Transferability of Sparse Subnetworks in Deep Reinforcement Learning

Sebastian Pusch[1] (s5488079), Mihai Damian[1] (s5508983), Yitong Lin Yang[1] (s5460611), Riandika Marshequila Dinu[1] (s5603838)
{s.pusch,m.damian,y.lin.39,riandika.marshequila.dinu}@student.rug.nl

[1]**University of Groningen - Reinforcement Learning Practical - Group 9**

Abstract

Sparse subnetworks such as lottery tickets can match or outperform dense performance in supervised learning, but it remains unclear whether such sparsified representations transfer across reinforcement learning (RL) tasks. We study the transferability of pruned subnetworks in value-based deep reinforcement learning by training Double Deep Q-Network (DDQN) agents on Atari environments and transferring sparse encoders between tasks. We compare subnetworks obtained via Lottery Ticket Hypothesis (LTH) pruning and Gradual Magnitude Pruning (GMP), and evaluate two target task adaptation regimes: Static transfer (no further pruning) and continued magnitude pruning during target training to restore plasticity. Across *Pong*, *Breakout*, and *Space Invaders*, we find that while winning tickets can exist within individual tasks, transferring sparse subnetworks across games yields limited or negative transfer relative to training from scratch, and reintroducing pruning during target task training further degrades performance compared to Static transfer. These results suggest that magnitude-based sparsity alone does not mitigate the representation-value misalignment that arises during cross-task transfer in DDQN, and that effective RL transfer requires mechanisms that preserve or re-establish encoder-head co-adaptation rather than solely relying on sparsity.

1 Introduction

Modern neural networks in supervised learning are typically heavily overparameterized, a phenomenon already noted in classic work such as Optimal Brain Damage (LeCun et al., 1990), which showed that many parameters can be removed with little loss in performance. The Lottery Ticket Hypothesis (LTH) (Frankle & Carbin, 2018) uses iterative magnitude pruning to identify sparse subnetworks—"winning tickets"—inside dense, randomly initialized networks, then resets their surviving weights to their original values and retrains them in isolation. These subnetworks can match or exceed the accuracy of the full model, often training faster and more reliably, and follow-up work shows that they can sometimes transfer across related datasets (Ganesh et al., 2023), suggesting that they encode meaningful inductive biases rather than merely overfitting to a single task.

In deep reinforcement learning, transfer has been far less successful. Sabatelli and Geurts (2021) examine transfer learning in Atari by pre-training a Dense Deep Q-Network (DDQN) on a source game and transferring the convolutional encoder to a target game while reinitializing the final value layer. They report limited or negative transfer and conjecture that the tight co-evolution of the feature extractor and value function approximator during training causes negative transfer: reinitializing the head breaks this coupling, leaving the pretrained features detached from the new value function. This highlights the importance of preserving representational alignment during transfer.

At the same time, recent work shows that sparsity itself can be effective in reinforcement learning (RL) when handled appropriately. Obando-Ceron et al. (2023) demonstrate that pruned value-based agents can match or surpass dense performance even at high sparsity levels, indicating that significant parameter reductions do not necessarily degrade policy quality. Todorov et al. (2025) further show that dynamically sparse methods such as Gradual Magnitude Pruning (GMP) improve robustness, plasticity, and generalization in multi-task RL settings, outperforming both dense agents and static sparse models. Together, these results suggest that sparse RL networks can perform well when sparsity is applied in a manner that preserves learning dynamics.

Despite these encouraging results, it remains unclear how sparsity interacts with transfer. Prior work has either examined transfer in dense agents (Sabatelli & Geurts, 2021) or studied sparsity in isolation (Obando-Ceron et al., 2023; Todorov et al., 2025), without considering whether sparse subnetworks identified on one RL task—such as winning tickets—can be effectively reused on another. In particular, the role of dynamic sparsification during adaptation to a new task has not been explored. Understanding this interaction is key to determining whether transferable sparse representations can be constructed in RL.

