generalize. The benefit is also not restricted to shifts in the goal distribution: Appendix A.4 shows that increasing depth can improve generalization under shifts in the initial-state distribution.

4.4 OOD Generalization Favors Depth over Width

Increasing depth also increases the number of parameters, raising a natural question: are the generalization gains simply a consequence of larger models? To separate depth from model size, we compare depth scaling at a fixed width of 256 with width scaling at a fixed depth of 4. Since the dominant parameter count scales approximately as $\mathcal{O}(d)$ with depth and $\mathcal{O}(w^2)$ with width, doubling the depth is approximately matched by multiplying the width by $\sqrt{2}$. We therefore compare

$$d \in \{4, 8, 16, 32\} \qquad \text{against} \qquad w \in \{256, 362, 512, 724\},$$

with corresponding configurations having approximately matched parameter counts.

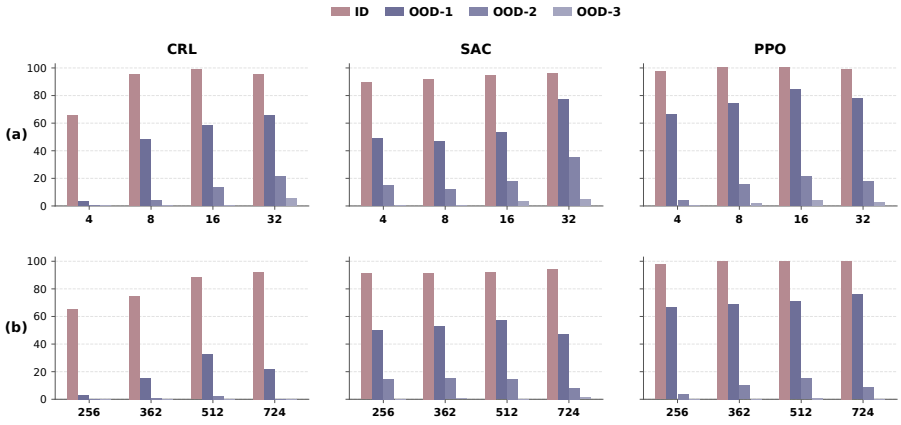


Figure 3: **Effect of depth and width scaling on goal generalization.** Success rates are shown across ID and OOD goal regions, with approximately matched parameter counts between corresponding depth- and width-scaled models. (a) Depth scaling with width fixed at 256. (b) Width scaling with depth fixed at 4.

Figure 3 shows that model size alone does not reproduce the effect of depth. For CRL, widening improves ID performance and provides some gains on nearby OOD goals, but these gains largely disappear as the distribution shift increases. In contrast, increasing depth progressively extends success into more distant OOD regions; at the largest matched scale, the depth-32, width-256 policy generalizes substantially farther than the depth-4, width-724 policy.

The same distinction appears beyond CRL. For SAC, depth and width can perform similarly at smaller shifts, but their behavior separates sharply at larger scales: the depth-32 model retains substantially more success in OOD-1 and OOD-2 than its width-scaled counterpart. PPO exhibits a similar advantage for depth, particularly in OOD-2, although the gains saturate beyond depth 16. Thus, adding parameters is not sufficient to explain the observed generalization gains. As goals move farther from the training support, *where* capacity is added matters: allocating it to additional depth extends the generalization frontier more effectively than allocating it to width.

4.5 Extended Computation with a Repeated Residual Module

The depth–width comparison shows that parameter count alone does not explain the OOD generalization gains of deeper networks. Because deeper models also perform more sequential transformations of their intermediate representations, we isolate the contribution of this additional computation using a weight-tied architecture. Specifically, the architecture repeatedly applies the same residual module, increasing computational depth without adding trainable parameters.

Let F_θ denote the transformation implemented by the residual module, with parameters θ, and let $z^{(0)} = z$ denote its input representation. The standard model passes the representation through this module once. In the looped model, the same module is applied L times:

$$z^{(k+1)} = F_\theta\left(z^{(k)}\right), \qquad k = 0, \ldots, L-1.$$

The final representation is therefore

$$z^{(L)} = F_\theta^{(L)}(z) = \underbrace{(F_\theta \circ F_\theta \circ \cdots \circ F_\theta)}_{L \text{ times}}(z).$$

Because the module and its parameters are reused at every iteration, increasing L adds sequential computation without changing the number of trainable parameters. We apply this construction separately to the actor and critic, with parameters shared across iterations within each component but not between the two components.

Figure 4 shows that repeating the residual module improves both CRL and SAC. Extended computation also improves ID performance, but the gains are more pronounced on OOD goals. These results indicate that additional sequential computation contributes to the generalization advantage of deeper networks independently of parameter count. However, weight-tied looping does not fully match deeper networks with independently parameterized layers, suggesting that repeated computation explains only part of the benefit of depth.



Figure 4: **Effect of repeated computation on goal generalization.** Each standard network is compared with its weight-tied looped counterpart, which increases computational depth without adding trainable parameters.

4.6 Deep RL and Invariant Representation

The behavioral results show that increasing depth can substantially improve generalization beyond the training distribution, particularly for CRL and SAC on Humanoid. We next investigate what changes within these deeper policies might underlie these gains. We find that deeper networks increasingly encode the relative state-goal configuration in a manner that is invariant to translation and scale (i.e., $v = \frac{g-s}{\|g-s\|}$), suggesting that this emergent invariance is a key mechanism underlying their improved generalization beyond the training horizon.

We first examine how readily relative goal direction can be recovered from actor representations. For CRL and SAC Humanoid policies, we train a linear probe at the output of each residual block to predict the normalized direction from the agent to the goal. As shown in Figure 5, this direction can be decoded with lower error from the internal activations of deeper actors in both algorithms.

Linear decodability alone, however, does not establish whether this information actually influences policy behavior. We therefore directly intervene on the actor's representation to test whether shifting its encoded goal direction toward a different goal correspondingly redirects the policy. For each Humanoid actor and state, we construct two goals, g_A and g_B, at the same distance from the agent but in different directions. The environment always retains g_A as its physical goal. At the same physical state, we evaluate the actor under both g_A and the counterfactual goal g_B, obtaining their corresponding hidden representations. We then intervene on the representation associated with $v_A = \frac{g_A-s}{\|g_A-s\|}$ by selectively shifting them toward the corresponding components induced by g_B. Thus, the physical state, environment goal, and reward remain unchanged, while only the actor's internal representation is modified.

