1000 Layer Networks for Self-Supervised RL: Scaling Depth Can Enable New Goal-Reaching Capabilities

Kevin Wang [1] **Ishaan Javali** [1] **Michał Bortkiewicz** [2] **Tomasz Trzciński** [2] **Benjamin Eysenbach** [1]

Abstract

Scaling up self-supervised learning has driven breakthroughs in language and vision, yet comparable progress has remained elusive in reinforcement learning (RL). In this paper, we introduce a self-supervised RL paradigm that unlocks substantial improvements in scalability, with network depth serving as a critical factor. Whereas most RL papers in recent years have relied on shallow architectures (around 2 – 5 layers), we demonstrate that increasing the depth up to 512 layers can significantly boost performance. Our experiments are conducted in an unsupervised goal-conditioned setting, where no demonstrations or rewards are provided, so an agent must explore (from scratch) and learn how to maximize the likelihood of reaching commanded goals. Evaluated on a diverse range of simulated locomotion and manipulation tasks, our approach ranges from doubling performance to over $50\times$ on humanoid-based tasks. Increasing the model depth not only increases success rates but also qualitatively changes the behaviors learned, with more sophisticated behaviors emerging as model capacity grows.

1. Introduction

While scaling model size has been an effective recipe in many areas of machine learning, its role and impact in reinforcement learning (RL) remains unclear. The typical model size for state-based RL tasks is between 2 to 5 layers (Raffin et al., 2021; Huang et al., 2022). In contrast, it is not uncommon to use very deep networks in other domain areas; Llama 3 (Dubey et al., 2024) and Stable Diffusion 3 (Esser et al., 2024) have hundreds of layers. In fields such as vision (Radford et al., 2021; Zhai et al., 2021; Dehghani et al., 2023) and language (Srivastava et al., 2023), models often only acquire the ability to solve certain tasks once the model is larger than a critical scale. In the RL setting, many researchers have searched for similar "emergent" phenomena (Srivastava et al., 2023), but these papers typically report only small marginal benefits and typically only on tasks where small models already achieve some degree of success (Nauman et al., 2024b; Lee et al., 2024; Farebrother et al., 2024). A key open question in RL today is whether it is possible to achieve similar "jumps" in performance by scaling RL networks.

At first glance, it makes sense why training very large RL networks should be difficult: the RL problem provides very few bits of feedback (e.g., only a sparse reward after a long sequence of observations), so the ratio of feedback to parameters is very small. The conventional wisdom (LeCun, 2016) (which many recent models reflect (Radford, 2018; Chen et al., 2020; Goyal et al., 2019)) has been that large AI systems must be trained primarily in a self-supervised fashion and that RL should only be used to finetune these models. Indeed, many of the recent breakthroughs in other fields have been primarily achieved with *self-supervised* methods, whether in computer vision (Caron et al., 2021; Radford et al., 2021; Liu et al., 2024), NLP (Srivastava et al., 2023) or multimodal learning (Zong et al., 2024). Thus, if we hope to scale reinforcement learning methods, self-supervision will likely be a key ingredient.

In this paper, we will study *building blocks* for scaling reinforcement learning. Our first step is to rethink the conventional wisdom above: "reinforcement learning" and "self-supervised learning" are not diametric learning rules, but rather can be married together into self-supervised RL (Eysenbach et al., 2021; 2022; Lee et al., 2022) systems that explore and learn policies without reference to a reward function or demonstrations. In this work, we use one of the simplest self-supervised RL algorithms, contrastive RL (CRL) (Eysenbach et al., 2022). The second step is to recognize the importance of increasing available data. We will do this by building on recent GPU-accelerated RL frame-

[0]Code available at https://anonymous.4open.science/r/scaling-crl-04B6/

[*]Equal contribution [1]Princeton University [2]Warsaw University of Technology. Correspondence to: Kevin Wang <kw6487@princeton.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

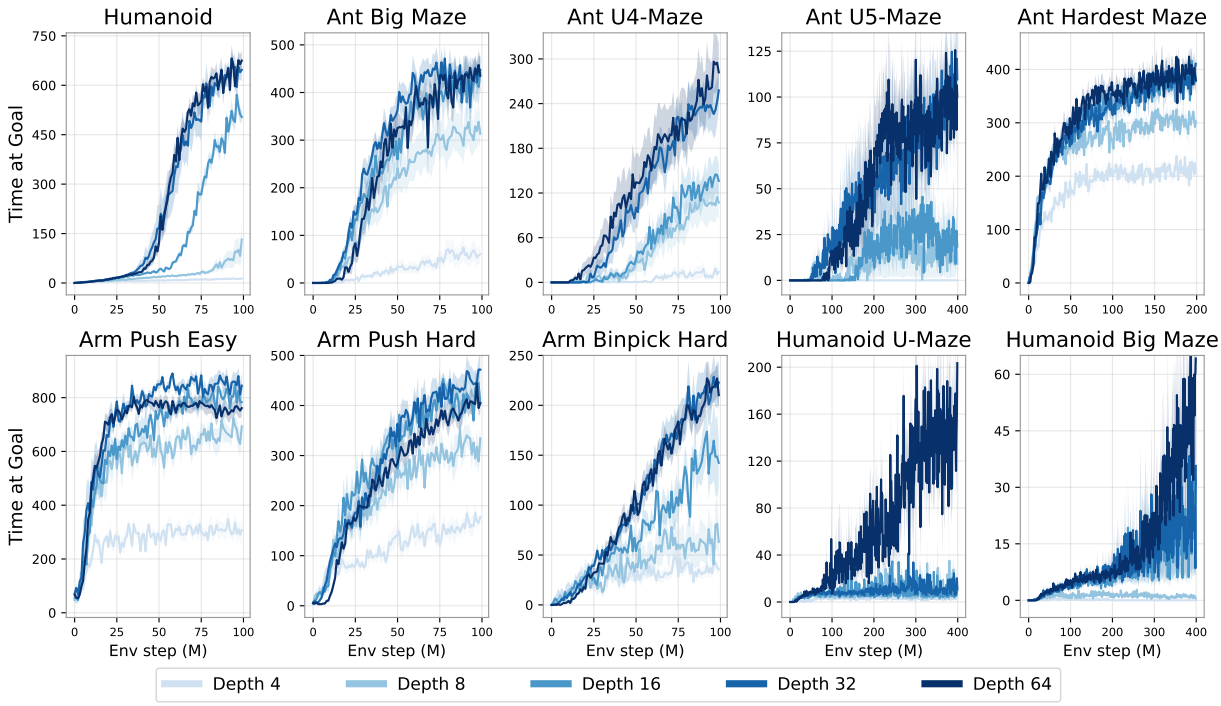

Figure 1. **Scaling network depth yields performance gains** across a suite of locomotion, navigation, and manipulation tasks, ranging from doubling performance to 50× improvements on Humanoid-based tasks. Notably, rather than scaling smoothly, performance often jumps at specific "critical" depths (e.g., 8 layers on Ant Big Maze, 64 on Humanoid U-Maze), which correspond to the emergence of qualitatively distinct learned policies (see Section 4).

works (Makoviychuk et al., 2021; Rutherford et al., 2023; Rudin et al., 2022; Bortkiewicz et al., 2024). The third component is increasing network depth, using networks that are up to 100× deeper than those typical in prior work. Stabilizing the training of such networks will require incorporating architectural techniques from prior work, including residual connections (He et al., 2015), layer normalization (Ba et al., 2016), and Swish activation (Ramachandran et al., 2018). Our experiments will also study the relative importance of batch size and network width.

