FIRE: Frobenius-Isometry Reinitialization for Balancing the Stability–Plasticity Tradeoff

Isaac Han[1] **Sangyeon Park**[1] **Seungwon Oh**[1] **Donghu Kim**[2]
Hojoon Lee[2]* **Kyung-Joong Kim**[1]*

[1]Gwangju Institute of Science and Technology (GIST), Korea
[2]Korea Advanced Institute of Science and Technology (KAIST), Korea
{lssac7778, tkddus0421, sw980907}@gm.gist.ac.kr

Abstract

Deep neural networks trained on nonstationary data must balance stability (i.e., retaining prior knowledge) and plasticity (i.e., adapting to new tasks). Standard reinitialization methods, which reinitialize weights toward their original values, are widely used but difficult to tune: conservative reinitializations fail to restore plasticity, while aggressive ones erase useful knowledge. We propose FIRE, a principled reinitialization method that explicitly balances the stability–plasticity tradeoff. FIRE quantifies stability through Squared Frobenius Error (SFE), measuring proximity to past weights, and plasticity through Deviation from Isometry (DfI), reflecting weight isotropy. The reinitialization point is obtained by solving a constrained optimization problem, minimizing SFE subject to DfI being zero, which is efficiently approximated by Newton–Schulz iteration. FIRE is evaluated on continual visual learning (CIFAR-10 with ResNet-18), language modeling (OpenWebText with GPT-0.1B), and reinforcement learning (HumanoidBench with SAC and Atari games with DQN). Across all domains, FIRE consistently outperforms both naive training without intervention and standard reinitialization methods, demonstrating effective balancing of the stability–plasticity tradeoff.

1 Introduction

Deep neural networks are typically trained under a fixed, stationary data distribution (Brown et al., 2020; Podell et al., 2024). However, many real-world applications require models to adapt continually as new data and shifting distributions emerge. In computer vision, autonomous driving systems must recognize unseen traffic signs, road layouts, or weather conditions that were absent during training (Verwimp et al., 2023). Large language models, trained once and deployed with a fixed knowledge cutoff date, quickly become outdated unless continually updated (Ke et al., 2023). Likewise, robots deployed in dynamic physical environments must adjust their perception and control policies as the environment changes (Wołczyk et al., 2021). In all of these domains, a central challenge is reliable adaptation to nonstationary data while preserving prior knowledge.

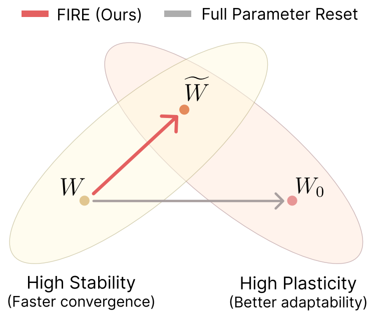

Figure 1: **Illustration of FIRE.** Solving a constrained optimization problem, FIRE places weights at the intersection of high-stability and high-plasticity manifolds.

This challenge is often framed as a balance between two competing properties: *stability*, the retention of learned knowledge, and *plasticity*, the ability to incorporate new information (Mermillod et al., 2013). Different research communities emphasize these properties to varying degrees. Conventional continual learning assumes limited access to past data, prioritizing stability to mitigate catastrophic forgetting (Kirkpatrick et al., 2017; Rebuffi et al., 2017; Rusu et al., 2016). In contrast, most

* Equally Corresponding

foundation models and robotic agents are trained on expanding datasets where past data remain accessible (Achiam et al., 2023; Team et al., 2025), making plasticity loss a central challenge (Lyle et al., 2023; Berariu et al., 2021). In this regime, stability is less about preserving past knowledge and more about accelerating adaptation to new tasks by leveraging prior representations. Motivated by these real world scenarios, we study the stability-plasticity tradeoff under the assumption of access to past data during continual learning.

Existing approaches to mitigating plasticity loss fall broadly into two categories: regularization-based and reinitialization-based. Regularization-based methods constrain parameters or features near their initialization (Kumar et al., 2025b; Lyle et al., 2022), or enforce weight orthogonality (Chung et al., 2024). While these methods can preserve a favorable geometry for future learning, overly strong constraints slow convergence, and overly weak ones fail to prevent plasticity degradation. Reinitialization-based methods instead reset weights to earlier checkpoints when new data arrive (Ash & Adams, 2020; Nikishin et al., 2022; Lee et al., 2024a; Shin et al., 2024). Their advantage lies in avoiding interference with current optimization, often yielding faster adaptation with lower overhead. However, they also suffer from a tuning dilemma: aggressive resets erase useful knowledge, while conservative ones provide little plasticity gain.

We aim to resolve this dilemma by treating reinitialization as a principled constrained optimization problem. Our approach relies on two complementary measures that capture the core dimensions of the stability–plasticity tradeoff. First, we define stability using the Squared Frobenius Error (SFE) between current and past weights, which measures the sum of squared differences across all weight entries. A smaller SFE indicates greater similarity, meaning the model remains closer to its previous representations. For plasticity, prior work has linked plasticity loss to sharp loss curvature (Lyle et al., 2023), dormant neurons (Sokar et al., 2023), and low rank features (Kumar et al., 2021a), but these metrics depend on incoming data and are non differentiable, limiting their use for optimization. We instead propose the Deviation from Isometry (DfI) (Pennington et al., 2017; Xiao et al., 2018), which measures how close weight matrices are to orthonormal. We show that reducing DfI simultaneously decreases curvature, prevents neuron dormancy, increases feature rank, and remains differentiable, making it a practical measure of plasticity to optimize. A formal proof is provided in Section 3.

We propose FIRE (Frobenius–Isometry REinitialization), which minimizes the SFE subject to the DfI being 0. As illustrated in Figure 1, FIRE avoids the pitfalls of either overly conservative or aggressive reinitialization by projecting weights onto the isotropy manifold while remaining close to their previous subspace. While directly solving this constrained optimization is costly, it can be implemented efficiently with the Newton–Schulz iteration, adding less than 1% to training time.

