SURVIVAL REINFORCEMENT LEARNING: TOWARD SCALABLE SELF-SUPERVISED RL

Anonymous authors
Paper under double-blind review

ABSTRACT

While self-supervised Contrastive Reinforcement Learning (CRL) has shown remarkable depth-scaling capabilities, successfully using networks over 64 layers, scaled CRL still struggles with long-horizon goal-conditioned planning due to the uniformity-tolerance dilemma inherent in contrastive losses. We introduce Survival Reinforcement Learning (SRL), an online classification-based alternative that extends the survival value learning framework by maximizing the agent's dwell time at target goals. SRL bypasses the structural constraints of CRL and mitigates the "bang-bang" control solutions inherent to survival frameworks, which often induce undesirable behavior in complex dynamical systems. Evaluated across diverse robotic benchmarks, scaled SRL matches state-of-the-art CRL on manipulation tasks and outperforms it by 2x to 8x on stable, long-horizon locomotion tasks. Our results provide strong additional evidence that classification-based methods may serve as a key primitive in the broader effort to scale reinforcement learning.

1 INTRODUCTION

In natural language processing and computer vision, the systematic scaling of neural network capacity and dataset size has driven unprecedented breakthroughs, establishing scale as a primary driver of generalist capabilities (Kaplan et al., 2020; Dosovitskiy et al., 2021). Empirical scaling laws suggest that performance improves predictably as a power law of compute, data, and parameter count (Henighan et al., 2020), with state-of-the-art models reaching trillions of parameters (Brown et al., 2020; Fedus et al., 2022; Hoffmann et al., 2022; DeepSeek-AI, 2026). A growing body of work in RL seeks to replicate this success by designing architectures and algorithms whose performance scales reliably with network capacity (Neumann & Gros, 2022; Schwarzer et al., 2023; Nauman et al., 2024b; Lee et al., 2024; Wang et al., 2025). By connecting the conventional wisdom on self-supervised learning to the self-supervised RL setting, Wang et al. (2025) provide the first compelling evidence of a genuine scaling phenomenon in self-supervised RL, demonstrating that increasing network depth in the classification-based approach Contrastive RL (CRL) yields substantial performance gains on challenging goal-reaching tasks. Yet, this scaling phenomenon remains specific to CRL as all non-contrastive goal-conditioned baselines fail to consistently benefit from increased depth. Even the contrastive approach, scaled to 64 layers, remains a bottleneck for complex, long-horizon planning. This raises a natural question: *can we design alternative self-supervised RL objectives that scale as favorably with network depth as CRL or unlock even stronger performance gains?*

The limited performance of scaled CRL on long-horizon tasks can be attributed in part to a well-known limitation. The uniformity-tolerance dilemma (Wang & Liu, 2021; Huang et al., 2023) arises because InfoNCE-style losses encourage representations to be uniformly spread across the embedding space, penalizing semantically similar states and destroying the geometric structure necessary for long-horizon planning. A promising alternative is survival learning-based RL (Tiofack et al., 2026), which frames policy evaluation as a first-hitting-time estimation problem. Rather than relying on contrastive objectives, it trains the critic by classifying whether a given state-action pair belongs to trajectories leading toward a goal, and at which time step along the trajectory the goal is reached. Its objective is grounded in maximum-likelihood estimation over observed and right-censored trajectories, a formulation that avoids the pitfalls of representation collapse associated with InfoNCE-style losses and positions survival learning-based RL as a compelling candidate for a scalable, online self-supervised RL algorithm.

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

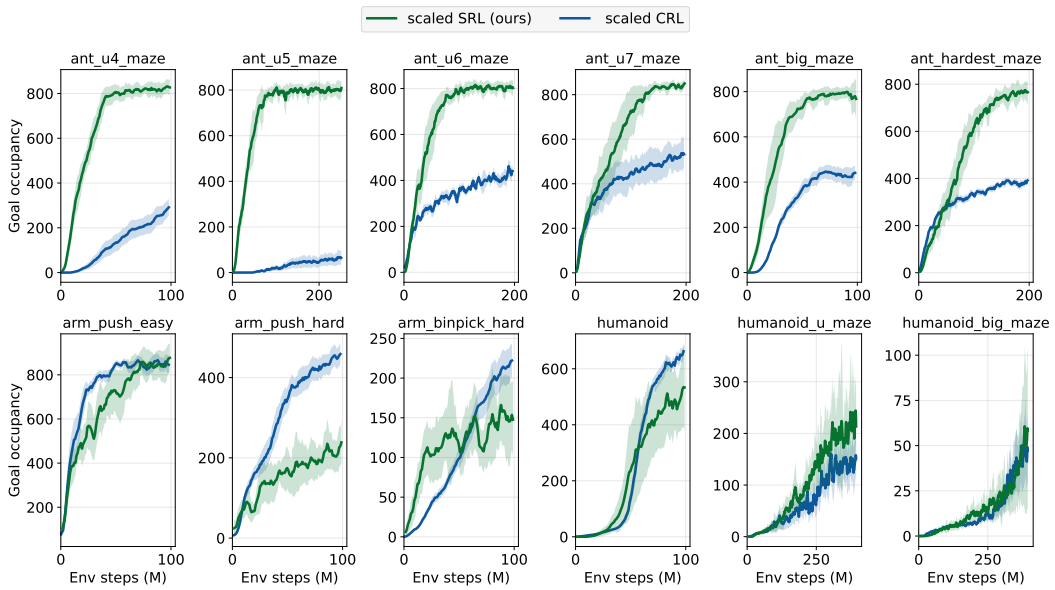


Figure 1: **Performance comparisons between SRL and CRL.** Time at goal of **SRL** (depth 32) against the state-of-the-art **CRL** (scaled up to depth 64, Wang et al. (2025)) on JaxGCRL.

In this paper, we introduce Survival Reinforcement Learning (SRL), an online self-supervised RL algorithm that extends the survival learning framework (Tiofack et al., 2026) to account for post-goal reachability. In the standard formulation, the agent minimizes a functional of the minimum time to reach the goal from a given state, a criterion that induces bang-bang control solutions (Evans, 1983; Maurer & Osmolovskii, 2004). Indeed, as in prior work (Wang et al., 2025), a more desirable policy may be one that not only drives the agent to reach the goal, but also stabilizes it there. To address this, we extend the survival learning framework with a dwell time at goal formulation that explicitly incentivizes the agent to stabilize at the goal state upon reaching it. Leveraging the network architecture proposed by prior work and GPU-accelerated simulation (Bortkiewicz et al., 2024; Wang et al., 2025), we conduct an empirical study of SRL's scaling behavior with network depth, compare it against state-of-the-art scaled CRL, and ablate the contribution of the dwell time at goal formulation. As reported Fig. 1, scaled SRL achieves competitive results on challenging goal-reaching tasks, with particularly strong performance in the AntMaze environments, providing additional evidence that classification-based objectives could be a good choice for unlocking the scaling potential of RL.

2 Related work

Designing architectures and algorithms that establish scaling laws for performance as a function of model and dataset size has been a long-standing challenge in the RL community (Neumann & Gros, 2022; Ota et al., 2024; Nauman et al., 2024b; Lee et al., 2024; Tuyls et al., 2023). Layer normalization (Nauman et al., 2024b; Ball et al., 2023; Nauman et al., 2024a; Lee et al., 2024) and residual connections (Lee et al., 2024; Nauman et al., 2024b; Wang et al., 2025; Ota et al., 2024) have emerged as the principal building blocks for stabilizing the training of wide and deep networks in RL. Wang et al. (2025) further highlights the role of activation functions, empirically showing that Swish (Ramachandran et al., 2018) is preferable to the widely adopted ReLU (Nair & Hinton, 2010) when scaling contrastive RL. We adopt the architecture of Wang et al. (2025) as our backbone to evaluate the depth-scaling behavior of SRL. Additionally, to encode the temporal structure inherent to the survival-based formulation, we follow Tiofack et al. (2026) and employ the temporal basis described in Fig. 6.

