Measure gradients, not activations! Enhancing neuronal activity in deep reinforcement learning

Jiashun Liu[1]* **Zihao Wu**[1]* **Johan Obando-Ceron**[2,3]* **Pablo Samuel Castro**[2,3]
Aaron Courville[2,3] **Ling Pan**[1]

[1] Hong Kong University of Science and Technology
[2] Mila - Québec AI Institute [3] Université de Montréal

Abstract

Deep reinforcement learning (RL) agents frequently suffer from neuronal activity loss, which impairs their ability to adapt to new data and learn continually. A common method to quantify and address this issue is the τ-dormant neuron ratio, which uses activation statistics to measure the expressive ability of neurons. While effective for simple MLP-based agents, this approach loses statistical power in more complex architectures. To address this, we argue that in advanced RL agents, maintaining a neuron's *learning capacity*, its ability to adapt via gradient updates, is more critical than preserving its expressive ability. Based on this insight, we shift the statistical objective from activations to gradients, and introduce `GraMa` (Gradient Magnitude Neural Activity Metric), a lightweight, architecture-agnostic metric for quantifying neuron-level learning capacity. We show that `GraMa` effectively reveals persistent neuron inactivity across diverse architectures, including residual networks, diffusion models, and agents with varied activation functions. Moreover, **re**setting neurons guided by `GraMa` (`ReGraMa`) consistently improves learning performance across multiple deep RL algorithms and benchmarks, such as MuJoCo and the DeepMind Control Suite. **We make our code available**[2].

1 Introduction

Deep reinforcement learning (Deep RL) has achieved remarkable success across a variety of domains, including robotics [Liu et al., 2021], foundation model fine-tuning [Shao et al., 2024, Liu et al., 2025c, Yu et al., 2025, Li et al., 2025, Lu et al., 2025, Liu et al., 2025b], and game playing [Berner et al., 2019, Schwarzer et al., 2023]. These advancements have been driven by the expressive power and adaptive learning ability of neural networks which effectively approximate and optimize value functions and/or policies [Sokar et al., 2023]. However, recent studies have uncovered a critical and often underexplored challenge: as training progresses, subsets of neurons in these networks often experience a progressive loss of activity and become dormant [Sokar et al., 2023, Ma et al., 2024, Qin et al., 2024]. This phenomenon reduces the learning capacity of the network, comprising its ability to adapt to non-stationary data distributions [Nikishin et al., 2022], which in turn hinders their ability to acquire new knowledge and adapt to evolving environments [Abbas et al., 2023]. Despite its importance, quantifying and mitigating neuronal activity remains challenging due to its complex underlying mechanisms [Lyle et al., 2023, Obando Ceron et al., 2023, Nauman et al., 2024a, Lyle et al., 2024].

To address this problem, a primary principle has been to restore a network's learning ability by reactivating or resetting inactive neurons. These approaches span multiple granularities: model-level

[*] Equal contribution. Correspondence to: `{ljshasdream, jobando0730}@gmail.com`
[2] Code: `https://github.com/torressliu/grad-based-plasticity-metrics`

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

and layer-level resets that reinitialize specific layers, while straightforward, often lead to catastrophic forgetting [Nikishin et al., 2023]. On the other hand, neuron-dependent resets (e.g., ReDo [Sokar et al., 2023]) target specific underperforming neurons [Xu et al., 2023] and provide a finer-grained approach by selectively reinitializing a subset of neurons identified as dormant, which mitigates forgetting and maintains computational efficiency.

The effectiveness of neuron-dependent resets hinges critically on having a reliable criterion to identify which neurons require initialization. Existing methods primarily rely on activation-based metrics, such as the τ dormant neuron metric [Sokar et al., 2023], which measures neuronal inactivity based on activation values (i.e., the output of activation functions). It has demonstrated utility in standard architectures, and has been used to guide targeted neuron resets to restore activity in simple settings such as serial MLPs with ReLU activations [Agarap, 2018], providing a simple means of maintaining learning capability without relying on auxiliary networks [Nikishin et al., 2023] or models [Lee et al., 2024]. By identifying neurons with weak or no activation and selectively reinitializing them, the activation-based dormancy metrics become a widely adopted tool for restoring learning capabilities and preventing performance plateau in standard deep RL settings [Farias and Jozefiak, 2025, Liu et al., 2025a, Juliani and Ash, 2024].

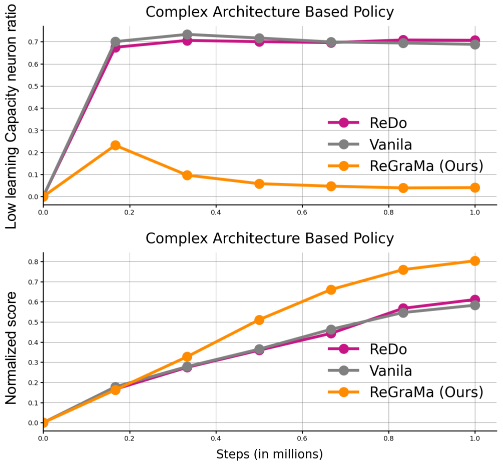


Figure 1: **Dormant neuron metric struggles in advanced vision RL BRO-net agent [Nauman et al., 2024b]**. Neuron resetting based on the dormant neuron index (ReDo) cannot restore the learning capacity of the agent, which limits its effectiveness. The curves record normalized score across 3 image-input tasks (15 runs per method), i.e., Dog Stand, Dog Walk, Dog Run.

However, deep RL architectures have rapidly evolved beyond these simple architectures, integrating advanced components such as residual connections [Nauman et al., 2024b, Lee et al., 2025a], mixture of experts (MoEs) [Obando-Ceron et al., 2024, Sokar et al., 2024, Willi et al.], and diffusion-based models [Ren et al., 2025, Wang et al., 2024] to improve scalability and performance for tackling complex tasks involving continuous control and high-dimensional observations. Additionally, they also imperceptibly alter the statistical properties of neuron activations, creating a significant mismatch with traditional activity metrics. Through our extensive analysis, we find that resetting based on τ-dormant neuron ratio (ReDo) reduces effectiveness in these newer architectures.

As shown in Fig. 1, ReDo struggles to identify inactive neurons that have lost their learning capacity in the more advanced BRO-net architecture, which features residual paths and layer normalization, ultimately failing to achieve the intended enhancement. This mismatch arises because activation-based metrics focus solely on a neuron's current output, evaluating its expressive capacity (how strongly it activates), while neglecting its learning capacity (how effectively it can adapt to new data distributions). Consequently, it becomes particularly problematic as deep RL architectures evolve beyond simple feedforward networks to incorporate advanced structural elements, diverse activation functions, and sophisticated normalization techniques. In these modern architectures, a neuron's activation magnitude often fails to accurately reflect its true learning potential.

To address this mismatch, we propose a fundamental shift in perspective: from evaluating neurons based on their outputs to evaluating their learning potential by leveraging gradient magnitude. While activation values only capture what a neuron currently expresses, gradients tend to measure a neuron's capacity in response to the given situation that directly drives parameter updates. This makes it a general and natural proxy for neuronal health across architectural variations. Building upon this insight, we introduce `GraMa` (**Gra**dient **Ma**gnitude based Neuronal Activity Metric), a robust and lightweight framework for quantifying neuronal activity via the gradient magnitudes. `GraMa` maintains validity across diverse architectural patterns, making it well-suited for modern deep RL agents.

In addition, `GraMa` imposes negligible computational and memory overhead by utilizing information already present in the optimization pipeline, which is a critical consideration for resource-intensive RL training. Leveraging `GraMa`'s efficiency, we develop a targeted neuron reset mechanism, (`ReGraMa`), that selectively reinitializes inactive neurons that have lost their learning capacity during training. This mechanism demonstrates robust efficiency across various architectures, including the SAC variant [Nadimpalli et al., 2025], the residual BRO-net [Nauman et al., 2024b], and the diffusion-based policy DACER [Wang et al., 2024].