Then, our research question is: *To what extent do sparse networks discovered on a source reinforcement learning task transfer to related target tasks, and how does introducing sparsification during target task tuning affect transfer performance when compared against dense and sparse-on-target-only baselines?*

We believe that sparse subnetworks (winning tickets) identified on a source RL task encode task-relevant inductive bias that can partially transfer to a related target task. We hypothesize that reintroducing dynamic sparsification during target task training, via GMP, will lead to positive transfer relative to static sparse transfer and dense baselines, by restoring network plasticity and, at the same time, enabling re-alignment between transferred representations and the target value function.

To test this hypothesis, we adopt the following experimental approach. We use DDQN agents with a fixed convolutional architecture, and compare sparse masks obtained via Lottery Ticket Hypothesis pruning and Gradual Magnitude Pruning. Our main intervention is the reintroduction of magnitude-based sparsification during target task training, allowing us to compare static sparse transfer against continued pruning as a mechanism for restoring plasticity. Transfer performance is evaluated using the interquartile mean (IQM) of episodic return with bootstrapped confidence intervals, and quantifies transfer benefit via area ratio, for which we also report 95% bootstrap confidence intervals (CIs), relative to training from scratch and sparse-only baselines.

2 Experimental setup

2.1 Benchmarks

We addressed the research question on three Atari environments from the Arcade Learning Environment (ALE) (Bellemare et al., 2013): *Pong*, *Breakout*, and *Space Invaders*. These were selected to represent a diverse range of visuals and game mechanics, while also considering computational feasibility. Additionally, games from ALE were also chosen in past studies on transferability (Sabatelli & Geurts, 2021). *Pong* and *Breakout* share similar dynamics, involving the player moving a paddle to hit a ball. The main difference between them is that in *Pong*, the player plays against another player, and in *Breakout*, the player has to guide the ball to break all of the bricks on the screen. *Space Invaders* is more complex as the player has to take into consideration the enemies' firing and movement patterns. All three games are single-task, fully observable environments and have been widely used in prior reinforcement learning research (Obando-Ceron et al., 2023).

2.2 Algorithms, Architecture and Hyperparameters

We used the Double Deep Q-Networks (DDQN) (Van Hasselt et al., 2016) as our main algorithm across all experiments. DDQN is used to address the overestimation bias in Q-learning by splitting the max operation in the target into action selection and action evaluation, which makes it a robust baseline for studying sparsity and transfer. We are using a shared convolutional architecture, consisting of three convolutional layers, followed by a fully connected layer. We avoid architecture ablations to isolate the effects of sparsity and transfer learning, ensuring that any performance differences arise from our pruning methods rather than architectural variations.

2.3 Pruning Methods

To discover sparse subnetworks, we investigate two magnitude-based pruning schedules:

Lottery Ticket Hypothesis (LTH) schedule (Frankle & Carbin, 2018): We prune in discrete rounds at regular intervals, removing the lowest-magnitude weights to reach a target sparsity level. After each pruning round, surviving weights are reset to their initialization values θ_0 from step k (rewind step, $k = 8 \times 10^4$). This iterative prune and reset procedure is continuously executed until the desired sparsity level is reached.

Gradual Magnitude Pruning (GMP) (Zhu & Gupta, 2017): We prune the weights gradually during training, removing the lowest-magnitude weights without resetting. The sparsity level increases smoothly from 0% to the target sparsity over the course of training.

Both methods use magnitude-based pruning, and they only differ in timing (discrete vs. continuous) and whether we reset the weights (reset vs. retain). We explore sparsity levels of 60% and 75% to determine how sparsity affects both source task performance and transferability.

We evaluate the existence and quality of winning tickets using a Random LTH baseline. This control uses random unstructured pruning rather than magnitude-based selection and randomly reinitializes surviving weights instead of rewinding to θ_0. This allows us to determine if a subnetwork's performance is due to specific learned connectivity or simply the resulting sparsity level.

