Sparsity-Driven Plasticity in Multi-Task Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Plasticity loss, a diminishing capacity to adapt as training progresses, is a critical challenge in deep reinforcement learning. We examine this issue in multi-task reinforcement learning (MTRL), where higher representational flexibility is crucial for managing diverse and potentially conflicting task demands. This paper specifically explores gradual magnitude pruning as a mechanism to enhance plasticity and consequently improve performance in MTRL agents. We systematically evaluate this approach across distinct MTRL architectures, including shared backbones with task-specific heads, Mixture of Experts (MoE), and Mixture of Orthogonal Experts (MOORE) on standardized MiniGrid benchmarks, comparing against dense baselines and alternative plasticity-inducing techniques. Our results demonstrate that pruning effectively mitigates key indicators of plasticity degradation, such as neuron dormancy and representational collapse. Consequently, these plasticity improvements from pruning directly correlate with enhanced multi-task performance, with sparse agents often outperforming both dense counterparts and alternative methods designed to induce plasticity. We further show that the benefits and specific dynamics induced by pruning are architecture-dependent, offering insights into the interplay between plasticity, network sparsity, and specific MTRL designs.

1 Introduction

Although deep reinforcement learning (DRL) agents have shown impressive results in a variety of applications (Levine et al., 2016; Silver et al., 2017; Bellemare et al., 2020; Mathieu et al., 2023), these achievements come with notable trade-offs. Attaining state-of-the-art performance often relies on large-scale computational resources and heavily overparameterized models (Botvinick et al., 2019; Glanois et al., 2022; Thompson et al., 2022), which may lead to agents that either generalize poorly (Kirk et al., 2023) or struggle to adapt to new tasks or data over time. The former issue is a topic of interest within the transfer learning literature (Farebrother et al., 2020; Sabatelli & Geurts, 2021; Sasso et al., 2023; Zhu et al., 2023), while the latter is commonly referred to as plasticity loss (Nikishin et al., 2022; Lyle et al., 2023; Dohare et al., 2024). Plasticity loss manifests through several interconnected optimization pathologies: gradient interference leading to premature convergence (Lyle et al., 2024a), representational collapse, limiting the diversity of learned features (Moalla et al., 2024), and neuronal saturation or dormancy that reduces effective network capacity (Bjorck et al., 2021; Sokar et al., 2023). While these challenges have been primarily investigated within single-task RL, (Nikishin et al., 2023; Abbas et al., 2023; Klein et al., 2024; Nauman et al., 2024a; Dohare et al., 2024), in this paper, we study them under the lens of multi-task reinforcement learning (MTRL), where maintaining representational flexibility across diverse tasks with potentially conflicting demands is even more crucial (Teh et al., 2017; Sodhani et al., 2021; D'Eramo et al., 2024). It naturally follows that this increased need for dynamic adaptation can make MTRL agents especially vulnerable to plasticity loss, as networks must simultaneously accommodate varied objectives without experiencing negative task interference (Liu et al., 2023). The necessity of determining which knowledge to share across tasks, and how to share it without harmful interference, further complicates the learning process (Devin et al., 2016). Moreover, this challenge can be even further exacerbated by inefficient use of network capacity (Kumar et al., 2021), with significant

portions of large networks becoming underutilized during training, ultimately hindering the acquisition of a universal policy capable of addressing multiple tasks concurrently.

Recent work in neural network pruning offers a promising direction beyond mere compression, showing that sparse agents can match or exceed dense counterparts in single-task RL (Livne & Cohen, 2020; Graesser et al., 2022; Obando-Ceron et al., 2024). Notably, pruning has shown positive effects on both single-task performance and plasticity (Obando-Ceron et al., 2024), suggesting it could address the optimization pathologies that undermine effective multi-task learning. Nonetheless, its impact on MTRL, where representational flexibility demands are significantly higher (Devin et al., 2016), remains largely unexplored.

This paper investigates whether gradual magnitude pruning enhances plasticity in MTRL agents, thereby improving performance across multiple tasks simultaneously. We evaluate this on Proximal Policy Optimization (PPO) (Schulman et al., 2017) agents with various multi-task architectures, including shared backbones with task-specific heads (MTPPO), Mixture of Experts (MoE) (Ceron et al., 2024), and Mixture of Orthogonal Experts (MOORE) (Hendawy et al., 2024), using MiniGrid (Chevalier-Boisvert et al., 2023), a collection of partially observable environments with sparse reward. Our central aim is to understand if the benefits of pruning can be primarily attributed to the mitigation of key plasticity loss indicators.

Our main contributions are threefold:

- We establish that gradual magnitude pruning, even at sparsity levels as high as 95%, serves as an effective mechanism for mitigating key indicators of plasticity loss in MTRL, with the most pronounced benefits observed in specific architectural configurations.
- We empirically show that pruning-induced plasticity improvements directly correlate with enhanced multi-task performance, often outperforming both dense baselines and alternative methods specifically designed to induce plasticity.
- We demonstrate that the effect of pruning on plasticity and performance is architecture-dependent, revealing insights into when and why sparsity interventions can complement existing architectural configurations for multi-task learning.

These findings highlight the potential role of network sparsity for developing adaptable and efficient RL agents for complex, real-world scenarios demanding generalization, continuous learning, and resource efficiency (Thompson et al., 2022).

2 Background

This section provides the necessary context for our approach. We begin by outlining the mathematical preliminaries underlying our framework, including key concepts and notations from reinforcement learning. Subsequently, we review related work, focusing on recent advances in sparsity in deep reinforcement learning, plasticity loss, and multi-task learning.

2.1 Preliminaries

We consider the Partially Observable Markov Decision Process (POMDP), defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$, consisting of a state space $\mathcal{S}$, an action space $\mathcal{A}$, transition dynamics $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, observation space Ω, observation probability function $\mathcal{O} : \mathcal{S} \times \mathcal{A} \to \Delta(\Omega)$, and discount factor $\gamma \in [0, 1)$. At each timestep t, the agent is situated in the true state $s_t \in \mathcal{S}$ and performs an action $a_t \in \mathcal{A}$. This causes the agent to transition to a new state $s_{t+1} \in \mathcal{S}$, receiving an observation $o_{t+1} \in \Omega$, and a reward $r_{t+1} = \mathcal{R}(s_t, a_t)$. The objective is to learn a policy $\pi_\theta(a_t|o_t)$ with parameters θ that maximizes the expected sum of discounted future rewards $J(\theta)$. In MTRL, the agent must learn a policy for a distribution of tasks $\mathcal{T}$. We adopt the Block Contextual POMDP framework (Sodhani et al., 2021; Hendawy et al., 2024), defined as $(\mathcal{C}, \mathcal{S}, \mathcal{A}, \mathcal{M}')$, where $\mathcal{C}$ represents the contextual space such that $c \in \mathcal{C}$ identifies a specific task $\tau \sim \mathcal{T}$. The mapping $\mathcal{M}'(c)$ provides the task-specific POMDP components $\{\mathcal{R}^c, \mathcal{P}^c, \mathcal{S}^c, \Omega^c, \mathcal{O}^c, \gamma^c\}$. The policy is now conditioned on the current observation $o \in \Omega^c$ and task context $c \in \mathcal{C}$. The objective is to maximize the expected return across all tasks $\mathbb{E}_{\tau \sim \mathcal{T}}[J_\tau(\theta)]$.

2.2 Related Work

Sparsity in Reinforcement Learning Recent work challenges the necessity of large, overparameterized networks in DRL, showing that sparse networks can achieve comparable or superior performance Livne & Cohen (2020); Graesser et al. (2022). This suggests that standard DRL agents may not fully utilize their capacity (Kumar et al., 2021) or may overfit to initial experiences (Nikishin et al., 2022), hence benefiting from sparsity. Neural network pruning can act as a regularizer (Jin et al., 2022), potentially yielding more robust policies. Our work focuses on Dense-to-Sparse Training, where connections in an initially dense network are gradually removed. This approach is motivated by the fact that gradual pruning schedules (Zhu & Gupta, 2017) have proven effective in RL (Obando-Ceron et al., 2024; Graesser et al., 2022), often matching or exceeding more complex pruning methods (Gale et al., 2019).

Plasticity Loss Reinforcement learning's inherent non-stationarity, arising from evolving policies and value estimates, can lead to plasticity loss: a reduced capacity of the network to adapt to new information (Lyle et al., 2022; Dohare et al., 2024). This can cause premature performance plateaus and instability (Igl et al., 2021; Berariu et al., 2023; Klein et al., 2024; Lyle et al., 2024b). This non-stationarity in learning targets is considered the primary driver, leading to challenging optimization landscapes (Dohare et al., 2022; Lyle et al., 2023). Associated pathologies include gradient collinearity (Lyle et al., 2024a), representational collapse (Moalla et al., 2024), and neuron saturation or dormancy (Bjorck et al., 2021; Sokar et al., 2023). While methods like network resets (Nikishin et al., 2022; 2023; Sokar et al., 2023), regularization, or modifying learning objectives (Farebrother et al., 2024) have been proposed, (Obando-Ceron et al., 2024) demonstrated that gradual pruning outperformed various plasticity-inducing methods in single-task settings, aligning with the view that general-purpose regularization might be more effective than methods explicitly designed to combat plasticity loss (Klein et al., 2024; Nauman et al., 2024a).

Multi-Task Reinforcement Learning MTRL aims to train agents on multiple tasks, facing the challenge of balancing shared learning for positive transfer against task interference. Common multi-task strategies include shared model parameters with task-specific components (Teh et al., 2017), modular architectures (Yang et al., 2020), addressing varying reward scales (Hessel et al., 2018), compositional policies (Sun et al., 2022), gradient projection techniques (Yu et al., 2020), and MoE models (Ceron et al., 2024), often with modifications such as attention-based (Cheng et al., 2023) or orthogonalized (Hendawy et al., 2024) experts. Unlike in some domains, simply scaling up model size in MTRL does not guarantee better performance and can be detrimental (Hansen et al., 2023; Ceron et al., 2024; Nauman et al., 2024b). While pruning enhances single-task plasticity and performance, and is complementary to scaling (Obando-Ceron et al., 2024), its role in MTRL is not yet systematically explored.

3 Experimental Setup

Our experiments compare the effects of different pruning levels against dense baselines and alternative plasticity-enhancing methods (ReDo (Sokar et al., 2023), Reset (Nikishin et al., 2022), and Weight Decay applied to the dense agent) across three multi-task architectures, specifically MTPPO, MoE (Ceron et al., 2024), and MOORE (Hendawy et al., 2024). Unless otherwise specified, we report the normalized interquantile mean (IQM) of the episode return from 30 trials. To ensure fair comparison across tasks with inherently different reward scales, raw episodic returns are normalized with respect to the maximum achievable reward in each environment (see Appendix B.3). Shaded regions indicate 95% stratified bootstrap confidence intervals, calculated using the `rliable` library (Agarwal et al., 2021).

Environment and Benchmarks We consider the three multi-task MiniGrid (Chevalier-Boisvert et al., 2023) benchmarks proposed by Hendawy et al. (2024) – MT3, MT5, and MT7. All environment details are outlined in Appendix B.