Our contributions are summarized as follows:

- We show that the widely-adopted activation-based neuronal health measurements lose statistical power in complex architectures and provide a qualitative analysis of the underlying causes.
- We reframe neuronal health evaluation through `Grama`, a gradient-based metric that quantifies learning potential independently of architectural complexity, and demostrate that neuronal learning capacity degradation affects even state-of-the-art network developments.
- We develop (`ReGraMa`), an efficient neuron resetting mechanism guided by `GraMa`, which effectively restores neuronal activity across a wide range of network architectures.
- We conduct extensive experiments on MuJoCo [Brockman et al., 2016], DeepMind Control Suite [Tassa et al., 2018], showing that `GraMa`-guided resetting improves performance and learning stability across diverse architectures.

2 Background

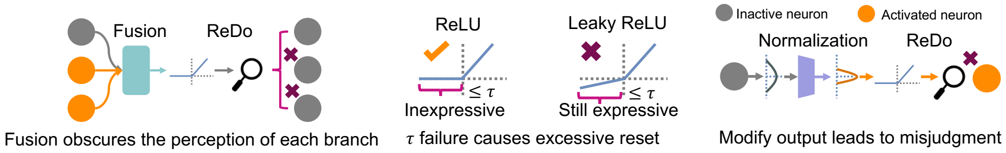
A reinforcement learning (RL) problem is typically formalized as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the transition probability function $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, $\mathcal{R}$ the reward function $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and $\gamma \in [0, 1)$ the discount factor. The state-action value function under policy π is given by:$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \mid s_0 = s, a_0 = a\right]$. The objective is to find a policy π that maximizes the expected return $Q^{\pi}(s, a)$ for each state s. In actor-critic-based deep RL, the Q-function is approximated by a neural network Q_θ with parameters θ. During training, the agent interacts with the environment and stores trajectories in a replay buffer D. Mini-batches sampled from D are used to update Q_θ by minimizing the temporal difference loss:$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,r,s') \sim D}\left[(Q_\theta(s, a) - Q^{\mathcal{T}}(s, a))^2\right]$, where the target is given by $Q^{\mathcal{T}}(s, a) = \mathcal{R}(s, a) + \gamma Q_{\bar{\theta}}(s', \pi_{\bar{\phi}}(s'))$, and $\bar{\theta}$, $\bar{\phi}$ denote the parameters of target critic and actor networks, respectively.

Neuronal activity measurement based on activation value. Recent studies have identified that the dynamic and non-stationary nature of RL objectives can cause neurons to permanently lose their activity [Lu et al., 2018], impairing the network's ability to fit new data and thereby limiting learning progress [Nikishin et al., 2022, Ceron et al., 2024]. This phenomenon, referred to as the *dormant neuron phenomenon*, has been quantitatively characterized using the $\tau-$*dormant neuron ratio* [Sokar et al., 2023], and serves as a core metric for assessing neuron-level plasticity [Xu et al., 2023, Qin et al., 2024, Liu et al., 2025a]. Specifically, a neuron i in layer ℓ is considered τ-dormant if its normalized activation (see Eq. 1) falls below a threshold τ, where H^ℓ is the number of neurons in layer ℓ, and D denotes the data distribution. $h_i^\ell(x)$ represents the activation value of neuron i given input x.

$$S_i^\ell = \frac{\mathbb{E}_{x \sim D} \left| h_i^\ell(x) \right|}{\frac{1}{H^\ell} \sum_{k \in h} \mathbb{E}_{x \sim D} \left| h_k^\ell(x) \right|} \tag{1}$$

3 Related Work

Dynamic objectives may cause irreversible damage to the neuronal activity of networks during training [Lyle et al., 2023, Nauman et al., 2024a, Castanyer et al., 2025], which is also significant in the field of multi-agent RL [Qin et al., 2024] and visual deep RL [Ma et al., 2024]. This issue may lead to the phenomenon where deep learning agents progressively lose their ability to fit new data [Abbas et al., 2023], and limit their capacity for continual learning [Elsayed and Mahmood, 2024]. In the field of Deep RL, there are some factors that have been found to have a direct correlation with the activity of neurons [Juliani and Ash, 2024, Mayor et al., 2025, Castanyer et al., 2025]. Among these, activation functions play a particularly important role, recent findings indicate that replacing ReLU activation can help preserve neuron-level learning capacity [Abbas et al., 2023].



(a) The multi-branch structure (b) Non-ReLU activation functions (c) Normalization

Figure 2: **Key techniques involved in advanced models may reduce the efficiency of activation-based metrics.** (a) The final activation values observed after branch fusion no longer accurately reflect the individual contributions of neurons in each branch. (b) A near-zero value fails to represent the expressivity with non-ReLU activation function. (c) Normalization confuses ReDo by modifying the neuron's outlier output.

Certain loss functions, especially those incorporating L2 regularization, have been found to mitigate the network's activity degradation [Kumar et al., 2023]. Another relevant line of research explores architectural adjustments such as scaling up [Nikishin et al., 2023] or topology growth [Liu et al., 2025a]. These methods can introduce modules that enhance the proportion of active, high-quality neurons. A separate and increasingly influential line of research focuses on directly manipulating model parameters to recover their activity. Resetting neural networks, either periodically or selectively, has been shown to be both effective and easy to implement [Farias and Jozefiak, 2025]. For instance, Nikishin et al. [2022] demonstrates the benefits of periodic resets for continual learning.

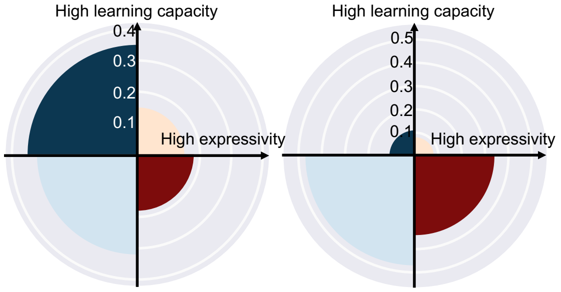
Similarly, Ma et al. [2024] shows that selectively resetting specific layers can improve stability without sacrificing expressive capacity. At a finer granularity, ReDO [Sokar et al., 2023] introduces a neuron-level resetting scheme that outperforms earlier coarse strategies in terms of stability and precision. However, ReDO's reliance on activation-based quantization limits its applicability to more complex architectures [Lyle et al., 2023, 2024]. To address this limitation, we introduce a novel neuronal activity quantification approach based on *gradient magnitudes*, enabling generalized activity estimation and recovery across arbitrary network designs. Notably, Ji et al. [2024] provided the first empirical evidence in model-free reinforcement learning of a strong correlation between the internal gradient dynamics of the agent's policy network and its learning capacity, offering compelling motivation for the present study.

4 Gradient Magnitude based Neuronal Activity Metric (`GraMa`)

In this section, we qualitatively investigate ReDo's limitations from the perspective of architectural composition Sec. 4.1. We then introduce `GraMa` (Sec. 4.2), a novel neuronal activity metric that redefines the statistical objective from activation values to gradient magnitudes. Next, we perform an in-depth analysis of the characteristics of low learning capacity neurons using `GraMa`, and provide both theoretical and empirical analyses of the performance similarity between `GraMa` and ReDo on simple architectures. Finally, we analyze the advantages of `GraMa` in complex architectures.

4.1 Misalignment of activation-based neuron activity detection in modern architectures

The widely adopted activation-based neuron activity measures [Sokar et al., 2023] operate on the assumption that a neuron's activation magnitude directly corresponds to its contribution to learning. Motivated by the development of more complex network structures in language and vision domains, recent trends in deep RL have moved beyond simple serial MLP architectures with ReLU activations towards more sophisticated network designs for scaling. However, this assumption becomes increasingly fragile in modern complex architectures which are rapidly becoming standard practice. We reveal a misalignment between activation values and actual learning potential, leading to inefficient neuron identification.