To obtain a sparse basis for this intervention, we train a Sparse Autoencoder (SAE) on hidden actor activations (Gao et al., 2025). The SAE maps the dense hidden representation to a sparse latent code, allowing us to selectively intervene on a small subset of representation components. While sparse feature representations have been used for causal analysis and model steering in language

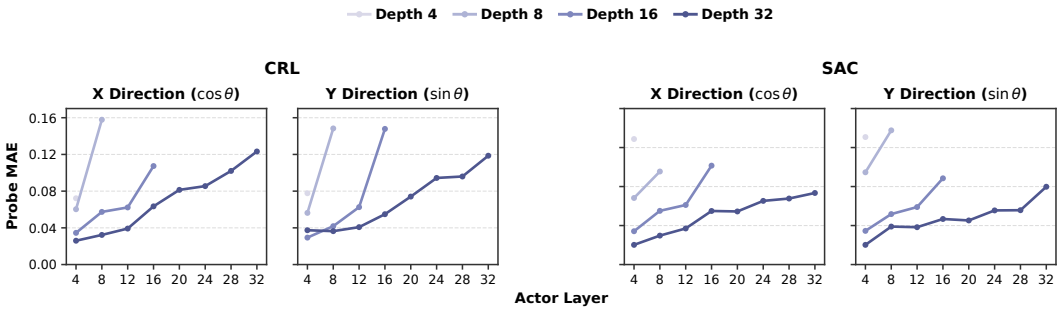


Figure 5: **Linear probing of relative goal direction across actor layers.** Linear probes predict the normalized planar direction from the agent to the goal using residual-block activations of CRL and SAC Humanoid actors. Lower held-out MAE indicates greater linear accessibility of relative goal direction.

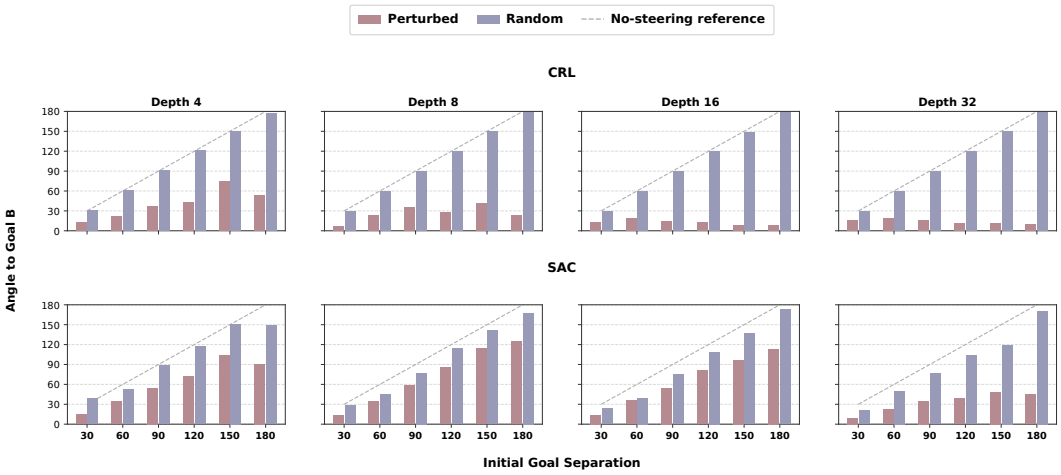


Figure 6: **Causal intervention on relative goal representations.** Rows show CRL and SAC, and columns correspond to network depths. Bars report the angle between the agent's final displacement and the direction toward the counterfactual goal g_B under targeted and matched random interventions across increasing initial goal separations. Lower angles indicate stronger steering toward g_B; the dashed line shows the no-steering reference.

models (Makelov et al., 2025; Marks et al., 2025), here we use them to intervene on the internal representations of continuous-control policies.

At each time step, we identify direction-relevant sparse features with the same pretrained direction probe used in the linear probing analysis and *cumulative greedy ablation*. Starting from the natural sparse code, we temporarily ablate active candidate features and select the feature whose removal most increases the probe's normalized relative direction error. Previously selected features remain ablated while selecting the next one, and the procedure is applied separately to the representations induced by g_A and g_B at each time step. Additional details on SAE training and the formal probe-guided feature-selection procedure are provided in Appendix C.

We then shift the representation associated with g_A toward that induced by g_B by replacing the selected sparse feature values with their corresponding values under g_B. We apply only the resulting change to the actor's original hidden representation, preserving the natural representation outside the intervention. As a matched random control, we apply the same procedure to the same number of randomly selected features.

As shown in Figure 6, we measure behavioral steering by the angle between the agent's final displacement and the direction toward the counterfactual goal g_B. Smaller angles indicate stronger steering toward g_B, even though the environment continues to use g_A as its physical goal. For

CRL, targeted interventions consistently produce smaller angles than matched random interventions across goal separations, showing that selectively manipulating direction-relevant sparse components systematically redirects subsequent policy behavior.

SAC exhibits the same qualitative behavior, indicating that the effect is not specific to CRL. The distinction between targeted and random interventions is particularly pronounced for the deepest actors. Notably, the strongest causal steering effects in Figure 6 occur at depths where CRL and SAC also exhibit substantial improvements in OOD goal generalization on Humanoid (Figure 2).

Figure 7 illustrates the intervention on a representative goal pair across network depths. Natural rollouts toward g_A and g_B provide behavioral references, while targeted and random interventions are applied with g_A retained as the physical goal.

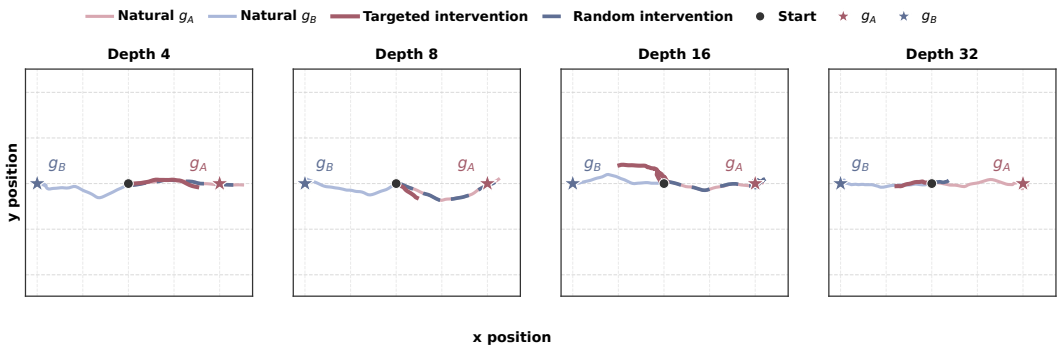


Figure 7: **Representative trajectories under causal intervention.** Targeted interventions on direction-relevant representations shift policy behavior toward the counterfactual goal, illustrating the behavioral effect of the intervention.

These results reveal a consistent change in how deeper policies represent and use relative goal information. Deeper actors better encode relative goal direction, while interventions on direction-relevant components shift behavior toward the counterfactual goal. This provides a possible representational explanation for the generalization benefits of depth: better encoding of relative state–goal geometry may support generalization to goals at longer horizons and to unseen initial states (Appendix A.4).