We evaluate FIRE on continual learning benchmarks in vision, language, and reinforcement learning, assuming access to past data. For vision, we split CIFAR-10, CIFAR-100, and Tiny-ImageNet into chunks under random or class-incremental protocols using ResNet He et al. (2016) and Vision Transformer Dosovitskiy et al. (2020). For language, we use a warm-start setup where GPT-0.1B Karpathy (2023) is pretrained on WikiText-103 and then continually trained on a mixture of OpenWebText and WikiText-103. For reinforcement learning, we test continuous control with SAC Haarnoja et al. (2018) on HumanoidBench Sferrazza et al. (2024) and discrete control with DQN Mnih et al. (2015) on Atari Bellemare et al. (2013). For vision and language tasks, reinitialization is applied whenever new data arrive, while in reinforcement learning it is applied once at the midpoint of training. Across all domains, FIRE consistently outperforms naive training and standard reinitialization, showing its effectiveness as a unified solution to the stability plasticity tradeoff.

2 Related Work

2.1 Stability-Plasticity tradeoff

The stability–plasticity tradeoff (Mermillod et al., 2013; Kim & Han, 2023) is a fundamental challenge in continual learning. Stability refers to the ability of a model to preserve previously acquired knowledge and avoid catastrophic forgetting when exposed to new data. Plasticity refers to the ability of a model to adapt flexibly and effectively to novel tasks. These two properties often conflict with each other since strong stability can make the model rigid and resistant to new learning while excessive plasticity can lead to the loss of past knowledge.

Research in continual learning has therefore focused on methods that balance these competing requirements such as constraining parameter updates through regularization (Kirkpatrick et al., 2017), revisiting earlier data through replay (Rebuffi et al., 2017; Rolnick et al., 2019; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018; Aljundi et al., 2019), or designing architectures that separate parameters across tasks (Rusu et al., 2016; Mallya & Lazebnik, 2018; Mallya et al., 2018; Yoon et al., 2017; Wortsman et al., 2020). Their aim is to develop models that can maintain previously learned skills while remaining adaptive to new experiences.

2.2 Loss of plasticity

Deep learning has traditionally been studied under stationary datasets (Glorot & Bengio, 2010; He et al., 2015), yet real-world applications often involve non-stationary streams (Shen et al., 2024; Kumar et al., 2025a). Training in such environments leads to a loss of plasticity (Lyle et al., 2023; Dohare et al., 2024; Kumar et al., 2025b), where models fail to adapt to new distributions. Prior work has identified potential indicators of this phenomenon, including dormant neurons (Sokar et al., 2023; Xu et al., 2023), shifts in pre-activations (Lyle et al., 2024), feature rank collapse (Kumar et al., 2021a), and diverging weight magnitudes (Lyle et al., 2024).

Loss of plasticity hinders not only the ability to fit the training data, but also the ability to generalize to unseen data. Models trained incrementally often generalize worse than those trained from scratch (Ash & Adams, 2020; Berariu et al., 2021; Lyle et al., 2025), due to factors such as diminished gradient norms (Ash & Adams, 2020), weak feature changes (Lyle et al., 2025), and the compounding effects of small pretraining datasets or noisy labels (Lee et al., 2024a).

To counteract plasticity loss, reinitialization-based strategies such as S&P (Ash & Adams, 2020), DASH (Shin et al., 2024) reinitialize weights into an intermediate checkpoint, weight regularizers constrain parameters to initialization or specific subspaces (Kumar et al., 2025b; Elsayed et al., 2024; Lewandowski et al., 2024a), and spectral or rank-based approaches explicitly maintain representation quality (Kumar et al., 2021a;b; He et al., 2024). Another approach proposed reinitializing at the neuronal level, based on the utility of each neuron (Sokar et al., 2023; Dohare et al., 2024; Elsayed & Mahmood, 2024). Recent work further leverages the fact that linear networks do not suffer from plasticity loss (Dohare et al., 2024; Lewandowski et al., 2024b; Park et al., 2025).

3 Method

In this section we explain how we frame the stability-plasticity tradeoff as a constrained optimization problem. To do this we first need two metrics: one for stability and one for plasticity loss. With these two pieces in place the optimization problem naturally emerges, and from this formulation we introduce our method FIRE, which provides an efficient approximation to the solution.

3.1 Measure for stability

To measure stability, we propose a simple yet effective metric, the *Squared Frobenius Error (SFE)*. SFE provides a natural way to quantify the preservation of learned information by comparing an original weight matrix W with its modified counterpart $\widetilde{W}$. Formally,

$$\text{SFE}(W, \widetilde{W}) = \|W - \widetilde{W}\|_F^2, \tag{1}$$

which captures the element-wise squared deviation between the two weight configurations. However, it remains unclear whether SFE can be used as a metric that can meaningfully capture similarity of feature representations. To clarify this point, we establish a theoretical link between SFE and the normalized feature covariance, a metric widely used in prior work to measure representation similarity (Lyle et al., 2025; Yang et al., 2022). In particular, we show that SFE provides an upper bound on the discrepancy between the normalized feature covariances of two distinct neural networks' output features (Theorem 1).

Theorem 1 (SFE bounds output feature covariance between two deep neural networks). *Let $\Theta = \{W^1, \ldots, W^L\}$ and $\widetilde{\Theta} = \{\widetilde{W}^1, \ldots, \widetilde{W}^L\}$ be the parameters of two depth-L feedforward networks with elementwise activations σ_ℓ (Lipschitz constants L_{σ_ℓ}).*

For an input batch $Z \in \mathbb{R}^{n\times d_0}$, we denote the layer outputs recursively by $H^0_\Theta(Z) = Z$ and $H^\ell_\Theta(Z) = \sigma_\ell(H^{\ell-1}_\Theta(Z)W^\ell)$. Let $B_\ell = \max\{\|W^\ell\|_2, \|\widetilde{W}^\ell\|_2\}$ be the maximum spectral norm of the weights in layer ℓ and $B^\ell_\Pi = \prod_{k=1}^\ell B_k$ the product across all layers. We further define $m_\ell = \min\{\|H^\ell_\Theta(Z)\|_F, \|H^\ell_{\widetilde{\Theta}}(Z)\|_F\} > 0$. The normalized feature covariances of the two networks are given by $C^\ell_\Theta(Z) = H^\ell_\Theta(Z)H^\ell_\Theta(Z)^\top/\|H^\ell_\Theta(Z)\|_F^2, \quad C^\ell_{\widetilde{\Theta}}(Z) = H^\ell_{\widetilde{\Theta}}(Z)H^\ell_{\widetilde{\Theta}}(Z)^\top/\|H^\ell_{\widetilde{\Theta}}(Z)\|_F^2$.

