Compressing Deep Reinforcement Learning via Evolving Target Sparsity

Rachana Tirumanyam[1][0009−0003−5944−583X],
André Biedenkapp[1][0000−0002−8703−8559], and
Jovita Lukasik[2][0000−0003−4243−9188]

[1] University of Freiburg
[2] University of Siegen

Abstract. Deep Reinforcement Learning (DRL) models are heavily overparameterized, hindering edge deployment. Analyzing neural redundancy across Atari and MuJoCo reveals a stark divide: while discrete policies withstand 95% static sparsity, continuous tasks suffer catastrophic collapse. To achieve stable extreme compression, we propose Population-Based Dynamic Sparse Training (PBT-DST). By treating target sparsity as an evolving hyperparameter, PBT-DST uses an exploit-and-mutate loop to decouple topology search from non-stationary RL gradients. Benchmarked against static post-hoc compression and Gradual Magnitude Pruning (GMP), PBT-DST eliminates the need for predetermined sparsity targets. Instead, it autonomously evolves the network to its optimal task-specific sparsity, successfully rescuing continuous tasks from static failure while maintaining stable performance.

Keywords: Deep Reinforcement Learning · Dynamic Pruning · Population Based Training · Edge AI

1 Introduction

Deep Reinforcement Learning (DRL) has demonstrated immense potential in robotics [12,9], autonomous systems [1,2,23], and healthcare applications such as prosthetic control [21] and insulin regulators [4]. At the heart of these successes lie deep neural networks used for function approximation [11,16,7,17,5]. Yet several lines of work suggest that the representational capacity of commonly used architectures is largely underutilized [10,6]: even tiny linear networks [20] or simple oscillators [14] suffice to solve popular MuJoCo locomotion benchmarks[18]. This overparameterization is a barrier to on-device RL, where large policies trained in simulation must be compressed to fit available hardware.

While sparse "winning tickets" exist in DRL [22], transferring compression to the online RL setting is challenging. Sparse training methods from supervised learning frequently introduce optimization instabilities under non-stationary RL data distributions [6]. Yet, AutoRL [13] successfully adapts architectures during training [19], but typically focuses on *growing* rather than *compressing* networks to boost performance. While network growth expands capacity at the cost of

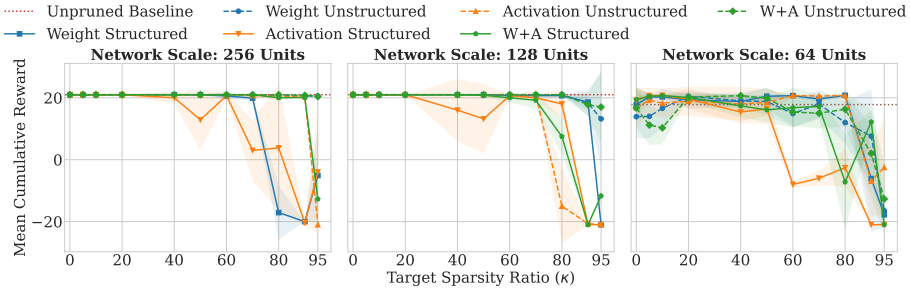


Fig. 1: **Post-hoc policy degradation sweeps on Pong** across three network scales. Extreme sparsity tolerance reveals massive architectural redundancy before a cliff-like drop. Curves report mean cumulative reward across 10 evaluation episodes for a converged policy; shaded regions denote one standard deviation.

hardware overhead, leveraging automated population-level search to discover compact sub-networks remains an open frontier.

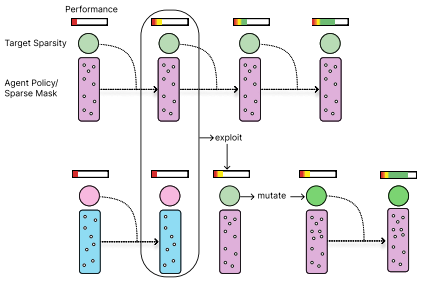
We close this gap with the following contributions: **(1)** We propose **PBT-DST**, a dynamic sparse training approach based on Population Based Training [8] that treats target sparsity as an actively evolving hyperparameter, eliminating the need for predetermined sparsity targets. **(2)** We benchmark weight-based, activation-based, and hybrid pruning metrics against static post-hoc and gradual magnitude pruning (GMP) [24]. **(3)** We demonstrate massive architectural redundancy in standard DRL environments, showing optimal performance requires only a fraction of original capacity (Fig. 1).