5 Conclusion

We studied how network depth affects horizon generalization in reinforcement learning across locomotion, manipulation, and navigation tasks using contrastive, off-policy, and on-policy methods. Increasing depth produced limited or algorithm-dependent gains on goals from the training distribution, but generally improved success on unseen goals as their distance from the training range increased. Deeper networks also generalized better than wider shallow networks with similar parameter counts, while weight-tied looping showed that additional sequential computation contributes to these gains. These benefits also extend beyond unseen goals, with deeper policies showing improved generalization under shifts in the initial-state distribution. Our mechanistic analysis further showed that deeper actors encode relative goal direction more accurately, while causal interventions on direction-relevant representations systematically shift policy behavior toward a counterfactual goal. Overall, our results suggest that the main value of depth in reinforcement learning may lie in extending policy generalization rather than improving performance on familiar tasks.

Limitations. Our study establishes the generalization benefits of depth across multiple RL algorithms, task domains, and forms of distribution shift. An important direction for future work is to determine how broadly these findings extend beyond the settings considered here, including to offline RL, additional task domains, and other forms of OOD generalization.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work to assist with language editing and polishing, code development and debugging, and figure preparation. All experiments and evaluations reported in this paper were designed and conducted by the authors. AI-assisted code was reviewed and tested by the authors, and all AI-assisted manuscript content was carefully reviewed and verified before inclusion in the paper.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Michał Bortkiewicz, Władysław Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating goal-conditioned reinforcement learning algorithms and research. In *International Conference on Learning Representations (ICLR)*, 2025.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1282–1289. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cobbe19a.html.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021. doi: 10.1007/s10994-021-05961-4.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35603–35620, 2022. doi: 10.52202/068431-2580. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/e7663e974c4ee7a2b475a4775201ce1f-Paper-Conference.pdf.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax – a differentiable physics engine for large scale rigid body simulation. *arXiv preprint arXiv:2106.13281*, 2021.

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

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, 2025.

Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P. Adams, and Sergey Levine. Why generalization in RL is difficult: Epistemic POMDPs and implicit partial observability. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/d5ff135377d39f1de7372c95c74dd962-Abstract.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 *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4904–4916. PMLR, 2021. URL https://proceedings.mlr.press/v139/jia21b.html.

Leslie Pack Kaelbling. Learning to achieve goals. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1993.

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.

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 (ICLR)*, 2025.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, 2018.

Aleksandar Makelov, Georg Lange, and Neel Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control. In *International Conference on Learning Representations*, 2025.

Samuel Marks, Can Rager, Eric Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations*, 2025.

Vivek Myers, Catherine Ji, and Benjamin Eysenbach. Horizon generalization in reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

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 (NeurIPS)*, 2024.

Kei Ota, Devesh K. Jha, and Asako Kanezaki. A framework for training larger networks for deep reinforcement learning. *Machine Learning*, 113:6115–6139, 2024. doi: 10.1007/s10994-024-06547-6.

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-Real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018. doi: 10.1109/ICRA.2018.8460528.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2817–2826. PMLR, 2017. URL https://proceedings.mlr.press/v70/pinto17a.html.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320, Lille, France, 2015. PMLR. URL https://proceedings.mlr.press/v37/schaul15.html.

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

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

Kevin Wang, Ishaan Javali, MichałBortkiewicz, Tomasz Trzcinski, and Benjamin Eysenbach. 1000 layer networks for self-supervised rl: Scaling depth can enable new goal-reaching capabilities. In *Advances in Neural Information Processing Systems (NeurIPS 2025)*, pp. 157643–157670, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/e74ee34cc0f2d0780f34ee77d8fba25b-Abstract-Conference.html.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

A Additional Experimental Results

A.1 In-Distribution Learning Across Environments

To complement the Humanoid results in Figure 1, we report the in-distribution learning curves for the remaining environments. Figure 8 shows success rate throughout training for CRL, SAC, and PPO across network depths.

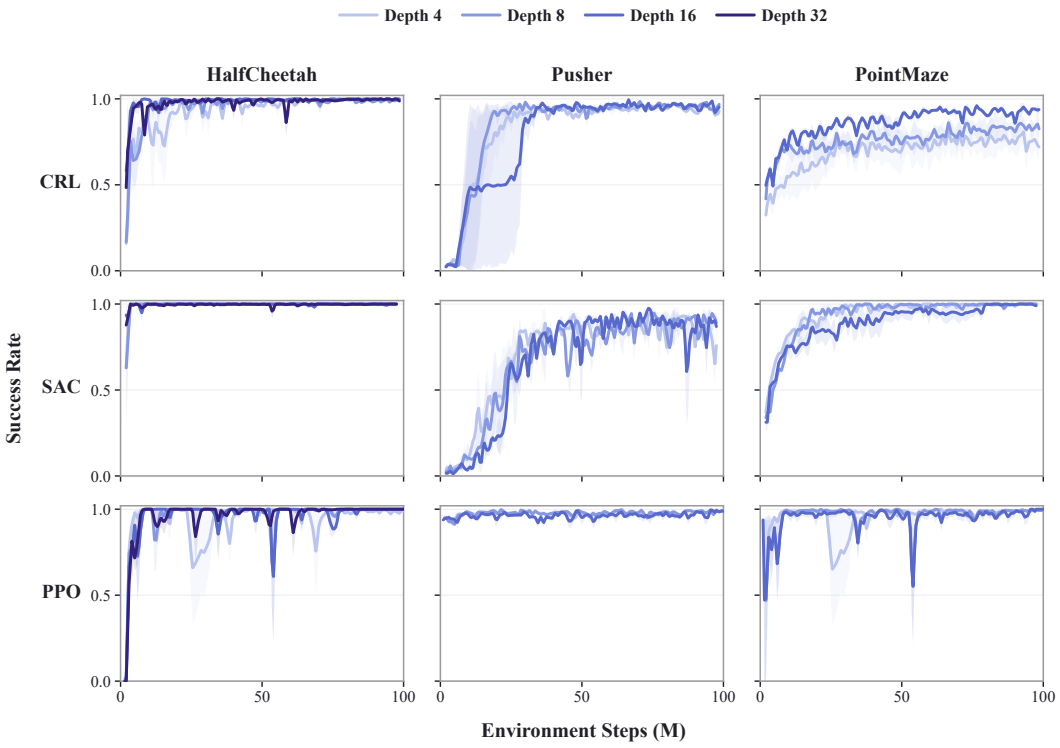


Figure 8: **In-distribution learning across environments.** Success rate during training for CRL, SAC, and PPO on HalfCheetah, Pusher, and PointMaze across network depths.

Across environments, increasing depth does not consistently improve final ID performance. Although depth can affect learning speed in some settings, the different depths often converge to similar success rates. Together with the Humanoid results, these findings show that the effect of depth is comparatively limited under in-distribution evaluation, in contrast to the substantially larger differences observed under OOD goal shifts.