Implementation and Training We use the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) via the `mushroom_rl` library (D'Eramo et al., 2021) and the code provided by Hendawy et al. (2024) for multi-task architectures. We outline training details and hyperparameters in Appendix A. Performance is measured by the episodic return across all tasks within the respective benchmark. Tasks are sampled

randomly with replacement at the beginning of each episode during training. Agents are evaluated every 2000 steps throughout training for 25 episodes per task. All code is publicly available.[1]

Pruning Procedure Following the procedure described by Zhu & Gupta (2017) and its application in single-task RL (Graesser et al., 2022; Obando-Ceron et al., 2024), we progressively increase the network's sparsity level ρ_t from 0 to a final target sparsity ρ_F over a predefined range of training timesteps $[t_{\text{start}}, t_{\text{end}}]$. The sparsity ρ_t at timestep t is defined by:

$$\rho_t = \begin{cases} 0 & \text{if } t < t_{\text{start}}, \\ \rho_F \left[1 - \left(1 - \frac{t - t_{\text{start}}}{t_{\text{end}} - t_{\text{start}}} \right)^3 \right] & \text{if } t_{\text{start}} \leq t \leq t_{\text{end}}, \\ \rho_F & \text{if } t > t_{\text{end}}. \end{cases}$$

Every 500 timesteps during training, we mask (set to zero) network parameters with the smallest magnitudes. We determine the number of parameters to be masked so that the ρ_t sparsity goal is reached. Pruning starts after 5% and ends at 80% of the total training timesteps for each benchmark. We note that we chose to work with unstructured pruning as achieving a speed-up in inference is not the main goal of our work, therefore, making this approach more suitable over its structured version (Hoefler et al., 2021).

Plasticity Measures We monitor three metrics during training, interpreted as correlative indicators of plasticity based on recent surveys and analyses (Berariu et al., 2023; Lyle et al., 2023; Klein et al., 2024; Falzari & Sabatelli, 2025), namely dormant neuron percentage, effective rank, and the trace of the Fisher Information Matrix. The computation of these metrics is detailed in Appendix C. Our analysis focuses on observing consistent patterns between pruning interventions, changes in these metrics, and MTRL performance, rather than claiming direct causality.

4 Results

This section details our empirical findings, beginning with the impact of gradual pruning on multi-task performance, followed by an analysis of its effects on plasticity indicators, and concluding with a comparison against alternative plasticity-inducing methods. All performance comparisons refer to the aggregated final outcomes presented in Table 1. Plasticity metric analyses are primarily illustrated using the MT3 benchmark, as trends were generally consistent across other benchmarks unless otherwise stated. Detailed learning curves and plasticity metrics are available in Appendix D and Appendix E, respectively.

4.1 Gradual Magnitude Pruning Improves Task Performance

Our findings indicate that gradual magnitude pruning generally leads to improvements in multi-task performance, an observation consistent with a significant body of research in supervised learning, where appropriately pruned sparse networks have been shown to match, outperform, and generalize better than their dense counterparts (Guo et al., 2019; Morcos et al., 2019; Hoefler et al., 2021). In our multi-task settings, the extent of these benefits from pruning varies with the underlying agent architecture and desired sparsity level.

For MTPPO and MoE architectures, pruning consistently resulted in improved final aggregate returns compared to their respective dense baselines across all tested benchmarks (MT3, MT5, MT7), as shown in Table 1. Higher sparsity levels often yielded the most substantial performance gains in these configurations, suggesting that these common MTRL architectures frequently contain considerable overparameterization that pruning can effectively address. This hints at a direct link between pruning intervention and improved MTRL outcomes.

In contrast, the impact of pruning on MOORE was more nuanced and dependent on benchmark-specific network configurations. On the MT3 benchmark, pruned agents, which had narrower networks, exhibited a decline in performance compared to the dense baseline, characterized by a *rank collapse* (see Section 4.2). On the MT5 and MT7 benchmarks, which feature wider actor and critic networks, pruned agents achieved

[1]The source code used in this study will be made publicly available upon publication.

performance comparable to the dense baseline, without substantial gains, with rank collapse observed in the 99% sparsity configuration. Nonetheless, even though the performance differences were negligible on those benchmarks, the ability to prune to high levels of sparsity (up to 95%) still indicates that even sophisticated architectures are significantly overparameterized.

Table 1: Final aggregate performance at epoch 100 across architectures and agent treatments on MT3, MT5, and MT7 benchmarks. **Gold** marks the best performance within the respective multi-task architecture and benchmark, while **blue** marks an improvement over the dense counterpart of each. Full learning curves illustrating training progression are available in Appendix D.

	MT3		MT5		MT7	
Agent Treatment	**IQM (↑)**	**95% CI**	**IQM (↑)**	**95% CI**	**IQM (↑)**	**95% CI**
Multi-Task PPO (MTPPO)						
Dense	0.57	(0.50, 0.65)	0.59	(0.51, 0.66)	0.62	(0.56, 0.67)
80% Sparsity	**0.58**	(0.50, 0.65)	**0.66**	(0.61, 0.71)	**0.64**	(0.59, 0.70)
95% Sparsity	**0.61**	(0.55, 0.66)	**0.64**	(0.60, 0.69)	**0.71**	(0.66, 0.75)
99% Sparsity	**0.62**	(0.57, 0.66)	**0.68**	(0.61, 0.73)	**0.67**	(0.62, 0.72)
ReDo	0.57	(0.53, 0.60)	**0.68**	(0.62, 0.73)	**0.68**	(0.64, 0.73)
Reset	0.56	(0.51, 0.60)	**0.71**	(0.67, 0.74)	**0.69**	(0.65, 0.73)
Weight Decay	0.55	(0.49, 0.62)	0.55	(0.48, 0.61)	**0.64**	(0.58, 0.70)
Mixture of Experts (MoE)						
Dense	0.70	(0.63, 0.74)	0.74	(0.69, 0.79)	0.70	(0.65, 0.75)
80% Sparsity	0.69	(0.63, 0.73)	0.74	(0.69, 0.80)	0.68	(0.63, 0.73)
95% Sparsity	**0.71**	(0.66, 0.73)	**0.78**	(0.70, 0.83)	**0.74**	(0.72, 0.77)
99% Sparsity	0.60	(0.57, 0.64)	0.71	(0.63, 0.76)	**0.74**	(0.69, 0.79)
ReDo	0.67	(0.64, 0.70)	0.72	(0.67, 0.78)	**0.77**	(0.74, 0.80)
Reset	0.61	(0.54, 0.67)	0.70	(0.66, 0.73)	0.70	(0.66, 0.74)
Weight Decay	**0.71**	(0.64, 0.76)	0.72	(0.67, 0.78)	**0.71**	(0.67, 0.75)
Mixture of Orthogonal Experts (MOORE)						
Dense	**0.75**	(0.67, 0.77)	0.81	(0.76, 0.85)	0.81	(0.76, 0.84)
80% Sparsity	0.54	(0.47, 0.58)	0.81	(0.76, 0.83)	0.80	(0.77, 0.83)
95% Sparsity	0.47	(0.44, 0.52)	**0.82**	(0.78, 0.84)	**0.82**	(0.79, 0.86)
99% Sparsity	0.47	(0.42, 0.53)	0.69	(0.64, 0.73)	0.70	(0.68, 0.72)
ReDo	0.67	(0.62, 0.70)	0.79	(0.75, 0.81)	**0.82**	(0.80, 0.84)
Reset	0.58	(0.53, 0.66)	0.72	(0.66, 0.76)	0.75	(0.71, 0.79)
Weight Decay	**0.75**	(0.70, 0.77)	**0.83**	(0.79, 0.85)	0.78	(0.74, 0.82)

4.2 Pruning Mitigates Plasticity Loss

The observed performance improvements, particularly within the MTPPO and MoE architectures, strongly correlate with pruning's mitigation of common indicators of plasticity loss, while displaying distinct learning dynamics. Notably, Figure 1 shows that MTPPO and MoE agents pruned to high levels of sparsity maintained substantially lower percentages of dormant neurons, particularly within the actor network components, and a maintenance of higher mean effective rank in representations compared to dense counterparts. Furthermore, the trace of the Fisher Information Matrix (FIM) in pruned agents typically stabilized at lower values post-initial learning, unlike the persistently high trace in dense agents, suggesting convergence to less sensitive parameter configurations in the later stages of training. Collectively, these observations regarding plasticity indicators support the hypothesis that gradual magnitude pruning enhances the learning capability of MTRL agents, plausibly through the mitigation of processes associated with plasticity degradation in dense networks.

While plasticity improvements were observed in MTPPO and MoE agents, the effect in MOORE was less pronounced and benchmark-specific, a trend consistent with the varied performance as detailed in Section 4.1 On the MT3 benchmark, the performance degradation in pruned MOORE agents was highly correlated with a deterioration in plasticity metrics, characterized by a sudden drop in the mean effective rank and a significant increase in the FIM, and the percentage of dormant neurons both in the actor and critic components. This implies that aggressive pruning in MOORE may have removed parameters vital for maintaining distinct expert representations. Conversely, on the MT5 and MT7 (see Appendix E, Figure 6 and Figure 7) benchmarks, plasticity metrics did not show significant changes compared to the dense baseline and sometimes even degraded, with rank collapse being observed in the 99% sparsity configuration. This suggests that MOORE's inherent design, particularly its emphasis on representation orthogonalization, may already address some aspects of plasticity, which we showed to be influenced by pruning. Consequently, dense MOORE agents already exhibited stable plasticity characteristics, highlighting that the utility of pruning as a plasticity enhancer is highly dependent on the specific mechanisms within the network architecture.