2.4 Training and Evaluation

Each experiment is conducted with 5 seeds to ensure statistical reliability. Source-task agents are trained for 10 million steps, which provides sufficient convergence for DDQN on these Atari environments (Mnih et al., 2015). During target task transfer, we initialize networks with the sparse masks and (where applicable) weight values discovered on the source task, then continue training for an additional 10 million timesteps. For the target task, we ablate two conditions: continued GMP or no additional pruning.

We evaluate performance using the Interquartile Mean (IQM) of episodic returns (Agarwal et al., 2021), which provides a robust, fair, and stable performance estimate by filtering out extreme outliers and focusing only on the middle 50% of our data. Using IQM is important given the high variance of Atari environments. We also report 95% bootstrap confidence intervals to quantify uncertainty. As a secondary metric, we report the area ratio between transfer and baseline learning curves. It is used to qualitatively and quantitatively indicate positive or negative transfer benefit.

2.5 Transfer and Baseline Condition Naming

To clearly distinguish between experimental conditions, we adopt the following naming convention:

Source task training conditions:

- Dense: full network trained without pruning
- LTH{X}: Lottery Ticket Hypothesis pruning to X% sparsity (e.g., LTH60)
- GMP{X}: Gradual Magnitude Pruning to X% sparsity (e.g, GMP75)

Target task transfer conditions:

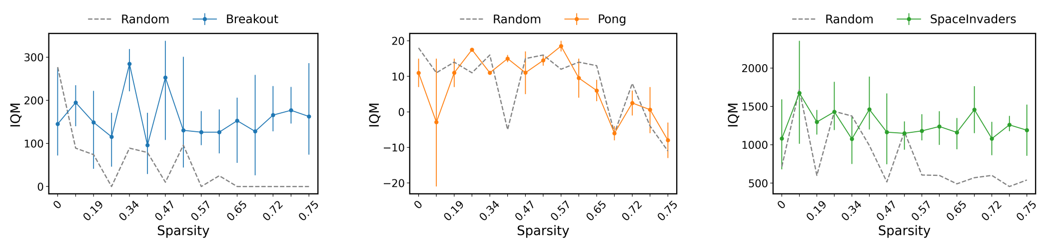
- {Method}Static: Sparse mask and weights from source task are transferred, and no additional pruning occurs during target task training (e.g, LTH60 Static)

- {Method}+GMP: Sparse mask from source task is transferred as initialization, then Gradual Magnitude Pruning is applied during target task training to further increase sparsity.

Baseline conditions (trained from scratch on the target task):

- Dense: full network trained without pruning
- GMP{X}: Gradual Magnitude Pruning applied from random initialization to X% sparsity

3 Results



(a) Breakout (b) Pong (c) Space Invaders

Figure 1: Existence and stability of winning tickets across Atari environments. Curves show the interquartile mean (IQM) of episodic return across seeds evaluated at the end of each pruning iteration, with vertical bars indicating 95% bootstrap confidence intervals. Performance is reported for increasing sparsity levels and compared against randomly pruned baselines.

We organize the results around our main research question: to what extent sparse subnetworks discovered on a source reinforcement learning task transfer to related target tasks, and whether allowing further sparsification during target task training improves transfer performance. We first establish the existence and stability of winning tickets on individual tasks, then analyze transfer performance relative to dense and sparse baselines, and finally examine ablations that isolate the role of pruning schedules and sparsity levels.

3.1 Existence of Winning Tickets

Figure 1 summarizes the existence of winning tickets across environments. For *Breakout* and *Space Invaders*, performance at the final pruning iteration exceeds that of the initial dense model up to 75% sparsity. Moreover, the performance gap relative to the random baseline widens with increasing sparsity, indicating that the identified subnetworks encode task-relevant structure rather than behaving as random sparse initializations.

In contrast, *Pong* exhibits a marked degradation in performance beyond 60% sparsity. At higher sparsity levels, the agent underperforms relative to the initial dense model, suggesting a failure to identify winning tickets at these sparsities. Furthermore, across all sparsity levels, the LTH pruning trajectory for *Pong* closely follows the random baseline, indicating that subnetworks identified at moderate sparsity do not retain meaningful inductive bias compared to randomly pruned counterparts.