The primary contribution of this work is to aggregates these "building block" scaling components into a single RL approach exhibits strong scalability. We anticipate that future research may build on this foundation by incorporating or uncovering additional "building block" elements.

Specifically, we summarize our contributions as follows:

- **Empirical Scalability:** We observe a significant performance increase, more than 20x in half of the environments, corresponding to qualitatively distinct policies that emerge with scale.
- **Scaling Depth in Network Architecture:** While many prior RL works have primarily focused on increasing network width, they often report limited or even negative returns when expanding depth (Lee et al., 2024; Nauman et al., 2024b). In contrast, our approach unlocks the ability to scale along the axis of depth, yielding performance improvements that surpass those from scaling width alone (see Section 4). As such, the experimental results of this paper largely focus on the depth dimension.
- **Extended Empirical Analyses**: In additional experiments, we show that deeper networks exhibit improved stitching capabilities, and learn more sophisticated state-action-goal relationships. We also find that deeper networks unlock the benefits of larger batch sizes. Compared to prior work, our approach benefits from both critic and actor scale.

2. Related Work

Natural Language Processing (NLP) and Computer Vision (CV) have recently converged in adopting similar architectures (i.e. transformers) and shared learning paradigms (i.e self-supervised learning), which together have enabled transformative capabilities of large scale models (Vaswani et al., 2017; Srivastava et al., 2023; Zhai et al., 2021; Dehghani et al., 2023; Wei et al., 2022). In contrast, achieving similar advancements in reinforcement learning (RL) remains

challenging. Several studies have explored the obstacles to scaling large RL models, including parameter underutilization (Obando-Ceron et al., 2024), plasticity and capacity loss (Lyle et al., 2024; 2022), data sparsity (Andrychowicz et al., 2017; LeCun, 2016), and training instabilities (Ota et al., 2021; Henderson et al., 2018; Van Hasselt et al., 2018; Nauman et al., 2024a). As a result, current efforts to scale RL models are largely restricted to specific problem domains, such as imitation learning (Tuyls et al., 2024), multi-agent games (Neumann & Gros, 2022), language-guided RL (Driess et al., 2023; Ahn et al., 2022), and discrete action spaces (Obando-Ceron et al., 2024; Schwarzer et al., 2023).

Recent approaches suggest several promising directions, including new architectural paradigms (Obando-Ceron et al., 2024), distributed training approaches (Ota et al., 2021; Espeholt et al., 2018), distributional RL (Kumar et al., 2023), and distillation (Team et al., 2023). Compared to these approaches, our method makes a simple extension to an existing self-supervised RL algorithm. The most recent works in this vein include Lee et al. (2024) and Nauman et al. (2024b), which leverage residual connections to facilitate the training of wider networks. These efforts primarily focus on network width, noting limited gains from additional depth, thus both works use architectures with only four MLP layers. In our experiments, we find that scaling width indeed improves performance (Section 4.2); however, our approach also enables scaling along depth, proving to be more powerful than width alone.

One notable effort to train deeper networks is described by Farebrother et al. (2024), who cast value-based RL into a classification problem by discretizing the TD objective into a categorical cross-entropy loss. This approach draws on the conjecture that classification-based methods can be more robust and stable and thus may exhibit better scaling properties than their regressive counterparts (Torgo & Gama, 1996; Farebrother et al., 2024). The CRL algorithm that we use effectively uses a cross entropy loss as well (Eysenbach et al., 2022). Its InfoNCE objective is a generalization of the cross-entropy loss, thereby performing RL tasks by effectively classifying whether current states and actions belong to the same or different trajectory that leads toward a goal state. In this vein, our work serves as a second piece of evidence that classification, much like cross-entropy's role in the scaling success in NLP, could be a potential "building block" in RL.

3. Preliminaries

This section introduces notation and definitions for goal-conditioned RL and contrastive RL. Our focus is on online RL, where a replay buffer stores the most recent trajectories, and the critic is trained in a self-supervised manner.

Goal-Conditioned Reinforcement Learning We define a goal-conditioned MDP as tuple $\mathcal{M}_g = (\mathcal{S}, \mathcal{A}, p_0, p, p_g, r_g, \gamma)$, where the agent interacts with the environment to reach arbitrary goals (Kaelbling, 1993; Andrychowicz et al., 2017; Blier et al., 2021). At every time step t agent observes state $s_t \in \mathcal{S}$ and performs a corresponding action $a_t \in \mathcal{A}$. The agent starts interaction in states sampled from $p_0(s_0)$, and interaction dynamics is defined by transition probability distribution $p(s_{t+1} \mid s_t, a_t)$. Goals $g \in \mathcal{G}$ are defined in a goal space $\mathcal{G}$, which is related to $\mathcal{S}$ via a mapping $f : \mathcal{S} \to \mathcal{G}$. For instance, $\mathcal{G}$ may correspond to a subset of state dimensions. The prior distribution over goals is defined by $p_g(g)$. The reward function is defined as the probability density of reaching the goal at the next time step $r_g(s_t, a_t) \triangleq (1-\gamma)p(s_{t+1} = g \mid s_t, a_t)$, with discount factor γ.

In this setting, goal-conditioned policy $\pi(a \mid s, g)$ receives both the current observation of the environment as well as a goal. We define the discounted state visitation distribution as $p_\gamma^{\pi(\cdot|\cdot,g)}(s) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^{\pi(\cdot|\cdot,g)}(s)$, where $p_t^\pi(s)$ is the probability that policy π visits s after exactly t steps, when conditioned with g. This last expression is precisely the Q-function of the policy $\pi(\cdot \mid \cdot, g)$ for the reward r_g: $Q_g^\pi(s,a) \triangleq p_\gamma^{\pi(\cdot|\cdot,g)}(g \mid s, a)$. The objective is to maximize the expected reward:

$$\max_\pi \mathbb{E}_{p_0(s_0), p_g(g), \pi(\cdot|\cdot,g)}\left[\sum_{t=0}^{\infty}\gamma^t r_g(s_t, a_t)\right]. \quad (1)$$

Contrastive Reinforcement Learning. Our experiments will use a contrastive RL algorithm (Eysenbach et al., 2022) to solve goal-conditioned problems. Contrastive RL is an actor-critic method; we will use $f_{\phi,\psi}(s,a,g)$ to denote the critic and $\pi_\theta(a \mid s, g)$ to denote the policy. The critic is parametrized with two neural networks that return state, action pair embedding $\phi(s,a)$ and goal embedding $\psi(g)$. The critic's output is defined as the l^2-norm between these embeddings: $f_{\phi,\psi}(s,a,g) = \|\phi(s,a) - \psi(g)\|_2$. The critic is trained with the InfoNCE objective (Sohn, 2016) as in previous works (Eysenbach et al., 2022; 2021; Zheng et al., 2023; 2024; Myers et al., 2024; Bortkiewicz et al., 2024). Training is conducted on batches $\mathcal{B}$, where s_i, a_i, g_i represent the state, action, and goal (future state) sampled from the same trajectory, while g_j represents a goal sampled from a different, random trajectory. The objective function is defined as:

$$\min_{\phi,\psi} \mathbb{E}_{\mathcal{B}}\left[-\sum_{i=1}^{|\mathcal{B}|}\log\left(\frac{e^{f_{\phi,\psi}(s_i,a_i,g_i)}}{\sum_{j=1}^{K} e^{f_{\phi,\psi}(s_i,a_i,g_j)}}\right)\right].$$

The policy $\pi_\theta(a \mid s, g)$ is trained to maximize the critic:

$$\max_{\pi_\theta} \mathbb{E}_{\substack{p_0(s_0), p(s_{t+1}|s_t,a_t),\\ p_g(g), \pi_\theta(a|s,g)}}\left[f_{\phi,\psi}(s,a,g)\right].$$

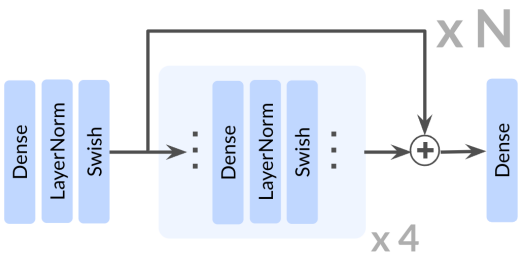

Figure 2. **Architecture.** Our approach integrates residual connections into both the actor and critic networks of the Contrastive RL (CRL) algorithm. The depth of this residual architecture is defined as the total number of Dense layers across the residual blocks, which with our residual block size of 4, equates to $4N$.