Then the difference between the output feature covariances of the two networks is bounded as follows:

$$\|C^\ell_\Theta(Z) - C^\ell_{\widetilde{\Theta}}(Z)\|_F^2 \;\le\; \frac{16\,\|Z\|_F^2}{m_\ell^2}\Big(\prod_{k=1}^{\ell} L_{\sigma_k}\Big)^2 B_\Pi^2 \left(\sum_{j=1}^{\ell} \tfrac{1}{B_j^2}\right) \mathrm{SFE}(\Theta, \widetilde{\Theta}).$$

In particular, if each activation is 1-Lipschitz ($L_{\sigma_k} \le 1$) and each spectral norm is bounded ($B_j \le S$), then

$$\|C^\ell_\Theta(Z) - C^\ell_{\widetilde{\Theta}}(Z)\|_F \;\le\; \frac{4\,\|Z\|_F}{m_\ell}\,\sqrt{\ell}\,S^{\ell-1}\,\sqrt{\mathrm{SFE}(\Theta, \widetilde{\Theta})}.$$

3.2 Measure for plasticity loss

In this work, we view plasticity loss as the situation where weights learned from previous data fail to serve as a favorable initialization point for new data. This perspective is particularly useful when designing reinitialization strategies that partially reinitialize weights before the arrival of new data. Prior analyses of plasticity loss suggest that one of its main causes is the increasing sharpness of the loss landscape curvature with respect to new data during training on the current task, which destabilizes optimization (Lyle et al., 2023). In addition, an increase in dormant neurons (Sokar et al., 2023) and a collapse in effective rank (Kumar et al., 2021a) have also been known to indicate plasticity loss.

However, as these measures are data-independent and non-differentiable, they are inappropriate for the optimization objective. Instead, we propose Deviation from Isometry (DfI) (Pennington et al., 2017; Xiao et al., 2018), an optimizable metric that closely connected to previous plasticity measure.

$$\mathrm{DfI}(W) = \|W^\top W - I\|_F^2. \tag{2}$$

Our theoretical analysis reveals that minimizing DfI results in a smoother loss landscape curvature (Theorem 2), a smaller number of dormant neurons (Theorem 4), and a higher effective rank (Theorem 3). This supports the use of DfI as a suitable measure for plasticity measure. In addition, minimizing DfI also makes the weights close to isotropy, which is known as a key property of favorable neural network initializations, termed as dynamical isometry (Xiao et al., 2018).

Theorem 2 (Hessian spectral norm bounded by layerwise DfIs)**.** *We assume that*

- *the inputs are whitened, so that the empirical covariance $\Sigma_Z = \frac{1}{n}Z^\top Z$ is approximately the identity: $\Sigma_Z := \frac{1}{n}Z^\top Z \approx I$*
- *for every sample i and relevant parameter vector u, the Hessian norm satisfies $|\nabla_u^2 \ell_i(u)|_2 \le \beta$, while the gradient norm is bounded by $|\nabla_u \ell_i(u)|_2 \le \gamma$.*
- *finally, we focus on a fixed ReLU activation pattern at the point of interest, so that the network is piecewise linear in that region. In particular, each diagonal gating matrix arising from the ReLU has operator norm ≤ 1.*

Let $\nu_k = 1 + \sqrt{\mathrm{DfI}(W_k)}$, then the Hessian spectral norm can be bounded as follows:

$$\big\|\nabla_\theta^2 \mathcal{L}(W_{1:L})\big\|_2 \;\le\; \beta \sum_{k=1}^{L} \prod_{j\ne k} \nu_j \;+\; 2\gamma \sum_{1\le k<\ell\le L} \prod_{j\notin\{k,\ell\}} \nu_j.$$

Theorem 3 (DfI controls effective rank)**.** *Let $Z \in \mathbb{R}^{n\times a}$ and $W \in \mathbb{R}^{a\times b}$ be an input and weight matrix, respectively, $\Phi = ZW$ be a feature matrix. Let $\Sigma_Z = \frac{1}{n}Z^\top Z$ be the empirical covariance of the inputs and let $W = QS$ be the right polar decomposition of W, where $Q \in \mathbb{R}^{a\times b}$ has* *orthonormal columns. Then* $\eta_1 \geq \cdots \geq \eta_d > 0$ *denote the positive eigenvalues of* $Q^\top \Sigma_Z Q$*, with* $d = \operatorname{rank}(Q^\top \Sigma_Z Q)$*, and* $\operatorname{srank}_\delta(\Phi)$ *be defined from the nonzero singular values* $\{\sigma_i(\Phi)\}_{i=1}^d$ *by* $\operatorname{srank}_\delta(\Phi) = \min\Big\{k : \ (\sum_{i=1}^k \sigma_i(\Phi) / \sum_{i=1}^d \sigma_i(\Phi)) \geq 1 - \delta\Big\}$.