On the algorithmic side, classification-based approaches have yielded promising results toward scalable RL. Based on the conjecture that classification objectives are inherently more stable than their regression-based counterparts (Torgo & Gama, 1996), and thus better suited to deep networks, Farebrother et al. (2024) reformulate value-based RL as a classification problem by discretizing the temporal-difference target into a categorical cross-entropy loss. (Wang et al., 2025) advances a related

argument, observing that the InfoNCE objective underlying CRL is a generalization of cross-entropy loss: rather than regressing value estimates, it learns by classifying whether state-action pairs belong to trajectories leading toward a given goal. Our work follows this line of reasoning. The survival learning objective frames policy evaluation as classification, predicting not only whether state-action pairs belong to trajectories leading toward a given goal, but also the time instant along the trajectory at which the goal is reached.

Several previous works formulate self-supervised RL as a classification problem (Ghosh et al., 2021; Eysenbach et al., 2021; 2022; Yang et al., 2022; Tiofack et al., 2026; Mezghani et al., 2023), but diverge in how they treat post-goal behavior. Contrastive methods (Eysenbach et al., 2021; 2022) optimize goal-occupancy, training the agent to reach and stabilize at the goal. A complementary line of work (Ghosh et al., 2021; Yang et al., 2022; Mezghani et al., 2023; Tiofack et al., 2026) focuses on goal-reaching, leaving post-goal behavior unmodeled, an omission that can produce undesirable outcomes for complex or unstable problems such as humanoid locomotion. This pathology is acute in the survival learning formulation (Tiofack et al., 2026), where the policy minimizes a functional of the first-hitting time, whose optimal solution leads to bang-bang control (Evans, 1983; Maurer & Osmolovskii, 2004) that drives the agent toward the goal as quickly as possible, with no incentive to decelerate, stabilize, or remain upon arrival. The dwell time at goal formulation introduced in this work mitigates this limitation by extending the survival learning objective to explicitly incentivize the agent to stabilize at the goal.

3 BACKGROUND

Goal-conditioned RL. We consider a goal-conditioned Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{G}, \mathcal{P}, r_g, \varphi, p_g, \mu, \gamma)$, where $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{G}$ denote the state, action, and goal spaces, respectively. Let $\Delta(\mathcal{X})$ denote the set of probability distributions over a set $\mathcal{X}$. The transition dynamics $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ governs the environment; $\varphi : \mathcal{S} \to \mathcal{G}$ projects states into the goal space; $p_g \in \Delta(\mathcal{G})$ is the goal distribution; $\mu : \mathcal{G} \to \Delta(\mathcal{S})$ is the goal-conditioned initial-state distribution; and $\gamma \in [0, 1)$ is the discount factor. At the start of each episode, a goal $g \sim p_g$ is sampled, and the agent is initialized at $s_0 \sim \mu(\cdot \mid g)$. The agent then follows a goal-conditioned policy $\pi : \mathcal{S} \times \mathcal{G} \to \Delta(\mathcal{A})$. Following prior work (Andrychowicz et al., 2017; Chane-Sane et al., 2021; Blier et al., 2021), we adopt a sparse reward formulation by setting $r_g(s_t, a_t) = -\mathbf{1}_{\{\varphi(s_{t+1}) \notin \mathcal{B}_\epsilon(g)\}}$ with $\mathcal{B}_\epsilon(g) = \{s \in \mathcal{S} \mid \|\varphi(s) - g\| \leq \epsilon\}$. For a given goal-conditioned policy π, the corresponding state-action value function is defined as:

$$Q_g^\pi(s, a) = \mathbb{E}_{\tau \sim \rho^\pi(\cdot|g)} \left[\sum_{t=0}^{\infty} \gamma^t r_g(s_t, a_t) |_{a_0 = a}^{s_0 = s} \right], \tag{1}$$

where $\rho^\pi(\tau|g) = \mu(s_0 \mid g) \prod_{t=0}^{\infty} \pi(a_t \mid s_t, g) \mathcal{P}(s_{t+1} \mid s_t, a_t)$ denotes the likelihood of the rollout trajectory $\tau = (s_0, a_0, s_1, a_1, \dots)$ under π.

Metrics. We evaluate a goal-conditioned policy π over fixed horizon H using two metrics:

$$\mathrm{SR}(\pi) = \mathbb{E}_{\substack{g \sim p_g \\ \tau \sim \rho^\pi(\cdot|g)}} \left[\mathbf{1}_{\{\exists\, t \leq H : \varphi(s_t) \in \mathcal{B}_\epsilon(g)\}} \right] \text{ and } \mathrm{GO}(\pi) = \mathbb{E}_{\substack{g \sim p_g \\ \tau \sim \rho^\pi(\cdot|g)}} \left[\sum_{t=1}^{H} \mathbf{1}_{\{\varphi(s_t) \in \mathcal{B}_\epsilon(g)\}} \right]. \tag{2}$$

The success rate $\mathrm{SR}(\pi)$ measures the probability that the agent reaches the goal region at least once during an episode, capturing task completion regardless of subsequent behavior. Goal-occupancy $\mathrm{GO}(\pi)$ measures the expected number of timesteps spent within the goal region, capturing the agent's ability to reach *and sustain* goal satisfaction. The two metrics can diverge significantly: an agent may achieve a high success rate by transiently passing through the goal region while accumulating negligible goal occupancy due to overshooting or instability upon arrival. Depending on the algorithmic design, the learned policy may explicitly target goal-occupancy or success rate.

Policy evaluation via survival analysis. In the GCRL setting, Tiofack et al. (2026) reformulate policy evaluation through survival analysis framework (Kaplan & Meier, 1958; Katzman et al., 2018). They model the first-hitting time of the goal region, a random variable defined as the minimum number of steps required to transition from state s to goal g when taking action a and subsequently following policy π:

$$T_g^\pi(s, a) = \inf \Big\{ t \geq 0 : \varphi(s_{t+1}) \in \mathcal{B}_\epsilon(g) \mid {}_{a_t \sim \pi(\cdot|s_t, g),\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}^{s_0 = s,\ a_0 = a} \Big\}, \tag{3}$$

with the convention that $T_g^\pi(s,a) = \infty$ if the goal is never reached. Under the assumption that each episode terminates upon reaching the goal region – equivalently, that every goal state is absorbing – the goal-conditioned action-value function admits the following closed-form relationship with $T_g^\pi(s,a)$:

$$Q_g^\pi(s,a) = -\sum_{t=0}^{\infty} \gamma^t \Pr\big(T_g^\pi(s,a) > t\big). \tag{4}$$

This identity, obtained by adapting Proposition 4.1 of Tiofack et al. (2026) to the action-value function, shows that critic learning reduces to estimating the survival probability $S^\pi(t \mid s,a,g) \triangleq \Pr(T_g^\pi(s,a) > t)$. This is estimated via a parametrized hazard model

$$h_\theta^\pi(t \mid s,a,g) \triangleq \Pr\big(T_g^\pi(s,a) = t \mid T_g^\pi(s,a) \geq t\big), \tag{5}$$

which gives the probability that the goal is reached at step t, given that it has not been reached before. The survival probability is then recovered through the product formula $S^\pi(t \mid s,a,g) = \prod_{k=0}^{t}\big(1 - h_\theta^\pi(k \mid s,a,g)\big)$, and the hazard model is trained by maximizing the right-censored likelihood over collected trajectories:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(s,a,g,t_h,c,\delta)\sim\mathcal{D}}\big[\delta\,\ell_{t_h}(\theta) + (1-\delta)\,\ell_c(\theta)\big] \text{ with } \ell_c(\theta) = \sum_{j=0}^{c} \log\big(1 - h_\theta^\pi(j \mid s,a,g)\big)$$
$$\text{and } \ell_{t_h}(\theta) = \log h_\theta^\pi(t_h \mid s,a,g) + \sum_{j=0}^{t_h-1} \log\big(1 - h_\theta^\pi(j \mid s,a,g)\big), \tag{6}$$

where $\delta \in \{0,1\}$ indicates whether the goal is reached within c steps, t_h denotes the hitting time when $\delta = 1$. For a complete derivation, see Tiofack et al. (2026). Note that this formulation does not explicitly model post-goal achievement and primarily focuses on success rate.