Residual connections We incorporate residual connections (He et al., 2015) into our architecture, following their successful use in RL (Farebrother et al., 2024; Lee et al., 2024; Nauman et al., 2024b). A residual block transforms a given representation $\mathbf{h}_i$ by adding a learned residual function $F_i(\mathbf{h}_i)$ to the original representation. Mathematically, this is expressed as:

$$\mathbf{h}_{i+1} = \mathbf{h}_i + F_i(\mathbf{h}_i)$$

where $\mathbf{h}_{i+1}$ is the output representation, $\mathbf{h}_i$ is the input representation, and $F_i(\mathbf{h}_i)$ is a transformation learned through the network (e.g., using one or more layers). The addition ensures that the network learns modifications to the input rather than entirely new transformations, helping to preserve useful features from earlier layers. Residual connections improve gradient propagation by introducing shortcut paths (He et al., 2016; Veit et al., 2016), enabling more effective training of deep models.

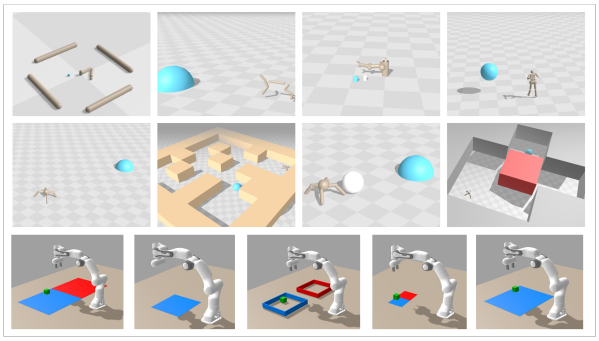

Figure 3. The scaling results of this paper are demonstrated on the JaxGCRL benchmark, showing that they replicate across a diverse range of locomotion, navigation, and manipulation tasks. These tasks are set in the goal-conditioned setting where there are no auxiliary rewards or demonstrations. Figure taken from (Bortkiewicz et al., 2024).

Table 1. Increasing network depth (depth $D = 4 \rightarrow 64$) increased performance (Figure 1). Scaling depth exhibits the greatest benefits on tasks with the largest observation dimension (*Dim*).

Task	Dim	*D* = 4	*D* = 64	Imprv.
Arm Binpick Hard	*17*	38 ±4	219 ±15	5.7×
Arm Push Easy		308 ±33	762 ±30	2.5×
Arm Push Hard		171 ±11	410 ±13	2.4×
Ant U4-Maze	*29*	11.4 ±4.1	286 ±36	25×
Ant U5-Maze		0.97 ±0.7	61 ±18	63×
Ant Big Maze		61 ±20	441 ±25	7.3×
Ant Hardest Maze		215 ±8	387 ±21	1.8×
Humanoid	*268*	12.6 ±1.3	649 ±19	52×
Humanoid U-Maze		3.2 ±1.2	159 ±33	50×
Humanoid Big Maze		0.06 ±0.04	59 ±21	1051×

4. Experiments

In this section, we detail the experiments conducted to evaluate the scalability and effectiveness of our approach across a diverse set of tasks and configurations. We describe the experimental setup before presenting the key empirical results, analyses and ablations.

4.1. Experimental Setup

Environments. All RL experiments use the JaxGCRL codebase (Bortkiewicz et al., 2024), which facilitates fast online GCRL experiments based on Brax (Freeman et al., 2021) and MJX (Todorov et al., 2012) environments. The specific environments used are a range of locomotion, navigation, and robotic manipulation tasks, as shown in Figure 3.

While no rewards are using during training, for evaluation we will measure the number of time steps (out of 1000) that the agent is near the goal. When reporting an algorithm's performance as a single number, we will compute the average score over the last five epochs of training.

Architectural Components We employ residual connections from the ResNet architecture (He et al., 2015), with each residual block consisting of four repeated units of a Dense layer, a Layer Normalization (Ba et al., 2016) layer, and Swish activation (Ramachandran et al., 2018). We apply the residual connections immediately following the final activation of the residual block, as shown in Figure 2. In this paper, we define the depth of the network as the total number of Dense layers across all residual blocks in the architecture. In all experiments, the depth refers to the configuration of the actor network and both critic encoder networks, which are scaled jointly, except for the ablation experiment in Section 4.9.

4.2. Depth Scaling in Contrastive RL

We start by studying how increasing network depth can increase performance. Both the JaxGCRL benchmark and

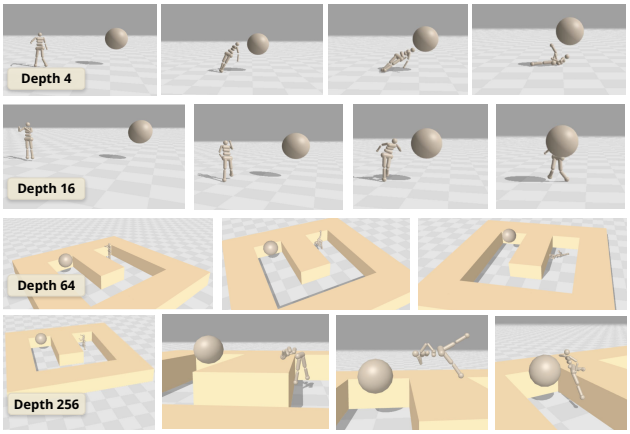

Figure 4. **Increasing depth results in new capabilities:** **Row 1**: A humanoid agent trained with network depth 4 collapses and throws itself towards the goal, as opposed to in **Row 2**, where the depth 16 agent gains the ability to walk upright. **Row 3**: At depth 64, the humanoid agent in U-Maze struggles to reach the goal and falls. **Row 4**: An impressively novel policy emerges at depth 256, as the agent exhibits an acrobatic strategy of compressing its body to vault over the maze wall.