A.2 Spatial Coverage on Training Goals

Figure 1 shows that the depth-4 CRL policy learns substantially more slowly and reaches lower in-distribution performance than deeper CRL policies, whereas SAC and PPO show much smaller differences across depth. To better understand this behavior, we examine the spatial regions visited by the final Humanoid policies on training-distribution goals.

For each policy, we record its horizontal position (x_t, y_t) throughout the evaluation episodes and discretize the common spatial region into a 120×120 grid. A cell is considered visited if the policy reaches it at least once. We use the same spatial limits and discretization across all algorithms and depths.

Figure 9 reveals a pronounced difference for CRL. The depth-4 policy remains concentrated within a relatively small region, whereas deeper CRL policies traverse substantially larger portions of the training-goal space and extend their trajectories across a broader range of directions. This behavior is

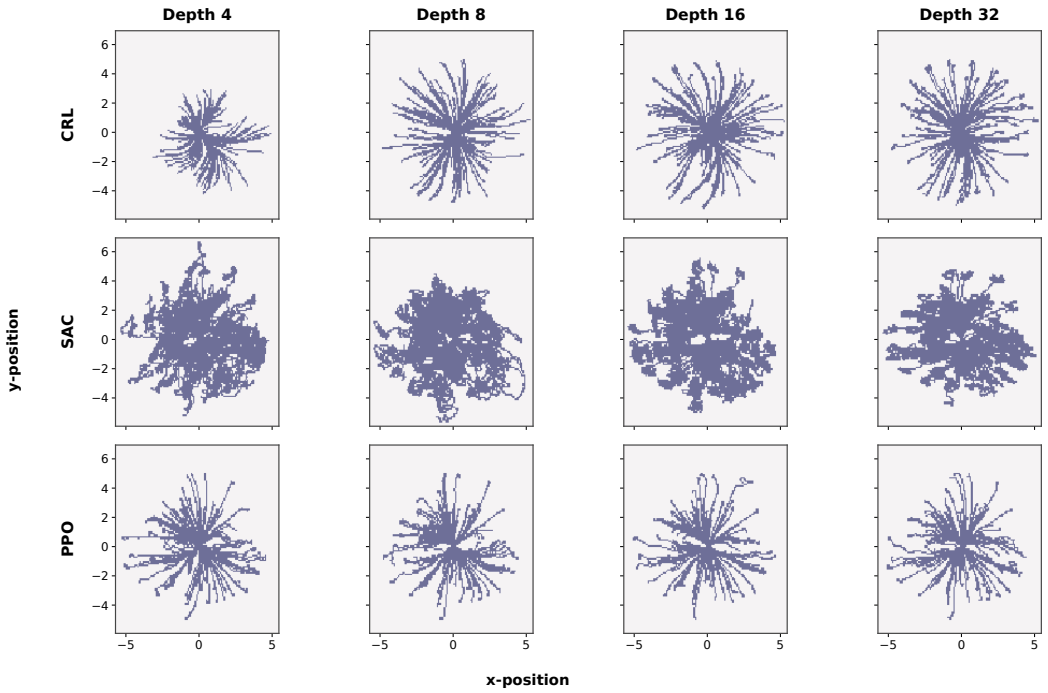


Figure 9: **Spatial coverage on Humanoid training goals.** Rows show CRL, SAC, and PPO, and columns correspond to network depth.

consistent with the large ID performance gap observed for CRL in Figure 1: the shallow CRL policy does not reliably cover the range of behaviors required to reach the full training-goal distribution.

In contrast, SAC and PPO already exhibit broad spatial coverage at shallow depths, with considerably smaller changes as depth increases. This mirrors their ID learning curves, where performance is much less sensitive to network depth. The coverage analysis therefore provides a behavioral explanation for why the strongest in-distribution depth effect in Figure 1 appears specifically for CRL.

A.3 Generalization Dynamics During Training on Humanoid

To examine how OOD generalization develops during learning, we evaluate intermediate Humanoid checkpoints for CRL, SAC, and PPO at depths $d \in \{4, 8, 16, 32\}$. Each checkpoint is evaluated on the training goal distribution and the same four OOD regions used in the main experiments.

Figure 10 shows a clear separation between learning the training distribution and generalizing beyond it. In CRL, shallow policies improve steadily on ID goals but generalize little, whereas deeper policies develop substantially stronger OOD performance as training progresses. A similar pattern appears in SAC and PPO: ID success often becomes high early, while OOD performance continues to improve and increasingly separates across depths. In particular, deeper policies can achieve markedly stronger OOD-1 and OOD-2 success despite nearly identical ID performance.

Depth therefore affects more than final performance: it changes how generalization emerges during training. Consequently, ID learning curves alone can substantially underestimate the benefit of network depth.

A.4 Generalization to Unseen Initial States

In our main experiments, we move the goal beyond the training distribution while keeping the initial-state distribution fixed. Here, we perform the complementary experiment and instead shift the initial-state distribution. The trained policies are evaluated without adaptation on initial states that move

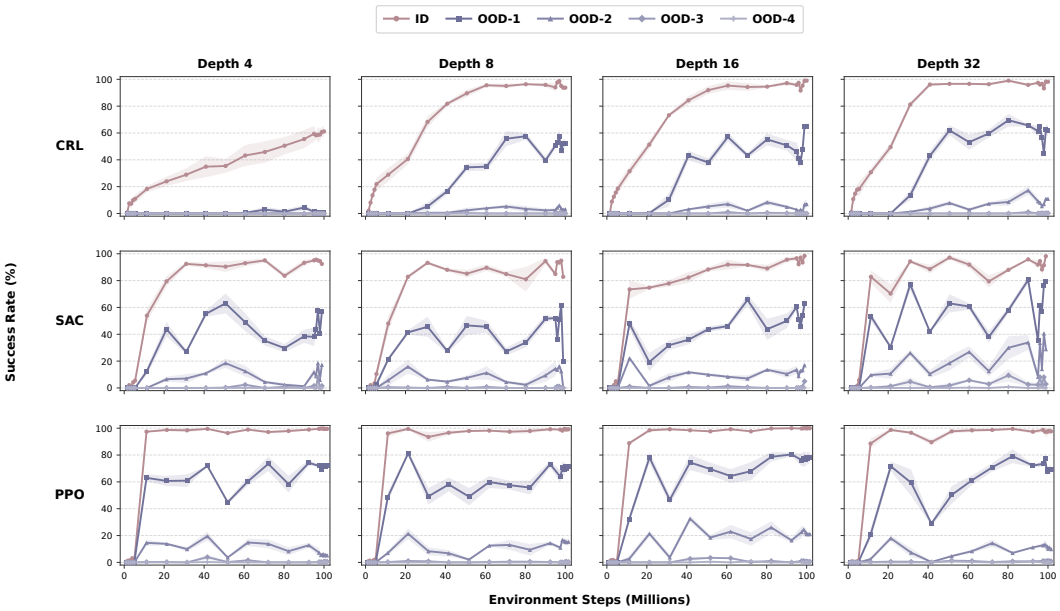


Figure 10: **Generalization during training on Humanoid.** Rows show CRL, SAC, and PPO, while columns correspond to depths 4, 8, 16, and 32. Depth substantially changes the emergence of OOD success even when ID performance is similar or already saturated.

progressively beyond the training distribution, while the environment dynamics, episode horizon, and success criterion remain unchanged.