Then $\varepsilon = \sqrt{\mathrm{DfI}(W)} < 1$ *gives a lower bound on the srank as left inequality below. If additionally* $\Sigma_Z = I$*, the bound simplifies to the right inequality below.*

$$\operatorname{srank}_\delta(\Phi) \ \geq \ \left\lceil \frac{(1-\delta)\, d}{\delta \sqrt{\frac{1+\varepsilon}{1-\varepsilon}} \sqrt{\frac{\eta_1}{\eta_d}} + (1-\delta)} \right\rceil, \qquad \operatorname{srank}_\delta(\Phi) \ \geq \ \left\lceil \frac{(1-\delta)\, d}{\delta \sqrt{\frac{1+\varepsilon}{1-\varepsilon}} + (1-\delta)} \right\rceil. \quad (3)$$

Theorem 4 (DfI bounds neuron dormancy)**.** *Let* $W \in \mathbb{R}^{a \times b}$ *and let* σ *be positive-homogeneous* ($\sigma(\alpha t) = \alpha \sigma(t)$ *for* $\alpha \geq 0$*). Assume the input vector* $z \sim \mathcal{N}(0, I_a)$ *(isotropic Gaussian). For neuron* j *with column* w_j *of* W*, define dormancy score of neuron* j *as* $s_j = \frac{\mathbb{E}_z[|\sigma(\langle z, w_j\rangle)|]}{\frac{1}{b}\sum_{k=1}^b \mathbb{E}_z[|\sigma(\langle z, w_k\rangle)|]}$*. If* $\varepsilon := \sqrt{\mathrm{DfI}(W)} < 1$*, then for all* j*,*

$$\sqrt{\frac{1-\varepsilon}{1+\varepsilon}} \ \leq \ s_j \ \leq \ \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}.$$

We provided detailed proofs in Appendix A.1. Theorem 2, 3, 4 demonstrates that reducing DfI is directly associated with maintaining a smoother loss landscape, a high effective rank, and large number of active units.

3.3 Balancing between stability and plasticity

To achieve low DfI while minimizing the loss of information, we formulate the problem as a constrained optimization. Specifically, we minimize the SFE between the original weights W and their orthogonalized counterpart $\widetilde{W}$, subject to the orthogonality constraint $\widetilde{W}^\top \widetilde{W} = I$, which is equivalent to requiring $\mathrm{DfI}(\widetilde{W}) = \| \widetilde{W}^\top \widetilde{W} - I \|_F^2 = 0$.

$$\min_{\widetilde{W}} \ \|W - \widetilde{W}\|_F^2 \quad \text{s.t.} \quad \widetilde{W}^\top \widetilde{W} = I. \quad (4)$$

This formulation is mathematically equivalent to the well-studied *Orthogonal Procrustes Problem* (Schönemann, 1966), whose solution can be expressed in closed form via the polar decomposition:

$$\widetilde{W}^\star = W\big(W^\top W\big)^{-\frac{1}{2}}. \quad (5)$$

While the optimization itself is classical, our contribution lies in leveraging this operation as a principled mechanism to balance stability and plasticity in neural networks. In particular, we reinterpret equation 5 as a projection that simultaneously drives the spectrum of W toward isotropy (low DfI) while maintaining stability (low SFE).

3.4 Approximating the solution

Directly computing $\widetilde{W}^\star$ exactly can be expensive, we efficiently approximate it using the Newton–Schulz iteration, making the approach scalable to large networks.

Our method, FIRE, orthogonalizes neural network weights after training on the current dataset but before learning on new data, using the Newton–Schulz iteration (Shown in Algorithm 1.) to efficiently approximate the above solution. Specifically, given a weight matrix $W \in \mathbb{R}^{m \times n}$, we apply the Newton–Schulz update defined as $X_{k+1} = aX_k + bX_k(X_k^\top X_k)$ with constants $a = 1.5$ and $b = -0.5$, where $X_0 = W / \|W\|_F$. This iterative process progressively drives the singular values of W toward 1, thereby enforcing approximate orthonormality.

Algorithm 1: Newton–Schulz Iteration (Pytorch-like)

```
# X: a two-dimensional matrix
# N: number of iteration
a, b = (1.5, -0.5)
X = X / X.norm()
for _ in range(N):
    A = X.T @ X
    X = a * X + b * (X @ A)
return X
```

In convolutional layers, the update is applied kernel-wise along the spatial dimensions, ensuring that each convolutional filter is orthogonalized independently.

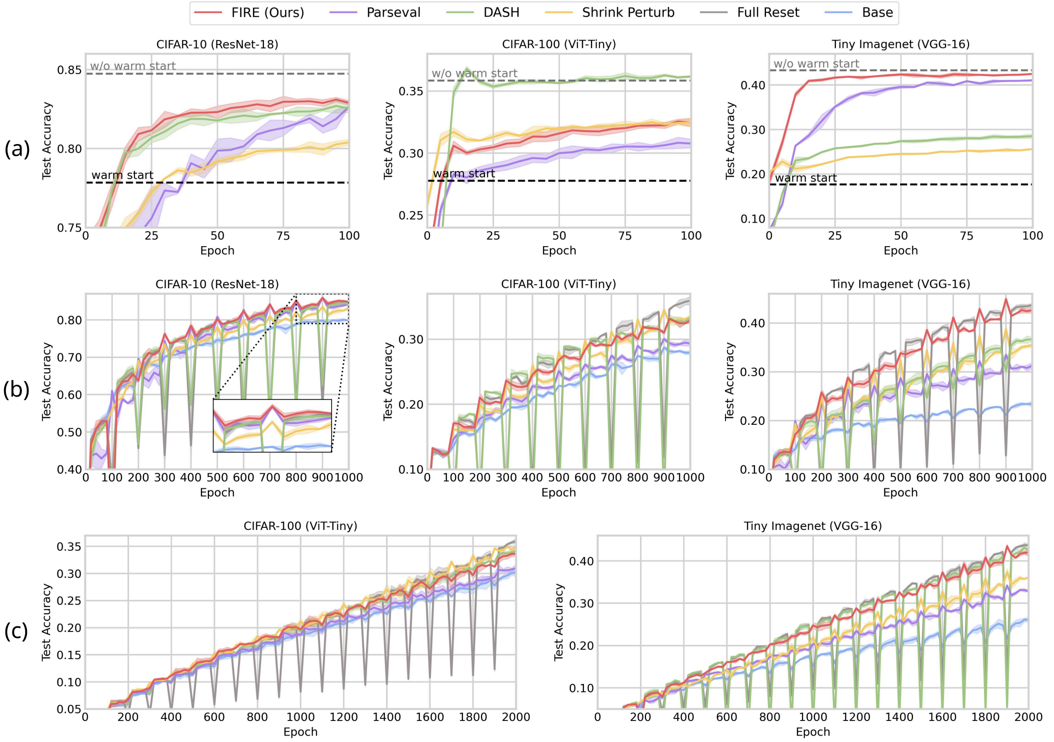

Figure 2: **Continual visual learning results.** Warm-start setting (a): training begins with only 10% of the data before continuing on the full dataset. Continual setting (b): the dataset is revealed in ten stages, expanding from 10% to 100% in 10% increments. Class-incremental setting (c): new classes are introduced over 20 phases, with an equal number of classes added at each phase.