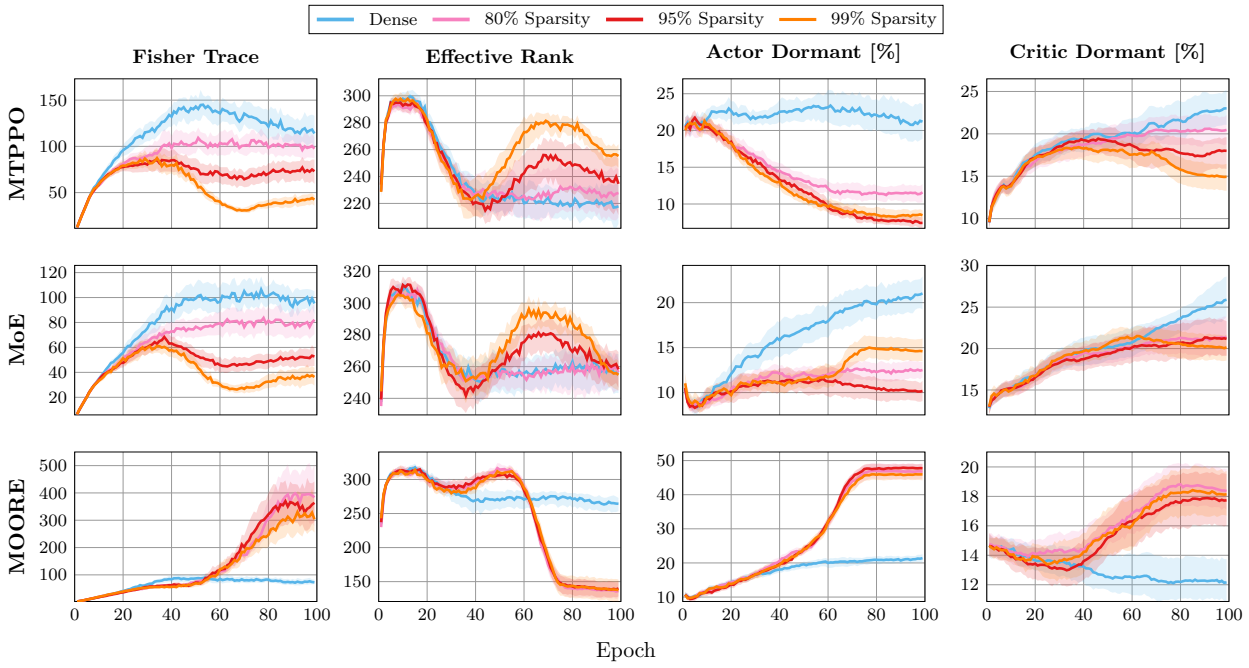

Figure 1: Gradual magnitude pruning positively influences plasticity markers in MTPPO and MoE architectures in MT3. Increasing sparsity (especially 95-99%) generally correlates with a lower Fisher Trace peak, sustained or recovered Effective Rank, and notably reduced Actor Dormancy compared to Dense baselines. In contrast, MOORE shows distinct dynamics, with pruning leading to plasticity degradation (e.g., rank collapse and increased FIM trace), aligning with its performance on this benchmark.

4.3 Pruning versus Alternative Plasticity-Inducing Methods

Our investigation reveals that gradual magnitude pruning frequently provides a more effective path to enhanced MTRL performance, correlating with distinct and consistent patterns in terms of plasticity when compared to established plasticity-inducing interventions like ReDo, Reset, and Weight Decay.

In terms of task performance, pruned agents, particularly MTPPO and MoE architectures, generally achieved better or comparable final returns to those employing alternative methods (Table 1). This aligns with the broader observation in deep learning that general regularization techniques can outperform methods aimed at specific symptoms of suboptimal learning (Nauman et al., 2024a; Klein et al., 2024). For MOORE agents, while some alternatives occasionally led to slightly higher performance (Weight Decay on MT5; ReDo on

MT7), these differences were generally small, similar to the modest improvements observed with pruning on these benchmarks for MOORE. This further suggests MOORE's architecture inherently addresses some optimization challenges, making additional interventions less impactful compared to MTPPO and MoE.

Examining the plasticity profiles, illustrated in Figure 2, all methods exhibited distinct dynamics. For instance, Reset, by its nature, tended to introduce instability in plasticity markers like the FIM and mean effective rank, particularly post-reset, consistent with previous work (Falzari & Sabatelli, 2025). Notably, even when ReDo consistently reduced the percentage of dormant neurons to very low levels across all architectures, this often did not inherently translate to superior task performance, suggesting that pruning addresses more comprehensive underlying optimization issues rather than isolated pathologies. Weight decay, as a modification to optimizer dynamics, typically resulted in plasticity profiles and performance that were largely similar to the dense baseline.

Overall, these comparisons indicate that pruning, by simultaneously addressing plasticity loss and optimizing network capacity, often provides a more robust and effective means of enhancing MTRL agent performance than alternative methods that may target plasticity through more isolated mechanisms.

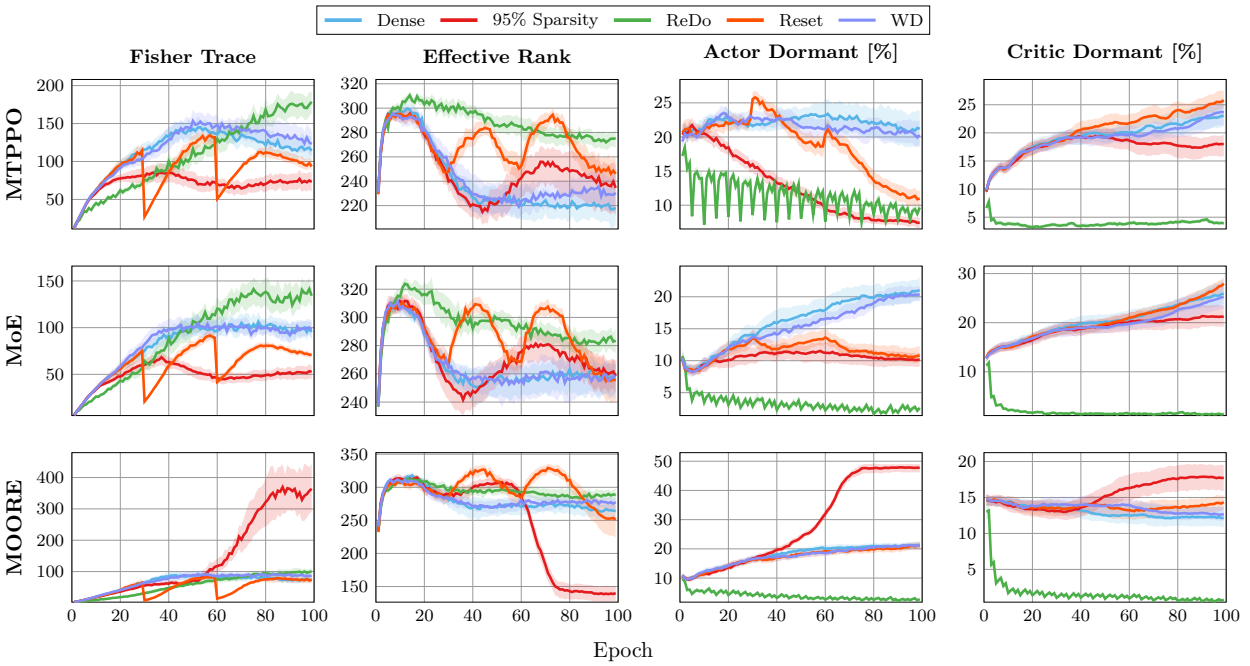

Figure 2: Comparative plasticity dynamics of 95% Sparsity Pruning versus alternative interventions in MT3. For MTPPO and MoE, ReDo achieves notably lower neuron dormancy and often a higher mean effective rank than 95% pruning. Pruning, in contrast, offers a distinct profile that correlates strongly with performance improvements, differing from the oscillations induced by Reset or the minimal impact of Weight Decay relative to the Dense baseline. For MOORE, 95% pruning degrades plasticity compared to its relatively stable dense baseline; other interventions show less detrimental effects.

5 Considerations of Pruning Efficacy

While this study is primarily empirical, the observed benefits of gradual magnitude pruning in MTRL can be interpreted through several established concepts from the sparsity, optimization, and multi-task learning literature.

Optimization Perspective Gradual magnitude pruning, through its iterative removal of low-magnitude weights, effectively guides the network towards sparse solutions. This mechanism can generally be viewed

as L_0 regularization, which encourages sparsity by penalizing the number of non-zero parameters (Louizos et al., 2018). Such regularization, along with direct pruning, reduces the model's degrees of freedom, which can confine the optimization process to lower-dimensional subspaces where optimizers might not be able to navigate around landscape obstacles that would otherwise be circumventable with more dimensions (Gao & Jojic, 2016; Hoefler et al., 2021). Despite this, achieving sparse solutions is often associated with residing in "flatter" minima of the loss landscape (Peste, 2023; Shah et al., 2024). Flatter minima are a topic of interest as they are characterized by lower sensitivity to parameter perturbations (Foret et al., 2021; Lee et al., 2025) and are widely believed to result in better generalization and robustness to distribution shifts (Hochreiter & Schmidhuber, 1997; Jiang et al., 2019; Kaddour et al., 2023; Li et al., 2024), the associated primary driver of plasticity loss. The convergence to such local flat minima is often indicated by specific dynamics in the curvature of the loss – for instance, the maximal Hessian eigenvalue typically grows, peaks, and then declines during training (Fort & Ganguli, 2019). Aligning with this, our empirical results for MTPPO and MoE agents showed that the Fisher trace, a proxy for curvature (Lewandowski et al., 2024), exhibited a similar peak-and-decline pattern for the 95% and 99% sparsity configurations. This observation suggests that our pruned agents reached flatter local minima, consequently contributing to their improved performance. Nonetheless, an excessive reduction in degrees of freedom through aggressive pruning could also make it challenging to satisfy specific architectural demands, such as maintaining expert orthogonality in models like MOORE, particularly if the network's capacity becomes too constrained to adequately represent the required orthogonal subspaces.

Sparsity and Generalization Within supervised learning, sparse networks have been recognized for their reduced tendency to overfit, better handling of noisy data, and often generalizing better to their dense counterparts (Gopalakrishnan et al., 2018; Cosentino et al., 2019; Guo et al., 2019; Liu et al., 2019; Liu, 2020). The iterative nature of the pruning schedule employed in this study, proposed by Zhu & Gupta (2017), has been noted for its effectiveness in supervised learning, often outperforming more sophisticated pruning methods (Gale et al., 2019; Graesser et al., 2022). In general, iterative pruning schemes are thought to help models evade suboptimal local minima that standard optimizers might otherwise converge to (Jin et al., 2016; Hoefler et al., 2021), principles with which our empirical findings align. Moreover, the success of pruning strategies in models, including shared backbones and task-specific heads in multi-task learning, is analogous to the findings reported by Xiang et al. (2024). They, similarly to us, demonstrate the effectiveness of pruning schedules that implicitly adapt the sparsity distribution during training, followed by periods of continued training or fine-tuning, leading to enhanced learning outcomes.

6 Discussion and Conclusion

This work investigated gradual magnitude pruning as a means to mitigate plasticity loss and enhance performance in multi-task reinforcement learning (MTRL), building on successes in single-task RL (Graesser et al., 2022; Obando-Ceron et al., 2024). Our findings demonstrate pruning as an effective mechanism against plasticity degradation, improving MTRL agents across certain architectures. Pruned agents showed improved plasticity profiles (e.g., reduced dormancy, increased representation diversity, stable dynamics) and often outperformed dense baselines and other plasticity-focused methods. This aligns with the *bitter lesson* of plasticity (Nauman et al., 2024a; Klein et al., 2024), favoring general mechanisms impacting learning dynamics and resource use over methods targeting specific symptoms and effects of plasticity loss. Nonetheless, benefits were architecture-dependent: Multi-Task Proximal Policy Optimization and Mixture of Experts agents consistently benefited, while the impact on Mixture of Orthogonal Experts architectures was more variable and dependent on network capacity. This finding suggests that pruning is not a one-size-fits-all solution but rather a tool whose application requires consideration depending on the intrinsic properties of the network.