4 Survival Reinforcement Learning

We introduce Survival Reinforcement Learning (SRL) by first extending the survival learning framework with a dwell-time-at-goal formulation that accounts for post-goal-reaching behavior, and then building an online goal-conditioned actor-critic algorithm on top of this extended critic objective.

4.1 Dwell-time-at-goal formulation

By modeling only the first-hitting time, the original survival learning formulation is indifferent to post-achievement behavior. As is well understood in optimal control theory, minimizing a first-hitting-time functional typically yields bang-bang solutions (Evans, 1983; Maurer & Osmolovskii, 2004; Bellman et al., 1956). The agent reaches the goal as rapidly as possible, with no incentive to decelerate, stabilize, or remain. In practice, as illustrated in Section 5, this manifests as a humanoid that lunges toward the target at high velocity and falls upon arrival, or as a manipulator that overshoots the target configuration. Post-goal stability is therefore necessary for reliable performance on locomotion and manipulation tasks. We address this by reformulating the hitting-time criterion to require the agent to match a short sequence of consecutive goals.

Extended hitting time. Given a state trajectory $(s_0, s_1, \ldots)$, we define the *k-length goal sequence* sampled at offset $u \geq 0$ as:

$$\mathbf{g}^{(k)} = \big(\varphi(s_u), \varphi(s_{u+1}), \ldots, \varphi(s_{u+k-1})\big) \in \mathcal{G}^k. \tag{7}$$

The first-hitting time is then extended to require k consecutive goal matches:

$$T_{\mathbf{g}^{(k)}}^\pi(s,a) = \inf\left\{ t \geq 0 : \varphi(s_{t+i+1}) \in \mathcal{B}_\epsilon(g_i),\ \forall i \in \{0,\ldots,k-1\} \,\middle|\, \begin{smallmatrix} s_0=s,\ a_0=a \\ a_t\sim\pi(\cdot|s_t,g_0) \\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t) \end{smallmatrix} \right\}, \tag{8}$$

where g_i denotes the i-th component of $\mathbf{g}^{(k)}$. The original formulation of Eq. 3 is recovered as the special case $k = 1$. When $g_i = g$ for all $i \in \{0,\ldots,k-1\}$, the event requires the agent to remain within $\mathcal{B}_\epsilon(g)$ for k consecutive timesteps. An agent that enters the goal region and immediately exits does not trigger the survival event and is penalized accordingly.

Extended value function. This formulation inherits the full survival analysis structure of Tiofack et al. (2026). Assuming episode termination upon $\mathbf{g}^{(k)}$ achievement, we define the extended action-value function evaluating the ability of policy π to achieve $\mathbf{g}^{(k)}$ as:

$$Q^{\pi}_{\mathbf{g}^{(k)}}(s,a) = -\sum_{t=0}^{\infty} \gamma^t \Pr\big(T^{\pi}_{\mathbf{g}^{(k)}}(s,a) > t\big), \tag{9}$$

which extends the identity in Eq. 4 to the dwell time at goal setting. Following (Tiofack et al., 2026), we estimate $Q^{\pi}_{\mathbf{g}^{(k)}}(s,a)$ by training a parametrized hazard model

$$h^{\pi}_{\theta}(t \mid s,a,\mathbf{g}^{(k)}) \triangleq \Pr\big(T^{\pi}_{\mathbf{g}^{(k)}}(s,a) = t \mid T^{\pi}_{\mathbf{g}^{(k)}}(s,a) \geq t\big) \tag{10}$$

via the right-censored negative log-likelihood objective of:

$$\mathcal{L}_{\text{hazard}}(\theta) = -\mathbb{E}_{(s,a,\mathbf{g}^{(k)},t_h,c,\delta)\sim\mathcal{D}}\left[\delta\, \ell_{t_h}(\theta) + (1-\delta)\, \ell_c(\theta)\right], \tag{11}$$

where the event log-likelihood ℓ_{t_h} and censored log-likelihood ℓ_c are given by:

$$\ell_{t_h}(\theta) = \log h^{\pi}_{\theta}(t_h \mid s,a,\mathbf{g}^{(k)}) + \sum_{j=0}^{t_h-1} \log\big(1 - h^{\pi}_{\theta}(j \mid s,a,\mathbf{g}^{(k)})\big),\ \ \ell_c(\theta) = \sum_{j=0}^{c} \log\big(1 - h^{\pi}_{\theta}(j \mid s,a,\mathbf{g}^{(k)})\big) \tag{12}$$

The survival probability $\Pr(T^{\pi}_{\mathbf{g}^{(k)}}(s,a) > t) = \prod_{j=0}^{t}\big(1 - h^{\pi}_{\theta}(j \mid s,a,\mathbf{g}^{(k)})\big)$ is then substituted into Eq. 9 to recover the extended value function.

4.2 Policy update

This extension requires no new loss function or architectural modification beyond the original survival learning framework. The actor $\pi_{\psi}(\cdot \mid s,g)$ is conditioned on a single point goal $g \in \mathcal{G}$, preserving the standard goal-conditioned RL interface. During actor updates, we instantiate the goal sequence as $\mathbf{g}^{(k)} = (g,g,\ldots,g)$, which aligns the critic signal with the dwell time at goal objective and incentivizes the agent to reach and remain at g. The stochastic Gaussian policy is trained by minimizing the maximum-entropy objective:

Algorithm 1 Survival Reinforcement Learning

```
Initialize critic θ, actor ψ, replay buffer D
while not converged do
    while Data collection do
        Sample goal g ~ p_g
        Policy rollout τ ~ ρ^{π_ψ}(· | g)
        Construct (s,a,g^(k),t_h,c,δ), store in D
    end while
    for each gradient step do
        Sample (s, a, g^(k), t_h, c, δ) from D
        θ ← θ − λ_θ ∇_θ L_hazard(θ)
        ψ ← ψ − λ_ψ ∇_ψ L_actor(ψ)
    end for
end while
```

$$\mathcal{L}_{\text{actor}}(\psi) = \mathbb{E}_{(s,g)\sim\mathcal{D},\ a\sim\pi_{\psi}(\cdot|s,g)}\Big[-Q^{\pi}_{(g,\ldots,g)}(s,a) + \alpha \log \pi_{\psi}(a \mid s,g)\Big], \tag{13}$$

where $\alpha > 0$ is the entropy temperature (Haarnoja et al., 2018). Although SRL is formally on-policy since the hazard is estimated under the policy that collected the data, we follow standard practice (Haarnoja et al., 2018; Fujimoto et al., 2018; Eysenbach et al., 2022) and perform multiple gradient steps per transition, treating the replay buffer as an off-policy dataset. The pseudocode is provided in Algorithm 1, and we refer to the Appendix for additional details.