4 Experiments

To demonstrate the effectiveness of FIRE, we evaluated it on three settings: continual visual learning, continual pretraining of LLMs, and reinforcement learning.

4.1 Continual Visual Learning

In continual visual learning experiments, we evaluated FIRE on various dataset–architecture pairs: the CIFAR-10 dataset with ResNet-18, the CIFAR-100 dataset with ViT-Tiny, and the Tiny ImageNet dataset with VGG-16.

To evaluate both the ability of FIRE to recover plasticity and its capacity to restore performance after a reset, we compare against two representative reset-based baselines: S&P (Ash & Adams, 2020), and DASH (Shin et al., 2024). Also, to evaluate the impact of regularization on the convergence speed, we adopt Parseval regularization (Chung et al., 2024), which constrains the weights to remain close to orthogonal, as a baseline. While FIRE minimizes the Deviation from Isometry (DfI) when resetting, Parseval regularization enforces the same constraint continuously throughout training. Thus, both FIRE and Parseval regularization share the same underlying optimization objective. Detailed experiment settings for continual visual learning are provided in Appendix A.4.1.

Following Ash & Adams (2020) and Lee et al. (2024a), we evaluate our approach in the warm-start setting, where the model is first trained on a subset of the dataset and then on the full dataset. Since plasticity loss is most severe when the subset ratio is small (Lee et al., 2024a), we warm-start with only 10% of the data before continuing on the entire dataset. As shown in Figure 2 (a), FIRE provides consistent performance gains across all three benchmarks. In particular, on CIFAR-10 with ResNet-18 and Tiny ImageNet with VGG-16, FIRE outperforms all baselines, demonstrating a strong ability to recover plasticity. On CIFAR-100 with ViT-Tiny, however, the

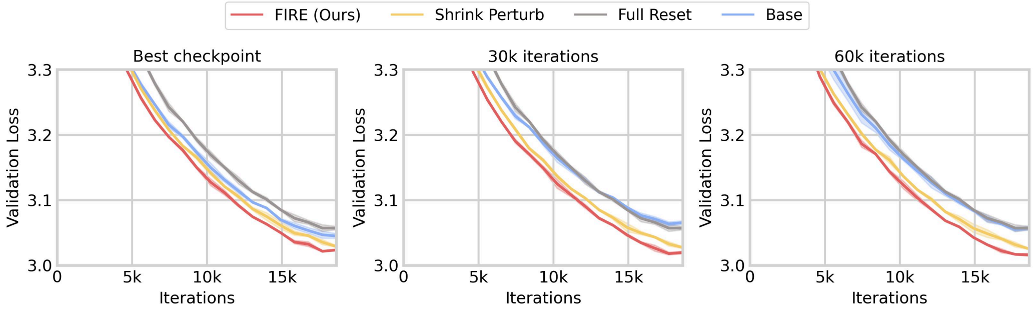

Figure 3: **Continual pretraining of GPT-0.1B.** Models are first pretrained on WikiText-103 and then continually trained on a new dataset consisting of a mixture of OpenWebText and WikiText-103. From left to right, results correspond to models initialized from the best checkpoint during pretraining, from 30k pretraining iterations, and from 60k pretraining iterations.

improvement is less pronounced. In this setting, DASH, which employs a data-dependent shrinking strategy, proves especially effective, suggesting that guidance from data can be beneficial when reinitializing transformer architectures. Notably, Parseval regularization converges more slowly than FIRE, particularly on CIFAR-10 with ResNet-18 and Tiny ImageNet with VGG-16. This suggests that continuously enforcing the constraint during training can hinder convergence, leading to longer training and incurring additional computational cost.

To examine whether these findings hold in a setting where data are continuously added, which is a more realistic and natural setting, we evaluate FIRE in the continual setting (Lee et al., 2024a). Here, training is divided into ten stages, starting with 10% of the dataset and adding an additional 10% at each stage. In this way, data gradually expand from 10% to the full 100%. As shown in Figure 2 (b), FIRE delivers consistent gains across all datasets. The improvements are particularly pronounced on CIFAR-10 with ResNet-18 and Tiny ImageNet with VGG-16, while on CIFAR-100 with ViT-Tiny it achieves performance comparable to the best alternatives. In contrast, full reset and DASH suffer a sharp drop immediately after each reset, and although S&P avoids such drops, its performance remains suboptimal compared to FIRE. In contrast, FIRE incurs only a slight or negligible drop, suggesting that it successfully balances stability and plasticity, thereby achieving high performance with minimal drop in performance.

To assess the effectiveness of FIRE under large distribution shifts, we conducted experiments in a class-incremental learning scenario, which is widely used setup in the continual learning literature (Rebuffi et al., 2017; Dohare et al., 2024; Lewandowski et al., 2024b). New classes were gradually introduced at regular intervals. The training process was divided into 20 phases, with an equal number of classes added in each phase. Since CIFAR-10 does not contain a sufficient number of classes for this setting, we exclude it in this experiment. Figure 2 (c) reports the results in the class-incremental setting. Consistent with our earlier findings, FIRE shows strong performance without exhibiting a performance drop after resets by effectively balancing stability and plasticity, and full reset and DASH show sharp drop after reset while S&P show suboptimal performance.

4.2 CONTINUAL PRETRAINING OF LLMS

Setup. We also tested FIRE in the continual pretraining of LLMs. We first pretrained a GPT-0.1B model on WikiText-103 and then trained on a combination of OpenWebText and WikiText-103. For the second phase, we used the best, 30k, and 60k checkpoints from initial pretraining to examine how plasticity loss worsens beyond the best checkpoint and how effectively FIRE mitigates this degradation at different stages. We present the detailed settings for the LLM experiments in Appendix A.4.2.