The findings in this study also naturally define several exciting directions for future research. Our evaluation on the MiniGrid benchmarks (Chevalier-Boisvert et al., 2023) proposed Hendawy et al. (2024), chosen for efficient experimentation, could be complemented by validation in more complex domains like Meta-World (Yu et al., 2021) to assess scalability. The current focus on gradual magnitude pruning also invites investigations into the array of sparsity techniques and their potential within MTRL. Alongside empirical extensions, developing a robust theoretical framework to precisely explain the interactions between sparsity, plasticity, and overall benefits in reinforcement learning would be invaluable. Finally, exploring whether pruning-induced

sparsity can enhance the interpretability of internal representations and decision-making processes, similar to benefits observed in other machine learning domains, presents another compelling research direction.

In conclusion, this paper provides empirical evidence that gradual magnitude pruning is an effective and architecture-sensitive technique for enhancing plasticity and improving overall performance in multi-task reinforcement learning. By mitigating common indicators of plasticity loss and promoting more efficient use of network capacity, pruning offers a robust approach to developing more adaptable MTRL systems. Our findings underscore the significant potential of network sparsity as a more general-purpose tool in the RL toolkit. This research opens the way for future investigations into more sophisticated sparsity methods and applications to more complex multi-task challenges, ultimately contributing to the creation of skilled, efficient, and potentially more understandable agents.

References

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C. Machado. Loss of Plasticity in Continual Deep Reinforcement Learning, March 2023. URL http://arxiv.org/abs/2303.07507. arXiv:2303.07507 [cs].

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep Reinforcement Learning at the Edge of the Statistical Precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

Marc G. Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C. Machado, Subhodeep Moitra, Sameera S. Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, December 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2939-8. URL https://www.nature.com/articles/s41586-020-2939-8. Publisher: Nature Publishing Group.

Tudor Berariu, Wojciech Czarnecki, Soham De, Jorg Bornschein, Samuel Smith, Razvan Pascanu, and Claudia Clopath. A study on the plasticity of neural networks, October 2023. URL http://arxiv.org/abs/2106.00042. arXiv:2106.00042 [cs].

Johan Bjorck, Carla P. Gomes, and Kilian Q. Weinberger. Is High Variance Unavoidable in RL? A Case Study in Continuous Control. October 2021. URL https://openreview.net/forum?id=9xhgmsNVHu.

Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement Learning, Fast and Slow. *Trends in Cognitive Sciences*, 23(5):408–422, May 2019. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2019.02.006. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(19)30061-0. Publisher: Elsevier.

Johan Samir Obando Ceron, Ghada Sokar, Timon Willi, Clare Lyle, Jesse Farebrother, Jakob Nicolaus Foerster, Gintare Karolina Dziugaite, Doina Precup, and Pablo Samuel Castro. Mixtures of Experts Unlock Parameter Scaling for Deep RL. June 2024. URL https://openreview.net/forum?id=X9VMhfFxwn.

Guangran Cheng, Lu Dong, Wenzhe Cai, and Changyin Sun. Multi-Task Reinforcement Learning With Attention-Based Mixture of Experts. *IEEE Robotics and Automation Letters*, 8(6):3812–3819, June 2023. ISSN 2377-3766. doi: 10.1109/LRA.2023.3271445. URL https://ieeexplore.ieee.org/document/10111062.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks, June 2023. URL http://arxiv.org/abs/2306.13831. arXiv:2306.13831 [cs].

Justin Cosentino, Federico Zaiter, Dan Pei, and Jun Zhu. The Search for Sparse, Robust Neural Networks, December 2019. URL http://arxiv.org/abs/1912.02386. arXiv:1912.02386 [cs].

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. MushroomRL: Simplifying Reinforcement Learning Research. *Journal of Machine Learning Research*, 22(131):1–5, 2021. ISSN 1533-7928. URL http://jmlr.org/papers/v22/18-056.html.

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Sharing Knowledge in Multi-Task Deep Reinforcement Learning, January 2024. URL http://arxiv.org/abs/2401.09561. arXiv:2401.09561 [cs].

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning Modular Neural Network Policies for Multi-Task and Multi-Robot Transfer, September 2016. URL http://arxiv.org/abs/1609.07088. arXiv:1609.07088 [cs].

Shibhansh Dohare, Richard S. Sutton, and A. Rupam Mahmood. Continual Backprop: Stochastic Gradient Descent with Persistent Randomness, May 2022. URL http://arxiv.org/abs/2108.06325. arXiv:2108.06325 [cs].

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Maintaining Plasticity in Deep Continual Learning, April 2024. URL http://arxiv.org/abs/2306.13812. arXiv:2306.13812 [cs].

Massimiliano Falzari and Matthia Sabatelli. Fisher-Guided Selective Forgetting: Mitigating The Primacy Bias in Deep Reinforcement Learning, February 2025. URL http://arxiv.org/abs/2502.00802. arXiv:2502.00802 [cs].

Jesse Farebrother, Marlos C. Machado, and Michael Bowling. Generalization and Regularization in DQN, January 2020. URL http://arxiv.org/abs/1810.00123. arXiv:1810.00123 [cs].

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop Regressing: Training Value Functions via Classification for Scalable Deep RL, March 2024. URL http://arxiv.org/abs/2403.03950. arXiv:2403.03950 [cs].

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-Aware Minimization for Efficiently Improving Generalization, April 2021. URL http://arxiv.org/abs/2010.01412. arXiv:2010.01412 [cs].

Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes, October 2019. URL http://arxiv.org/abs/1910.05929. arXiv:1910.05929 [cs].

Trevor Gale, Erich Elsen, and Sara Hooker. The State of Sparsity in Deep Neural Networks, February 2019. URL http://arxiv.org/abs/1902.09574. arXiv:1902.09574 [cs].

Tianxiang Gao and Vladimir Jojic. Degrees of freedom in deep neural networks. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, pp. 232–241, Arlington, Virginia, USA, June 2016. AUAI Press. ISBN 978-0-9966431-1-5.

Claire Glanois, Paul Weng, Matthieu Zimmer, Dong Li, Tianpei Yang, Jianye Hao, and Wulong Liu. A Survey on Interpretable Reinforcement Learning, February 2022. URL http://arxiv.org/abs/2112.13112. arXiv:2112.13112 [cs].

Soorya Gopalakrishnan, Zhinus Marzi, Upamanyu Madhow, and Ramtin Pedarsani. Combating Adversarial Attacks Using Sparse Representations, July 2018. URL http://arxiv.org/abs/1803.03880. arXiv:1803.03880 [stat].

Laura Graesser, Utku Evci, Erich Elsen, and Pablo Samuel Castro. The State of Sparse Training in Deep Reinforcement Learning, June 2022. URL http://arxiv.org/abs/2206.10369. arXiv:2206.10369 [cs].

Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse DNNs with Improved Adversarial Robustness, November 2019. URL http://arxiv.org/abs/1810.09619. arXiv:1810.09619 [cs].

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. October 2023. URL https://openreview.net/forum?id=Oxh5CstDJU.

Ahmed Hendawy, Jan Peters, and Carlo D'Eramo. Multi-Task Reinforcement Learning with Mixture of Orthogonal Experts, May 2024. URL http://arxiv.org/abs/2311.11385. arXiv:2311.11385 [cs].

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task Deep Reinforcement Learning with PopArt, September 2018. URL http://arxiv.org/abs/1809.04474. arXiv:1809.04474 [cs].

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks, January 2021. URL http://arxiv.org/abs/2102.00554. arXiv:2102.00554 [cs].

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient Non-Stationarity and Generalisation in Deep Reinforcement Learning, September 2021. URL http://arxiv.org/abs/2006.05826. arXiv:2006.05826 [cs].

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic Generalization Measures and Where to Find Them, December 2019. URL http://arxiv.org/abs/1912.02178. arXiv:1912.02178 [cs].

Tian Jin, Michael Carbin, Daniel M. Roy, Jonathan Frankle, and Gintare Karolina Dziugaite. Pruning's Effect on Generalization Through the Lens of Training and Regularization, October 2022. URL http://arxiv.org/abs/2210.13738. arXiv:2210.13738 [cs].

Xiaojie Jin, Xiaotong Yuan, Jiashi Feng, and Shuicheng Yan. Training Skinny Deep Neural Networks with Iterative Hard Thresholding Methods, July 2016. URL http://arxiv.org/abs/1607.05423. arXiv:1607.05423 [cs].

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J. Kusner. When Do Flat Minima Optimizers Work?, January 2023. URL http://arxiv.org/abs/2202.00661. arXiv:2202.00661 [cs].

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs].

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76:201–264, January 2023. ISSN 1076-9757. doi: 10.1613/jair.1.14174. URL http://arxiv.org/abs/2111.09794. arXiv:2111.09794 [cs].

Timo Klein, Lukas Miklautz, Kevin Sidak, Claudia Plant, and Sebastian Tschiatschek. Plasticity Loss in Deep Reinforcement Learning: A Survey, November 2024. URL http://arxiv.org/abs/2411.04832. arXiv:2411.04832 [cs].

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit Under-Parameterization Inhibits Data-Efficient Deep Reinforcement Learning, October 2021. URL http://arxiv.org/abs/2010.14498. arXiv:2010.14498 [cs].

Taehwan Lee, Kyeongkook Seo, Jaejun Yoo, and Sung Whan Yoon. Understanding Flatness in Generative Models: Its Role and Benefits, March 2025. URL http://arxiv.org/abs/2503.11078. arXiv:2503.11078 [cs] version: 1.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-End Training of Deep Visuomotor Policies, April 2016. URL http://arxiv.org/abs/1504.00702. arXiv:1504.00702 [cs].

Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C. Machado. Directions of Curvature as an Explanation for Loss of Plasticity, October 2024. URL http://arxiv.org/abs/2312.00246. arXiv:2312.00246 [cs] version: 4.

Aodi Li, Liansheng Zhuang, Xiao Long, Minghong Yao, and Shafei Wang. Seeking Consistent Flat Minima for Better Domain Generalization via Refining Loss Landscapes, December 2024. URL http://arxiv.org/abs/2412.13573. arXiv:2412.13573 [cs] version: 1.

Shiwei Liu. Learning Sparse Neural Networks for Better Generalization. volume 5, pp. 5190–5191, July 2020. doi: 10.24963/ijcai.2020/735. URL https://www.ijcai.org/proceedings/2020/735. ISSN: 1045-0823.

Shiwei Liu, Decebal Constantin Mocanu, and Mykola Pechenizkiy. On improving deep learning generalization with adaptive sparse connectivity, June 2019. URL http://arxiv.org/abs/1906.11626. arXiv:1906.11626 [cs].

Vincent Liu, Han Wang, Ruo Yu Tao, Khurram Javed, Adam White, and Martha White. Measuring and Mitigating Interference in Reinforcement Learning. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*, pp. 781–795. PMLR, November 2023. URL https://proceedings.mlr.press/v232/liu23a.html. ISSN: 2640-3498.