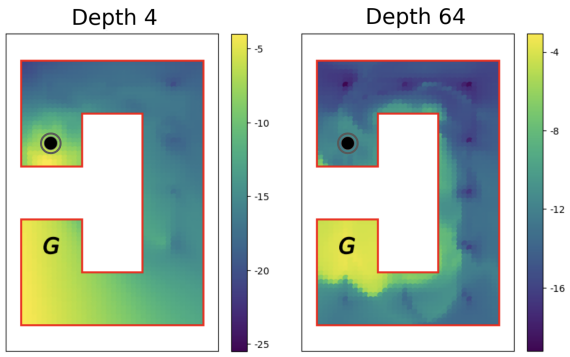

Figure 5. **Deeper Q-functions are qualitatively different.** In the U4-Maze, the start and goal positions are indicated by the ◉ and **G** symbols respectively, and the visualized Q values are computed via the L_2 distance in the learned representation space, i.e., $Q(s, a, g) = \|\phi(s, a) - \psi(g)\|_2$. The shallow depth-4 network *(left)* appears to naively rely on Euclidean proximity, exhibited by the high Q values of the semicircular gradient near the start position, despite the maze wall. In the depth-64 heatmap *(right)*, the highest Q values cluster at the goal, gradually tapering along the maze's interior boundary. These results highlight how increasing depth is important for learning value functions in goal-conditioned settings, which are characterized by long horizons and sparse rewards.

relevant prior work (Lee et al., 2024; Nauman et al., 2024b; Zheng et al., 2024) use MLPs with a depth of 4, and as such we adopt it as our baseline. In contrast, we will study networks of depth 8, 16, 32, and 64. The results in Figure 1 demonstrates that deeper networks achieve significant performance improvements across a diverse range of locomotion, navigation, and manipulation tasks. Compared to the 4-layer models typical in prior work, deeper networks achieve $2-5\times$ gains in robotic manipulation tasks, over $20\times$ gains in long-horizon maze tasks such as Ant U4-Maze and Ant U5-Maze, and over $50\times$ gains in humanoid-based tasks. The full table of performance increases up to depth 64 is provided in Table 1.

4.3. Novel Learned Policies Emerge

A closer examination of the results from the performance curves in Figure 1 reveals a notable pattern: instead of a gradual improvement in performance as depth increases, there are pronounced "jumps" that occur once a *critical depth* threshold is reached (also shown in Figure 9). The critical depths vary by environment, ranging from 8 layers (e.g. Ant Big Maze) to 64 layers in the Humanoid U-Maze task, with further "jumps" occurring even at depths of 512 layers (see the Testing Limits section, Section 4.6).

Prompted by this observation, we visualized the learned policies at various depths and found qualitatively distinct skills and behaviors exhibited. This is particularly pronounced in the humanoid-based tasks, as illustrated in Figure 4. Networks with a depth of 4 exhibit rudimentary policies where the agent either "falls" or "throws itself" toward the target. Only at a critical depth of 16 does the agent develop the ability to walk upright into the goal. In the Humanoid U-Maze environment, networks of depth 64 struggle to navigate around the intermediary wall, collapsing on the ground. Remarkably at a depth of 256, the agent learns unique behaviors on Humanoid U-Maze. These behaviors include folding forward into a leveraged position to propel itself over walls and shifting into a seated posture over the intermediary obstacle to "worm" its way toward the goal (one of these policies is illustrated in the fourth row of Figure 4). To the best of our knowledge, this is the first goal-conditioned approach to document such behaviors on the humanoid environment.

4.4. Scale Learns Better Contrastive Representations

The long-horizon setting has been a long-standing challenge in RL particularly in unsupervised goal-conditioned settings where there is no auxiliary reward feedback (Gupta et al., 2019). The family of U-Maze environments requires a global understanding of the maze layout for effective navigation. We consider a variant of the Ant UMaze environment, the U4-maze, in which the agent must initially move in the direction opposite the goal to loop around and ultimately reach it. As shown in Figure 5, we observe a qualitative difference in the behavior of the shallow network (depth 4) compared to the deep network (depth 64). The visualized

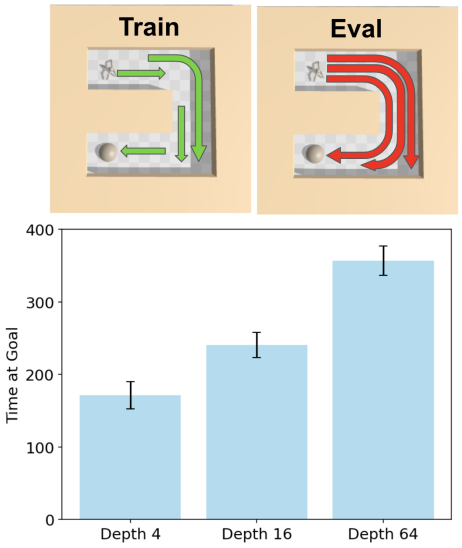

Figure 6. **Deeper networks exhibit improved generalization.** *(Top left)* We modify the training setup of the Ant U-Maze environment such that start-goal pairs are separated by ≤ 3 units. This design guarantees that no evaluation pairs *(top right)* were encountered during training, testing the ability for combinatorial generalization via "stitching." *(Bottom)* Generalization ability improves as network depth grows from 4 to 16 to 64 layers.

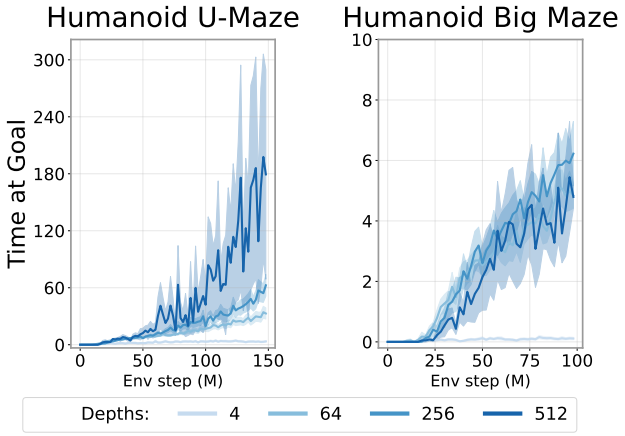

Figure 7. **Testing the limits of scale.** We extend the results from Figure 1 by scaling networks even further on the difficult Humanoid maze environments. We observe continued performance improvements with network depths of 256 and 512 layers on Humanoid U-Maze.

Q-values computed from the critic encoder representations reveal that the depth-4 network seemingly relies on Euclidean distance to the goal as a proxy for the Q value, even when a wall obstructs the direct path. In contrast, the depth-64 critic network learns richer representations, enabling it to effectively capture the topology of the maze as visualized by the trail of high Q values along the inner edge. These findings suggest that scaling network depth may be an important ingredient for reasoning over longer horizons in sparse reward settings.

4.5. Increasing Depth Results in (Some) Stitching

Another key challenge in reinforcement learning is learning policies that can generalize to tasks unseen during training. To evaluate this setting, we designed a modified version of the Ant U-Maze environment. As shown in Figure 6 (top right), the original JaxGCRL benchmark assesses the agent's performance on the three farthest goal positions located on the opposite side of the wall. However, instead of training on all possible subgoals (a superset of the evaluation state-goal pairs), we modified the setup to train on start-goal pairs that are at most 3 units apart, ensuring that none of the evaluation pairs ever appear in the training set. Figure 6 demonstrates that depth-4 networks show limited generalization, solving only the easiest goal (4 units away from the start). Depth-16 networks achieve moderate success, while depth-64 networks excel, sometimes solving the most challenging goal position. These results suggest that the increasing network depth results in some degree of "stitching", combining $\leq$3-unit pairs to navigate the 6-unit span of the U-Maze. These results demonstrate that scaling depth can improve generalization.

4.6. Testing the Limits of Scale

We next study whether further increasing depth beyond 64 layers further improves performance. We use the Humanoid maze tasks as these are both the most challenging environments in the benchmark and also seem to benefit from the deepest scaling. The results, shown in Figure 7, indicate that performance continues to substantially improve as network depth reaches 256 and 512 layers in the Humanoid U-Maze environment. While we were unable to scale beyond 512 layers due to computational constraints, we expect to see continued improvements with even greater depths, especially on the most challenging tasks.