Results. As shown in Figure 3, the gap between the base model and full reset narrows as pretraining progresses, since the base model's validation loss increases with longer training. This aligns with prior findings that plasticity loss becomes more severe as pretraining duration grows (Ash & Adams, 2020). While S&P improves performance by moving parameters toward intermediate trade-off points between stability and plasticity, it remains suboptimal compared to FIRE, which achieves a more principled balance. Notably, FIRE was applied without any tuning, using a fixed 5 iterations, whereas

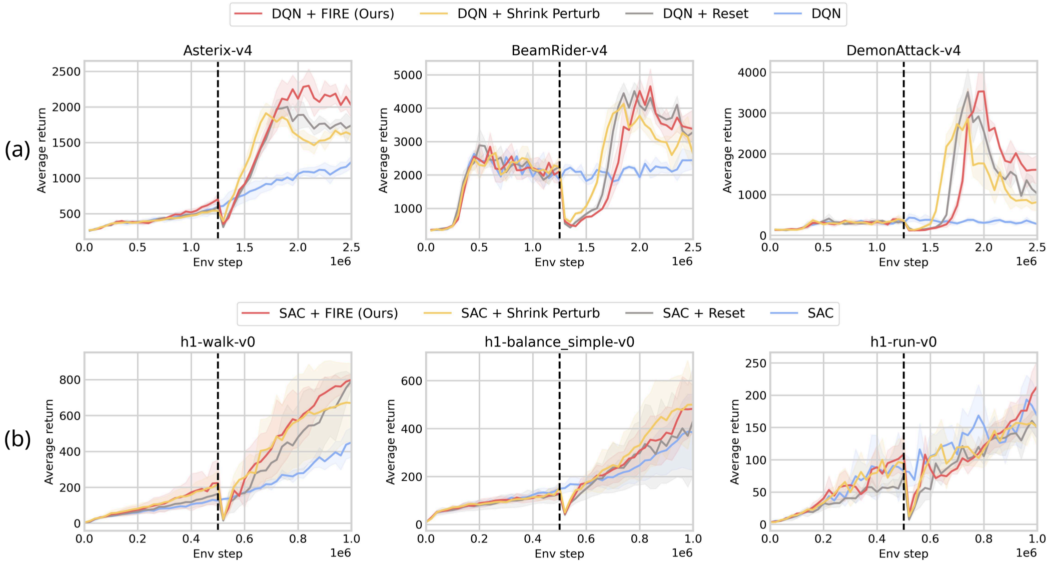

Figure 4: **Reinforcement learning results.** Discrete control with DQN on three Atari environments that suffer from severe plasticity loss (a) and continuous control with SAC on three HumanoidBench tasks (b). The black dashed line indicates the point at which reinitialization is applied.

S&P was carefully tuned over varying reinitialization degrees. Moreover, while the performance of the base model deteriorates with longer pretraining, FIRE maintains strong performance even when initialized from the 60k checkpoint. This demonstrates that FIRE can effectively balance the stability–plasticity trade-off even under severe plasticity loss.

In addition, unlike in continual visual learning (Section 4.1), full reset performs poorly in this setting. The main reason is the lack of stability inherent to full reset. Consequently, the full resetted model cannot outperform the base model, even though the base model itself already suffers from plasticity loss. In other words, the instability introduced by erasing all past information outweighs the potential benefit of restoring plasticity. These findings indicate that full parameter resetting is not an effective strategy for mitigating plasticity loss in continual pretraining of LLMs. Instead of providing a stable improvement, it wastes useful prior knowledge and leads to extreme inefficiency, making it an impractical choice in this setting.

4.3 REINFORCEMENT LEARNING

Setup. Finally, we evaluated FIRE in reinforcement learning. We evaluated the effectiveness of FIRE in a high Replay Ratio (RR) setting (Nikishin et al., 2022; Sokar et al., 2023), where loss of plasticity is severe and acts as a critical bottleneck for sample efficiency. For a comprehensive evaluation, we consider both continuous and discrete control environments. For discrete control, we focus on three Atari 2600 (Bellemare et al., 2013) games (Asterix, BeamRider, and DemonAttack), which have been reported to suffer from severe plasticity loss (Sokar et al., 2023). We use standard nature CNN with DQN algorithm (Mnih et al., 2015). For continuous control, we choose three primary tasks from HumanoidBench (Mnih et al., 2015): balance, walk, and run. We use SimBa (Lee et al., 2024b) with SAC algorithm (Haarnoja et al., 2018) as our baseline, whose replay ratio has failed to scale beyond 1 without resets (Lee et al., 2025).

Results. As shown in Figure 4, FIRE achieves superior or competitive performance across environments compared to both full reset and S&P. In DQN, FIRE consistently outperforms S&P, surpasses full reset in Asterix, and remains competitive in other environments. Although S&P provides a slight improvement in convergence speed, it is still suboptimal relative to both full reset and FIRE. In continuous control tasks, S&P performs competitively, but it falls short of FIRE in all Atari environments. These results suggest that manually tuning hyperparameters to balance stability and plasticity is less effective in visual reinforcement learning—where plasticity loss is particularly severe—than our principled approach, FIRE, which explicitly balances the two.

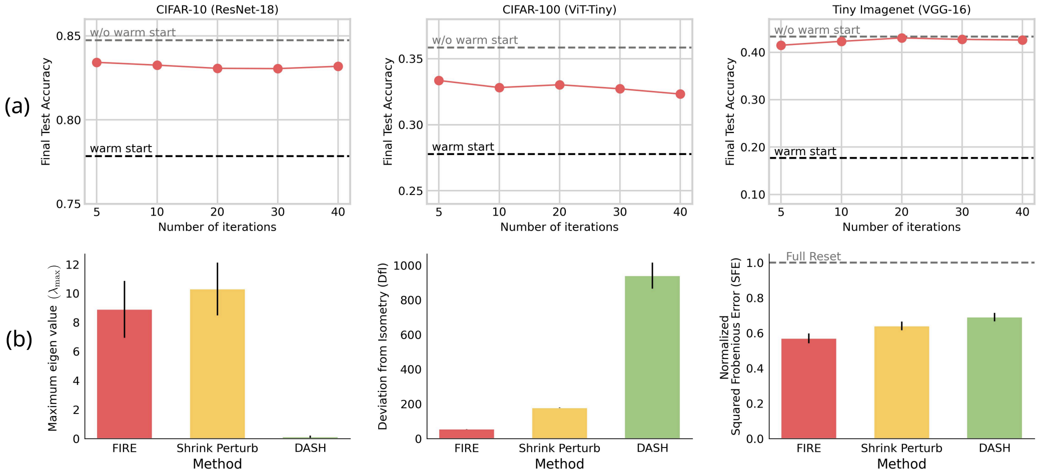

Figure 5: **Ablation study results.** Final performance of FIRE with varying numbers of iterations for Netwon-Schulz algorithm (a). Comparison of FIRE and baselines in terms of loss curvature (maximum eigenvalue of the Hessian), plasticity (DfI), and stability (normalized SFE) (b).