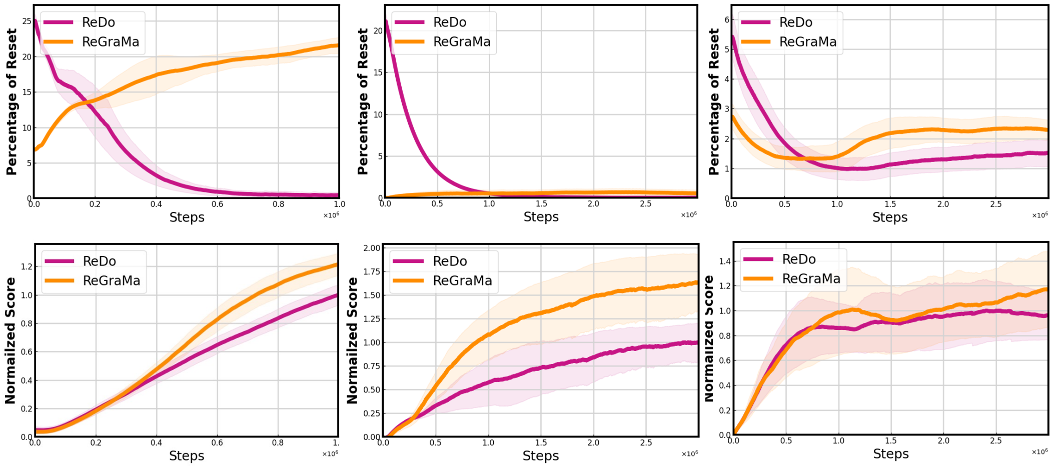
(a) ReDo (b) ReGraMa

Figure 3: **ReDo fails to reset neurons with low learning capacity (red), while also incorrectly resets the neurons with high learning capacity (dark blue)**, including those with both high learning capacity and high expressiveness (white). This plot shows the proportion of resetting 4 neuron types during the same reset step in the Dog Walk task.

Motivating example. To systematically analyze this misalignment, we evaluate BRO-net [Nauman et al., 2024b], a recently proposed architecture, on the challenging Dog Walk task [Tassa et al., 2018]. We categorized neurons into four quadrants based on their expressive capacity (measured by activation magnitude) and learning capacity (measured by gradient magnitude), which we introduce in more detail in Sec. 4.2. Neurons are ranked according to these two criteria, and we select the top 25% of neurons in each ranking, so as to provide an understanding of the types of neurons reset by each method. The results in Fig. 3 demonstrate that ReDo cannot accurately identify neurons that have genuinely lost their learning capacity (red) based on their activation values. More importantly, ReDo mistakenly resets a significant number of neurons with high learning capacity that are less expressive at the moment (dark blue).

Analysis. In addition to the primary factor of statistical objectives, we identify three key architectural features that may undermine the reliability of activation-based neuron dormancy detection. Fig. 2 provides an intuitive illustration of each. (i) *Multi-branch network structures.* In modern architectures like ResNets [He et al., 2015], information flows through multiple branches before being fused and passing through activation functions. This architectural pattern introduces a critical problem: the final activation values observed after branch fusion no longer accurately reflect the individual contributions of neurons in each branch. Our experiments with Resnet-SAC [Shah and Kumar, 2021] shown in Fig. 4 (a) confirm this effect, where the results suggest that activation-based methods lose their ability to identify dormant neurons and limits the performance of the agent. (ii) *Non-ReLU activation function.* The interpretability of activation values is highly dependent on the specific activation function used. While ReLU creates a clear distinction between "dead" (output=0) and "active" neurons, this clarity breaks down when considering alternative activation functions. For example, with Leaky ReLU, neurons may remain expressive and contribute to learning even when their pre-activation values are negative.

As a result, solely relying on activation values to measure neuron dormancy becomes unreliable and may lead to misidentification. Our experiments replacing ReLU with Leaky ReLU in SAC (Fig. 4 (b)) demonstrate this problem, where the activation-based method (ReDo) shows considerable reset activity during early learning with lower performance, indicating that it struggles to establish meaningful dormancy criteria. (iii) *Normalization layers.* Normalization techniques modify the distribution of neuron outputs before they pass through activation functions. This process adjusts outliers and rescales values across entire layers, causing post-normalization values to lose a direct correspondence with individual neuron functionality. Thus, activation values measured after normalization may no longer accurately reflect a neuron's learning capacity. Our experiments incorporating layer normalization into SAC, Fig. 4 (c) reveal that the activation-based method struggles to maintain consistent neuron assessment, whereas our method consistently identifies relevant neurons throughout the learning process.



(a) The multi-branch structure (b) Non-ReLU activation functions (c) Normalization

Figure 4: **Empirical validation corresponding to the three cases of Fig. 2**. Top row records the proportion of reset neurons. Bottom row shows the performance. Each curve represents the average over 3 seeds (Dog Walk).

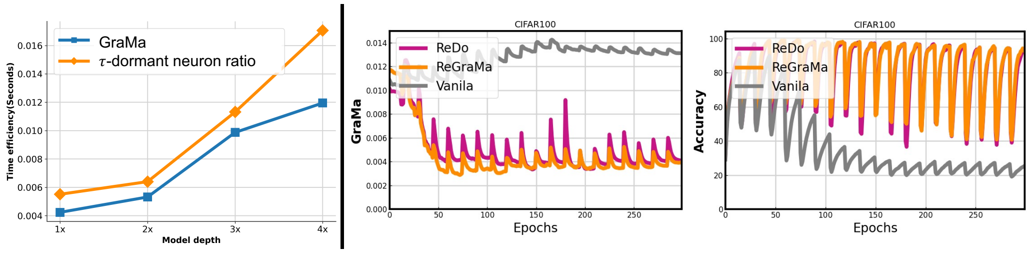


Figure 5: (Left) Execution time comparison based on BRO-net (RTX3090 GPU); (Right) The results verify that ReGraMa is as effective as ReDo on traditional network architectures. The backbone, hyperparameter settings and number of seeds are the same for all experiments.

4.2 Gradient Magnitude based Neuronal Activity Metric (GraMa)

Learning activity is directly related to the gradient received by a neuron during training [Zhou and Ge, 2024]. A natural and simple way to enhance ReDo-like metrics is to count how many neurons in each network layer fail to obtain meaningful gradients from the current sample batch. Given an input distribution D, let $|\nabla_{h_i^l} L(x)|$ denote the gradient magnitude of the neuron i in layer ℓ under input $x \in D$ and H^ℓ be the number of neurons in layer ℓ. Based on Eq. 1, we compute a learning capacity score for each neuron using the normalized average of the corresponding layer ℓ, as shown in Eq. 2.

$$G_i^\ell = \frac{\mathbb{E}_{x\in D}\, |\nabla_{h_i^l} L(x)|}{\frac{1}{H^l} \sum_{k\in h} \mathbb{E}_{x\in D}\, |\nabla_{h_i^l} L(x)|}. \tag{2}$$

GraMa determines that neuron i in layer ℓ is inactive when $G_i^\ell \leq \tau$. GraMa has the same form as the dormant neuron ratio, but redefines its core signal, using gradient magnitudes $|\nabla_{h_i^l} L(x)|$ rather than activation values $h_i^l(x)$. The pre-set threshold τ allows us to detect neurons with outlier gradient magnitude. Since gradient information is available in the tensor after backpropagation at each step, there is no need to store the intermediate outputs of each neuron during forward computation,which is required by activation-based metrics. This makes GraMa lightweight, as verified in Fig. 5 (left).