2 Method: Population-Based Dynamic Sparse Training

We consider an MDP where the policy $\pi_\theta(a|s)$ uses parameters $\theta \in \mathbb{R}^D$. Structural compression applies a binary mask $M \in \{0,1\}^D$, yielding $\pi_{M\odot\theta}$, subject to $\|M\|_0 \leq (1-S)D$ for a target sparsity $S \in [0,1)$. PBT-DST decouples optimization: θ updates via local gradients, while M and S evolve across a population via trajectory-level feedback.

Pruning criteria. In fully connected bottleneck layers, connection importance I_{ij} is scored via: **(1) Weight-Magnitude:** $I_{ij} = |w_{ij}|$; **(2) Activation-Driven:** $I_{ij} = |w_{ij}| \cdot \bar{x}_j$, where $\bar{x}_j$ tracks layer inputs via an exponential moving average ($\alpha_{\text{ema}} = 0.99$); or **(3) Hybrid Alternating:** Toggling Modes 1 and 2 per interval. Connections scoring lowest are masked to meet S (convolutional encoders remain dense).

Evolutionary loop. Following PBT [8], we initialize $P = 8$ agents with initial sparsities $S^{(k)} \in \{0, 0.1, 0.2\}$. To let weights learn meaningful state representations, a dense **lockout period** precedes evolution. After which, every T steps,



Algorithm 1 PBT-DST Exploit & Mutate

Require: Agent k (bottom quantile), Top k^*
if $\text{Fitness}(k^*) - \text{Fitness}(k) > \text{Var\_Threshold}$ **then**
  Exploit: $\theta^{(k)}, M^{(k)} \leftarrow \theta^{(k^*)}, M^{(k^*)}$
  $S^{(k)}, \text{Cycle}^{(k)} \leftarrow S^{(k^*)}, \text{Cycle}^{(k^*)}$
  Mutate: $S^{(k)} \leftarrow \min(S^{(k)} + \Delta_S, 0.95)$
  if Hybrid Mode **then**
    $\text{Mode} \leftarrow \text{Cycle}^{(k)} \pmod 2$
  end if
  $M^{(k)} \leftarrow \text{MaskTopK}(\mathbf{I}_{\text{Mode}}, S^{(k)})$
end if

Fig. 2: **Left:** Timeline of topology transfer and mutation. **Right:** Exploit & Mutate update applied to underperforming agents, gated by a variance threshold.

agents are evaluated using:

$$\text{Fitness}(k) = \bar{R}^{(k)} + \mathbb{I}(\bar{R}^{(k)} \geq \rho) \cdot \alpha \cdot (S^{(k)})^2, \tag{1}$$

where $\bar{R}^{(k)}$ is rolling reward, ρ a task-specific competence threshold, and α scales the quadratic sparsity reward.

During the **exploit-and-mutate** cycle (Fig. 2), bottom-quantile agents *exploit* by cloning a top performer's state, *only* if the fitness gap exceeds a significance margin (`Var_Threshold`), protecting stable policies from being prematurely overwritten. They then *mutate* by increasing $S^{(k)}$ by Δ_S (capped at 0.95) and re-masking. By exclusively mutating underperforming agents that just inherited competent weights, the population autonomously discovers optimal task-specific sparsity.

3 Experiments

Setup. We evaluate Atari (*Pong, Freeway*) using Nature-DQN architectures [11] trained with PPO [16], and MuJoCo (*Ant, HalfCheetah*) using MLPs trained with SAC [7] via Stable-Baselines3 [15]. We restrict hidden bottlenecks to 256, 128, or 64 units to show even reduced architectures harbor redundancy. Target sparsity S applies globally across linear layers. *(Hyperparameters:* $P = 8$*,* $\Delta_S = 0.05$*;* $\rho \in [18, 4500]$*,* $\alpha \in [5, 200]$*, and Var_Threshold* $\in [1.5, 2500]$ *depending on task, detailed in code).* We compare: (i) dense upper bound, (ii) static post-hoc pruning swept over $S \in [0.05, 0.95]$, (iii) dynamic GMP [24], and (iv) PBT-DST.