4.7. Scaling Width vs. Depth

Past literature has shown that scaling network width can be effective (Lee et al., 2024; Nauman et al., 2024b). In Figure 8, we find that scaling width is also helpful in our experiments: wider networks (2048 and 4096) consistently outperform narrower networks across three diverse environment types (depth held constant at 4). However, depth seems to be a more effective axis for scaling: simply doubling the depth to 8 (width held constant at 256) layers outperforms the widest networks on the Humanoid environment. The advantage of depth scaling is most pronounced in the Humanoid environment (observation dimension 268), followed by Ant Big Maze (dimension 29) and Arm Push Easy (dimension 17), suggesting that the comparative benefit may

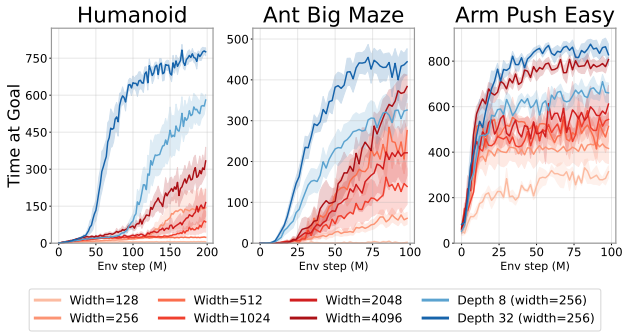

Figure 8. **Scaling network width vs. depth**. Here, we reflect findings from previous works (Lee et al., 2024; Nauman et al., 2024b) which suggest that increasing network width can enhance performance. However, in contrast to prior work, our method is able to scale depth, yielding more impactful performance gains. For instance, in the Humanoid environment, raising the width to 4096 (depth=4) fails to match the performance achieved by simply doubling the depth to 8 (width=256). This comparative advantage of scaling depth seems more pronounced as the dimensionality of the observations increases.

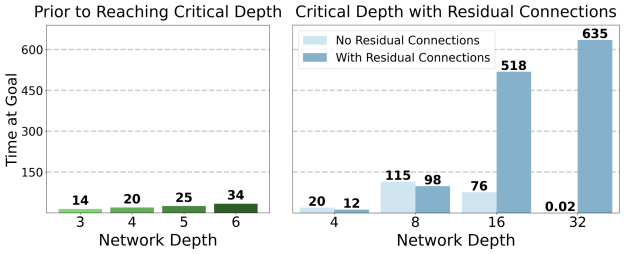

Figure 9. **Critical depth and residual connections.** Incrementally increasing the depth of shallow networks results in marginal performance gains *(left)*. However, when the network depth reaches a critical threshold, performance improves dramatically *(right)*. Residual connections are essential to stabilize the training of large networks in order to fully leverage the benefits of greater depth.

increase with higher observation dimensionality.

Note that the parameter count scales linearly with width but quadratically with depth. For comparison, a network with 4 MLP layers and 4096 hidden units has roughly 70M parameters, while one with a depth of 32 and 256 hidden units has only around 2M. Therefore, when operating under a fixed FLOP compute budget or specific memory constraints, depth scaling may be a more computationally efficient approach to improving network performance.

4.8. Deeper Networks Unlock Batch Size Scaling

Scaling batch size has played a well-established role in other areas of machine learning (Chen et al., 2022; Zhang et al., 2024). However, this approach has not translated as effectively to reinforcement learning (RL), and prior work has

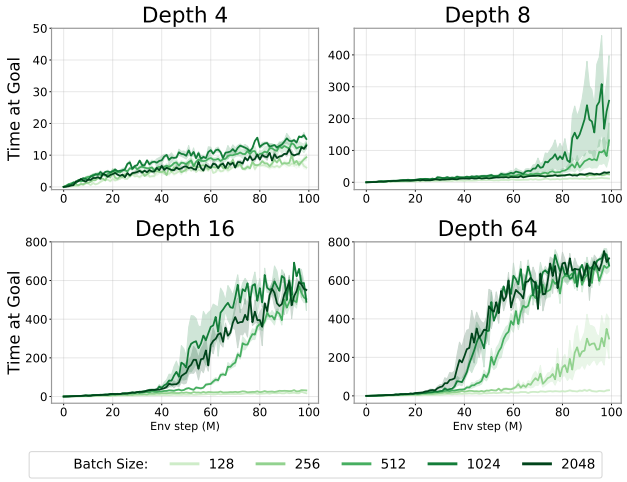

Figure 10. **Deeper networks unlock batch size scaling.** Scaling batch size has been an effective mechanism in many areas in ML (Chen et al., 2022; Zhang et al., 2024), but it has not demonstrated the same effectiveness in reinforcement learning (Obando-Ceron et al., 2023). Our findings indicate that, while scaling batch size provides only marginal benefits at the original network capacity, larger networks can effectively leverage batch size scaling to achieve further improvements.

even reported negative impacts on value-based RL (Obando-Ceron et al., 2023). Indeed, in our experiments, simply increasing the batch size for the original CRL networks yields only marginal differences in performance (Figure 10, top left).

At first glance, this might seem counterintuitive: since reinforcement learning typically involves fewer "informational bits" per training data (LeCun, 2016), one might expect higher variance in batch loss or gradients, suggesting the need for larger batch sizes to compensate. At the same time, this possibility hinges on whether the model in question can actually make use of a bigger batch size—in domains of ML where scaling has been successful, larger batch sizes usually bring the most benefit when coupled with sufficiently large models (Zhang et al., 2024; Chen et al., 2022). One hypothesis is that the small models traditionally used in RL may obscure the underlying benefits of larger batch size.

To test this hypothesis, we study the effect of increasing the batch size for networks of varying depths. As shown in Figure 10, scaling the batch size becomes effective as network depth grows. This finding offers evidence that by scaling network capacity, we may simultaneously unlock the benefits of larger batch size, potentially making it an important component in the broader pursuit of scaling self-supervised RL.

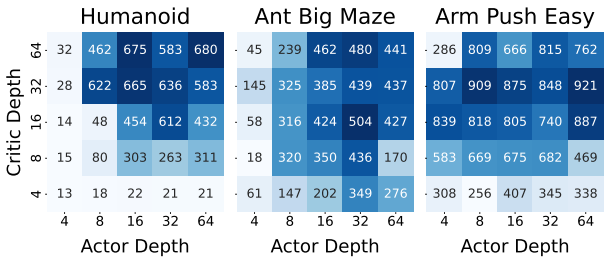

Figure 11. **Actor vs. Critic.** In the Arm Push Easy environment, scaling the critic is more effective, whereas in the Ant Big Maze, scaling the actor has a greater impact. In the Humanoid environment, scaling both actor and critic together is essential. Overall, the results suggest that scaling both actor and critic networks can complement each other to enhance performance.

Summary of key empirical findings:

- CRL is scalable to depths unattainable by other RL proprioceptive algorithms (500+ layers), perhaps due to its self-supervised nature.
- Both width and depth are key factors influencing CRL's performance, but depth achieves greater performance and better parameter-efficiency (similar performance for $50\times$ smaller models).
- We observe signs of emergent behaviors in CRL with deep neural networks, such as humanoid learning to walk and navigate a maze.
- Scale unlocks learning difficult maze topologies.
- Batch size scaling occurs in CRL for deep networks.
- CRL benefits from both the actor and critic scale.