Dor Livne and Kobi Cohen. PoPS: Policy Pruning and Shrinking for Deep Reinforcement Learning. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):789–801, May 2020. ISSN 1932-4553, 1941-0484. doi: 10.1109/JSTSP.2020.2967566. URL http://arxiv.org/abs/2001.05012. arXiv:2001.05012 [cs].

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning Sparse Neural Networks through L_0 Regularization, June 2018. URL http://arxiv.org/abs/1712.01312. arXiv:1712.01312 [stat].

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and Preventing Capacity Loss in Reinforcement Learning, May 2022. URL http://arxiv.org/abs/2204.09560. arXiv:2204.09560 [cs].

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks, November 2023. URL http://arxiv.org/abs/2303.01486. arXiv:2303.01486 [cs].

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the Causes of Plasticity Loss in Neural Networks, February 2024a. URL http://arxiv.org/abs/2402.18762. arXiv:2402.18762 [cs].

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, James Martens, Hado van Hasselt, Razvan Pascanu, and Will Dabney. Normalization and effective learning rates in reinforcement learning, July 2024b. URL http://arxiv.org/abs/2407.01800. arXiv:2407.01800 [cs].

Michaël Mathieu, Sherjil Ozair, Srivatsan Srinivasan, Caglar Gulcehre, Shangtong Zhang, Ray Jiang, Tom Le Paine, Richard Powell, Konrad Żołna, Julian Schrittwieser, David Choi, Petko Georgiev, Daniel Toyama, Aja Huang, Roman Ring, Igor Babuschkin, Timo Ewalds, Mahyar Bordbar, Sarah Henderson, Sergio Gómez Colmenarejo, Aäron van den Oord, Wojciech Marian Czarnecki, Nando de Freitas, and Oriol Vinyals. AlphaStar Unplugged: Large-Scale Offline Reinforcement Learning, August 2023. URL http://arxiv.org/abs/2308.03526. arXiv:2308.03526 [cs].

Skander Moalla, Andrea Miele, Daniil Pyatko, Razvan Pascanu, and Caglar Gulcehre. No Representation, No Trust: Connecting Representation, Collapse, and Trust Issues in PPO, November 2024. URL http://arxiv.org/abs/2405.00662. arXiv:2405.00662 [cs].

Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers, October 2019. URL http://arxiv.org/abs/1906.02773. arXiv:1906.02773 [stat].

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, June 2024a. URL http://arxiv.org/abs/2403.00514. arXiv:2403.00514 [cs].

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, Regularized, Optimistic: scaling for compute and sample efficient continuous control. November 2024b. URL https://openreview.net/forum?id=fu0xdh4aEJ.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The Primacy Bias in Deep Reinforcement Learning, May 2022. URL http://arxiv.org/abs/2205.07802. arXiv:2205.07802 [cs].

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep Reinforcement Learning with Plasticity Injection, October 2023. URL http://arxiv.org/abs/2305.15555. arXiv:2305.15555 [cs].

Johan Obando-Ceron, Aaron Courville, and Pablo Samuel Castro. In value-based deep reinforcement learning, a pruned network is a good network, June 2024. URL http://arxiv.org/abs/2402.12479. arXiv:2402.12479 [cs].

Elena-Alexandra Peste. *Efficiency and generalization of sparse neural networks*. thesis, 2023. URL https://research-explorer.ista.ac.at/record/13074. ISSN: 2663-337X.

Matthia Sabatelli and Pierre Geurts. On The Transferability of Deep-Q Networks, November 2021. URL http://arxiv.org/abs/2110.02639. arXiv:2110.02639 [cs].

Remo Sasso, Matthia Sabatelli, and Marco A. Wiering. Multi-Source Transfer Learning for Deep Model-Based Reinforcement Learning, April 2023. URL http://arxiv.org/abs/2205.14410. arXiv:2205.14410 [cs].

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs].

Aditya Shah, Aditya Challa, Sravan Danda, Archana Mathur, and Snehanshu Saha. A Granger-Causal Perspective on Gradient Descent with Application to Pruning, December 2024. URL http://arxiv.org/abs/2412.03035. arXiv:2412.03035 [cs].

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, October 2017. ISSN 1476-4687. doi: 10.1038/nature24270. URL https://www.nature.com/articles/nature24270. Publisher: Nature Publishing Group.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-Task Reinforcement Learning with Context-based Representations, June 2021. URL http://arxiv.org/abs/2102.06177. arXiv:2102.06177 [cs].

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The Dormant Neuron Phenomenon in Deep Reinforcement Learning, June 2023. URL http://arxiv.org/abs/2302.12902. arXiv:2302.12902 [cs].

Lingfeng Sun, Haichao Zhang, Wei Xu, and Masayoshi Tomizuka. PaCo: Parameter-Compositional Multi-Task Reinforcement Learning, October 2022. URL http://arxiv.org/abs/2210.11653. arXiv:2210.11653 [cs].

Yee Whye Teh, Victor Bapst, Wojciech Marian Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust Multitask Reinforcement Learning, July 2017. URL http://arxiv.org/abs/1707.04175. arXiv:1707.04175 [cs].

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. The Computational Limits of Deep Learning, July 2022. URL http://arxiv.org/abs/2007.05558. arXiv:2007.05558 [cs].

Mingcan Xiang, Steven Jiaxun Tang, Qizheng Yang, Hui Guan, and Tongping Liu. AdapMTL: Adaptive Pruning Framework for Multitask Learning Model, August 2024. URL http://arxiv.org/abs/2408.03913. arXiv:2408.03913 [cs].

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-Task Reinforcement Learning with Soft Modularization, December 2020. URL http://arxiv.org/abs/2003.13661. arXiv:2003.13661 [cs].

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient Surgery for Multi-Task Learning, December 2020. URL http://arxiv.org/abs/2001.06782. arXiv:2001.06782 [cs].

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning, June 2021. URL http://arxiv.org/abs/1910.10897. arXiv:1910.10897 [cs].

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression, November 2017. URL http://arxiv.org/abs/1710.01878. arXiv:1710.01878 [stat].

Zhuangdi Zhu, Kaixiang Lin, Anil K. Jain, and Jiayu Zhou. Transfer Learning in Deep Reinforcement Learning: A Survey, July 2023. URL http://arxiv.org/abs/2009.07888. arXiv:2009.07888 [cs].

A Hyperparameters

This appendix details the hyperparameters used for the experimental evaluations presented in this study. Table 2 provides a list covering the general experimental settings, architecture of the used networks, and specific hyperparameters used for MoE and MOORE. All architectures are multi-headed, with task-specific heads. Hyperparameters were largely adopted from Hendawy et al. (2024), with the only modification being in the number of evaluation episodes per task.

B Environment Details

This appendix provides details on the MiniGrid (Chevalier-Boisvert et al., 2023) environments used in our multi-task benchmarks. We use standard environments from the MiniGrid suite, which are designed to test various capabilities such as navigation, memory, and problem-solving in partially observable grid-world settings with sparse reward.

B.1 Composition

Our experiments use three multi-task benchmarks – MT3, MT5, and MT7, as proposed by Hendawy et al. (2024), composed as follows:

- MT3: `LavaGapS7-v0 + RedBlueDoors-6x6-v0 + MemoryS11-v0`
- MT5: `MT3 + DoorKey-6x6-v0 + DistShift1-v0`
- MT7: `MT5 + SimpleCrossingS9N2 + MultiRoom-N2-S4`

B.2 Descriptions

Below are descriptions for each unique environment used in the benchmarks, adapted from Chevalier-Boisvert et al. (2023). In all environments `S` specifies the size of the map `SxS`.

- `DoorKey-6x6-v0`: The agent must pick up a key, navigate to a locked door, and open it to reach a goal square.
- `DistShift1-v0`: The agent starts in the top-left corner and must reach the goal, which is in the top-right corner, but has to avoid stepping into lava on its way. Stepping into lava terminates the episode.
- `RedBlueDoors-6x6-v0`: The agent is in a room with two doors, one red and one blue. The agent has to open the red door and then open the blue door, in that order.
- `MemoryS11-v0`: The agent starts in a small room where it sees an object. It then has to go through a narrow hallway, which ends in a split. At each end of the split, there is an object, one of which is the same as the object in the starting room. The agent has to remember the initial object and go to the matching object at the split.
- `SimpleCrossingS9N2-v0`: The agent has to reach the green goal square on the other corner of the room while avoiding walls. Walls run across the room either horizontally or vertically, and have `N` crossing points which can be safely used; the path to the goal is guaranteed to exist.
- `MultiRoom-N2-S4-v0`: This environment has a series of connected rooms with doors that must be opened to get to the next room. The final room has the green goal square that the agent must get to. `N` specifies the number of rooms.
- `LavaGapS7-v0`: The agent has to reach the green goal square at the opposite corner of the room, and must pass through a narrow gap in a vertical strip of deadly lava. Touching the lava terminates the episode with a zero reward.

Table 2: Core experimental setup, agent architecture, and algorithm hyperparameters. The choice for hyperparameters is largely borrowed from Hendawy et al. (2024), while following their exact training configuration (except number of evaluation episodes).

Hyperparameter	**Value**
General:	
Number of environments	[3, 5, 7]
Steps per environment	1 step per environment
Number of epochs	100
Steps per epoch	2000
Total number of timesteps	200000
Train frequency	2000 timesteps
Evaluation episodes	25 per task
Evaluation frequency	2000 timesteps
Shared Feature Extractor:	
Type	Conv2D
Channels per Layer	[16, 32, 64]
Kernel Size	[(2,2), (2,2), (2,2)]
Activations	[ReLU, ReLU, Tanh]
PPO:	
Optimizer	Adam (Kingma & Ba, 2017)
Critic Loss	MSE
Actor Learning Rate	1×10^{-3}
Critic Learning Rate	1×10^{-3}
Critic Network Hidden Size	128
Actor Network Hidden Size	128
Number of Linear Layers	$2 \times \lvert\mathcal{T}\rvert$ (number of tasks)
Number of Output Units	$\lvert\mathcal{A}\rvert$ for actor, 1 for critic
Output Activations	[Tanh, Linear]
GAE λ	0.95
Entropy Term Coefficient	0.01
Clipping ϵ	0.2
Epochs for Policy	8
Epochs for Critic	1
Batch Size for Policy	256
Batch Size for Critic	2000
Discount Factor (γ)	0.99
Task Encoder (for MoE/MOORE):	
k Experts	[2, 3, 4]
Encoder Linear Layers	1
Encoder Output Units	k (number of experts)
Encoder Use Bias	False
Encoder Activation	Linear

B.3 Reward Normalization

To ensure fair comparison across tasks with inherently different reward scales, raw episodic returns are normalized with respect to the maximum achievable reward in each environment. The standard MiniGrid reward for successful task completion is calculated as

$$1 - 0.9 \times (\texttt{steps\_taken}/\texttt{max\_episode\_steps}),$$

while failure results in a score of 0. We further normalize this score by performing a Min-Max scaling with respect to the maximum performance obtainable in each environment. We use the default maximum timesteps of each environment and estimate how many steps an optimal agent can solve the environment. This normalization procedure scales the performance such that a score of 1.0 represents achieving the optimal (shortest path) solution, facilitating comparisons of learning efficacy across environments with varying complexities and step horizons and addressing reward scales. Table 3 presents the optimal steps, maximum allowed steps, and the maximum achievable reward used for the score normalization of each environment.