For Humanoid and HalfCheetah, we translate the start and goal jointly so that the relative goal remains unchanged while the absolute configuration moves outside the training distribution. In Humanoid, the agent and goal are translated together on the ground plane by a common vector Δ:

$$s'_{0,xy} = s_{0,xy} + \Delta, \qquad g'_{xy} = g_{xy} + \Delta, \qquad \Delta = r_s[\cos\theta, \sin\theta].$$

For ID, $r_s = 0$, while for OOD-k, $r_s \sim \mathcal{U}(k, k+1)$. Thus, OOD-1 through OOD-4 correspond to radial translations of $1-2$, $2-3$, $3-4$, and $4-5$, respectively.

In HalfCheetah, the initial root position and target are jointly translated along the x-axis:

$$x'_0 = x_0 + \delta_x, \qquad g'_x = g_x + \delta_x.$$

For ID, $\delta_x = 0$, while for OOD-k, the translation is sampled from $(2k, 2k+2)$. Accordingly, OOD-1 through OOD-4 correspond to translations of $2-4$, $4-6$, $6-8$, and $8-10$, respectively.

For Pusher, we instead shift only the initial object position while keeping the goal distribution unchanged. We denote the training start region by S_0. OOD initial states are constructed using progressively larger nested rectangles around this region. Each OOD level corresponds to the new outer shell introduced by the next rectangle, so that higher OOD levels place the object progressively farther outside the initial-position region observed during training.

Figure 11 shows that the benefit of depth is not limited to generalization to unseen goals. Increasing depth can also substantially improve robustness to shifts in the initial-state distribution, with the clearest gains appearing for CRL in Humanoid and HalfCheetah, and for SAC and PPO in HalfCheetah. In these settings, deeper policies maintain high success under substantially larger shifts, while shallow policies degrade much earlier. Together with our goal-OOD results, these findings provide further evidence that increasing depth can expand the range over which learned behavior generalizes beyond the training distribution.

A.5 Disentangling Actor and Critic/Value Depth

In our main experiments, the actor and the corresponding critic or value network are scaled jointly. To disentangle their contributions, we independently vary their depths over $\{4, 8, 16, 32\}$ on Hu-

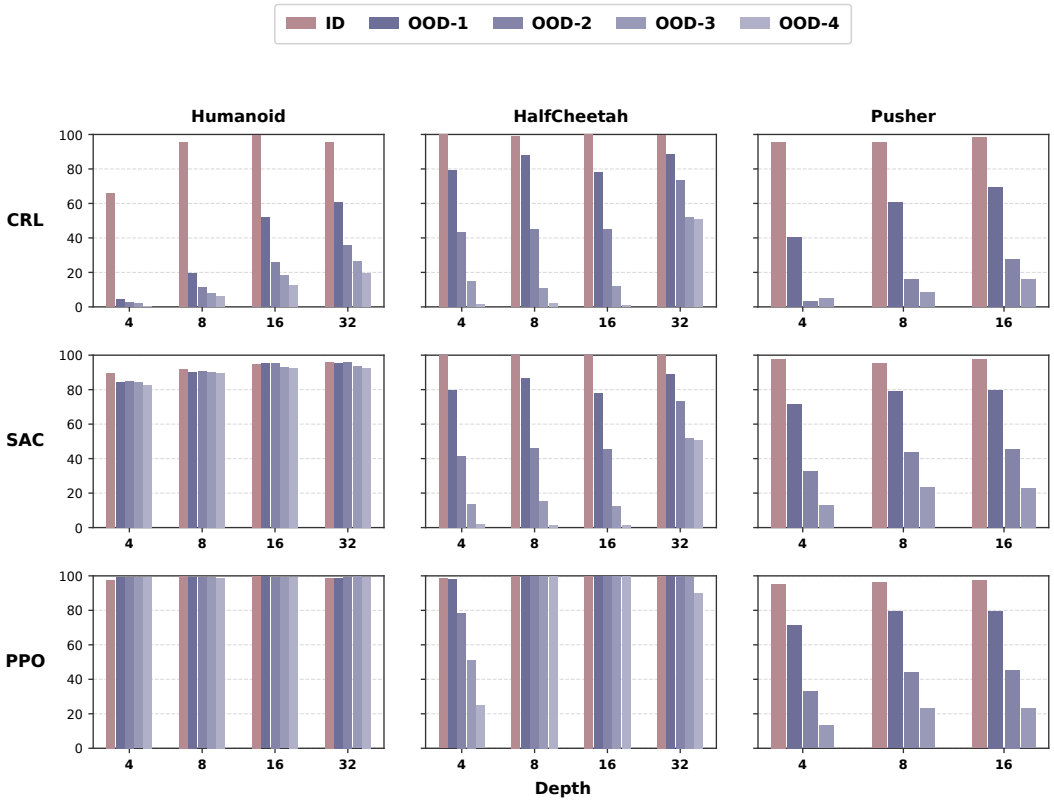


Figure 11: **Success under increasing initial-state distribution shift.** Rows correspond to CRL, SAC, and PPO, while columns correspond to Humanoid, HalfCheetah, and Pusher. Bars show success on the training initial-state distribution (ID) and progressively shifted OOD initial states.

manoid while keeping the remaining setup fixed. In Figures 12–14, rows correspond to actor depth and columns to critic depth for CRL and SAC, and value-network depth for PPO. Each panel reports success rate on the ID and progressively more distant OOD goal distributions; diagonal panels recover the jointly scaled configurations used in the main experiments.

Across the three algorithms, changing either actor or critic/value depth can affect OOD generalization, indicating that the gains from joint depth scaling cannot be attributed exclusively to either component. For CRL and SAC, actor depth is also associated with substantial changes in the generalization profile. Together with the actor's direct role in determining goal-conditioned behavior at evaluation time, this observation motivates our focus on actor representations in the mechanistic analyses of CRL and SAC.