4.9. Ablating the Actor vs. Critic Networks

To investigate the role of scaling in the actor and critic networks, Figure 11 presents the final performance for various combinations of actor and critic depths across three environments. Prior work (Nauman et al., 2024b; Lee et al., 2024) focuses on scaling the critic network, finding that scaling the actor degrades performance. In contrast, while we do find that scaling the critic is more impactful in two of the three environments (Humanoid, Arm Push Easy), our method benefits from scaling the actor network jointly, with one environment (Ant Big Maze) demonstrating actor scaling to be more impactful. These findings suggest that scaling both the actor and critic networks can play a complementary role in enhancing performance.

4.10. Does scaling depth help in the offline setting?

In preliminary experiments, we also evaluated depth scaling in the offline goal-conditioned setting using OGBench (Park et al., 2024). We found no evidence that increasing the network depth of CRL improves performance in this offline setting. To further investigate this, we conducted ablations: *(1)* scaling critic depth while holding the actor at 4 or 8 layers, and *(2)* applying cold initialization to the final layers of the critic encoders (Zheng et al., 2024). In all cases, baseline depth-4 networks had the highest success. This finding hints that the reason why scaling depth is useful in the online setting might be partially because it aids in exploration.

5. Conclusion

Arguably, much of the success of vision and language models today is due to the emergent capabilities they exhibit from scale (Srivastava et al., 2023), leading to many systems reducing the RL problem to a vision or language problem. A critical question for large AI models is: where does the data come from? Unlike supervised learning paradigms, RL methods inherently address this by jointly optimizing both the model and the data collection process through exploration. Ultimately, determining effective ways of building RL systems that demonstrate "emergent" capabilities may be important for transforming the field into one that trains its own large models. We believe that our work is a step towards these systems. By integrating key components for scaling up RL into a single approach, we show that model performance consistently improves as scale increases in complex tasks. In addition, deep models exhibit qualitatively better behaviors which might be interpreted as implicitly acquired skills necessary to reach the goal.

Limitations. The primary limitations of our results are that scaling network depth comes at the cost of compute. An important direction for future work is to study how distributed training might be used to leverage even more compute, and how techniques such as pruning and distillation might be used to decrease the computational costs.

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N., Julian, R. C., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., and Yan, M. Do as i can, not as i say: Grounding language in robotic affordances. *Conference on Robot Learning*, 2022.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. Hindsight Experience Replay. In *Neural Information Processing Systems*, volume 30, 2017.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv: 1607.06450*, 2016.

Blier, L., Tallec, C., and Ollivier, Y. Learning Successor States and Goal-Dependent Values: A Mathematical Viewpoint, January 2021.

Bortkiewicz, M., Pałucki, W., Myers, V., Dziarmaga, T., Arczewski, T., Kuciński, Ł., and Eysenbach, B. Accelerating goal-conditioned rl algorithms and research. *arXiv preprint arXiv:2408.11052*, 2024.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv: 2104.14294*, 2021.

Chen, C., Zhang, J., Xu, Y., Chen, L., Duan, J., Chen, Y., Tran, S. D., Zeng, B., and Chilimbi, T. Why do we need large batchsizes in contrastive learning? a gradient-bias perspective. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=T1dhAPdS--.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A., Caron, M., Geirhos, R., Alabdulmohsin, I. M., Jenatton, R., Beyer, L., Tschannen, M., Arnab, A., Wang, X., Riquelme, C., Minderer, M., Puigcerver, J., Evci, U., Kumar, M., van Steenkiste, S., Elsayed, G. F., Mahendran, A., Yu, F., Oliver, A., Huot, F., Bastings, J., Collier, M., Gritsenko, A., Birodkar, V., Vasconcelos, C., Tay, Y., Mensink, T., Kolesnikov, A., Paveti'c, F., Tran, D., Kipf, T., Luvci'c, M., Zhai, X., Keysers, D., Harmsen, J., and Houlsby, N. Scaling vision transformers to 22 billion parameters. *International Conference on Machine Learning*, 2023. doi: 10.48550/arXiv.2302.05442.

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., and Florence, P. R. Palm-e: An embodied multimodal language model. *International Conference on Machine Learning*, 2023. doi: 10.48550/arXiv.2303.03378.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pp. 1407–1416. PMLR, 2018.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Eysenbach, B., Salakhutdinov, R., and Levine, S. C-Learning: Learning to Achieve Goals via Recursive Classification. In *International Conference on Learning Representations*. arXiv, 2021.

Eysenbach, B., Zhang, T., Levine, S., and Salakhutdinov, R. R. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Farebrother, J., Orbay, J., Vuong, Q., Taïga, A. A., Chebotar, Y., Xiao, T., Irpan, A., Levine, S., Castro, P. S., Faust, A., Kumar, A., and Agarwal, R. Stop Regressing: Training Value Functions via Classification for Scalable Deep RL, 2024.

Freeman, C. D., Frey, E., Raichuk, A., Girgin, S., Mordatch, I., and Bachem, O. Brax – a Differentiable Physics Engine for Large Scale Rigid Body Simulation. In *NeurIPS Datasets and Benchmarks*. arXiv, 2021. URL http://arxiv.org/abs/2106.13281.

Goyal, P., Mahajan, D., Gupta, A., and Misra, I. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the ieee/cvf International Conference on computer vision*, pp. 6391–6400, 2019.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *Conference on Robot Learning*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/cvpr.2016.90.

He, K., Zhang, X., Ren, S., and Sun, J. *Identity Mappings in Deep Residual Networks*, pp. 630–645. Springer International Publishing, 2016. doi: 10.1007/978-3-319-46493-0_38. URL http://link.springer.com/content/pdf/10.1007/978-3-319-46493-0_38.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Huang, S., Dossa, R. F. J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., and Araújo, J. G. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Kaelbling, L. P. Learning to achieve goals. In *IJCAI*, volume 2, pp. 1094–8. Citeseer, 1993.

Kumar, A., Agarwal, R., Geng, X., Tucker, G., and Levine, S. Offline q-learning on diverse multi-task data both scales and generalizes. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=4-k7kUavAj.

LeCun, Y. Predictive learning. Invited talk at the 30th Conference on Neural Information Processing Systems (NIPS), December 2016. URL https://www.youtube.com/watch?v=Ount2Y4qxQo. Barcelona, Spain.

Lee, H., Hwang, D., Kim, D., Kim, H., Tai, J. J., Subramanian, K., Wurman, P. R., Choo, J., Stone, P., and Seno, T. SimBa: Simplicity Bias for Scaling Up Parameters in Deep Reinforcement Learning, 2024.

Lee, K.-H., Nachum, O., Yang, M., Lee, L., Freeman, D., Xu, W., Guadarrama, S., Fischer, I., Jang, E., Michalewski, H., and Mordatch, I. Multi-Game Decision Transformers, 2022.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Lyle, C., Rowland, M., and Dabney, W. Understanding and preventing capacity loss in reinforcement learning. *arXiv preprint arXiv:2204.09560*, 2022.

Lyle, C., Zheng, Z., Khetarpal, K., van Hasselt, H., Pascanu, R., Martens, J., and Dabney, W. Disentangling the causes of plasticity loss in neural networks. *arXiv preprint arXiv:2402.18762*, 2024.

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Myers, V., Zheng, C., Dragan, A., Levine, S., and Eysenbach, B. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. *International Conference on Machine Learning*, 2024. doi: 10.48550/arXiv.2406.17098.