3.2 Transfer

3.2.1 LTH

Figure 2 presents training curves for all baseline models and LTH-based transfer conditions across source–target environment pairs. Across all tasks, winning tickets at 60% sparsity transferred without further pruning on the target task (LTH60 Static) achieve the strongest overall transfer performance, both in terms of final IQM and learning stability. Contrary to our hypothesis, reintroducing

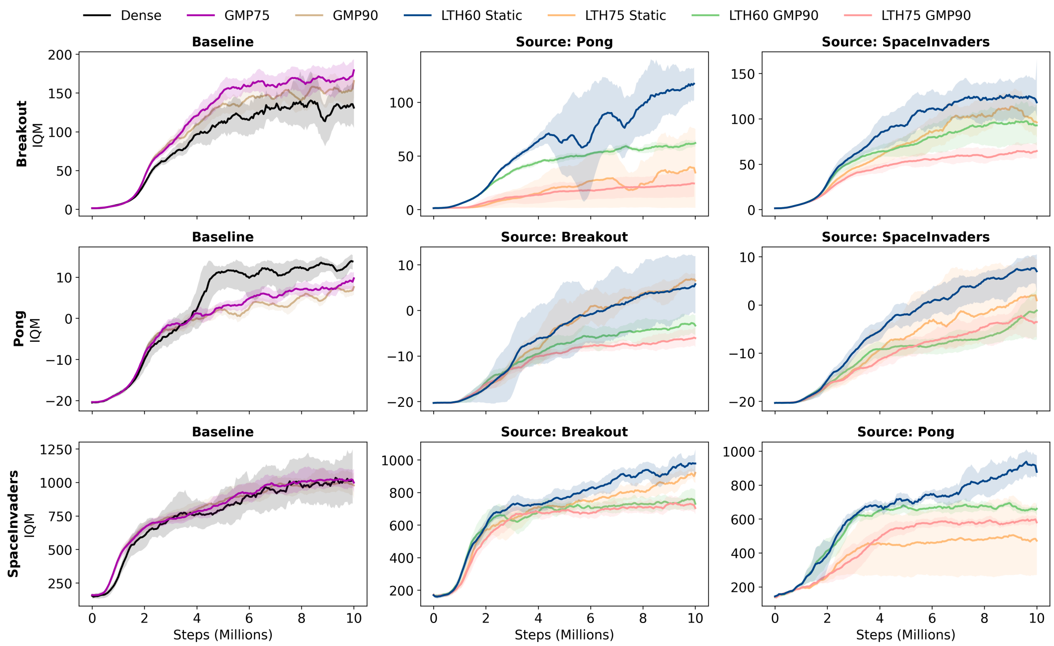


Figure 2: Target task learning curves under transfer for three Atari environments. Curves show the interquartile mean (IQM) of episodic return across seeds, with shaded regions indicating 95% bootstrap confidence intervals. Each row corresponds to a target task; the first column shows the baseline, and each subsequent column corresponds to a source task.

gradual magnitude pruning during target task training does not improve adaptation and, in most cases, degrades performance relative to Static transfer.

The LTH75 transfer conditions with *Pong* as the source consistently exhibit lower final IQM and a delayed learning onset across all target environments. As shown in Figure 1b, winning tickets could not be identified for *Pong* at 75% sparsity. This observation suggests that effective winning tickets encode inductive biases that are, at least in part, relevant to the target task.

Baseline performance varies across environments. For *Breakout*, the GMP75 baseline achieves the highest final performance, while for *Pong*, the dense baseline remains strongest. In *Space Invaders*, no statistically significant differences are observed among baseline configurations. These observations motivate comparing the best-performing transfer configuration (LTH60 Static) against the strongest baseline for each task.

Table 1: Area ratios comparing LTH60 Static transfer against (left) Dense and (right) GMP75 baselines. Rows are source and columns are target environments. † indicates statistically significant negative transfer (95% bootstrap confidence interval does not overlap zero).