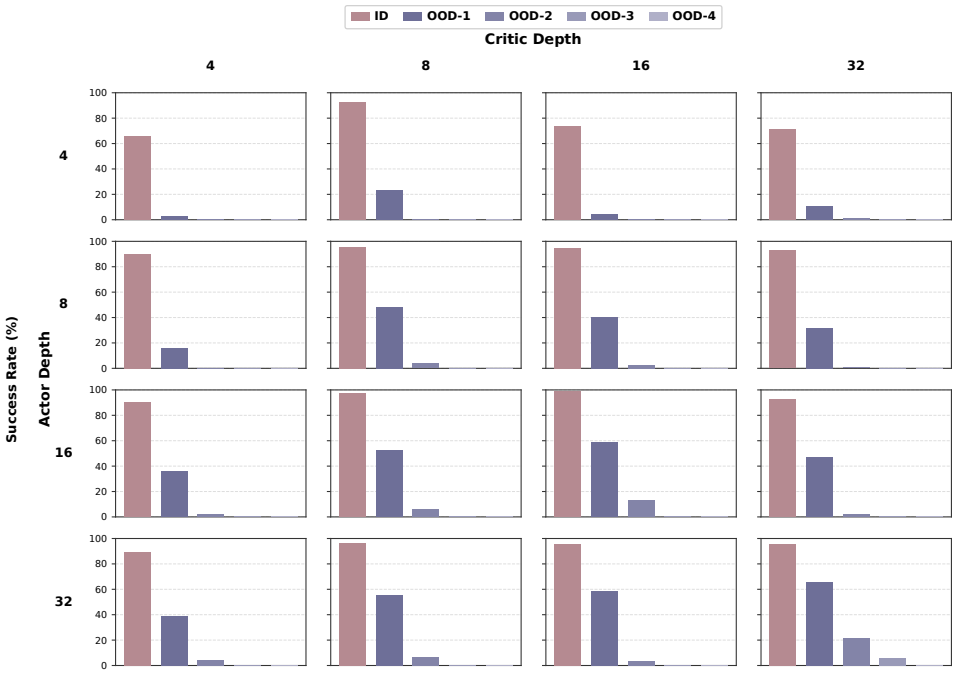


Figure 12: **Actor–critic depth ablation for CRL on Humanoid.**

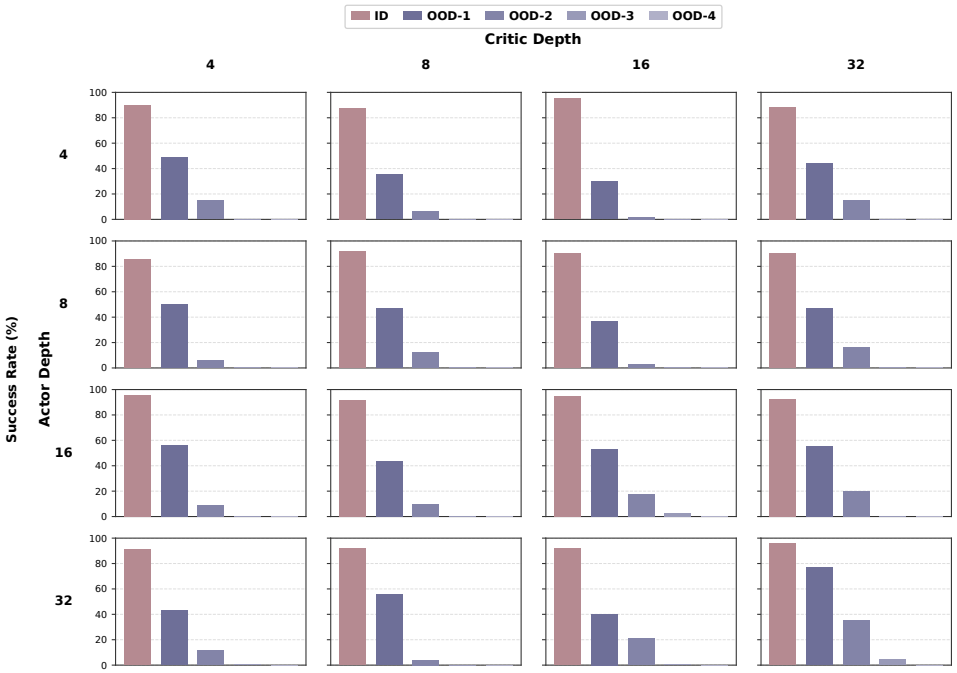


Figure 13: **Actor–critic depth ablation for SAC on Humanoid.**

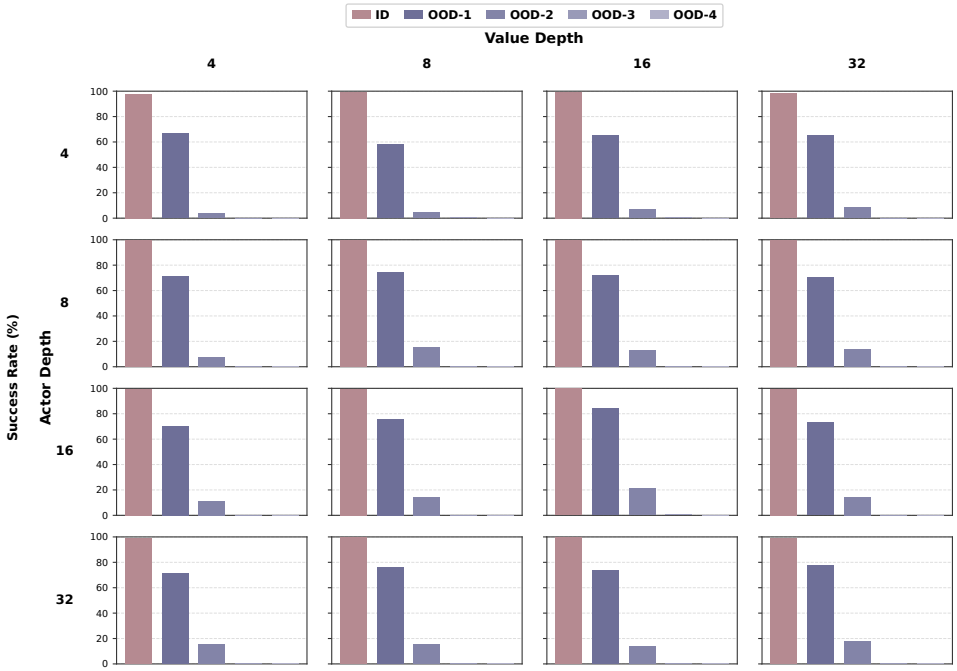


Figure 14: **Actor–value depth ablation for PPO on Humanoid.**

B Experimental Details

B.1 Environments and Goal Distributions

Our experiments build on the JaxGCRL benchmark (Bortkiewicz et al., 2025) and use continuous-control environments based on Brax (Freeman et al., 2021) and MuJoCo (Todorov et al., 2012). We consider four goal-conditioned tasks spanning locomotion, manipulation, and navigation: Humanoid, HalfCheetah, Pusher, and PointMaze, illustrated in Figure 15. For each environment, we define a training goal distribution together with progressively shifted OOD goal distributions used exclusively for evaluation. The exact construction of these distributions is described below.