4.3 Goal-occupancy bound

The following proposition justifies our dwell-time formulation: goal-occupancy is lower-bounded by the remaining horizon upon arrival, discounted by the probability of premature exit. Sec. 5.5 illustrates that, in practice, minimizing the hitting time $T^{\pi}_{\mathbf{g}^{(k)}}$ decreases the exit probability p^{π}_{k}.

Proposition 1 (Goal-occupancy lower bound)**.** *For a given policy π and fixed episode length H, let*

$$p^{\pi}_{k} = \max_{t\in\{1,\ldots,H\}} \mathbb{E}_{\substack{g\sim p_g,\ s_0\sim\mu(\cdot|g),\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^{\pi}(\cdot|g,s_0,a_0)}}\Big[\mathbf{1}_{\{\varphi(s_t)\notin\mathcal{B}_{\epsilon}(g)\}} \mid T^{\pi}_{\mathbf{g}^{(k)}}(s_0,a_0) < t\Big],$$

where $\mathbf{g}^{(k)} = (g,g,\ldots,g)$ and $T^{\pi}_{\mathbf{g}^{(k)}}(s_0,a_0)$ is the extended hitting time defined in Eq. 8. Then:

$$\mathbb{E}_{\substack{g\sim p_g\\ \tau\sim\rho^{\pi}(\cdot|g)}}\left[\sum_{t=1}^{H} \mathbf{1}_{\{\varphi(s_t)\in\mathcal{B}_{\epsilon}(g)\}}\right] \geq (1-p^{\pi}_{k})\left(H - \mathbb{E}_{\substack{g\sim p_g,\ s_0\sim\mu(\cdot|s)\\ a_0\sim\pi(\cdot|s_0,g)}}\Big[T^{\pi}_{\mathbf{g}^{(k)}}(s_0,a_0) \wedge H\Big]\right). \tag{14}$$

Proof. We refer readers to Appendix A.1.

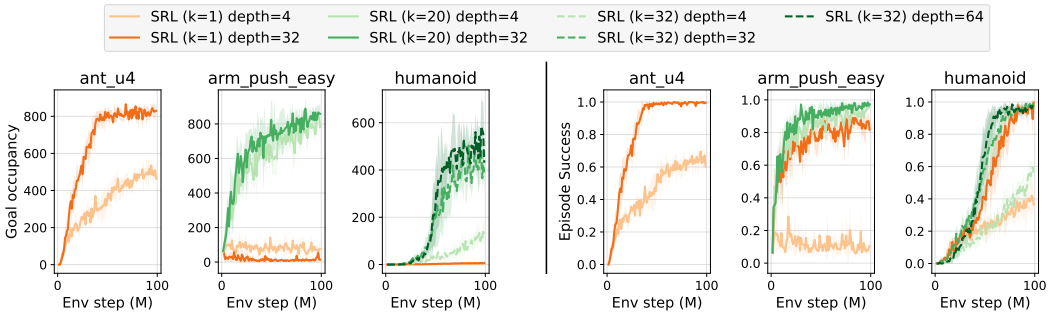


Figure 2: **SRL on two metrics** for varying network depth and goal sequence k. The dwell time at goal ($k > 1$) is essential for `Arm Push Easy` and `Humanoid` to exhibit scaling behavior in goal occupancy.

5 Experiments

We structure our evaluation around three main questions: (i) How important is the dwell time at goal formulation?; (ii) How does SRL scale with network depth?; (iii) How tight is the occupancy lower bound of Proposition 1? We use CRL (Eysenbach et al., 2022) as our baseline, as it was shown to be the principal goal-conditioned RL algorithm to consistently scale with network depth (Wang et al., 2025). SRL implementation will be released upon paper acceptance.

5.1 Experimental Setup

We implement SRL based on Wang et al. (2025), which adapts JaxGCRL (Bortkiewicz et al., 2024) to enable fast online GCRL experiments via Brax (Freeman et al., 2021). We evaluate on a diverse set of locomotion, navigation, and robotic manipulation tasks (see Fig. 8) using the two metrics defined in Eq. 2 and fix the episode length to $H = 1000$.

Following prior work (Wang et al., 2025; Lee et al., 2024; Nauman et al., 2024b), all networks use residual blocks (He et al., 2016) with four Dense layers each, Layer Normalization (Ba et al., 2016), Swish activation (Ramachandran et al., 2018), and temporal basis (Tiofack et al., 2026), see Appendix A.2. Depth is the total number of dense layers across all residual blocks. The actor and both critic networks have the same depth and are scaled jointly. We use the piecewise-constant survival parameterization.

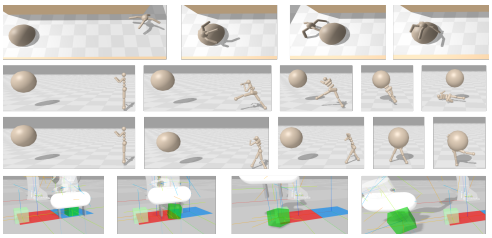


Figure 3: **Behavioral visualization of SRL.** *First row:* with $k = 1$, the `Ant` overshoots the target and must reverse direction to return. *Second row:* with $k = 1$, the `Humanoid` lunges toward the target at full speed and collapses upon arrival. *Third row:* with the dwell-goal mechanism ($k = 32$), the `Humanoid` decelerates as it approaches the target to maintain dynamic balance, achieving sustained goal occupancy. *Fourth row:* The arm strikes the cube with excessive force, causing it to overshoot the target position (Enlarged views in Fig. 9).

5.2 Why the dwell-time-at-goal formulation?

Stable locomotion (`Ant U4 Maze`). On this task, the standard survival learning formulation ($k = 1$) scales well with network depth and saturates on both metrics, as shown in the first and fourth columns of Fig. 2. With sufficient capacity (32 layers), the policy learns to move the ant as quickly as possible toward the goal, yielding a bang-bang control solution, as studied by optimal control theory (Maurer & Osmolovskii, 2004; Evans, 1983). The higher arrival speed causes a slight overshoot, followed by a correction, resulting in a 100% success rate and a goal occupancy above $800/1000$. This behavior is shown in the first row of Fig. 3.

Challenging locomotion (`Humanoid`). When $k = 1$, the success rate scales well with depth, rising from below 40% at depth 4 to 100% at depth 32. Goal-occupancy remains negligible at all depths, as shown in the third and sixth columns of Fig. 2. Again, the policy converges to a bang-bang solution: the `Humanoid` lunges toward the target at full speed with no incentive to decelerate or stabilize, and falls upon arrival. This behavior is illustrated in the second row of Fig. 3. The dwell time at goal formulation corrects this by incentivizing the agent to decelerate when approaching the goal,

enabling it to stabilize upon arrival, as shown in the third row of Fig. 3. With $k = 32$, both metrics scale consistently with depth, as depicted in the third and sixth columns of Fig. 2.

Manipulation (`Arm Push Easy`). A similar pattern emerges for manipulation, as shown in the second and fifth columns of Fig. 2. With $k = 1$, scaling the depth from 4 to 32 layers improves the success rate from roughly 40% to 90%, but goal occupancy remains near zero throughout. The standard objective incentivizes the arm to strike the cube with excessive force, satisfying the transient success condition while failing to achieve sustained occupancy. Setting $k = 20$ resolves this: the dwell time at goal formulation seamlessly saturates both metrics.