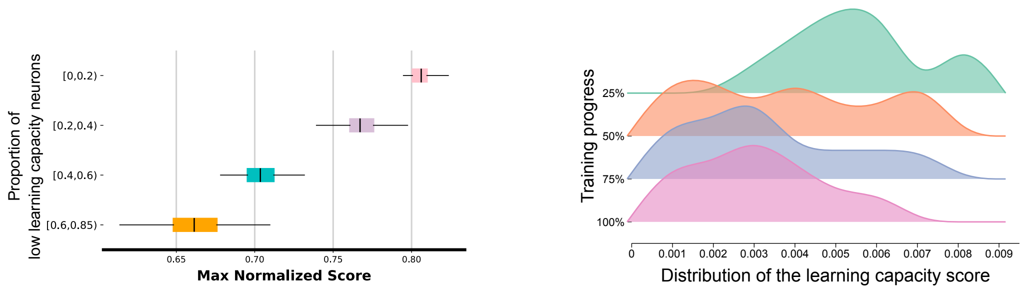
Algorithm 1: ReGraMa

```
Input : Model $\theta$, threshold $\tau$, frequency $\Delta_t$
while $t <$ maximum training time do
    Update $\theta$ with regular RL loss;
    if $t \mod \Delta_t == 0$ then
        for each layer $\ell$ do
            for eachneuron $i$ do
                Calculate $G_i^\ell$                ▷ Eq. 2
                if $G_i^\ell \leq \tau$ then
                    Reinitialize neuron $i$;
```

Resetting neurons guided by GraMa (ReGraMa). Neuron resetting is a simple technique widely used to preserve the learning capacity of deep RL agents [Nikishin et al., 2022]. Our approach follows the ReDo pipeline [Sokar et al., 2023], as outlined in Algorithm 1: during training, we periodically quantify the activity of neurons in all layers using GraMa; any neuron i with $G_i^\ell \leq \tau$ is considered inactive and reinitialized. Specifically, reinitialization involves resetting incoming weights to the original weight distribution, while outgoing weights are set to zero. It is worth noting that, empirically, we found that although LayerNorm layers contain only two learnable parameters, they tend to be quite inactive. As a result, ReGraMa is deployed for these parameters in our official implementation, leading to performance improvements that warrant further analysis. Another avenue worth exploring is the potential for resetting the input weights and biases of neurons.

The ratio of low learning capacity neurons is inversely related to performance. We control the number of reset neurons in ReGraMa to evaluate the performance of four agents with varying proportions of low learning capacity neurons on the Dog Walker task. Results in Fig. 6 (a) indicate that performance degrades significantly as the ratio of low learning capacity neurons increases.

Neuron loss of learning capacity is irreversible. We use GraMa with a threshold of $\tau = 0.0095$ to sample 1000 inactive neurons in the pre-training period of vanilla agent. We then trace the change



(a) (b)

Figure 6: (a): **Low learning capacity neurons have a direct negative impact on agent performance.** X-axis represents the normalized performance, and the Y-axis denotes the BRO-nets with different proportions of low learning capacity neurons. (b): **Neuron loss of learning capacity is irreversible.** x-axis denotes the number of neurons under each GraMa score, and the y-axis shows the training phase. (define > 0.0095 as active).

of their learning capacity scores and analyze the score distribution over time, as shown in Fig. 6 (b). The results indicate that none of the sampled neurons exceed the threshold of 0.0095 as training progresses. This suggests that such neurons are unable to recover their learning ability independently, further underscoring the importance of resetting neurons with low learning capacity.

Equivalence of ReGraMa to ReDo in traditional architectures. We theoretically analyze the similarities between ReGraMa and ReDo in traditional architectures (MLP with ReLU activations [Sokar et al., 2023]) and draw the following conclusions:

Theorem 1. *If neuron i is dormant ($s_i^\ell = 0$), then both $\nabla_{h_i^\ell} f = 0$ and $G_i^\ell = 0$.*

Proof. From the dormant neuron formula (Eq. 1), we can conclude that:

$$s_i^\ell = 0 \quad \Longleftrightarrow \quad \mathbb{E}_{x\in D}|h_i^\ell(x)| = 0.$$

Since $|h_i^\ell(x)| \geq 0$, $|h_i^\ell(x)| = 0 \iff h_i^\ell(x) = 0$. This means the neuron is almost never activated on the dataset D. The derivative of the ReLU [Agarap, 2018] activation function is:

$$\frac{\partial h_i^\ell(x)}{\partial z_i^\ell(x)} = \begin{cases} 1 & \text{if } z_i^\ell(x) > 0, \\ 0 & \text{if } z_i^\ell(x) \leq 0. \end{cases}$$

z represents the output after passing through the activation function. Thus, if $h_i^\ell(x) = 0$, then $z_i^\ell(x) \leq 0$, and during backpropagation, the gradient turns to zero:

$$\nabla_{z_i^\ell} f = \mathbb{E}_{x\in D}\left[\frac{\partial f}{\partial h_i^\ell(x)} \cdot \frac{\partial h_i^\ell(x)}{\partial h_i^\ell}\right] = \mathbb{E}_{x\in D}\left[\frac{\partial f}{\partial h_i^\ell(x)} \cdot \underbrace{\frac{\partial h_i^\ell(x)}{\partial z_i^\ell(x)}}_{=0} \cdot h^{\ell-1}(x)\right] = 0.$$

Takeaway. *We prove that, in the traditional architecture (MLP with ReLU), neurons that are identified as inactivate by ReDo will also be identified as such by* ReGraMa.

Empirical verification. We trained a traditional fully connected network with ReLU on the CIFAR-100 benchmark [Krizhevsky, 2009], following the continuous learning experimental setup in Dohare et al. [2024]. Every 15 epochs, a new category of data is added to the training set, requiring the agent to classify samples from the whole data distribution. The two **gray curves** in Fig. 5 (right) show that the vanilla agent's accuracy gradually declines as training progresses, indicating a loss of learning ability. Meanwhile, the proportion of inactive neurons detected by GraMa increases over time and fluctuates with the same periodicity as the accuracy curve, with both stabilizing around epoch 150. To further validate ReGraMa, we conducted a two-part intervention study: (1) We reset neurons identified as inactive by the $\tau-$dormant neuron ratio (ReDo) whenever a new data category is introduced. The resulting **pink curves** in Fig. 5 show that resetting dormant neurons can both improve performance and reduce GraMa ratio. (2) We then reset neurons flagged by ReGraMa as having low activity. This intervention produced improvements similar to those of ReDo, as illustrated by the **orange curves** in Fig. 5. These results suggest that ReGraMa is as effective as the ReDo metric in identifying neuronal inactivity in simple network architectures.

ReGraMa identifies inactive neurons more effectively in complex architectures. When moving to more complex architectures, resetting neurons based solely on activation values becomes less reliable. In deep networks with normalization layers, residual connections, or context-dependent features, activations can fluctuate substantially even for neurons that are not meaningfully contributing to learning. As a result, activation magnitude alone is a noisy indicator of long-term inactivity. To examine this limitation, we compared activation-based and gradient-based criteria by pruning the corresponding neurons from a complex BRO-net agent and evaluating the resulting performance.

The degree of post-pruning performance degradation reflects both the relevance of the pruned neurons and the reliability of the underlying metric. As shown in Fig. 7, pruning neurons identified by ReGraMa causes only minimal performance loss, indicating that gradient magnitude provides a more stable and task-relevant measure of neuronal inactivity. This finding suggests that ReGraMa detects structural inactivity—neurons that remain consistently uninformative during optimization—rather than transient activation noise. Such stability is particularly advantageous in large, modular architectures, where distinguishing genuine inactivity from context-specific silence is critical for effective pruning and interpretability.

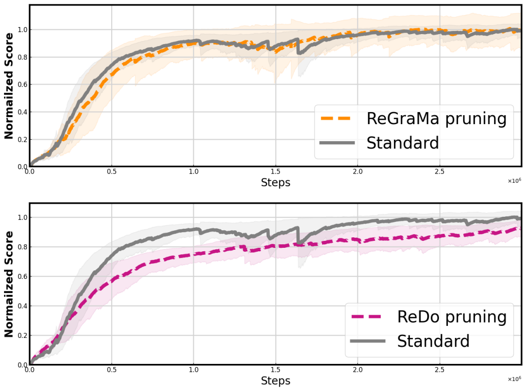


Figure 7: **Pruning neurons identified as inactive by ReGraMa during the training has less impact on the performance in Dog Walk.** Standard denotes vanilla agent. Curves show the average over four seeds.