(a) LTH60 Static vs. Dense

	Breakout	Pong	Space Invaders
Breakout		$-2.8796^{\dagger}$	$-0.0552^{\dagger}$
Pong	$-0.2978^{\dagger}$		$-0.1655^{\dagger}$
Space Invaders	-0.0930	$-2.3985^{\dagger}$	

(b) LTH60 Static vs. GMP75

	Breakout	Pong	Space Invaders
Breakout		17.2047	$-0.0877^{\dagger}$
Pong	$-0.4541^{\dagger}$		$-0.1951^{\dagger}$
Space Invaders	$-0.2957^{\dagger}$	12.6953	

Tables 1a and 1b report the corresponding area ratios. Across all source–target pairs, transfer results in negative area ratios, with most comparisons being statistically significant, indicating consistent negative transfer relative to training from scratch. The only apparent exception arises when comparing LTH60 Static against the GMP75 baseline on *Pong* as the target; however, this effect is not statistically significant, as the confidence interval overlaps with zero. We attribute this deviation to the high variability observed in the *Pong* environment during our experiments, combined with the comparatively weak performance of the GMP75 baseline relative to the dense baseline. Overall, these results provide a negative answer to our research question and contradict our hypothesis: the inductive bias encoded by sparse winning tickets identified on a source Atari task does not transfer effectively across environments under the considered settings.

3.2.2 GMP

	Breakout	Pong	Space Invaders
Breakout		$-1.7897^{\dagger}$	$-0.1827^{\dagger}$
Pong	$-0.2481^{\dagger}$		$-0.1872^{\dagger}$
Space Invaders	$-0.3828^{\dagger}$	$-1.4100^{\dagger}$	

Table 2: Area ratios comparing GMP60 Static transfer against Dense baseline.

Table 2 reports the area ratios for sparse networks identified using GMP on the source task and transferred without further pruning to the target task, compared against the dense baseline. All source-target combinations yield negative area ratios, indicating that GMP-based sparse networks also fail to transfer effectively across environments.

3.2.3 LTH vs. GMP

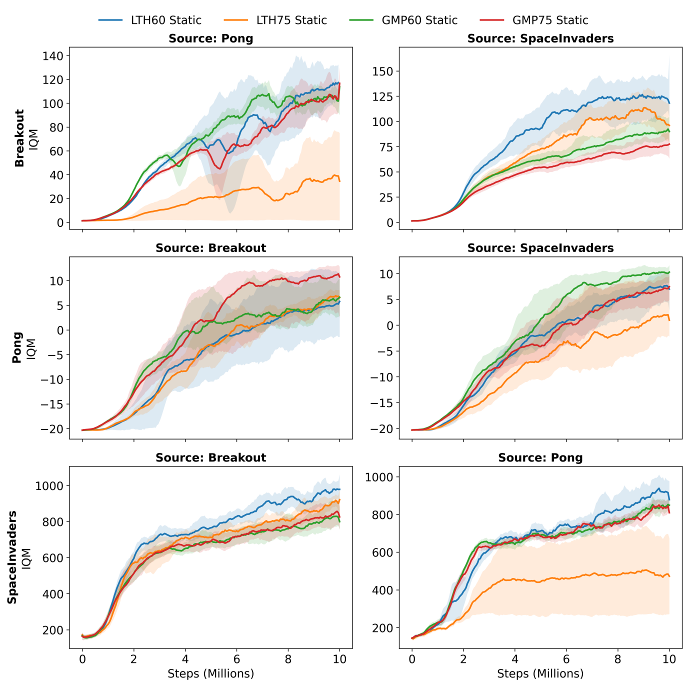


Figure 3: Comparison of Static LTH and Static GMP transfer conditions across source–target pairs. While individual source–target combinations exhibit differences in early learning dynamics or stability, neither pruning strategy consistently outperforms the other in terms of final performance across all environments.