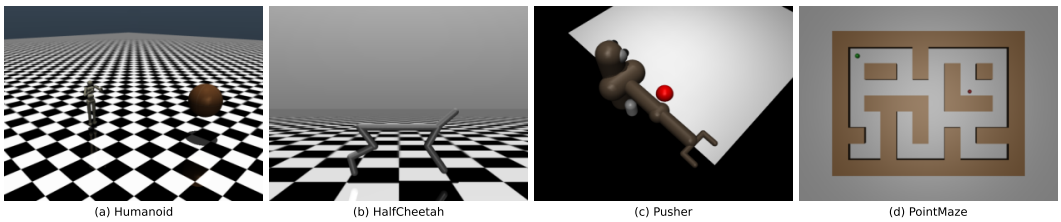


Figure 15: **Goal-conditioned environments used in our experiments.** From left to right: Humanoid and HalfCheetah for locomotion, Pusher for object manipulation, and PointMaze for navigation.

Humanoid. The agent is tasked with reaching a three-dimensional target while maintaining stable locomotion. The target height is fixed at $z = 1.25$, while its horizontal position is determined by a direction and radial distance from the agent. During training, the target direction is sampled uniformly and the radial distance is sampled from $r \in [1, 5]$. We construct four progressively more distant OOD goal distributions by sampling r from $(5, 6]$, $(6, 7]$, $(7, 8]$, and $(8, 9]$, denoted OOD-1 through OOD-4, respectively. Across these evaluations, the target-direction distribution and target height remain unchanged, isolating the effect of extrapolating to goal distances beyond the training support. An episode is considered successful if the agent reaches within 0.5 of the target.

HalfCheetah. The agent is tasked with reaching a target located along the forward horizontal axis. During training, the target position is sampled from $g_x \in [2, 8]$. We construct four progressively more distant OOD goal distributions by extending the target position beyond the training support: $g_x \in (8, 10]$, $(10, 12]$, $(12, 14]$, and $(14, 16]$, corresponding to OOD-1 through OOD-4, respectively. Thus, the OOD shift changes only the distance of the target along the forward axis. An episode is considered successful if the agent reaches within 0.5 of the target.

Pusher. The agent is tasked with manipulating an object to a desired two-dimensional goal position. Training goals are sampled uniformly from a rectangular region B_0, with x-range $[-0.55, -0.250001]$ and y-range $[-0.20, 0.20]$. To construct increasingly challenging OOD evaluations, we progressively expand this region along both spatial dimensions. The resulting outer rectangles have x-ranges $[-0.65, -0.15]$, $[-0.75, -0.05]$, and $[-0.80, 0.00]$, and corresponding y-ranges $[-0.30, 0.30]$, $[-0.40, 0.40]$, and $[-0.45, 0.45]$.

Rather than evaluating on the entire expanded rectangles, each OOD level contains only the newly introduced region. Specifically, OOD-1, OOD-2, and OOD-3 correspond to $B_1 \setminus B_0$, $B_2 \setminus B_1$, and $B_3 \setminus B_2$, respectively. Goals are sampled uniformly within each region, so successive OOD levels evaluate increasingly farther spatial extrapolation beyond the training support.

PointMaze. The agent is tasked with navigating from a fixed initial location to a two-dimensional goal within the maze. We use a fixed maze topology and reserve the free cells in the two rightmost internal columns exclusively as held-out evaluation goals. These cells remain free and traversable during training but are never sampled as training goals. All remaining eligible free cells form the training goal distribution. Thus, the OOD evaluation changes the goal support without modifying the maze topology or the initial location.

We further partition the held-out goals according to their shortest-path distance from the fixed start location. OOD-1 contains goals with shortest-path distances 12 and 13, while OOD-2 and OOD-3 contain goals at distances 14 and 15, respectively. Consequently, successive OOD levels evaluate generalization to unseen goal locations that require progressively longer navigation paths, while keeping the underlying environment unchanged.

B.2 Algorithm and Architecture Details

Shared Architecture. All methods use the residual MLP architecture described in the main text. An input projection is followed by $d/4$ residual blocks, each containing four Dense layers with LayerNorm and Swish activations. The residual core therefore contains $d \in \{4, 8, 16, 32\}$ Dense layers. Input projections and output heads are not included in the reported depth, and the actor and corresponding critic or value networks are scaled jointly.

Contrastive RL. CRL learns goal-directed behavior through a shared latent geometry rather than a conventional scalar value function. A state–action encoder $\phi(s, a)$ and a goal encoder $\psi(g)$ map their inputs into a common 64-dimensional representation space, with compatibility defined by

$$f(s, a, g) = -\left\|\phi(s, a) - \psi(g)\right\|_2 .$$

The encoders are trained with an InfoNCE objective over in-batch negatives, together with a log-sum-exp regularization term. The actor is a tanh-squashed Gaussian policy optimized to select actions whose state–action representations align with the desired goal representation, while retaining entropy regularization with an automatically learned temperature. Thus, CRL expresses goal-directed control directly through the geometry of its learned representations.

Soft Actor-Critic. We use a goal-conditioned implementation of SAC (Haarnoja et al., 2018). The actor is a tanh-squashed Gaussian policy, while the critic consists of two independent Q-functions and corresponding target networks. Each Q-function takes the goal-conditioned observation and action as input and predicts a scalar action value. The critic is trained using entropy-regularized Bellman targets constructed from the minimum of the two target Q-values. The actor is optimized against the minimum of the learned Q-functions with entropy regularization, and the temperature parameter is learned automatically. Target networks are updated using Polyak averaging.

Proximal Policy Optimization. We use a goal-conditioned implementation of PPO (Schulman et al., 2017). The actor is a tanh-squashed Gaussian policy with a trainable, state-independent log standard deviation, and the value function is parameterized by a separate network. Policy updates use the clipped PPO surrogate objective, with advantages estimated using generalized advantage estimation (GAE). Advantages are normalized over each on-policy batch, and the value function is optimized using an unclipped squared-error objective. We additionally apply observation normalization and an entropy bonus during training.

B.3 HYPERPARAMETERS

Table 1 summarizes the main training hyperparameters used in our experiments. Unless otherwise stated, these settings are shared across environments. Network depth and width are varied only in the corresponding scaling experiments.

C ADDITIONAL MECHANISTIC-ANALYSIS DETAILS

We provide additional details for the representation and causal analyses in Section 4.6. Our analysis combines linear probing with sparse feature decomposition: the probe identifies representations that encode relative goal direction, while a Sparse Autoencoder (SAE) provides a sparse basis on which we can perform targeted interventions.

Direction probe and representation selection. For an actor representation $h_t^{(\ell)} \in \mathbb{R}^{256}$ at residual block ℓ, we define the planar relative goal direction as

$$v_t = \frac{g_{xy} - s_{t,xy}}{\|g_{xy} - s_{t,xy}\|_2} = \begin{bmatrix} \cos\theta_t \\ \sin\theta_t \end{bmatrix}.$$

Following the standard linear-probing paradigm, we fit a linear map P_ℓ from the frozen actor representation to the two direction components,

$$P_\ell(h_t^{(\ell)}) = W_\ell h_t^{(\ell)} + b_\ell \approx \begin{bmatrix} \cos\theta_t \\ \sin\theta_t \end{bmatrix}.$$

The actor itself remains frozen throughout this analysis. Probes are trained on the training split and evaluated on episode-disjoint held-out activations. For the causal analysis at each network depth, we use the residual block with the lowest held-out direction-probe error.