Table 3: Environment-specific parameters for reward normalization. This table lists the optimal number of steps to solve each task, the maximum default permissible steps per episode, and the resulting maximum achievable raw reward score (used as the max score for normalization).

Environment Name	Optimal Steps	Max Steps	Achievable Reward
`DoorKey-6x6-v0`	11	360	0.9725
`DistShift1-v0`	11	252	0.9607
`RedBlueDoors-6x6-v0`	8	720	0.9900
`LavaGapS7-v0`	8	196	0.9633
`MemoryS11-v0`	15	605	0.9777
`SimpleCrossingS9N2-v0`	15	324	0.9583
`MultiRoom-N2-S4-v0`	5	40	0.8875

C Plasticity Metrics Implementation

This appendix details the methodology and interpretation of the plasticity metrics used in this work, serving as correlative indicators of an agent's learning capacity and adaptability. The computation of activations and gradients for these metrics relies on sampling from a plasticity replay buffer of training observations to approximate expected values via sample means. Hyperparameters specific to these calculations are detailed in Table 4.

C.1 Neuron Dormancy

We adapt the dormant neuron formalization from Sokar et al. (2023). A neuron's activity is assessed relative to other (non-masked) neurons in the same layer. Given an input distribution D (approximated by the plasticity replay buffer) and an activation $h_i^l(x)$ of a neuron i in layer l with H^l neurons under input $x \in D$, the normalized activation is

$$s_i^l = \frac{\mathbb{E}_{x \in D} |h_i^l(x)|}{\frac{1}{H^l} \sum_{k=1}^{H^l} \mathbb{E}_{x \in D} |h_k^l(x)|}.$$

Neuron i is called τ-dormant for some threshold $\tau > 0$ if $s_i^l \leq \tau$. If H_τ^l denotes the number of dormant neurons per layer, then the *dormancy ratio* β_τ is the ratio of dormant neurons and all neurons across all layers in the network L_{all} except the final L_{out}

$$\beta_\tau = \frac{\sum_{l \in L_{\text{all}} \setminus \{L_{\text{out}}\}} H_\tau^l}{\sum_{l \in L_{\text{all}} \setminus \{L_{\text{out}}\}} H^l}.$$

A high percentage of dormant neurons suggests significant underutilization of the network's capacity, potentially hindering its ability to learn complex functions or adapt to new data, a key aspect of plasticity.

C.2 Trace of the Fisher Information Matrix

The Fisher Information Matrix (FIM) F quantifies the sensitivity of a model's output (e.g., the policy) to changes in the parameters θ. For a policy π, its Fisher trace is given by

$$\mathrm{Tr}(F) = \mathbb{E}_{s,a\sim\pi}\left[\|\nabla_\theta \log \pi(a|s)\|_2^2\right].$$

The trace of the FIM can be viewed as a measure of the policy's sensitivity to parameter perturbations. A very high or persistently increasing trace might indicate that the policy is in a "sharp" region of the loss landscape, making it brittle to small changes and potentially indicative of overfitting or optimization instability. Conversely, a lower, stabilized trace, as observed in our pruned agents (see Section 4.2), can suggest convergence to "flatter" minima, implying a more robust policy that is less sensitive to parameter variations and more capable of sustained learning or adaptation.

C.3 Effective Rank

For a feature matrix Φ (e.g., a shared feature extractor) with d singular values σ_i sorted descendingly, the effective rank at tolerance δ is

$$\mathrm{srank}_\delta(\Phi) = \min_k \left\{ \frac{\sum_{i=1}^k \sigma_i}{\sum_{i=1}^d \sigma_i} \geq 1-\delta \right\}.$$

The effective rank measures the dimensionality of the space spanned by the features. A low effective rank suggests a representation collapse, where learned features are highly correlated and less diverse, limiting a network's ability to represent various information. Conversely, a high effective rank implies a richer, more diverse set of feature representations, implying a greater capacity to learn and distinguish between inputs.

Table 4: Configuration details for gradual magnitude pruning and the alternative plasticity-enhancing methods (ReDo, Reset, and Weight Decay) evaluated.

Hyperparameter	**Value**
Gradual Magnitude Pruning:	
Desired Sparsity ρ_F	[0%, 80%, 95%, 99%]
Pruning Frequency	500 timesteps
Pruning start interval t_{start}	0.05× number of timesteps
Pruning end interval t_{end}	0.80× number of timesteps
Sparsity ρ_t at timestep t	$\rho_F \left[1 - \left(1 - \frac{t - t_{\text{start}}}{t_{\text{end}} - t_{\text{start}}}\right)^3\right]$
Prune Bias	False
Pruned Layers	[Conv2D, Linear]
Plasticity:	
Plasticity Buffer Max Size	100000
ReDo (Sokar et al., 2023) Frequency	5000 timesteps
Dormant Neuron Threshold	0.01
Dormant Activation Batch Size	1024
Fisher Trace Batch Size	1024
Effective Rank Batch Size	1024
Effective Rank Target	Shared Feature Extractor
Reset (Nikishin et al., 2022) at timestep	[60000, 120000]
Reset (Nikishin et al., 2022) Target Layers	Output
Weight Decay Coefficient	1×10^{-3}

D Detailed Learning Curves

This appendix presents the complete learning curves for all agent configurations. These curves complement the aggregated final performance data analyzed in Table 1. In the subsequent figures:

- Figure 3 illustrates the normalized aggregate returns for agents under different pruning levels (Dense, 80%, 95%, 99% sparsity) across the MT3, MT5, and MT7 benchmarks for each of the three multi-task architectures: Multi-Task PPO (MTPPO), Mixture of Experts (MoE), and Mixture of Orthogonal Experts (MOORE).
- Figure 4 displays the corresponding learning curves comparing 95% sparsity pruning with alternative plasticity-inducing interventions (ReDo, Reset, and Weight Decay) on the same benchmarks and architectures.

The horizontal dashed line indicates the aggregated performance of single-task PPO agents. Each of the ST PPO agents was trained separately on a single environment from the respective benchmark for the full 200000 timesteps (the same total duration as the multi-task agents) across 30 runs. Consequently, this ST PPO performance should be viewed as a potentially **near-maximal** reference point from a single-task perspective, as multi-task agents faced the more challenging scenario of learning all tasks within a benchmark concurrently using the same total number of timesteps.

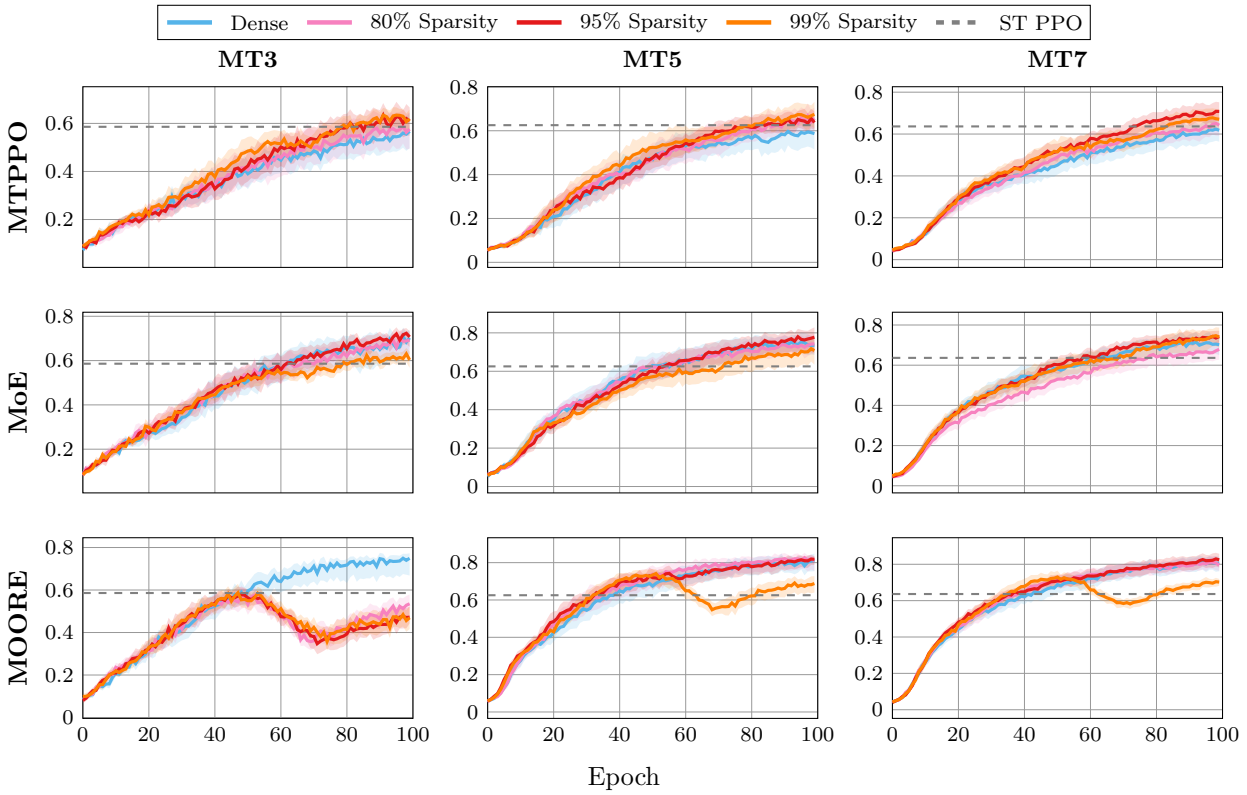

Figure 3: Learning curves comparing performance under various pruning levels across benchmarks and architectures. Normalized aggregate IQM is shown for MTPPO, MoE, and MOORE agents on the MT3, MT5, and MT7 benchmarks. Each plot compares Dense agents with those pruned to 80%, 95%, and 99% sparsity. The dashed horizontal line denotes the aggregated performance of single-task PPO agents trained for the full 200000 timesteps on each respective task, serving as the maximal achievable single-agent performance.