5 Experiments

We conduct a series of experiments to investigate whether ReGraMa can mitigate neuronal activity loss and enhance performance. Specifically, we evaluate the effectiveness of ReGraMa across three representative and widely adopted architecture types: (i) the residual network-based policy (Sec. 5.1), (ii) the online policy parameterized by a diffusion model (Sec. 5.2), and (iii) the MLP policy featuring various activation functions (Sec. 5.3). Finally, we verify the robustness of ReGraMa with respect to the threshold τ (Appendix B).

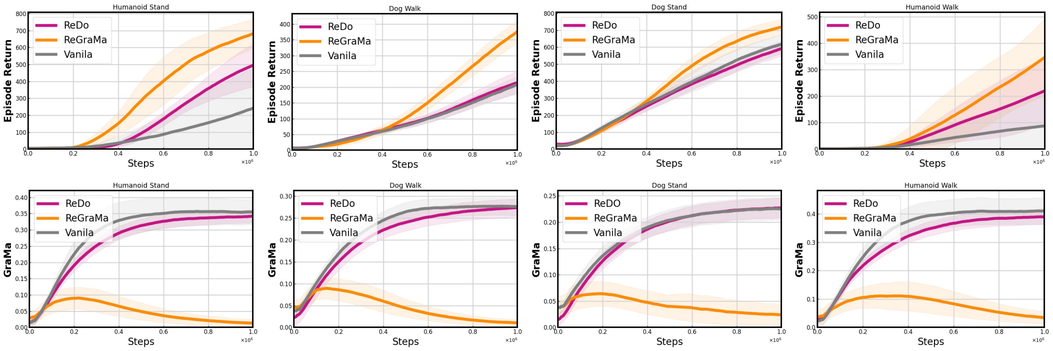


Figure 8: **Performance and neuron inactivity with the default BRO-net size.** (Top row) Episode return across four environments (Humanoid Stand, Dog Walk, Dog Stand, Humanoid Walk). (Bottom row) Corresponding proportion of inactive neurons. ReGraMa consistently achieves higher returns while maintaining fewer inactive neurons compared to ReDo and the vanilla baseline, demonstrating its effectiveness in stabilizing learning dynamics. Results are averaged over four seeds.

5.1 Residual Net-based Policy

Experiment setup. Recent studies [Nauman et al., 2024b, Lee et al., 2025a] have shown that integrating residual modules into deep RL agents can significantly improve representation capability on complex visual tasks. We choose BRO-net [Nauman et al., 2024b] as a representative baseline and evaluate all methods on four challenging tasks from the DeepMind Control Suite [Tassa et al., 2018]. All the algorithm parameters follow the default settings. We set the empirical threshold $\tau = 0.01$ for ReGraMa, and use the same ReDo's hyperparameters. The reset period is fixed at 1000 steps. Further details are provided in Appendix A.1.

Main results. Results in Fig. 8 show that ReGraMa can accurately reset the neurons with low activity in each stream of the multi-branch network, thus effectively maintaining the learning capacity of the deep RL agent on the four complex tasks and enabling continual learning. In contrast, ReDo performs poorly. This performance gap arises because, as discussed in Sec. 4.1, ReDo misidentifies inactive neurons in multi-branch networks, undermining the effectiveness of its reset schedule.

Network scaling. To assess scalability, we evaluate both ReDo and ReGraMa under increased network depth using the BRO-net architecture across four challenging environments: Dog Walk, Dog Stand, Humanoid Walk, and Humanoid Stand. As the network grows deeper, training stability and gradient signal propagation typically become harder to maintain, often leading to degraded performance or inefficient utilization of added capacity. In this context, we analyze whether the gradient-based reset criterion of ReGraMa remains effective under larger model scales.

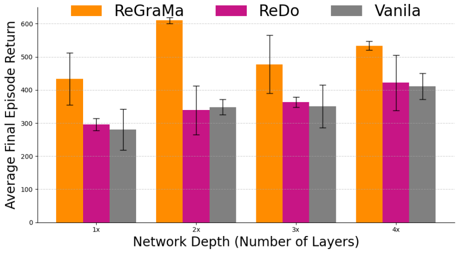


Figure 9: **ReGraMa is more robust under network scaling.** Results averaged over four DMC tasks with 12 seeds per method.

Results show that ReGraMa consistently preserves stable performance improvements as network depth increases, demonstrating that its gradient-driven measure generalizes well across scales. In contrast, activation-based resetting (ReDo) fails to exploit the additional representational capacity, showing marginal gains at best and even degrading performance in the 2× model. These findings indicate that ReGraMa scales more gracefully with model size, effectively leveraging deeper architectures without introducing instability.

5.2 Diffusion Model-Based Policy

Experiment setup. Recent works have demonstrated that diffusion models, due to their strong expressiveness over multi-modal distributions, can significantly improve RL performance on complex control tasks [Chi et al., 2023]. We use DACER [Wang et al., 2024], a recent online diffusion policy, as a baseline and test ReDo and ReGraMa on two MuJoCo-v4 tasks from the original paper. DACER relies on a Unet backbone with Swish activations. All parameters follow official defaults. Hyperparameters for ReGraMa and ReDo are the same as in Sec. 5.1, and the reset period is fixed at 1,000 steps. Full details are in Appendix A.2.

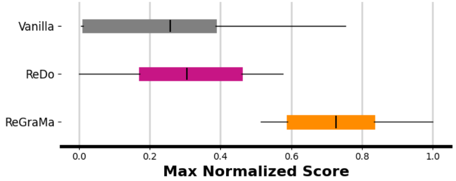


Figure 10: **Normalized scores for Ant and Walker2d.** Boxes show 4 seeds, whiskers indicate min/max, the midline denotes the median.

Results. As shown in Fig. 10, ReGraMa maintains robust and consistent performance on this complex architecture, whereas ReDo provides only marginal improvements. This discrepancy is explained by the ratio of inactive neurons in Fig. 11, which reveals that ReDo fails to reset neurons that have lost learning capacity. In contrast, ReGraMa accurately identifies and resets low-activity neurons, preserving the agent's ability to learn and improving overall performance.

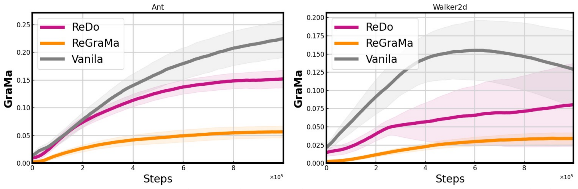
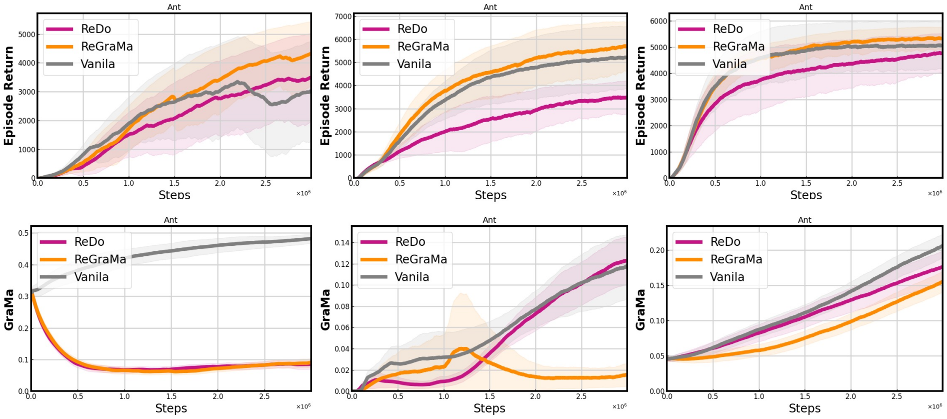


Figure 11: **Proportion of inactive neurons during training across two MuJoCo tasks** (Ant and Walker2d). ReGraMa maintains a consistently lower ratio of inactive neurons.



(a) ReLU (b) Sigmoid (c) Tanh

Figure 12: **Effect of activation functions on performance and neuron inactivity.** (Top row) Episode return across training under different activation functions (ReLU, Sigmoid, Tanh). (Bottom row) Corresponding proportion of inactive neurons. ReGraMa maintains higher performance and lower inactivity levels across all activations, indicating that gradient-based resetting generalizes effectively beyond specific nonlinearities. Curves are averaged over four seeds.