Nauman, M., Bortkiewicz, M., Milos, P., Trzcinski, T., Ostaszewski, M., and Cygan, M. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=5vZzmCeTYu.

Nauman, M., Ostaszewski, M., Jankowski, K., Miłoś, P., and Cygan, M. Bigger, Regularized, Optimistic: Scaling for compute and sample-efficient continuous control, 2024b.

Neumann, O. and Gros, C. Scaling laws for a multi-agent reinforcement learning model. *arXiv preprint arXiv:2210.00849*, 2022.

Obando-Ceron, J., Bellemare, M. G., and Castro, P. S. Small batch deep reinforcement learning. *Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2310.03882. Published at NeurIPS 2023.

Obando-Ceron, J., Sokar, G., Willi, T., Lyle, C., Farebrother, J., Foerster, J. N., Dziugaite, G., Precup, D., and Castro, P. S. Mixtures of experts unlock parameter scaling for deep rl. *International Conference on Machine Learning*, 2024. doi: 10.48550/arXiv.2402.08609.

Ota, K., Jha, D. K., and Kanezaki, A. Training larger networks for deep reinforcement learning. *arXiv preprint arXiv:2102.07920*, 2021.

Park, S., Frans, K., Eysenbach, B., and Levine, S. Ogbench: Benchmarking offline goal-conditioned rl. *arXiv preprint arXiv: 2410.20092*, 2024.

Radford, A. Improving language understanding by generative pre-training. 2018.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 2021.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop*

Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hkuq2EkPf.

Rudin, N., Hoeller, D., Reist, P., and Hutter, M. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pp. 91–100. PMLR, 2022.

Rutherford, A., Ellis, B., Gallici, M., Cook, J., Lupu, A., Ingvarsson, G., Willi, T., Khan, A., de Witt, C. S., Souly, A., et al. Jaxmarl: Multi-agent rl environments and algorithms in jax. *arXiv preprint arXiv:2311.10090*, 2023.

Schwarzer, M., Obando-Ceron, J. S., Courville, A. C., Bellemare, M. G., Agarwal, R., and Castro, P. S. Bigger, better, faster: Human-level atari with human-level efficiency. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 30365–30380. PMLR, 2023. URL https://proceedings.mlr.press/v202/schwarzer23a.html.

Sohn, K. Improved Deep Metric Learning With Multi-Class N-Pair Loss Objective. In *Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A. W., Safaya, A., Tazarv, A., Xiang, A., Parrish, A., Nie, A., Hussain, A., Askell, A., Dsouza, A., Slone, A., Rahane, A., Iyer, A. S., Andreassen, A., Madotto, A., Santilli, A., Stuhlmüller, A., Dai, A. M., La, A., Lampinen, A. K., Zou, A., Jiang, A., Chen, A., Vuong, A., Gupta, A., Gottardi, A., Norelli, A., Venkatesh, A., Gholamidavoodi, A., Tabassum, A., Menezes, A., Kirubarajan, A., Mullokandov, A., Sabharwal, A., Herrick, A., Efrat, A., Erdem, A., Karakas, A., Roberts, B. R., Loe, B. S., Zoph, B., Bojanowski, B., Özyurt, B., Hedayatnia, B., Neyshabur, B., Inden, B., Stein, B., Ekmekci, B., Lin, B. Y., Howald, B., Orinion, B., Diao, C., Dour, C., Stinson, C., Argueta, C., Ramírez, C. F., Singh, C., Rathkopf, C., Meng, C., Baral, C., Wu, C., Callison-Burch, C., Waites, C., Voigt, C., Manning, C. D., Potts, C., Ramirez, C., Rivera, C. E., Siro, C., Raffel, C., Ashcraft, C., Garbacea, C., Sileo, D., Garrette, D. H., Hendrycks, D., Kilman, D., Roth, D., Freeman, D., Khashabi, D., Levy, D., González, D. M., Perszyk, D., Hernandez, D., Chen, D., Ippolito, D., Gilboa, D., Dohan, D., Drakard, D., Jurgens, D., Datta, D., Ganguli, D., Emelin, D., Kleyko, D., Yuret, D., Chen, D., Tam, D., Hupkes, D., Misra, D., Buzan, D., Mollo, D. C., Yang, D., Lee, D.-H., Schrader, D., Shutova, E., Cubuk, E. D., Segal, E., Hagerman, E., Barnes, E., Donoway, E., Pavlick, E., Rodolà, E., Lam, E., Chu, E., Tang, E., Erdem, E., Chang, E., Chi, E. A., Dyer, E., Jerzak, E., Kim, E., Manyasi, E. E., Zheltonozhskii, E., Xia, F., Siar, F., Martínez-Plumed, F., Happé, F., Chollet, F., Rong, F., Mishra, G., Winata, G. I., de Melo, G., Kruszewski, G., Parascandolo, G., Mariani, G., Wang, G. X., Jaimovitch-López, G., Betz, G., Gur-Ari, G., Galijasevic, H., Kim, H., Rashkin, H., Hajishirzi, H., Mehta, H., Bogar, H., Shevlin, H., Schütze, H., Yakura, H., Zhang, H., Wong, H. M., Ng, I., Noble, I., Jumelet, J., Geissinger, J., Kernion, J., Hilton, J., Lee, J., Fisac, J., Simon, J. B., Koppel, J., Zheng, J., Zou, J., Kocoń, J., Thompson, J., Wingfield, J., Kaplan, J., Radom, J., Sohl-Dickstein, J. N., Phang, J., Wei, J., Yosinski, J., Novikova, J., Bosscher, J., Marsh, J., Kim, J., Taal, J., Engel, J., Alabi, J. O., Xu, J., Song, J., Tang, J., Waweru, J. W., Burden, J., Miller, J., Balis, J. U., Batchelder, J., Berant, J., Frohberg, J., Rozen, J., Hernández-Orallo, J., Boudeman, J., Guerr, J., Jones, J., Tenenbaum, J. B., Rule, J., Chua, J., Kanclerz, K., Livescu, K., Krauth, K., Gopalakrishnan, K., Ignatyeva, K., Markert, K., Dhole, K. D., Gimpel, K., Omondi, K., Mathewson, K., Chiafullo, K., Shkaruta, K., Shridhar, K., McDonell, K., Richardson, K., Reynolds, L., Gao, L., Zhang, L., Dugan, L., Qin, L., Ochando, L. C., Morency, L.-P., Moschella, L., Lam, L., Noble, L., Schmidt, L., He, L., Colón, L. O., Metz, L., Senel, L. K., Bosma, M., Sap, M., ter Hoeve, M., Farooqi, M., Faruqui, M., Mazeika, M., Baturan, M., Marelli, M., Maru, M., Ramírez-Quintana, M. J., Tolkiehn, M., Giulianelli, M., Lewis, M., Potthast, M., Leavitt, M. L., Hagen, M., Schubert, M., Baitemirova, M., Arnaud, M., McElrath, M., Yee, M. A., Cohen, M., Gu, M., Ivanitskiy, M., Starritt, M., Strube, M., Swedrowski, M., Bevilacqua, M., Yasunaga, M., Kale, M., Cain, M., Xu, M., Suzgun, M., Walker, M., Tiwari, M., Bansal, M., Aminnaseri, M., Geva, M., Gheini, M., MukundVarma, T., Peng, N., Chi, N. A., Lee, N., Krakover, N. G.-A., Cameron, N., Roberts, N., Doiron, N., Martinez, N., Nangia, N., Deckers, N., Muennighoff, N., Keskar, N., Iyer, N., Constant, N., Fiedel, N., Wen, N., Zhang, O., Agha, O., Elbaghdadi, O., Levy, O., Evans, O., Casares, P. A. M., Doshi, P., Fung, P., Liang, P., Vicol, P., Alipoormolabashi, P., Liao, P., Liang, P., Chang, P., Eckersley, P., Htut, P. M., Hwang, P., Milkowski, P., Patil, P., Pezeshkpour, P., Oli, P., Mei, Q., Lyu, Q., Chen, Q., Banjade, R., Rudolph, R. E., Gabriel, R., Habacker, R., Risco, R., Milliere, R., Garg, R., Barnes, R., Saurous, R., Arakawa, R., Raymaekers, R., Frank, R., Sikand, R., Novak, R., Sitelew, R., Bras, R. L., Liu, R., Jacobs, R., Zhang, R., Salakhutdinov, R., Chi, R., Lee, R., Stovall, R., Teehan, R., Yang, R., Singh, S., Mohammad, S., Anand, S., Dillavou, S., Shleifer, S., Wiseman, S., Gruetter, S., Bowman, S. R., Schoenholz, S., Han, S., Kwatra, S., Rous, S. A., Ghazarian, S., Ghosh, S., Casey, S., Bischoff, S., Gehrmann, S., Schuster, S., Sadeghi, S., Hamdan, S. S., Zhou, S., Srivastava, S., Shi, S., Singh, S., Asaadi, S., Gu, S., Pachchigar,