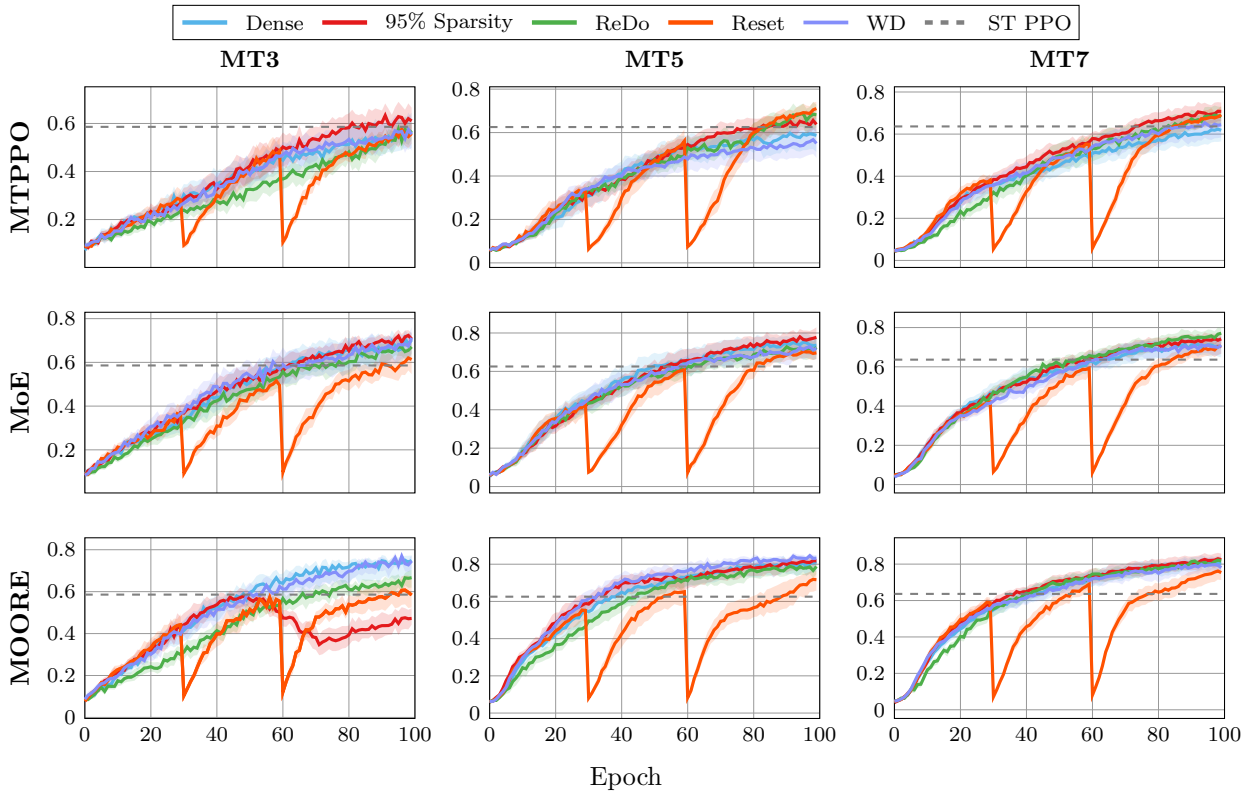

Figure 4: Learning curves comparing pruning with alternative plasticity-inducing intervention strategies. Normalized aggregate IQM for MTPPO, MoE, and MOORE agents on the MT3, MT5, and MT7 benchmarks. These plots compare the performance of Dense agents, agents with 95% sparsity pruning, and agents employing alternative methods: ReDo, Reset, and Weight Decay (WD). The dashed horizontal line denotes the aggregated performance of single-task PPO agents trained for the full 200000 timesteps on each respective task, serving the as maximal achievable single-agent performance.

E Detailed Plasticity Metrics

This appendix provides the detailed plasticity metric evolutions, complementing our analyses in Section 4.2.

- Figure 5, Figure 6, and Figure 7 show the plasticity dynamics of all architectures under varying levels of gradual magnitude pruning Dense, 80%, 95%, and 99% sparsity), shown for the MT3, MT5, and MT7 benchmarks, respectively.
- Figure 8, Figure 9, and Figure 10 compare the effects on plasticity of the dense and 95% sparsity agents against alternative intervention (ReDo, Reset, WD), across MT3, MT5, and MT7, respectively.

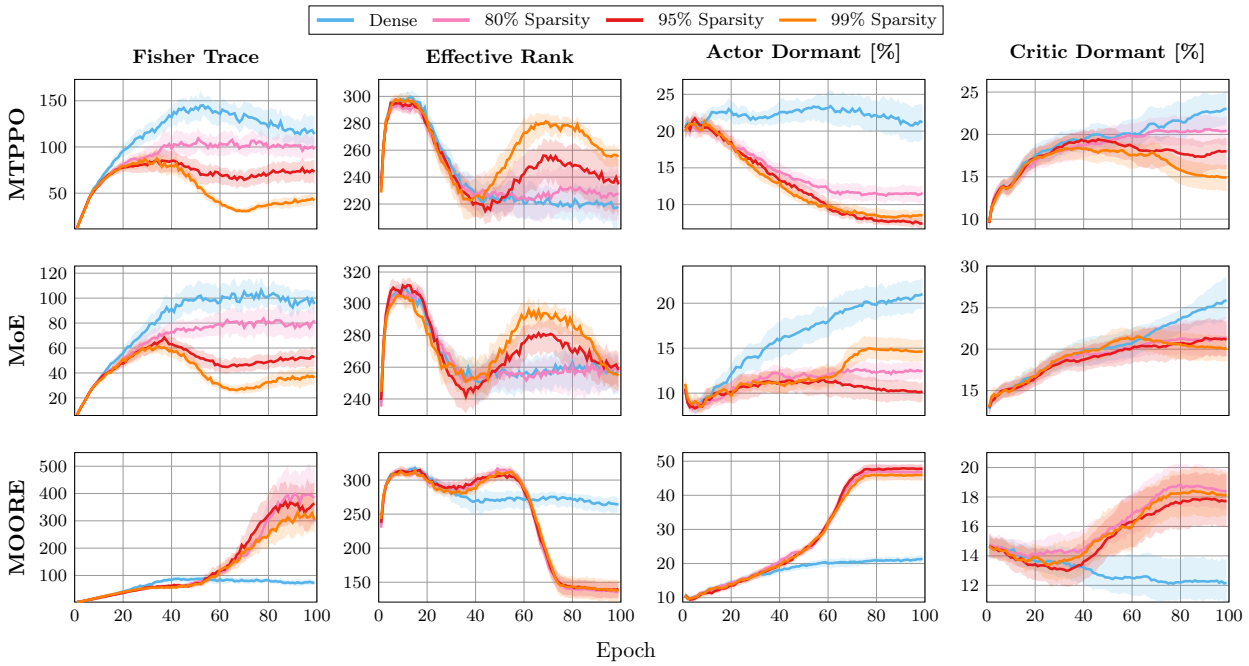

Figure 5: Gradual magnitude pruning positively influences plasticity markers in MTPPO and MoE architectures in MT3. Increasing sparsity (especially 95-99%) generally correlates with a lower Fisher Trace peak, sustained or recovered Effective Rank, and notably reduced Actor Dormancy compared to Dense baselines. In contrast, MOORE shows distinct dynamics, with pruning leading to plasticity degradation (e.g., rank collapse and increased FIM trace), aligning with its performance on this benchmark.

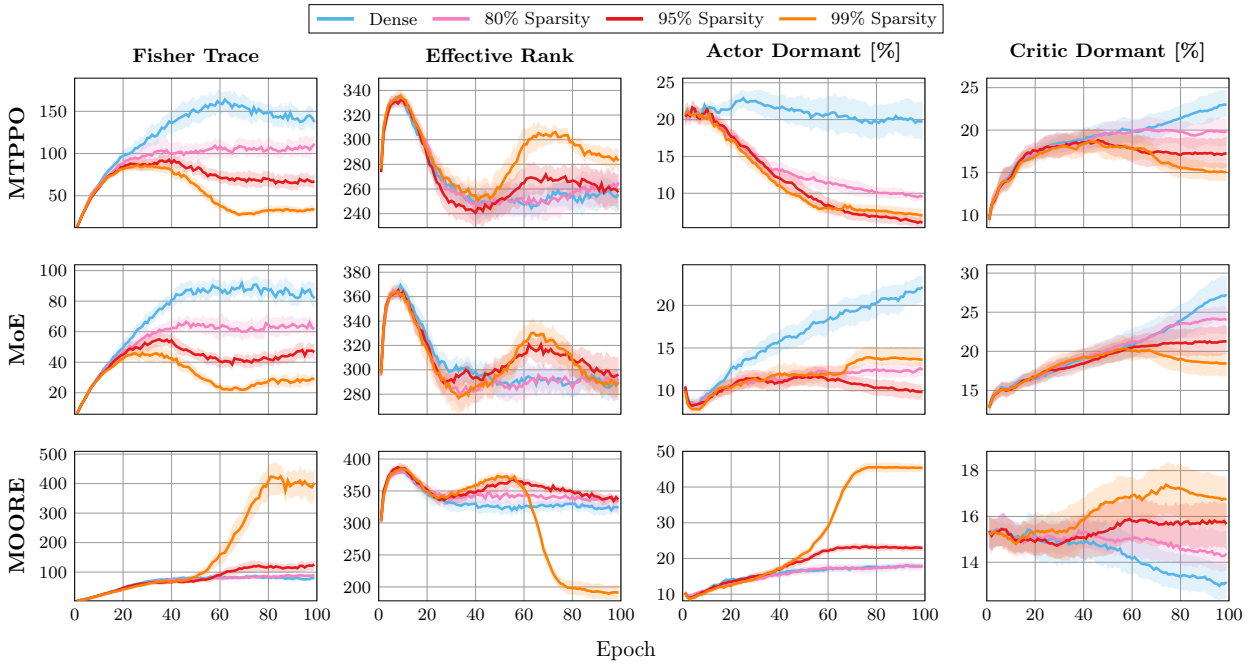

Figure 6: Plasticity dynamics under various pruning levels in MT5. MTPPO and MoE largely replicate the beneficial plasticity trends observed in MT3 with increasing sparsity (e.g., controlled FIM trace, reduced dormancy). For MOORE, unlike MT3, moderate pruning (80-95%) maintains plasticity markers comparable (or slightly degraded) to its Dense counterpart, while very high sparsity (99%) can still induce degradation like rank collapse.