5.3 Activation Function Variants

Experiment setup. Nadimpalli et al. [2025] shows that saturated activation functions in the hidden layers may improve RL agents' performance. Following their setup, we replace ReLU in SAC with Tanh and Sigmoid, keeping all other parameters as default. This enables us to evaluate the robustness of ReGraMa across various activation functions while minimizing the influence of extraneous factors on the experimental outcomes. We set $\tau = 0$ for ReGraMa, and configure ReDo as in Sec. 5.1. All methods are evaluated on the challenging Ant task. Hyperparameter are provided in Appendix A.3.

Results. Results in Fig. 12 show that ReGraMa accurately identifies low-quality neurons, mitigating neuronal inactivity and avoiding instability caused by false reset. While ReDo performs well under the ReLU, its performance degrades significantly with other activation functions, sometimes failing below vanilla SAC. In Fig. 12(c), although ReGraMa outperforms ReDo under the Tanh activation function, the proportion of inactive neurons gradually increases during training. We leave further investigation into whether this behavior stems from the perspective of the activation function or reset mechanism to future work.

6 Discussion

This research focuses on the critical issue of neuronal activity loss during training in deep RL agents. We show that the commonly used τ-dormant neuron ratio (ReDo), which relies on neuron activations, struggles to capture learning activity in modern, highly parameterized agents. This limitation arises because activation sparsity does not directly reflect a neuron's contribution to learning. We shift focus from activations to gradients, and introduce GraMa, a simple, efficient, and architecture-agnostic metric based on gradient magnitude. GraMa enables accurate tracking of neuron-level learning dynamics across a broad range of network types, including residual and diffusion-based policies, and substantially recovers learning activity via guiding neuron resetting (Sec. 4.2). Our findings suggest that even high-capacity policies in deep RL suffer from underutilization at the neuron level, which may hinder generalization, multi-task transfer, and continual adaptation as in supervised learning [Dohare et al., 2024]. By providing a lightweight diagnostic and intervention tool, GraMa opens the door to more principled approaches for maintaining continuous learning ability in deep RL agents.

Limitations. Our evaluation is limited to three representative neural architectures due to computational constraints. Future work will extend GraMa to more complex settings, including large-scale transformer policies and multi-task environments. We aim for GraMa to serve as a practical tool for diagnosing and preserving learning capacity in deep RL agents.

Acknowledgment This work is supported by the National Natural Science Foundation of China 62406266. The authors would like to thank the reviewers for providing valuable feedback on the paper. We would also like to thank the Python community Van Rossum and Drake Jr [1995], Oliphant [2007] for developing tools that enabled this work, including NumPy Harris et al. [2020], Matplotlib Hunter [2007], Jupyter Kluyver et al. [2016], and Pandas McKinney [2013].

Broader Impact This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C. Machado. Loss of plasticity in continual deep reinforcement learning. *ArXiv*, abs/2303.07507, 2023. URL https://api.semanticscholar.org/CorpusID:257504763.

Abien Fred Agarap. Deep learning using rectified linear units (relu). *ArXiv*, abs/1803.08375, 2018. URL https://api.semanticscholar.org/CorpusID:4090379.

Jordan T. Ash and Ryan P. Adams. On the difficulty of warm-starting neural network training. *ArXiv*, abs/1910.08475, 2019. URL https://api.semanticscholar.org/CorpusID:204788802.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub W. Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019. URL https://api.semanticscholar.org/CorpusID:209376771.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *ArXiv*, abs/1606.01540, 2016. URL https://api.semanticscholar.org/CorpusID:16099293.

Roger Creus Castanyer, Johan Obando-Ceron, Lu Li, Pierre-Luc Bacon, Glen Berseth, Aaron Courville, and Pablo Samuel Castro. Stable gradients for stable learning at scale in deep reinforcement learning, 2025. URL https://arxiv.org/abs/2506.15544.

Johan Samir Obando Ceron, Aaron Courville, and Pablo Samuel Castro. In value-based deep reinforcement learning, a pruned network is a good network. In *International Conference on Machine Learning*, pages 38495–38519. PMLR, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *ArXiv*, abs/2303.04137, 2023. URL https://api.semanticscholar.org/CorpusID:257378658.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, Ashique Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632:768 – 774, 2024. URL https://api.semanticscholar.org/CorpusID:259251905.

Xiaoyi Dong, Jian Cheng, and Xi Sheryl Zhang. Maximum entropy reinforcement learning with diffusion policy. *ArXiv*, abs/2502.11612, 2025. URL https://api.semanticscholar.org/CorpusID:276408472.

Mohamed Elsayed and A. Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sKPzAXoylB.

Vivek Farias and Adam Daniel Jozefiak. Self-normalized resets for plasticity in continual learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=G82uQztzxl.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. URL https://api.semanticscholar.org/CorpusID:206594692.

John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03): 90–95, 2007.

Tianying Ji, Yongyuan Liang, Yan Zeng, Yu Luo, Guowei Xu, Jiawei Guo, Ruijie Zheng, Furong Huang, Fuchun Sun, and Huazhe Xu. Ace : Off-policy actor-critic with causality-aware entropy regularization. *ArXiv*, abs/2402.14528, 2024. URL https://api.semanticscholar.org/CorpusID:267782426.

Arthur Juliani and Jordan T. Ash. A study of plasticity loss in on-policy deep reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=MsUf8kpKTF.

Thomas Kluyver, Benjain Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter Development Team. Jupyter Notebooks—a publishing format for reproducible computational workflows. In *IOS Press*, pages 87–90. 2016. doi: 10.3233/978-1-61499-649-1-87.

Alex Krizhevsky. Learning multiple layers of features from tiny images. In *Arxiv*, 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. In *CoLLAs*, 2023. URL https://api.semanticscholar.org/CorpusID:261076021.

Hojoon Lee, Hyeonseo Cho, Hyunseung Kim, Donghu Kim, Dugki Min, Jaegul Choo, and Clare Lyle. Slow and steady wins the race: Maintaining plasticity with hare and tortoise networks. *ArXiv*, abs/2406.02596, 2024. URL https://api.semanticscholar.org/CorpusID:270258586.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R. Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=jXLiDKsuDo.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. *ArXiv*, abs/2502.15280, 2025b. URL https://api.semanticscholar.org/CorpusID:276558261.

Yang Li, Zhichen Dong, Yuhan Sun, Weixun Wang, Shaopan Xiong, Yijia Luo, Jiashun Liu, Han Lu, Jiamang Wang, Wenbo Su, Bo Zheng, and Jun-Feng Yan. Attention illuminates llm reasoning: The preplan-and-anchor rhythm enables fine-grained policy optimization. 2025. URL https://api.semanticscholar.org/CorpusID:282102960.

Jiashun Liu, Johan Samir Obando Ceron, Aaron Courville, and Ling Pan. Neuroplastic expansion in deep reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=20qZK2T7fa.

Jiashun Liu, Johan S. Obando-Ceron, Han Lu, Yancheng He, Weixun Wang, Wenbo Su, Bo Zheng, Pablo Samuel Castro, Aaron C. Courville, and Ling Pan. Asymmetric proximal policy optimization: mini-critics boost llm reasoning. 2025b. URL https://api.semanticscholar.org/CorpusID:281724081.

Rongrong Liu, Florent Nageotte, Philippe Zanne, Michel de Mathelin, and Birgitta Dresp. Deep reinforcement learning for the control of robotic manipulation: A focussed mini-review. *ArXiv*, abs/2102.04148, 2021. URL https://api.semanticscholar.org/CorpusID:231846753.

Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, Jian Hu, Shengyi Huang, Siran Yang, Jiamang Wang, Wenbo Su, and Bo Zheng. Part i: Tricks or traps? a deep dive into rl for llm reasoning. *ArXiv*, abs/2508.08221, 2025c. URL https://api.semanticscholar.org/CorpusID:280566935.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Hao Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *ArXiv*, abs/2505.18719, 2025. URL https://api.semanticscholar.org/CorpusID:278904856.