5.3 Scaling depth in SRL

Fig. 1 compares scaled CRL and scaled SRL across 12 environments, with SRL network depth scaled up to 32 layers. Despite using half the network depth of scaled CRL, SRL outperforms it by a substantial margin in 7 of 12 environments, achieving an average goal occupancy of $800/1000$ in most of those environments and remaining competitive in the rest. For all environments except `Ant U6 Maze` and `Ant U7 Maze`, the scaled CRL results were provided by Wang et al. (2025). Fig. 7 shows the scaling performance of SRL across JaxGCRL for varying network depths.

Ant navigation. SRL excels across all AntMaze environments, independently of maze topology and planning horizon. At depth 32, SRL converges much faster and achieves goal-occupancy values two to four times higher than the contrastive baseline. SRL rapidly saturates at an average goal occupancy of $800/1000$ across all maze topologies from `Ant U4 Maze` to `Ant Hardest Maze`. Given a maximum episode horizon of $H = 1000$ steps, this represents near-optimal behavior: the agent requires only approximately 200 steps to traverse any maze and reach the target, after which it successfully occupies the goal region for the remainder of the episode.

Humanoid. On the challenging `Humanoid` environment, SRL begins to succeed at depth 8 and achieves a goal-occupancy of $600/1000$ when scaled to depth 32. The resulting policies stabilize at a given target position, as illustrated in the third row of Fig. 3. On `Humanoid U Maze` and `Humanoid Big Maze`, SRL exhibits scaling performance comparable to CRL (Wang et al., 2025).

Manipulation. On `Arm Push Easy`, SRL reaches near-optimal goal-occupancy ($800/1000$) at depth 4 and remains consistent across all depths, whereas CRL requires depth 16 to achieve comparable performance. On `Arm Binpick Hard`, performance jumps from depth 4 to depth 8. The `Arm Push Hard` task remains a significant challenge for SRL, due to exploration bottlenecks: since we use vanilla SAC, task-agnostic entropy maximization struggles to discover the initial cube contact (a known failure mode on this environment, leading to trivial HER targets (Bortkiewicz et al., 2025, Appendix A.5)).

5.4 SRL on extreme long-horizon tasks

The results of Sec. 5.3 demonstrate that scaled SRL achieves near-optimal behavior on every `AntMaze` environment, from `U4` to `Hardest`. To probe the limits of SRL on long horizons beyond prior settings (Wang et al., 2025), we use the geometric flexibility of the maze suite. Concretely, we take the layout requiring the longest planning horizon, `Ant Big Maze`, and scale the maze's physical size by increasing `maze_size_scaling` from its default value of 4.0 to 8.0 and 16.0, effectively doubling and quadrupling the maze dimensions. We scale data collection to **one billion** transitions and retrain both scaled CRL and scaled SRL (depth=32). This modification has two compounding effects on task difficulty. First, the minimum number of steps required to traverse the maze grows linearly with the scaling factor, since path lengths increase proportionally, at `maze_size_scaling` $= 16.0$, the agent must navigate an environment four times larger than the standard benchmark. Second, the goal

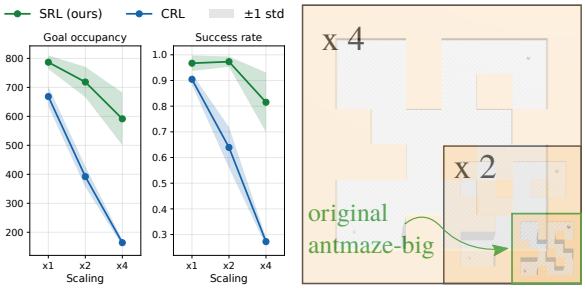
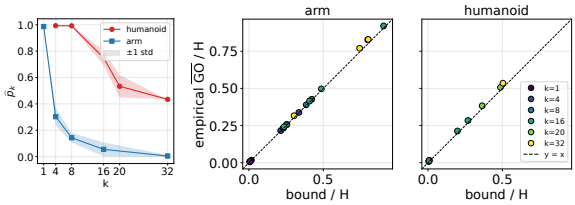


Figure 4: **SRL on extreme long-horizon tasks.** *(left)* As the planning horizon increases beyond standard longest horizon, scaled SRL maintains consistent performance against CRL. *(right)* Visual comparison of the three maze scales.

region $\mathcal{B}_\epsilon(g)$ remains fixed in absolute size, making it proportionally smaller and harder to reach precisely as the maze grows. Together, these effects place the task in the extreme long-horizon regime, exacerbating the curse of horizon (Park et al., 2025; Myers et al., 2025). As shown in Fig. 4, the CRL baseline collapses under this extended horizon ($< 30\%$ success) when the maze is quadrupled. Conversely, SRL demonstrates remarkable robustness, maintaining over 80% success and sustained goal occupancy even at the largest scale.

5.5 Ablation Study

We evaluate SRL on `Humanoid` and `Arm Push Easy` on 200 rollouts post-training at network depth 16 and $k \in \{1, 4, 8, 16, 32\}$.



(a) Exit probability (b) Lower bound vs. Goal occupancy

Figure 5: **Ablation study. (a)** $\hat{p}_k$ decreases monotonically with k; `Humanoid` requires larger k than `Arm Push Easy` to reduce exit probability, reflecting the difficulty of dynamic balancing versus quasi-static manipulation. **(b)** The lower bound from Proposition 1 tracks empirical goal-occupancy closely along the $y = x$ diagonal (axes normalized by H), with larger k shifting points toward higher occupancy.

Empirical exit probability. Fig. 5a reports the empirical exit probability $\hat{p}_k$. At $k = 1$, both environments yield $\hat{p}_k \approx 1.0$: a single in-goal step provides no incentive to stabilize. Increasing k reduces $\hat{p}_k$ at different rates. On `Arm Push Easy`, $k = 4$ already drives $\hat{p}_k$ to ≈ 0.30, decreasing monotonically to ≈ 0 at $k = 32$. On `Humanoid`, $\hat{p}_k$ remains near 1.0 for $k \leq 8$, drops to ≈ 0.75 at $k = 16$, and reaches ≈ 0.43 at $k = 32$. This contrast reflects task difficulty: the arm naturally holds the cube at rest once positioned, whereas the humanoid must continuously balance against gravity, requiring a larger k to achieve sustained occupancy.

Empirical occupancy lower bound. Fig. 5b validates Proposition 1 by plotting the estimated lower bound against the empirical goal-occupancy, both normalized by H. Data points lie closely along the $y = x$ line across k, confirming that increasing k improves the lower bound and drives higher occupancy. Runs with $k = 1$ cluster near the origin, while larger k shifts points along the diagonal, reaching $\overline{\text{GO}}/H \approx 0.9$ on `Arm Push Easy` and ≈ 0.5 on `Humanoid` at $k \in \{20, 32\}$.

6 Discussion and Conclusion

We introduced SRL, an online self-supervised RL algorithm that extends the survival-learning framework of Tiofack et al. (2026) with a dwell time at goal formulation. Without requiring additional loss terms or architectural changes, this formulation aligns the optimization target with the goal occupancy metric and recovers stable post-arrival behavior. Empirically, scaled SRL completely solves every `AntMaze` topology at depth 32, outperforming the depth-64 contrastive baseline of Wang et al. (2025). Our results extend prior findings of Wang et al. (2025) that the choice of self-supervised objective governs depth scaling by showing that the survival objective scales at least as gracefully as the contrastive one and substantially more favorably on long-horizon planning with stable dynamics.

We attribute this to two properties of the survival formulation. First, the hazard model is trained by maximum-likelihood classification over discrete hitting-time bins, inheriting the same advantages that motivated classification-based value learning Farebrother et al. (2024) and that underlie CRL's scaling behavior Wang et al. (2025). Second, unlike InfoNCE, the right-censored likelihood does not enforce uniform spread in the embedding space, sidestepping the uniformity-tolerance dilemma Wang & Liu (2021) inherent to CRL.