Figure 3 compares the training curves of the LTH Static and GMP Static transfer conditions. The results are mixed: neither pruning approach consistently outperforms the other across all source-target combinations, suggesting that the choice of static magnitude-based pruning strategy alone is not sufficient to produce positive transfer.

4 Discussion, Limitations, and Future Work

This work set out to investigate whether sparse subnetworks discovered on a source reinforcement learning task can transfer to related target tasks, and whether reintroducing sparsification during target task training can improve such transfer by restoring network plasticity. Across all ablations, our results provide a clear and consistent answer to both parts of this research question. Sparse subnetworks identified on a source task do not transfer effectively across Atari environments, and dynamic sparsification during target task training further degrades performance relative to Static transfer. Together, these findings provide strong evidence against our hypothesis that sparsity-driven plasticity can recover positive transfer in value-based deep reinforcement learning.

Beyond individual comparisons, several systematic trends emerge. First, higher initial sparsity on the target task leads to progressively worse transfer performance across environments. Second, Static transfer consistently outperforms configurations that further increase sparsity during adaptation. These patterns suggest that, while sparse subnetworks may encode task-relevant structure for the source environment, this structure is either insufficiently general or only marginally useful for the target task. Increasing sparsity further reduces network capacity, limiting the model's ability to form new task-specific representations during adaptation and exacerbating negative transfer.

Our findings align with prior observations by Sabatelli and Geurts (2021), who report limited or negative transfer when transferring convolutional encoders between Atari tasks. Their conjecture—that transfer breaks the co-adaptation between the feature extractor and the value head—provides a compelling explanation for our results. In our setting, this issue is likely amplified by sparsity. Pruning-induced constraints reduce representational flexibility in the encoder, while the reinitialized value head must adapt to a new reward structure. The combination of broken co-adaptation and reduced capacity appears to hinder the realignment between representations and value estimation required for successful transfer. In this light, sparsity does not merely fail to help transfer, but actively worsens the underlying mismatch.

While our results are consistent and robust across ablations, several limitations must be acknowledged. Our evaluation is restricted to three Atari environments and value-based DDQN agents, and our conclusions therefore apply primarily to this setting. Although we expect similar trends to hold across additional Atari environments, it would be inappropriate to generalize these findings to other architectures, algorithms, or continuous-control domains without further empirical evidence. Moreover, our study focuses exclusively on magnitude-based pruning methods. Alternative sparsification techniques—such as structured pruning, gradient-based pruning, or learned sparsity mechanisms—may interact differently with transfer and remain an open area for exploration.

Importantly, our results suggest that the core limitation lies not in sparsity per se, but in how representations are transferred and adapted. Future work should therefore shift focus away from sparsity-driven transfer alone and toward mechanisms that explicitly preserve or restore encoder–head co-adaptation. One potential direction is to investigate partial or structured transfer of the value head, rather than full reinitialization, potentially through action-space alignment methods. Such approaches may retain critical representational couplings while still accommodating differing action spaces.

Another concrete direction is to combine sparse transfer with interpretability or representation analysis techniques to directly measure representational mismatch between source and target tasks. For example, probing activations, measuring representation similarity, or analyzing sparsity patterns across layers could help identify which aspects of sparse subnetworks fail to generalize. This, in turn, could inform targeted interventions such as selective layer transfer or hybrid dense–sparse adaptation strategies.

Finally, extending this investigation to multi-task or continual learning settings—where encoder–head co-adaptation is preserved across tasks—may help clarify why dynamic sparsity improves robustness in those regimes but fails in transfer. Understanding this distinction could bridge the gap between successful sparsity-driven plasticity in multi-task learning and its failure in cross-task transfer.

In summary, our results demonstrate that magnitude-based sparsity-driven transfer is not a promising approach for improving transfer performance in value-based deep reinforcement learning. Rather than enabling generalization, sparsity exacerbates representational misalignment and capacity limitations. Effective transfer in RL likely requires interventions that directly address encoder–head co-adaptation, rather than relying on sparsity as a proxy for plasticity.