Lu Lu, Yanhui Su, and George Em Karniadakis. Collapse of deep and narrow neural nets. *ArXiv*, abs/1808.04947, 2018. URL https://api.semanticscholar.org/CorpusID:81981236.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, H. V. Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. *ArXiv*, abs/2402.18762, 2024. URL https://api.semanticscholar.org/CorpusID:268063557.

Guozheng Ma, Lu Li, Sen Zhang, Zixuan Liu, Zhen Wang, Yixin Chen, Li Shen, Xueqian Wang, and Dacheng Tao. Revisiting plasticity in visual reinforcement learning: Data, modules and training stages. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=0aR1s9YxoL.

Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Soft diffusion actor-critic: Efficient online reinforcement learning for diffusion policy. *ArXiv*, abs/2502.00361, 2025. URL https://api.semanticscholar.org/CorpusID:276094359.

Walter Mayor, Johan Obando-Ceron, Aaron Courville, and Pablo Samuel Castro. The impact of on-policy parallelized data collection on deep reinforcement learning networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=cnqyzuZhSo.

Wes McKinney. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 1 edition, February 2013. ISBN 9789351100065. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1449319793.

Kalyan Varma Nadimpalli, Shashank Reddy Chirra, Pradeep Varakantham, and Stefan Bauer. Evolving RL: Discovering new activation functions using LLMs. In *Towards Agentic AI for Science: Hypothesis Generation, Comprehension, Quantification, and Validation*, 2025. URL https://openreview.net/forum?id=H2x9juCuJg.

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. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample-efficient continuous control. In *NeurIPS 2024*, 2024b. URL https://openreview.net/forum?id=WTxWRQlk8x.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron C. Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:248811264.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and Andre Barreto. Deep reinforcement learning with plasticity injection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=jucDLW6G9l.

Johan Obando Ceron, Marc Bellemare, and Pablo Samuel Castro. Small batch deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36:26003–26024, 2023.

Johan Obando-Ceron, Ghada Sokar, Timon Willi, Clare Lyle, Jesse Farebrother, Jakob Foerster, Karolina Dziugaite, Doina Precup, and Pablo Samuel Castro. Mixtures of experts unlock parameter scaling for deep rl. In *Proceedings of the 41st International Conference on Machine Learning*, pages 38520–38540, 2024.

Travis E. Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3): 10–20, 2007. doi: 10.1109/MCSE.2007.58.

Haoyuan Qin, Chennan Ma, Mian Deng, Zhengzhu Liu, Songzhu Mei, Xinwang Liu, Cheng Wang, and Siqi Shen. The dormant neuron phenomenon in multi-agent reinforcement learning value factorization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=4NGrHrhJPx.

Allen Z. Ren, Justin Lidard, Lars Lien Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mEpqHvbD2h.

Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level Atari with human-level efficiency. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 30365–30380. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/schwarzer23a.html.

Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. *ArXiv*, abs/2107.03380, 2021. URL https://api.semanticscholar.org/CorpusID:235755271.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024. URL https://api.semanticscholar.org/CorpusID:267412607.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pages 32145–32168. PMLR, 2023.

Ghada Sokar, Johan Obando-Ceron, Aaron Courville, Hugo Larochelle, and Pablo Samuel Castro. Don't flatten, tokenize! unlocking the key to softmoe's efficacy in deep rl. *arXiv preprint arXiv:2410.01930*, 2024.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *ArXiv*, abs/1801.00690, 2018. URL https://api.semanticscholar.org/CorpusID:6315299.

Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang WU, Jingliang Duan, and Shengbo Eben Li. Diffusion actor-critic with entropy regulator. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=l0c1j4QvTq.

Timon Willi, Johan Samir Obando Ceron, Jakob Nicolaus Foerster, Gintare Karolina Dziugaite, and Pablo Samuel Castro. Mixture of experts in a mixture of rl settings. In *Reinforcement Learning Conference*.

Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, Shuzhen Li, Yanjie Ze, Hal Daum'e, Furong Huang, and Huazhe Xu. Drm: Mastering visual reinforcement learning through dormant ratio minimization. *ArXiv*, abs/2310.19668, 2023. URL https://api.semanticscholar.org/CorpusID:264796749.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *ArXiv*, abs/2107.09645, 2021. URL https://api.semanticscholar.org/CorpusID:236134152.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Honglin Yu, Weinan Dai, Yuxuan Song, Xiang Wei, Haodong Zhou, Jingjing Liu, Wei Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yong-Xu Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *ArXiv*, abs/2503.14476, 2025. URL https://api.semanticscholar.org/CorpusID:277104124.

Mo Zhou and Rong Ge. How does gradient descent learn features — a local analysis for regularized two-layer neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=XYw051ZmUn.

A Experimental Details

A.1 Residual network based policy

BRO-net [Nauman et al., 2024b] is the first model-free algorithm to achieve near-optimal policies in the notoriously challenging Dog and Humanoid tasks. Its residual module-based architecture also gives it a powerful ability to scale up, making it widely concerned [Lee et al., 2025b,a]. To this end, we choose BRO-net as the advanced agent for the multi-branch architecture to test the effectiveness of our metrics. And our implementation of BRO-net is based on the official implementation [3].

BRO-net Architecture The detailed structure of the core block used in BRO-net is shown in Tab. 1. The comprehensive Actor-Critic structure build by the above blocks is outlined in Tab. 2.

Table 1: BroNetBlock Structure. H_b denotes the block's internal hidden dimension (e.g., 256).

Step	Layer Configuration
1. FC Layer	Linear(H_b, H_b)
2. Norm + Act	LayerNorm(H_b), ReLU
3. FC Layer	Linear(H_b, H_b)
4. Norm	LayerNorm(H_b)
5. Residual	Output = Step 4 Output + Block Input

Table 2: Whole network architectures. The structure of BroNetBlock is detailed in Table 1.

Layer	Actor Network	Critic Network (per Critic)
Fully Connected	(state dim,256)	(state dim + action dim, 256)
LayerNorm	LayerNorm	LayerNorm
Activation	ReLU	ReLU
BroNetBlock	$N\times$ BroNetBlock	$N\times$ BroNetBlock
Fully Connected	(256, 2 $\times$ action dim)	(256, 1)
Activation	Tanh	None

Hyperparameter setting The shared hyperparameters of the BRO-net algorithm utilized in all our experiments are outlined in Tab. 3. Both Redo and `Grama` were configured with the recommended values for τ (0.01 for `Grama` and 0.02 for ReDo) and the same reset frequency (every 1000 steps). To reproduce the learning curves shown in the main text, we advise using seeds ranging from 0 to 4. For the scale experiments, we increased the number of BroNetBlock from 1 to 4 in both actor and critic.

Table 3: Hyperparameter settings for BRO

Hyperparameter	Value
Actor Learning Rate	1×10^{-4}
Critic Learning Rate	1×10^{-3}
Replay Ratio	2
Discount Factor (γ)	0.99
Batch Size	128
Buffer Size	1×10^{6}
Actor BroNetBlock	1
Critic BroNetBlock	2
Reset Specific Parameters	
Reset τ	0.01
Reset Frequency	1000

[3] `https://github.com/naumix/BiggerRegularizedOtimistic_Torch`

A.2 Diffusion model based policy

Network Architecture As one of the recent works to successfully construct online policies based on the diffusion model, Dacer has received extensive attention from the community and has been used as a baseline by some recent studies [Dong et al., 2025, Ma et al., 2025]. To test the effectiveness of our metrics in the advanced diffusion model policies, we chose the official Swish activation function and U-net based Dacer as a baseline with complex architecure. We reproduce the version of the code that introduces the two neuronal metrics into the policy model, based on the official code of DACER[4]. Our detailed Structures were shown in Tab. 4.

Table 4: Network Structures for DACER.