4.4 Ablation study

To better understand the underlying factor of FIRE's strong performance, we conducted an ablation study. To verify whether FIRE indeed effectively balances stability and plasticity, we evaluated the stability metric (SFE) and the plasticity metric (DfI), and compared FIRE against reinitialization baselines. In addition, we measured the loss landscape curvature with respect to upcoming data immediately after a reset, to examine whether our theoretical findings are also reflected in practice.

As shown in Figure 5 (b), FIRE achieves the lowest DfI while maintaining the lowest SFE, which suggests that FIRE successfully balances stability and plasticity in practice. Moreover, FIRE produces a smoother loss landscape compared to S&P, while still preserving a lower SFE. This indicates that our theoretical insights on DfI and loss curvature are indeed manifested in practice. Although DASH is particularly effective in smoothing the loss landscape, it also exhibits the highest SFE, which may contribute to an erasure of useful learned knowledge, thereby leading to instability after reset.

In addition, we evaluated FIRE with various hyperparameters in the warm-start setting to assess its sensitivity. The only hyperparameter in FIRE is the number of iterations for the Newton–Schulz iteration. As the number of iterations increases, we obtain a more accurate estimate of the solution to the constrained optimization problem discussed in Section 3.3. Therefore, our interest is to identify the minimum number of iterations that provides a sufficiently accurate estimate of the solution to yield performance benefits. As shown in Figure 5 (a), FIRE is highly robust to the number of iterations and already provides strong performance gains even with as few as five iterations.

5 Conclusion

In this work, we addressed stability–plasticity trade-off, which is the long-standing problem in continual learning, by introducing FIRE. By approaching stability-plasticity tradeoff as a constrained optimization problem, FIRE enables a principled reinitialization without heavy hyperparameter tuning. Across continual visual learning, reinforcement learning, and language learning benchmarks, FIRE achieved superior or competitive performance, underscoring the importance of effective stability–plasticity management for advancing continual learning.

The main limitation of our work is the assumption of access to past data. Since our focus is on balancing stability and plasticity when such access is available, we did not evaluate FIRE under restricted data scenarios. Future work should, therefore, examine FIRE under restricted access to past data. In addition, we only used relatively small models for continual pretraining of LLMs. Evaluating FIRE on the larger models and applying FIRE not only pretraining, but also continual fine-tuning of LLMs can be a promising direction for future works.

Reproducibility Statement

We provide hyperparameter configurations and implementation details of our experiments in Appendix A.4. The core algorithmic part of our method is described in Algorithm 1. The proofs and assumption of theoretical works provided in this paper are described in Appendix A.1.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. *Advances in neural information processing systems*, 32, 2019.

Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

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

Tudor Berariu, Wojciech Czarnecki, Soham De, Jorg Bornschein, Samuel Smith, Razvan Pascanu, and Claudia Clopath. A study on the plasticity of neural networks. *arXiv preprint arXiv:2106.00042*, 2021.

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=hErdffTsLu.

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

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Wesley Chung, Lynn Cherif, David Meger, and Doina Precup. Parseval regularization for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 37:127937–127967, 2024.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024.

Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Mohamed Elsayed and A Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. *arXiv preprint arXiv:2404.00781*, 2024.

Mohamed Elsayed, Qingfeng Lan, Clare Lyle, and A Rupam Mahmood. Weight clipping for deep continual and reinforcement learning. *arXiv preprint arXiv:2407.01704*, 2024.

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

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Qiang He, Tianyi Zhou, Meng Fang, and Setareh Maghsudi. Adaptive regularization of representation rank as an implicit constraint of bellman equation. *arXiv preprint arXiv:2404.12754*, 2024.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Andrej Karpathy. NanoGPT. `https://github.com/karpathy/nanoGPT`, 2023.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Dongwan Kim and Bohyung Han. On the stability-plasticity dilemma of class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20196–20204, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021a.

Aviral Kumar, Rishabh Agarwal, Tengyu Ma, Aaron Courville, George Tucker, and Sergey Levine. Dr3: Value-based deep reinforcement learning requires explicit regularization. *arXiv preprint arXiv:2112.04716*, 2021b.

Saurabh Kumar, Henrik Marklund, Ashish Rao, Yifan Zhu, Hong Jun Jeon, Yueyang Liu, Benjamin Van Roy, et al. Continual learning as computationally constrained reinforcement learning. *Foundations and Trends® in Machine Learning*, 18(5):913–1053, 2025a.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. In *Conference on Lifelong Learning Agents*, pp. 410–430. PMLR, 2025b.

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. In *International Conference on Machine Learning*, pp. 26416–26438. PMLR, 2024a.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R. Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. *arXiv preprint arXiv:2410.09754*, 2024b.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. *arXiv preprint arXiv:2502.15280*, 2025.

Alex Lewandowski, Michał Bortkiewicz, Saurabh Kumar, András György, Dale Schuurmans, Mateusz Ostaszewski, and Marlos C Machado. Learning continually by spectral regularization. *arXiv preprint arXiv:2406.06811*, 2024a.

Alex Lewandowski, Dale Schuurmans, and Marlos C Machado. Plastic learning with deep fourier features. *arXiv preprint arXiv:2410.20634*, 2024b.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. In *International Conference on Learning Representations*, 2022.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pp. 23190–23211. PMLR, 2023.