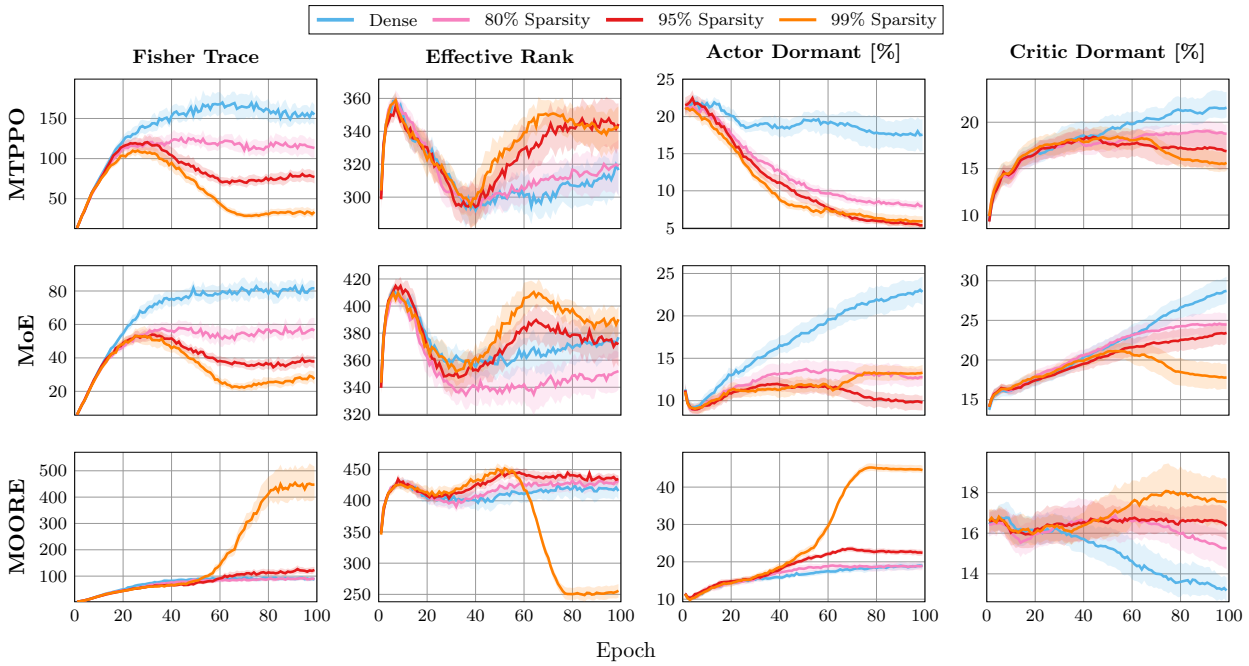

Figure 7: Impact of sparsity on plasticity metrics in MT7. Consistent with the two other benchmarks, pruning benefits plasticity in MTPPO and MoE. Similar to MT5, MOORE continues to show similar or slightly degraded plasticity measures from moderate pruning on this wider-network benchmark, with 99% sparsity experiencing rank collapse.

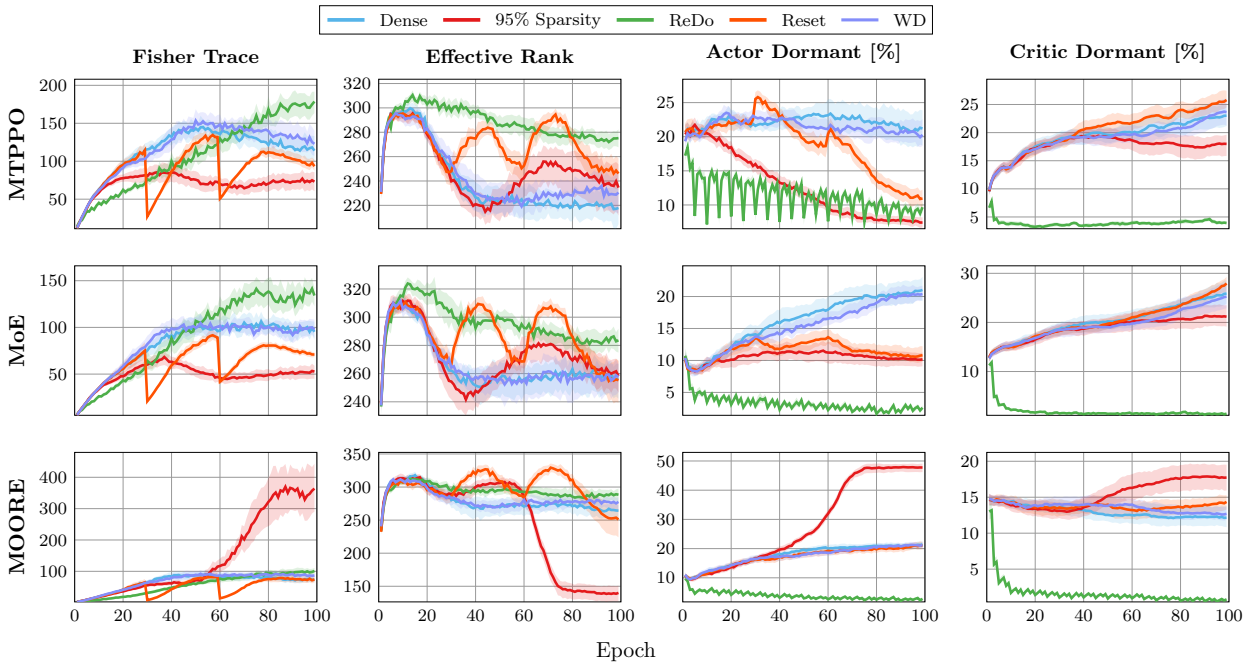

Figure 8: Comparative plasticity dynamics of 95% Sparsity Pruning versus alternative interventions in MT3. For MTPPO and MoE, ReDo achieves notably lower neuron dormancy and often a higher mean effective rank than 95% pruning. Pruning, in contrast, offers a distinct profile that correlates strongly with performance improvements, differing from the oscillations induced by Reset or the minimal impact of Weight Decay relative to the Dense baseline. For MOORE, 95% pruning degrades plasticity compared to its relatively stable dense baseline; other interventions show less detrimental effects.

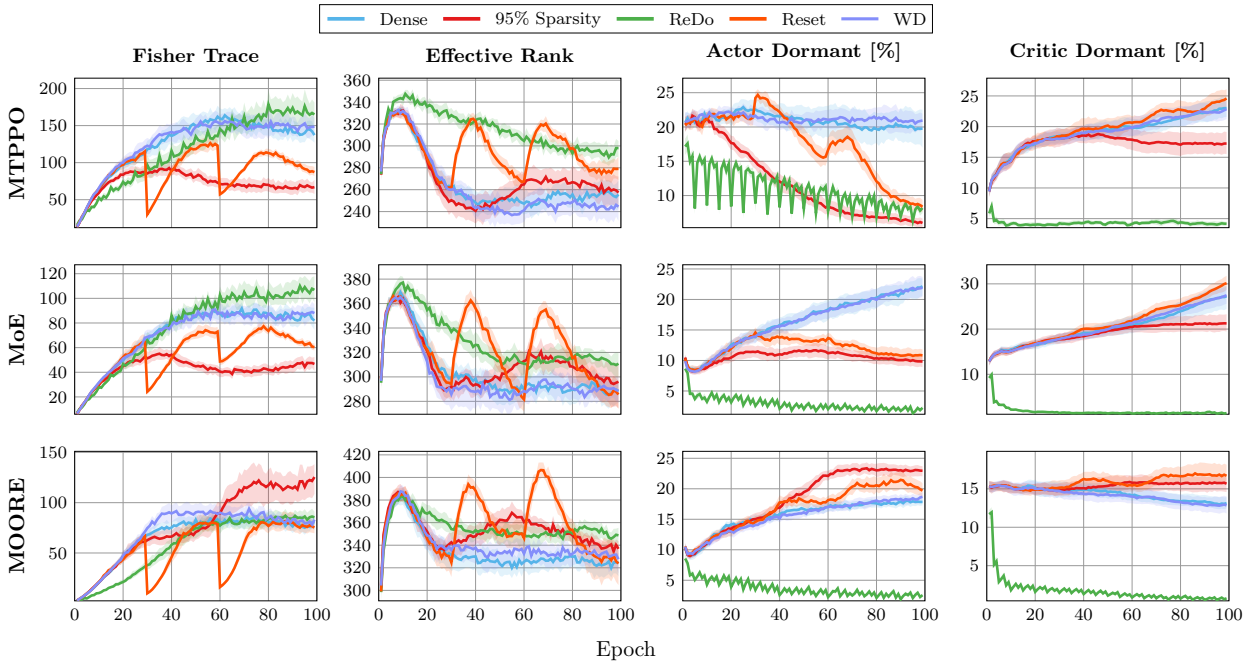

Figure 9: Plasticity profiles under different intervention strategies in MT5. ReDo generally leads to lower neuron dormancy and can exhibit a higher effective rank than 95% Sparsity Pruning. Reset's interventions continue to cause visible fluctuations. For MOORE, the dense baseline exhibits considerable stability; 95% pruning and Weight Decay generally maintain or slightly modify these stable characteristics without the pronounced broad enhancements or degradations seen in other architectures under various interventions.

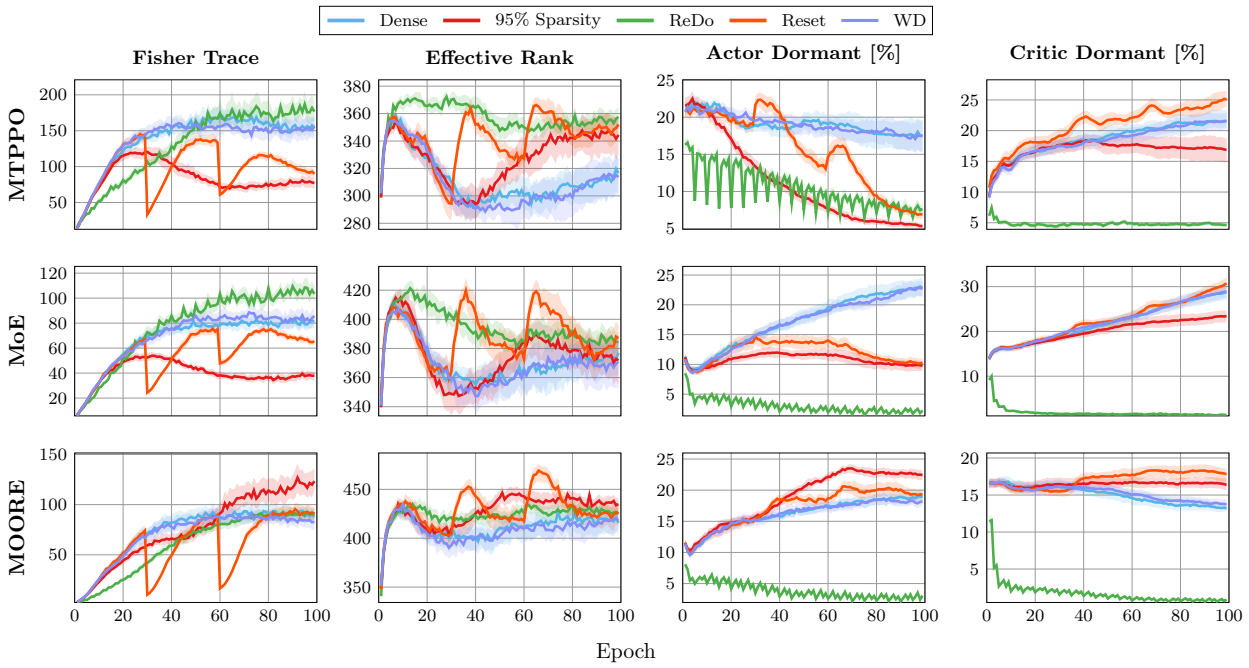

Figure 10: Plasticity profiles under different intervention strategies in MT7, largely similar to the ones observed in MT5. ReDo generally leads to lower neuron dormancy and can exhibit a higher effective rank than 95% Sparsity Pruning. Reset's interventions continue to cause visible fluctuations. For MOORE, the dense baseline exhibits considerable stability; 95% pruning and Weight Decay generally maintain or slightly modify these stable characteristics without the pronounced broad enhancements or degradations seen in other architectures under various interventions.