Redundancy in deep RL. Post-hoc pruning reveals a stark domain divide. Pruning bottleneck layers (keeping encoders dense) in Atari exhibits massive over-parameterization: *Pong* and *Freeway* maintain performance up to 90–95% unstructured sparsity. Conversely, continuous locomotion policies (MLPs) suffer catastrophic failure past 20–40% unstructured sparsity. Training MuJoCo tasks with PPO yields higher tolerance (up to 80%) but from a lower dense baseline.

Dynamic sparse training. As shown in Table 1, static pruning collapses catastrophically on continuous tasks when forced to 95% sparsity. Dynamic

Table 1: **Mean cumulative rewards** (± std) at 256-unit scale (unstructured pruning). Target sparsity is 95% unless explicitly noted. PBT-DST autonomously halts at 50–55% on continuous tasks. Evaluated over 7 agents (PBT-DST), 3 seeds (GMP), and 10 episodes of one policy (Static Post-Hoc). Bold indicates best sparse performance.

Method	**Pong**	**Freeway**	**Ant**	**HalfCheetah**
Dense Bound	21.0 ± 0.0	32.5 ± 1.1	6513 ± 78	14550 ± 36
Static Post-Hoc	$\mathbf{20.6 \pm 0.5}$	28.2 ± 0.9	-343 ± 498 (95%) 644 ± 669 (50%)	-328 ± 1 (95%) 1482 ± 367 (50%)
Dynamic GMP	6.2 ± 22.9	27.5 ± 5.2	$\mathbf{3579 \pm 1518}$ (95%)	13101 ± 172 (50%)
Ours (PBT-DST)	19.9 ± 0.3	$\mathbf{30.9 \pm 0.5}$	3029 ± 389 (50%)	$\mathbf{13807 \pm 103}$ (55%)

methods like GMP mitigate this, but rely on rigid, predefined targets. If this a priori guess is incorrect, the policy collapses or necessitates expensive hyperparameter sweeps (e.g., scaling GMP back to 50% for *HalfCheetah*). PBT-DST bypasses this. By treating target sparsity as an evolving parameter, the population autonomously discovers the environment's tolerance limit, halting at 50–55% for continuous control while safely pushing to 95% on discrete tasks. PBT-DST guarantees a stable, highly compressed agent in a single distributed run, eliminating blind searches for sparsity targets.

4 Conclusion

Our findings reinforce prior observations regarding the extensive overparameterization of deep RL policies, while highlighting its strong domain-dependent nature. While static pruning compresses Atari networks by 95% losslessly, it triggers catastrophic collapse in continuous locomotion. To overcome this, we introduced PBT-DST, treating target sparsity as an actively evolving trait. By decoupling topology search from non-stationary RL gradients via an exploit-and-mutate loop, PBT-DST dynamically adapts mask schedules based on agent competence. Rather than relying on rigid, predetermined pruning schedules, this enables the population to autonomously discover the environment's optimal sparsity equilibrium, eliminating expensive trial-and-error searches and rescuing continuous-control tasks from static failure. Future work includes establishing a unified, budget-matched evaluation protocol across baselines, evaluating the hybrid pruning criterion, and testing whether discovered sparse masks can be trained from scratch. We will also formally quantify the computational tradeoffs of the population search, explore bidirectional mutations to dynamically refine per-task sparsity equilibria, and combine this structural pruning with weight quantization for TinyRL deployment.

Acknowledgments

The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG.