Limitations. The dwell time at goal formulation introduces an additional hyperparameter k; however, a small set of values ($k \in \{1, 20, 32\}$) proved sufficient across all twelve JaxGCRL tasks. The only environment where SRL does not match scaled CRL is `Arm Push Hard`, where task-agnostic entropy maximization rarely produces the initial cube contact, causing the policy to collapse onto trivial Hindsight relabeling targets (Bortkiewicz et al., 2025), a known failure mode shared by most GCRL algorithms. Finally, while the hazard critic produces a full distribution over hitting times, the value function currently uses only a discounted summary, leaving rich temporal information underutilized for policy optimization.

AI USE STATEMENT

We used generative AI tools to polish the writing and to assist with parts of the code. We did not use them to generate research ideas, claims, or experimental results. The authors reviewed all AI-assisted text and code, and tested the code for correctness. We take full responsibility for the final content of this work.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pp. 1577–1594. PMLR, 2023.

Richard Bellman, Irving Glicksberg, and Oliver Gross. On the "bang-bang" control problem. *Quarterly of Applied Mathematics*, 14(1):11–18, 1956.

Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

Michał Bortkiewicz, Władysław Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating goal-conditioned rl algorithms and research. *arXiv preprint arXiv:2408.11052*, 2024.

Michał Bortkiewicz, Władysław Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating goal-conditioned reinforcement learning algorithms and research. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4gaySj8kvX.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International conference on machine learning*, pp. 1430–1440. PMLR, 2021.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Lawrence C Evans. An introduction to mathematical optimal control theory version 0.2. *Lecture notes available at http://math. berkeley. edu/evans/control. course. pdf*, 1983.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=tc5qisoB-C.

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

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, et al. Stop regressing: Training value functions via classification for scalable deep rl. *arXiv preprint arXiv:2403.03950*, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

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 Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Manon Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=rALA0Xo6yNJ.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Zizheng Huang, Haoxing Chen, Ziqi Wen, Chao Zhang, Huaxiong Li, Bo Wang, and Chunlin Chen. Model-aware contrastive learning: Towards escaping the dilemmas. In *International conference on machine learning*, pp. 13774–13790. PMLR, 2023.

Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

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. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):24, 2018.

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. *arXiv preprint arXiv:2410.09754*, 2024.

Helmut Maurer and Nikolai P Osmolovskii. Second order sufficient conditions for time-optimal bang-bang control. *SIAM journal on control and optimization*, 42(6):2239–2263, 2004.

Lina Mezghani, Sainbayar Sukhbaatar, Piotr Bojanowski, Alessandro Lazaric, and Karteek Alahari. Learning goal-conditioned policies offline with self-supervised reward shaping. In *Conference on robot learning*, pp. 1401–1410. PMLR, 2023.

Vivek Myers, Catherine Ji, and Benjamin Eysenbach. Horizon generalization in reinforcement learning. *arXiv preprint arXiv:2501.02709*, 2025.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzciński, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. *arXiv preprint arXiv:2403.00514*, 2024a.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample efficient continuous control. *Advances in neural information processing systems*, 37:113038–113071, 2024b.

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

Kei Ota, Devesh K Jha, and Asako Kanezaki. A framework for training larger networks for deep reinforcement learning. *Machine Learning*, 113(9):6115–6139, 2024.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. *arXiv preprint arXiv:2506.04168*, 2025.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2018. URL https://openreview.net/forum?id=SkBYYyZRZ.

Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pp. 30365–30380. PMLR, 2023.

Franki Nguimatsia Tiofack, Fabian Schramm, Théotime Le Hellard, and Justin Carpentier. SVL: Goal-Conditioned Reinforcement Learning as Survival Learning. *arXiv preprint arXiv:2604.17551*, 2026.

Luís Torgo and Joao Gama. Regression by classification. In *Brazilian symposium on artificial intelligence*, pp. 51–60. Springer, 1996.

Jens Tuyls, Dhruv Madeka, Kari Torkkola, Dean Foster, Karthik Narasimhan, and Sham Kakade. Scaling laws for imitation learning in single-agent games. *arXiv preprint arXiv:2307.09423*, 2023.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

Kevin Wang, Ishaan Javali, Michał Bortkiewicz, Tomasz Trzcinski, and Benjamin Eysenbach. 1000 layer networks for self-supervised RL: Scaling depth can enable new goal-reaching capabilities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=s0JVsx3bx1.

Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline RL. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=KJztlfGPdwW.

A Appendix

A.1 Proof of Proposition 1

$$
\begin{aligned}
\mathrm{GO}(\pi) &= \mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\sum_{t=1}^{H}\mathbf{1}_{\{\varphi(s_t)\in\mathcal{B}_\epsilon(g)\}}\right] \\
&\geq \mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\sum_{t=1}^{H}\mathbf{1}_{\{\varphi(s_t)\in\mathcal{B}_\epsilon(g),\, T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)<t\}}\right] \\
&= \mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\sum_{t=1}^{H}\mathbf{1}_{\{T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)<t\}} - \mathbf{1}_{\{T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)<t,\, \varphi(s_t)\notin\mathcal{B}_\epsilon(g)\}}\right] \\
&\geq (1-p_k^\pi)\sum_{t=1}^{H}\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)<t\}}\right] \\
&= (1-p_k^\pi)\left(H - \mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)\wedge H\right]\right). \quad \square
\end{aligned}
$$

A.1.1 Tightness and Empirical Remarks

Proposition 2 (Tight goal occupancy lower bound)**.** *For a given policy π and fixed episode length H, define the time-dependent exit probability*

$$
p_k^\pi(t) = \mathbb{E}_{\substack{g\sim p_g, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{\varphi(s_t)\notin\mathcal{B}_\epsilon(g)\}} \mid T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0) < t\right],
$$

where $\mathbf{g}^{(k)} = (g,\ldots,g)$ and $T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)$ is defined in Eq. 8. Then:

$$
\mathbb{E}_{\substack{g\sim p_g\\ \tau\sim\rho^\pi(\cdot|g)}}\left[\sum_{t=1}^{H}\mathbf{1}_{\{\varphi(s_t)\in\mathcal{B}_\epsilon(g)\}}\right] \geq \sum_{t=1}^{H}\big(1-p_k^\pi(t)\big)\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)<t\}}\right].
$$

Note. Proposition 1 is recovered from Proposition 2 by bounding $p_k^\pi(t) \leq p_k^\pi \triangleq \max_{t\in\{1,\ldots,H\}} p_k^\pi(t)$ for all t, and applying the identity:

$$
\sum_{t=1}^{H}\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)<t\}}\right] = H - \mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)}}\left[T^\pi_{\mathbf{g}^{(k)}}(s_0,a_0)\wedge H\right].
$$