S., Toshniwal, S., Upadhyay, S., Debnath, S., Shakeri, S., Thormeyer, S., Melzi, S., Reddy, S., Makini, S., Lee, S.-H., Torene, S. B., Hatwar, S., Dehaene, S., Divic, S., Ermon, S., Biderman, S., Lin, S., Prasad, S., Piantadosi, S. T., Shieber, S. M., Misherghi, S., Kiritchenko, S., Mishra, S., Linzen, T., Schuster, T., Li, T., Yu, T., Ali, T., Hashimoto, T., Wu, T.-L., Desbordes, T., Rothschild, T., Phan, T., Wang, T., Nkinyili, T., Schick, T., Kornev, T., Tunduny, T., Gerstenberg, T., Chang, T., Neeraj, T., Khot, T., Shultz, T., Shaham, U., Misra, V., Demberg, V., Nyamai, V., Raunak, V., Ramasesh, V., Prabhu, V. U., Padmakumar, V., Srikumar, V., Fedus, W., Saunders, W., Zhang, W., Vossen, W., Ren, X., Tong, X., Zhao, X., Wu, X., Shen, X., Yaghoobzadeh, Y., Lakretz, Y., Song, Y., Bahri, Y., Choi, Y., Yang, Y., Hao, Y., Chen, Y., Belinkov, Y., Hou, Y., Hou, Y., Bai, Y., Seid, Z., Zhao, Z., Wang, Z., Wang, Z. J., Wang, Z., and Wu, Z. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

Team, A. A., Bauer, J., Baumli, K., Baveja, S., Behbahani, F., Bhoopchand, A., Bradley-Schmieg, N., Chang, M., Clay, N., Collister, A., et al. Human-timescale adaptation in an open-ended task space. *arXiv preprint arXiv:2301.07608*, 2023.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A Physics Engine for Model-Based Control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, IEEE, 2012.

Torgo, L. and Gama, J. Regression by classification. In *Advances in Artificial Intelligence: 13th Brazilian Symposium on Artificial Intelligence, SBIA'96 Curitiba, Brazil, October 23–25, 1996 Proceedings 13*, pp. 51–60. Springer, 1996.

Tuyls, J., Madeka, D., Torkkola, K., Foster, D., Narasimhan, K., and Kakade, S. Scaling Laws for Imitation Learning in Single-Agent Games, 2024.

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

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *nips*, 2017.

Veit, A., Wilber, M., and Belongie, S. Residual networks behave like ensembles of relatively shallow networks. *arXiv preprint arXiv: 1605.06431*, 2016.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022. doi: 10.48550/arXiv.2206.07682.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. *Computer Vision and Pattern Recognition*, 2021. doi: 10.1109/CVPR52688.2022.01179.

Zhang, H., Morwani, D., Vyas, N., Wu, J., Zou, D., Ghai, U., Foster, D., and Kakade, S. How does critical batch size scale in pre-training? *arXiv preprint arXiv: 2410.21676*, 2024.

Zheng, C., Salakhutdinov, R., and Eysenbach, B. Contrastive Difference Predictive Coding. In *Twelfth International Conference on Learning Representations*. arXiv, October 2023.

Zheng, C., Eysenbach, B., Walke, H., Yin, P., Fang, K., Salakhutdinov, R., and Levine, S. Stabilizing Contrastive RL: Techniques for Offline Goal Reaching. In *International Conference on Learning Representations*. arXiv, 2024.

Zong, Y., Aodha, O. M., and Hospedales, T. Self-supervised multimodal learning: A survey, 2024. URL https://arxiv.org/abs/2304.01008.

A. Additional Experiments

A.1. Python environment differences

During the development, we observed different behavior of physics between the environments we used for experiments, CleanRL version of JaxGCRL, and one recommended for installation in more recent commit of JaxGCRL (Bortkiewicz et al., 2024). Upon closer examination, this difference in learning performance (Figure 12) arises from using different versions of MJX and Brax—two actively developed packages that define physics simulation for GPU devices. Importantly, we observe a similar pattern in the monotonic increase in performance with depth scaling.

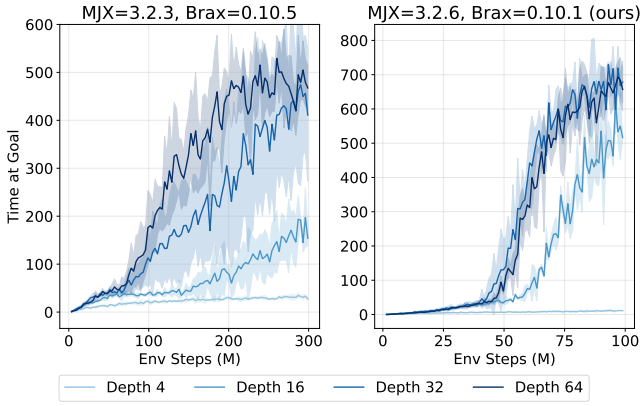

Figure 12. Scaling behavior for humanoid in two different python environments: MJX=3.2.3, Brax=0.10.5 and MJX=3.2.6, Brax=0.10.1 (ours) version of JaxGCRL. Scaling depth improves the performance significantly for both versions. However, in the environment we used, training requires fewer environment steps to reach a marginally better performance than in other Python environment.

B. Technical Details

Our experiments use JaxGCRL suite of GPU-accelerated environments and contrastive RL algorithm with hyperparameters reported in Table 2.

Table 2. Hyperparameters

Hyperparameter	Value
num_timesteps	100M-400M (varying across tasks)
update-to-data (UTD) ratio	1:40
max_replay_size	10,000
min_replay_size	1,000
episode_length	1,000
discounting	0.99
num_envs	512
batch_size	512 for the majority of experiments
action_repeat	1
policy_lr	3e-4
critic_lr	3e-4
activation function	Swish
contrastive_loss_function	InfoNCE
energy_function	L2
logsumexp_penalty	0.1
Network depth	depends on the experiment
Network width	depends on the experiment
representation dimension	64