References

1. Bellemare, M.G., Candido, S., Castro, P.S., Gong, J., Machado, M.C., Moitra, S., Ponda, S.S., Wang, Z.: Autonomous navigation of stratospheric balloons using reinforcement learning. Nature **588**(7836), 77–82 (2020)
2. Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de las Casas, D., Donner, C., Fritz, L., Galperti, C., Huber, A., Keeling, J., Tsimpoukelli, M., Kay, J., Merle, A., Moret, J.M., Noury, S., Pesamosca, F., Pfau, D., Sauter, O., Sommariva, C., Coda, S., Duval, B., Fasoli, A., Kohli, P., Kavukcuoglu, K., Hassabis, D., Riedmiller, M.: Magnetic control of tokamak plasmas through deep reinforcement learning. Nature **602**(7897), 414–419 (2022)
3. Dy, J., Krause, A. (eds.): Proceedings of the 35th International Conference on Machine Learning (ICML'18), vol. 80. Proceedings of Machine Learning Research (2018)
4. Fox, I., Wiens, J.: Reinforcement learning for blood glucose control: Challenges and opportunities. arXiv preprint arXiv:2012.05451 (2020)
5. Fujimoto, S., Hoof, H., Meger, D.: Addressing Function Approximation Error in Actor-Critic Methods. In: Dy and Krause [3], pp. 1582–1591
6. Graesser, L., Evci, U., Elsen, E., Castro, P.: The state of sparse training in deep reinforcement learning. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning (ICML'22). Proceedings of Machine Learning Research, vol. 162. PMLR (2022)
7. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Dy and Krause [3]
8. Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., Kavukcuoglu, K.: Population based training of neural networks. arXiv:1711.09846 [cs.LG] (2017)
9. Jayawardana, V., Freydt, B., Qu, A., Hickert, C., Yan, Z., Wu, C.: Intersectionzoo: Eco-driving for benchmarking multi-agent contextual reinforcement learning. In: The Thirteenth International Conference on Learning Representations, ICLR 2025 (2025)
10. Kumar, A., Agarwal, R., Ghosh, D., Levine, S.: Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
11. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (02 2015)

12. OpenAI, Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., Schneider, J., Tezak, N., Tworek, J., Welinder, P., Weng, L., Yuan, Q., Zaremba, W., Zhang, L.: Solving rubik's cube with a robot hand. arXiv:1910.07113 [cs.LG] (2019)
13. Parker-Holder, J., Rajan, R., Song, X., Biedenkapp, A., Miao, Y., Eimer, T., Zhang, B., Nguyen, V., Calandra, R., Faust, A., Hutter, F., Lindauer, M.: Automated reinforcement learning (AutoRL): A survey and open problems. Journal of Artificial Intelligence Research (JAIR) **74**, 517–568 (2022)
14. Raffin, A., Sigaud, O., Kober, J., Albu-Schäffer, A., Silvério, J., Stulp, F.: An open-loop baseline for reinforcement learning locomotion tasks. RLJ **1**, 92–107 (2024), https://rlj.cs.umass.edu/2024/papers/Paper18.html
15. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research **22**(268), 1–8 (2021)
16. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv:1707.06347 [cs.LG] (2017)
17. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. Adaptive computation and machine learning, MIT Press, 2 edn. (2018)
18. Todorov, E., Erez, T., Tassa, Y.: MuJoCo: A physics engine for model-based control. In: International Conference on Intelligent Robots and Systems (IROS'12). pp. 5026–5033. ieeecis, IEEE (2012)
19. Wan, X., Lu, C., Parker-Holder, J., Ball, P., Nguyen, V., Ru, B., Osborne, M.: Bayesian generational population-based training. In: Guyon, I., Lindauer, M., van der Schaar, M., Hutter, F., Garnett, R. (eds.) Proceedings of the First International Conference on Automated Machine Learning. Proceedings of Machine Learning Research (2022)
20. Wong, A., de Nobel, J., Bäck, T., Plaat, A., Kononova, A.V.: Solving deep reinforcement learning benchmarks with linear policy networks. arXiv:2402.06912 [cs.LG] (2024)
21. Wu, R., Li, M., Yao, Z., Liu, W., Si, J., Huang, H.: Reinforcement learning impedance control of a robotic prosthesis to coordinate with human intact knee motion. IEEE Robotics and Automation Letters **7**(3), 7014–7020 (2022). https://doi.org/10.1109/lra.2022.3179420
22. Yu, H., Edunov, S., Tian, Y., Morcos, A.S.: Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. arXiv (2019). https://doi.org/10.48550/arxiv.1906.02768
23. Zheng, H., Ma, Y., Araki, B., Chen, J., Wu, C.: Learning-guided prioritized planning for lifelong multi-agent path finding in warehouse automation. JAIR **85** (2026). https://doi.org/10.1613/JAIR.1.20611
24. Zhu, M., Gupta, S.: To prune, or not to prune: exploring the efficacy of pruning for model compression. In: International Conference on Learning Representations (ICLR) Workshop Track (2018), https://openreview.net/forum?id=S1lN69AT-