Sparse autoencoder training. We train an independent Top-K Sparse Autoencoder on the selected actor representation for each model (Gao et al., 2025). The SAE is trained on natural activations collected from successful rollouts; the held-out episodes are not used for fitting the SAE. Let $h \in \mathbb{R}^{256}$ denote an actor activation. We first normalize the activation using the training-set mean μ and a scalar root-mean-square scale ρ,

$$\widetilde{h} = \frac{h - \mu}{\rho}.$$

The encoder maps this representation to an overcomplete dictionary of 1024 latent features:

$$a = \mathrm{ReLU}\left(W_{\mathrm{enc}}\widetilde{h} + b_{\mathrm{enc}}\right), \qquad z = \mathrm{TopK}_K(a),$$

where TopK_K retains the $K = 32$ largest positive activations and sets all remaining coordinates to zero. The decoder reconstructs the normalized actor representation as

$$\widehat{\widetilde{h}} = W_{\mathrm{dec}} z + b_{\mathrm{dec}}, \qquad \widehat{h} = \mu + \rho\,\widehat{\widetilde{h}}.$$

Because sparsity is imposed directly by the Top-K operator, the SAE is trained using only the reconstruction objective

$$\mathcal{L}_{\mathrm{SAE}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\widetilde{h}_i - \widehat{\widetilde{h}}_i\right\|_2^2.$$

We optimize this objective for 60 epochs using AdamW with learning rate 3×10^{-4}, weight decay 10^{-6}, and batch size 1024. We use at most 100,000 training activations and 20,000 held-out activations per model and selected representation.

Table 1: **Training hyperparameters.** Hyperparameters used for CRL, SAC, and PPO. A dash denotes a parameter that is not applicable to the corresponding algorithm.

Hyperparameter	CRL	SAC	PPO
Shared training setup			
total_env_steps	100M	100M	100M
num_epochs	100	100	100
num_envs	512	512	512
num_eval_envs	128	128	128
episode_length	1,000	1,000	1,000
unroll_length	62	62	62
actor_lr	3×10^{-4}	3×10^{-4}	3×10^{-4}
critic/value_lr	3×10^{-4}	3×10^{-4}	3×10^{-4}
network_width	256	256	256
network_depth	$\{4, 8, 16, 32\}$	$\{4, 8, 16, 32\}$	$\{4, 8, 16, 32\}$
Activation	Swish	Swish	Swish
Normalization	LayerNorm	LayerNorm	LayerNorm
Off-policy optimization			
batch_size	256	256	–
discount γ	0.99	0.99	–
max_replay_size	10,000	10,000	–
min_replay_size	1,000	1,000	–
sgd_batches/update	800	800	–
alpha_lr	3×10^{-4}	3×10^{-4}	–
entropy_param	0.5	1.0	–
logsumexp_penalty	0.1	–	–
Representation dimension	64	–	–
target_update τ	–	0.005	–
reward_scale	–	1.0	–
PPO optimization			
batch_size	–	–	8,192
discount γ	–	–	0.97
gae_lambda	–	–	0.90
clip_coef	–	–	0.25
vf_coef	–	–	0.5
ent_coef	–	–	10^{-3}
max_grad_norm	–	–	0.5
update_epochs	–	–	8
num_rollouts/update	–	–	8
reward_scale	–	–	0.1
Observation normalization	–	–	Yes
obs_clip	–	–	10.0
Initial action std.	–	–	0.5
Learning-rate annealing	–	–	Linear

Probe-guided sparse feature selection. The linear probe is frozen before feature selection and is never optimized using SAE-reconstructed or intervened representations. Given a sparse code z, we decode it back to actor space and measure its predicted direction by

$$\widehat{\theta}(z) = \operatorname{atan2}\left([P(D(z))]_2,\, [P(D(z))]_1\right),$$

where $D(z)$ denotes the SAE decoder including the inverse activation normalization. We define the circular probe error with respect to a target direction θ as

$$E(z;\theta) = \left|\operatorname{wrap}\left(\widehat{\theta}(z) - \theta\right)\right|.$$

At every causal timestep, feature importance is recomputed using cumulative greedy ablation. Let S_{r-1} denote the features selected during the first $r-1$ greedy steps, and let $z^{(-S)}$ denote the sparse

code obtained by setting the coordinates in S to zero. The next feature is selected as

$$j_r = \arg \max_{j \in \mathcal{A}(z) \setminus S_{r-1}} \left[E\Big(z^{-(S_{r-1} \cup \{j\})}; \theta\Big) - E\big(z^{-S_{r-1}}; \theta\big) \right],$$

where $\mathcal{A}(z)$ is the set of currently active SAE features. After selecting j_r, it remains ablated during subsequent greedy steps. Thus, feature importance is evaluated conditionally on the features already selected rather than through independent single-feature ablations.

For each physical state, we construct two equal-distance goals g_A and g_B and obtain the corresponding actor representations and sparse codes (h_A, z_A) and (h_B, z_B). The greedy procedure is applied independently to the two codes using their respective directions θ_A and θ_B. With $m = 16$, we select $m/2$ features with respect to g_A and $m/2$ with respect to g_B, and use the unique union of these features as the targeted intervention set J.

Counterfactual feature intervention. We test the functional role of the selected sparse features using a counterfactual intervention inspired by feature-level model-control approaches (Makelov et al., 2025; Marks et al., 2025). The physical environment continues to use g_A throughout the rollout. Only the selected coordinates of the g_A sparse code are replaced by their values under g_B:

$$z_j^{\text{raw}} = \begin{cases} (z_B)_j, & j \in J, \\ (z_A)_j, & j \notin J. \end{cases}$$

To avoid changing the overall latent magnitude, we preserve the global ℓ_2 norm,

$$z_{\text{patch}} = z^{\text{raw}} \frac{\|z_A\|_2}{\max(\|z^{\text{raw}}\|_2, \epsilon)}.$$

Rather than replacing the natural actor representation with an imperfect SAE reconstruction, we decode only the intervention-induced change and add it to the original hidden state:

$$h_{\text{patch}} = h_A + [D(z_{\text{patch}}) - D(z_A)].$$

The remaining actor layers then operate normally on h_{patch}.

As a control, we construct a matched random intervention with the same effective number of modified coordinates. Random features are drawn from non-selected features while preferentially matching the activation frequency of the targeted features, and the same donor-value replacement, latent-norm preservation, and decoder-difference procedure is applied. Feature selection and matching are recomputed at every timestep as the physical state evolves. We quantify steering by the angle between the final trajectory displacement and the direction toward g_B; smaller angles indicate stronger behavioral steering toward the counterfactual goal.