Layer	Actor Network	Critic Network
Fully Connected	(state dim + time embedding dim, 256)	(state dim + action dim, 256)
Activation	ReLU	ReLU
Fully Connected	(256, 256)	(256, 256)
Activation	ReLU	ReLU
Fully Connected	(256, action dim)	(256, 2)

Hyperparameter setting Our experiments adhere to the hyperparameter listed in Tab. 5. τ for ReDo: 0.02; τ for `Grama`: 0.01; reset frequency (every 1000 steps). To reproduce the learning curves shown in the main text, we advise using seeds ranging from 0 to 4.

Table 5: Hyperparameters for DACER Training

Hyperparameter	Value
Actor Learning Rate	3×10^{-4}
Critic Learning Rate	3×10^{-4}
Alpha Learning Rate	3×10^{-2}
Discount Factor (γ)	0.99
Batch Size	256
Replay Buffer Size	1×10^{6}
Target Network Update Rate (τ)	5×10^{-3}
Policy Update Delay	2
Hidden Layer Size	256
Reward Scale	1.0
Reset Specific Parameters	
Reset τ	0.01
Reset Frequency	1000

A.3 MLP-based SAC

Network Architecture We use CleanRL for SAC (also Resnet SAC) implementation, which can be found at `https://github.com/vwxyzjn/cleanrl`. This library is a reliable open-source resource for deep reinforcement learning, designed in a PyTorch-friendly manner. And the detailed structure is shown in Tab. 6.

Hyperparameter setting The shared hyperparameters for the SAC algorithm are summarized in Tab. 7. Note: We impose a maximum reset percentage limitation of 5% exclusively for the Humanoid task.

[4] https://github.com/happy-yan/DACER-Diffusion-with-Online-RL

Table 6: Network Structures for SAC

Layer	Actor Network	Critic Network
Fully Connected	(state dim, 256)	(state dim + action dim, 256)
Activation	ReLU	ReLU
Fully Connected	(256, 256)	(256, 256)
Activation	ReLU	ReLU
Fully Connected	(256, 2× action dim)	(256, 1)
Activation	Tanh	None

Table 7: Hyperparameters for the SAC

Hyperparameter	Value
Total Timesteps	3×10^6
Replay Buffer Size	1×10^6
Discount Factor (γ)	0.99
Target Smoothing Coefficient (τ)	0.005
Batch Size	256
Learning Starts	5×10^3
Policy Learning Rate	3×10^{-4}
Q-Network Learning Rate	1×10^{-3}
Policy Update Frequency	2
Target Network Update Frequency	1
Automatic Entropy Tuning	True
Reset Specific Parameters	
Reset τ	0
Reset Frequency	1000
Max Reset percentage	5% (Humanoid)

B ReGraMa is more robust to the threshold τ

To assess the robustness of both metrics across varying thresholds, we use BRO-net as the backbone and evaluate performance on the challenging DeepMind Control Suite Humanoid Walk task. By systematically varying the τ following the recommended setup, we found that ReGraMa consistently outperformed ReDo (Fig. 13). This proves that, even with relaxed restrictions, ReGraMa has less tendency to reset incorrectly.

C Additional experiments

We further evaluate four challenging DeepMind Control (DMC) tasks with tensor-based observations. The results in Tab. 8 show that, under the BRO-net architecture, ReGraMa consistently outperforms all competing methods, highlighting its robustness in complex network architectures. ReBron [Qin et al., 2024], which explicitly accounts for both overactive and dormant neurons, also achieves strong performance, although it slightly degrades performance on the Dog Walk task. In contrast, S&P [Ash and Adams, 2019] and CBP [Dohare et al., 2024] provide modest but consistent improvements across tasks.

We additionally include new ablation studies in the updated appendix to investigate the interaction between ReGraMa and commonly used regularization techniques. Following the optimal weight decay configuration reported in Lyle et al. [2024], we evaluate the combination of ReGraMa with weight decay and L2 initialization ($\lambda = 10^{-2}$) on the DMC Quadruped Run task using DrQv2 [Yarats et al., 2021] for 3M environment steps.

The results in Tab. 9 indicate that combining ReGraMa with either L2 initialization or weight decay individually further improves learning efficiency. Among these variants, L2 initialization, which preserves plasticity by regularizing the initial parameters, appears more compatible with our method.

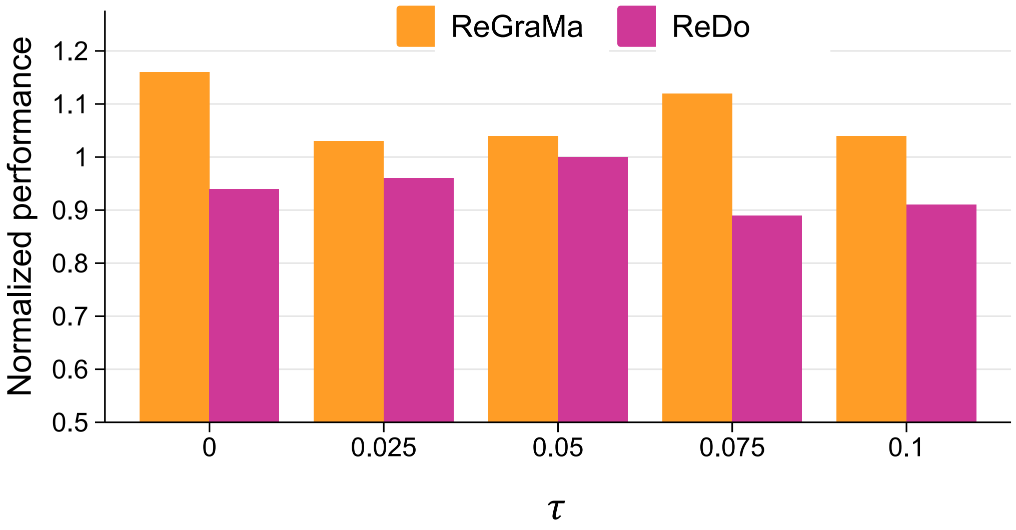


Figure 13: Sample uniformly from $[0, 0.1]$. Each bar is the average of 4 seeds.

In contrast, the simultaneous application of ReGraMa, L2 initialization, and weight decay does not yield additional gains. We leave a deeper investigation of the underlying interactions between ReGraMa and other regularization strategies to future work.

Table 8: Normalized performance relative to the vanilla BRO-net policy on DMC tasks (mean ± std over 3 runs).

Method	Humanoid stand	Humanoid Run	Dog stand	Dog walk
vanilla BRO-net policy (baseline)	1.0	1.0	1.0	1.0
ReGraMa	1.21 ± 0.03	1.16 ± 0.08	1.12 ± 0.08	1.08 ± 0.04
ReDo	1.17 ± 0.07	0.96 ± 0.03	0.92 ± 0.12	0.94 ± 0.06
S&P	1.05 ± 0.12	1.08 ± 0.04	0.95 ± 0.07	1.07 ± 0.02
ReBorn	1.13 ± 0.06	1.05 ± 0.07	1.02 ± 0.06	0.91 ± 0.13
CBP	1.18 ± 0.02	0.99 ± 0.12	1.06 ± 0.04	1.09 ± 0.07

Table 9: Effect of combining ReGraMa with L2 initialization and weight decay on DMC Quadruped Run (3M steps, DrQv2).

Method	Quadruped Run
vanilla policy	649.13 ± 182.43
ReGraMa	706 ± 127.25
ReGraMa + L2 init	742.36 ± 127.31
ReGraMa + weight decay	751.31 ± 94.26
ReGraMa + L2 init & weight decay	739.42 ± 104.58

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations can be seen in Conclusion of main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please refer to Sec.4 of the main paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We make our code available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code link can be found in abstrct, and the hyperparameter setting can be found in Appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental detail can be found in Experiment part. The hyperparameter setting can be found in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results are accompanied by error bars, confidence intervals. We detail the calculation of error bars, running steps and the number of seeds in the text description of each figure.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Experiments Compute Resources can be found in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We make sure to preserve anonymity.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper is only about promoting technological innovation and does not have social impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Ours paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the code package, and state which version of the asset is used (include a URL).

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.