Proof Proposition 2

$$
\begin{aligned}
\mathrm{GO}(\pi) &= \mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\sum_{t=1}^{H}\mathbf{1}_{\{\varphi(s_t)\in\mathcal{B}_\epsilon(g)\}}\right] \\
&\geq \mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\sum_{t=1}^{H}\mathbf{1}_{\{\varphi(s_t)\in\mathcal{B}_\epsilon(g),\, T^\pi_{\mathbf{g}(k)}(s_0,a_0)<t\}}\right] \\
&= \sum_{t=1}^{H}\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}(k)}(s_0,a_0)<t\}} - \mathbf{1}_{\{T^\pi_{\mathbf{g}(k)}(s_0,a_0)<t,\, \varphi(s_t)\notin\mathcal{B}_\epsilon(g)\}}\right] \\
&= \sum_{t=1}^{H}\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}(k)}(s_0,a_0)<t\}}\right] - \sum_{t=1}^{H}\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}(k)}(s_0,a_0)<t,\, \varphi(s_t)\notin\mathcal{B}_\epsilon(g)\}}\right] \\
&= \sum_{t=1}^{H}\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}(k)}(s_0,a_0)<t\}}\right] - \sum_{t=1}^{H}p_k^\pi(t)\,\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}(k)}(s_0,a_0)<t\}}\right] \\
&= \sum_{t=1}^{H}\big(1-p_k^\pi(t)\big)\,\mathbb{E}_{\substack{g\sim p_g,\, s_0\sim\mu(\cdot|g)\\ a_0\sim\pi(\cdot|s_0,g)\\ \tau\sim\rho^\pi(\cdot|g,s_0,a_0)}}\left[\mathbf{1}_{\{T^\pi_{\mathbf{g}(k)}(s_0,a_0)<t\}}\right]. \quad \square
\end{aligned}
$$

A.2 Network Architectures

Figure 6 illustrates the architecture of the hazard critic $h_\theta(t \mid s, a, g)$ and the Gaussian actor $\pi_\psi(a \mid s, g)$. Both networks are built from a common residual block consisting of four `(Linear, LayerNorm, Swish)` layers wrapped by an identity skip connection. Following Wang et al. (2025), we define the network depth as the total number of `Linear` layers across residual blocks, so an encoder of depth N contains $N/4$ residual blocks.

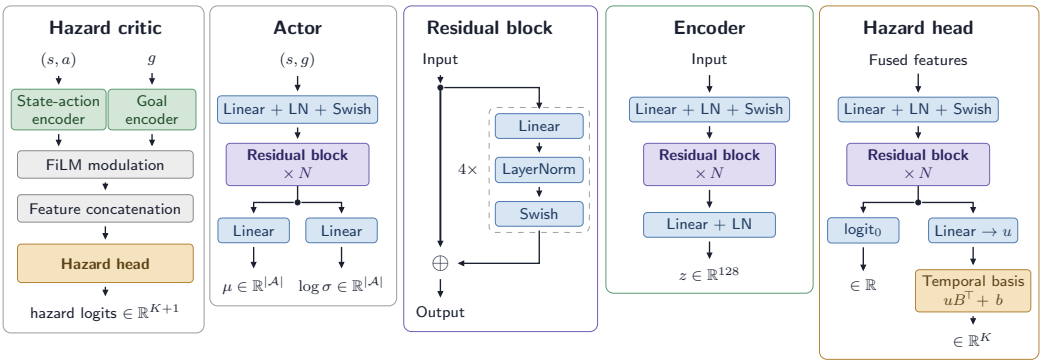


Figure 6: **Network architecture.** Hazard critic $h_\theta(t \mid s, a, g)$ (leftmost panel): a state–action encoder and a goal encoder produce 128-dimensional embeddings, which are fused via FiLM modulation followed by feature concatenation and passed to the hazard head. The head emits $K{+}1$ logits: each logit logit_t parameterized the probability $h_\theta(t \mid s, a, g) = \sigma(\mathrm{logit}_t)$. The K per-bin hazard logits $\ell_{1:K} = uB^\top + b$ are obtained by combining a feature vector u with a learned temporal basis B and bias b. Actor $\pi_\psi(a \mid s, g)$ (second panel): outputs the mean μ and log standard deviation $\log\sigma$ of a diagonal Gaussian policy. Residual block (center): four `(Linear, LayerNorm, Swish)` layers with an identity skip connection. Encoder and hazard head (right panels): a `(Linear+LN+Swish)` stem followed by residual blocks; the two encoders share this template.

Encoders. The state–action pair (s, a) and the desired goal g are processed by two independent encoders, each mapping its input to a 128-dimensional latent $z \in \mathbb{R}^{128}$.

Fusion. Encoder outputs are combined in two stages: the goal embedding first produces a pair of affine parameters (γ, β) applied channel-wise to the state–action embedding (FiLM modulation (Perez et al., 2018)). The modulated state–action embedding is then concatenated with the goal embedding before being passed to the hazard head.

Hazard head. The hazard head takes as input the concatenated features vector and outputs $K + 1$ logits, each logit parameterizing $\sigma(\text{logit}_t) = h_\theta(t \mid s, a, g)$. The K per-bin hazard logits $\ell_{1:K} = uB^\top + b$ are obtained by combining a feature vector $u \in \mathbb{R}^r$ with a learnable temporal basis $B \in \mathbb{R}^{K \times r}$ and bias $b \in \mathbb{R}^K$. For more details, see the architecture of Tiofack et al. (2026) (their Appendix A.4).

Actor. The actor takes (s, g) as input and outputs the mean $\mu \in \mathbb{R}^{|\mathcal{A}|}$ and log standard deviation $\log \sigma \in \mathbb{R}^{|\mathcal{A}|}$ of a diagonal Gaussian policy, which is trained with the entropy-regularised SAC objective in Eq. 13

Binning strategy and value computation. We adopt the geometric-time binning (Tiofack et al., 2026). Given horizon H and K bins, bin edges are $b_k = \lfloor \rho^k \rfloor$ with $\rho = H^{1/K}$, which yields short bins near $t = 0$ (where γ^t is large) and longer bins near the horizon. Among the discretization schemes introduced in Tiofack et al. (2026), we use piecewise-constant survival (PCS) parameterization, in which the survival function is kept constant within each bin, $S^\pi_\theta(t \mid s, a, g) = S^\pi_\theta(b_k \mid s, a, g)$ for $t \in [b_k, b_{k+1})$. The resulting Q-estimator is

$$Q_\theta(s, a, g) \;=\; -\sum_{k=0}^{K-1} D_k \, S^\pi_\theta(b_k \mid s, a, g), \qquad D_k \;=\; \sum_{t=b_k}^{b_{k+1}-1} \gamma^t, \tag{15}$$

with bin-start survivals computed recursively from logit_0 and the per-bin interval hazards $\tilde{h}_k = \sigma(\ell_k)$ through $\log S(b_0) = \log(1 - p_0)$ and $\log S(b_{k+1}) = \log S(b_k) + \log(1 - \tilde{h}_k)$. When $\gamma < 1$ we optionally add the geometric tail $\gamma^H S(b_K)/(1 - \gamma)$ to account for occupancy beyond the bin grid. The corresponding grouped-time log-likelihood used to train h_θ is the PCS objective derived in Appendix A.5 of Tiofack et al. (2026). It is important to note that when $H = K$ (which is the case for most of our experiments), each bin contains exactly one timestep.

Table 1: **Hyperparameters.** Parameters not specific to SRL are taken from Tiofack et al. (2026).

Hyperparameter	**Value**
Learning rate	3×10^{-4}
Optimizer	Adam
Batch size	512 (default), 1024 (`Humanoid`)
Environment steps	100M–1B (varies across experiments)
Update-to-data (UTD) ratio	1:40
MLP width	256
Discount factor γ	0.999 (default), 0.99 (`Arm_hard`)
Last interval edge b_K	1000
Number of bins K	$\min\{b_K, \texttt{env_episode_length}\}$
Value $(p^{\mathcal{D}}_{\text{cur}}, p^{\mathcal{D}}_{\text{traj}}, p^{\mathcal{D}}_{\text{rand}})$ ratio	(0.05, 0.85, 0.1)
Maximum replay size	10 000
Minimum replay size	1 000
Networks depth	depends on the experiment

The parameter `env_episode_length` in Tab. 1 corresponds to the environment predefine episode length in JaxGCRL.