5 Supplementary Materials / Appendix

This appendix reports the full set of hyperparameters and experimental settings used in our experiments.

5.1 DDQN Training Hyperparameters

Table 3: DDQN training hyperparameters used across all Atari experiments.

Parameter	**Value**
Algorithm	Double Deep Q-Network (DDQN)
Discount factor γ	0.99
Optimizer	Adam
Learning rate	1.0×10^{-4}
Adam ϵ	1.0×10^{-8}
Batch size	32
Replay buffer size	1.0×10^{6}
Learning starts	80,000 steps
Training frequency	Every 4 environment steps
Target network update frequency	1,000–5,000 steps
Gradient clipping norm	10.0
Hidden layer dimension	512
Network head	Linear

5.2 Exploration Schedule

Table 4: Epsilon-greedy exploration parameters.

Parameter	**Value**
Initial ϵ	1.0
Final ϵ	0.01 (0.05 for some baselines)
Decay schedule	Linear
Decay fraction	0.1–0.2 of total timesteps

5.3 Atari Environment and Preprocessing

Table 5: Atari environment configuration and observation preprocessing.

Setting	**Value**
Environment suite	Arcade Learning Environment (ALE v5)
Frame stacking	4 frames
Observation resolution	84×84
Grayscale observations	Yes
Reward clipping	Yes
No-op starts	Up to 30 steps
Observation scaling	Disabled (handled by network)
Number of parallel environments	1

5.4 Lottery Ticket Hypothesis (LTH) Pruning Hyperparameters

Table 6: Hyperparameters for Lottery Ticket Hypothesis (LTH) pruning.

Parameter	**Value**
Pruning method	Iterative magnitude pruning
Final sparsity	0.60, 0.75
Pruning rate per iteration	0.10
Rewind step	80,000 steps
Pruned layers	Encoder only
Weight reset	Yes (to rewind initialization)
Ticket evaluation window	IQM over last 20 evaluations

5.5 Gradual Magnitude Pruning (GMP) Hyperparameters

Table 7: Hyperparameters for Gradual Magnitude Pruning (GMP).

Parameter	**Value**
Pruning method	Gradual magnitude pruning
Initial sparsity	0.0
Final sparsity	0.60, 0.75, 0.90
Pruning start step	1,000,000
Pruning end step	6,000,000
Pruning update frequency	1,000–10,000 steps
Sparsity scheduler	Cubic
Pruned layers	Encoder only
Weight reset	No

5.6 Training Budget and Evaluation

Table 8: Training budget and evaluation protocol.

Parameter	**Value**
Total training timesteps	10,000,000
Evaluation metric	Interquartile Mean (IQM) of episodic return
Confidence intervals	95% bootstrap
Number of random seeds	5 (GMP), 3–5 (LTH)
Checkpoint interval	100,000 steps
Logging interval	100 steps

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47: 253–279, 2013.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

P. Ganesh, G. Tirumala, A. Singh, et al. One ticket to win them all: Generalizing lottery ticket initializations across datasets and optimizers. *Advances in Neural Information Processing Systems*, 2023.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in Neural Information Processing Systems*, 2, 1990.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, Marc G. Bellemere, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, feb 2015. ISSN 0028-0836. URL http://dx.doi.org/10.1038/nature14236.

Julio Obando-Ceron, Pablo Samuel Castro, and Marc G. Bellemare. Pruning deep reinforcement learning networks without performance loss. In *International Conference on Learning Representations*, 2023.

Matthia Sabatelli and Pierre Geurts. On the transferability of winning tickets in reinforcement learning. *arXiv preprint arXiv:2110.02639*, 2021.

Aleksandar Todorov, Juan Cardenas-Cartagena, Rafael F. Cunha, Marco Zullich, and Matthia Sabatelli. Sparsity-driven plasticity in multi-task reinforcement learning. *Transactions on Machine Learning Research*, 2025.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Michael Zhu and Suyog Gupta. To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *International Conference on Learning Representations (Workshop Track)*, 2017.