Algorithm 2 Survival Reinforcement Learning (SRL)

Input Stacking parameter k, rollout horizon c, discount factor γ
Initialize hazard critic h_θ, actor π_ψ, empty replay buffer $\mathcal{B}$
while not done **do**
  # 1. Data collection
  Sample task goal $g \sim p_g$ and initial state s_0
  Collect trajectory $(s_0, a_0, \ldots, s_c)$ using actor $\pi_\psi(\cdot \mid s_t, g)$
  # 2. Hindsight relabeling and survival evaluation
  for each timestep t and sampled hindsight goal g' **do**
    Construct k-length target sequence $\mathbf{g}^{(k)}$ (Eq. 7)
    Compute first hitting time τ_t requiring matching of $\mathbf{g}^{(k)}$
    Set censorship indicator $\delta_t = 1$ if sequence is matched within c, else 0
    Store survival tuple $(s_t, a_t, \mathbf{g}^{(k)}, t_h, c, \delta_t)$ in $\mathcal{B}$
  end for
  # 3. Updates
  for n_{grad} gradient steps **do**
    Sample mini-batch of survival tuples $(s_t, a_t, \mathbf{g}^{(k)}, t_h, c, \delta_t) \sim \mathcal{B}$
    Update hazard critic h_θ by minimizing the right-censored NLL (Eq. 11)
    Estimate action-value $Q^\pi_{\mathbf{g}^{(k)}}(s_t, a_t)$ via the survival identity (Eq. 9)
    Update actor π_ψ by minimizing the maximum entropy objective (Eq. 13)
  end for
end while
Output Optimized actor policy π_ψ

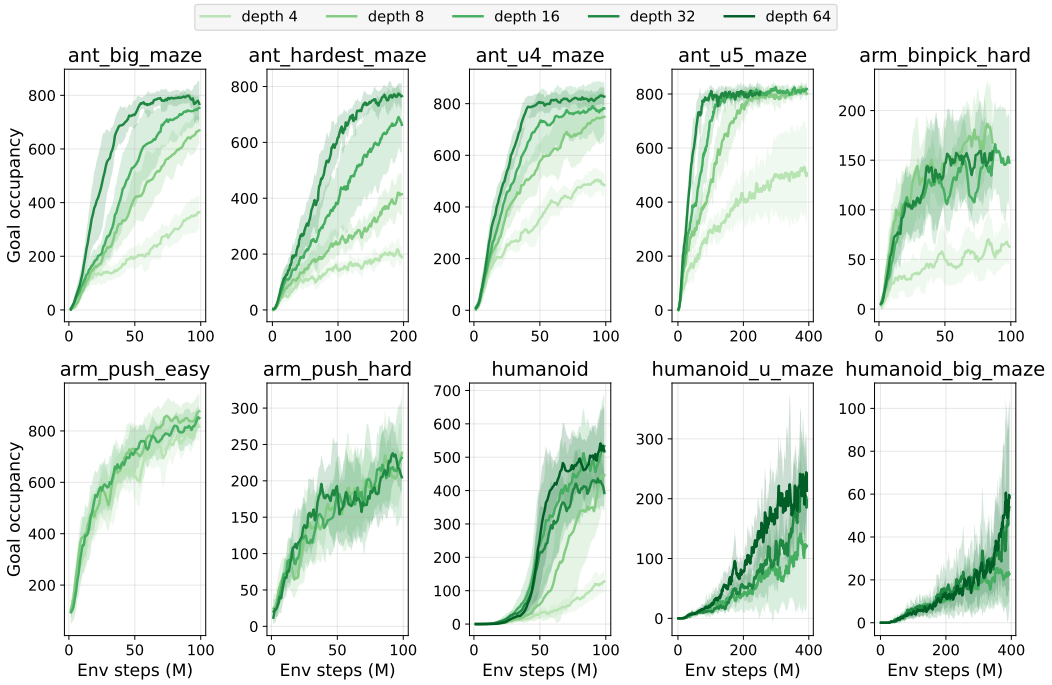


Figure 7: **Depth scaling of SRL** across the JaxGCRL suite. Goal occupancy (mean ± std over 4 seeds).

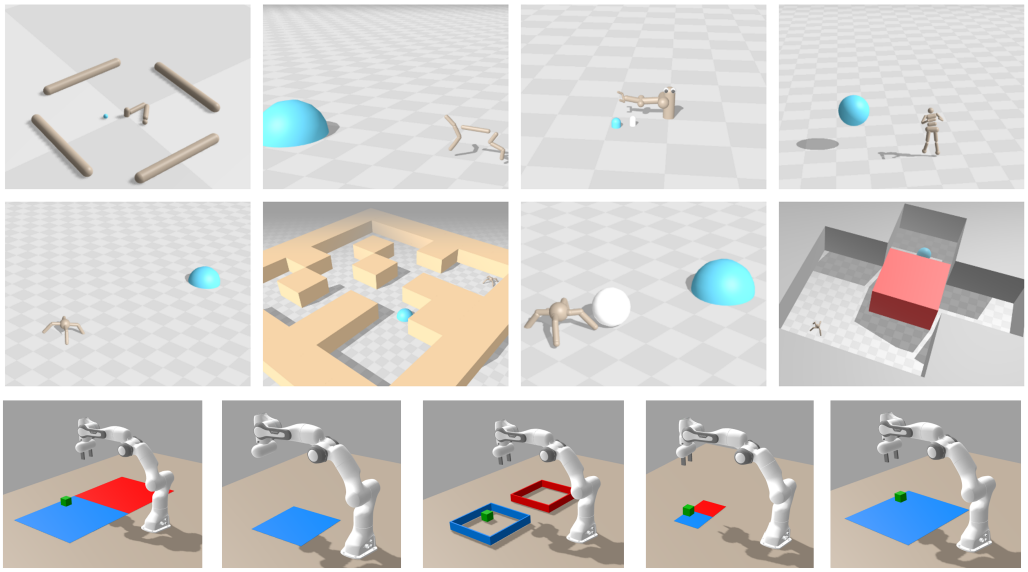


Figure 8: **Some environments and tasks available in JaxGCRL.** Image from Bortkiewicz et al. (2025).

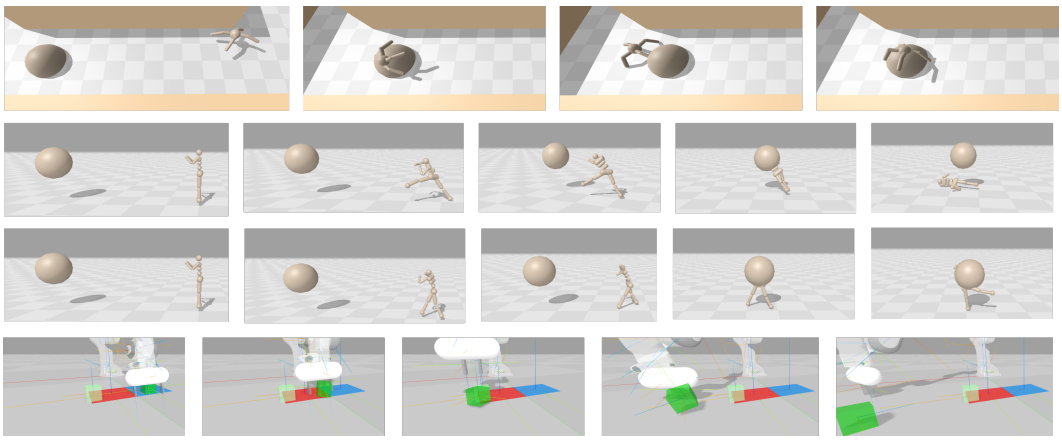


Figure 9: **Visualization of agent behavior.** Enlarged view of Fig. 3.