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. *arXiv preprint arXiv:2402.18762*, 2024.

Clare Lyle, Gharda Sokar, Razvan Pascanu, and Andras Gyorgy. What can grokking teach us about learning under nonstationarity? *arXiv preprint arXiv:2507.20057*, 2025.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 67–82, 2018.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pp. 16828–16847. PMLR, 2022.

Sangyeon Park, Isaac Han, Seungwon Oh, and KyungJoong Kim. Activation by interval-wise dropout: A simple way to prevent neural networks from plasticity loss. In *Forty-second International Conference on Machine Learning*, 2025.

Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation, 2024.

Maying Shen, Hongxu Yin, Pavlo Molchanov, Lei Mao, and Jose M Alvarez. Step out and seek around: On warm-start training with incremental data. *arXiv preprint arXiv:2406.04484*, 2024.

Baekrok Shin, Junsoo Oh, Hanseul Cho, and Chulhee Yun. Dash: Warm-starting neural network training in stationary settings without loss of plasticity. *Advances in Neural Information Processing Systems*, 37:43300–43340, 2024.

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

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Eli Verwimp, Kuo Yang, Sarah Parisot, Lanqing Hong, Steven McDonagh, Eduardo Pérez-Pellitero, Matthias De Lange, and Tinne Tuytelaars. Clad: A realistic continual learning benchmark for autonomous driving. *Neural Networks*, 161:659–669, 2023.

Maciej Wołczyk, Michał Zajac, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28496–28510, 2021.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in neural information processing systems*, 33:15173–15184, 2020.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International conference on machine learning*, pp. 5393–5402. PMLR, 2018.

Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. *arXiv preprint arXiv:2310.19668*, 2023.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

A Appendix

A.1 Proof of Theorems

Proof of Theorem 1

Proof. Unless explicitly subscripted, $\|\cdot\|$ denotes the Frobenius norm $\|\cdot\|_F$, and $\|\cdot\|_2$ denotes the spectral (operator) norm. We also recall

$$\mathrm{SFE}(\Theta,\widetilde{\Theta}) \;:=\; \sum_{j=1}^{L} \|W^j - \widetilde{W}^j\|_F^2.$$

We will use two elementary facts.

For any $F, \widetilde{F}$ with $m := \min\{\|F\|, \|\widetilde{F}\|\} > 0$,

$$\left\| \frac{F}{\|F\|} - \frac{\widetilde{F}}{\|\widetilde{F}\|} \right\| \;\le\; \frac{2}{m}\,\|F - \widetilde{F}\|.$$

If arbitrary two matrix U and V satisfies $\|U\| = \|V\| = 1$ then

$$\|UU^\top - VV^\top\| \;\le\; 2\,\|U - V\|.$$

Combining these two facts with $U := F/\|F\|$ and $V := \widetilde{F}/\|\widetilde{F}\|$ yields

$$\left\| \frac{FF^\top}{\|F\|^2} - \frac{\widetilde{F}\widetilde{F}^\top}{\|\widetilde{F}\|^2} \right\| \;\le\; \frac{4}{m}\,\|F - \widetilde{F}\|. \tag{6}$$

Applying equation 6 with $F = H_\Theta^\ell(Z)$ and $\widetilde{F} = H_{\widetilde{\Theta}}^\ell(Z)$, and $m_\ell := \min\{\|H_\Theta^\ell(Z)\|, \|H_{\widetilde{\Theta}}^\ell(Z)\|\}$ gives

$$\|C_\Theta^\ell(Z) - C_{\widetilde{\Theta}}^\ell(Z)\| \;\le\; \frac{4}{m_\ell}\,\|H_\Theta^\ell(Z) - H_{\widetilde{\Theta}}^\ell(Z)\|. \tag{7}$$

Introduce the *hybrid* outputs $H_{(\le j)}^\ell(Z)$: the first j layers use Θ and the remaining $j+1,\ldots,\ell$ layers use $\widetilde{\Theta}$. Note that $H_{(\le \ell)}^\ell(Z) = H_\Theta^\ell(Z)$ and $H_{(\le 0)}^\ell(Z) = H_{\widetilde{\Theta}}^\ell(Z)$. Then

$$H_\Theta^\ell(Z) - H_{\widetilde{\Theta}}^\ell(Z) = \sum_{j=1}^{\ell} \Big(H_{(\le j)}^\ell(Z) - H_{(\le j-1)}^\ell(Z) \Big),$$

so by the triangle inequality,

$$\|H_\Theta^\ell(Z) - H_{\widetilde{\Theta}}^\ell(Z)\| \;\le\; \sum_{j=1}^{\ell} \|H_{(\le j)}^\ell(Z) - H_{(\le j-1)}^\ell(Z)\|. \tag{8}$$

Each summand differs in *only the j-th layer weights*. Let $X_{j-1} := H_\Theta^{j-1}(Z)$ denote the shared input fed to layer j in both hybrids. Consider the *backend subnetwork*

$$\mathcal{T}_{j\to\ell}^{(\widetilde{\Theta})}(Y) := \sigma_\ell\big(\cdots \sigma_{j+1}(Y\,\widetilde{W}^{\,j+1}) \cdots \widetilde{W}^{\,\ell}\big).$$

For arbitrary Y_1 and Y_2, its input-Lipschitz constant is

$$\big\|\mathcal{T}_{j\to\ell}^{(\widetilde{\Theta})}(Y_1) - \mathcal{T}_{j\to\ell}^{(\widetilde{\Theta})}(Y_2)\big\| \;\le\; \Big(\prod_{k=j+1}^{\ell} L_{\sigma_k}\,\|\widetilde{W}^{\,k}\|_2 \Big)\,\|Y_1 - Y_2\|. \tag{9}$$

Hence

$$\|H^{\ell}_{(\leq j)}(Z) - H^{\ell}_{(\leq j-1)}(Z)\| \leq \Big(\prod_{k=j+1}^{\ell} L_{\sigma_k}\,\|\widetilde{W}^{k}\|_2\Big)\,L_{\sigma_j}\,\|X_{j-1}\|\,\|W^j - \widetilde{W}^{j}\|. \tag{10}$$

Meanwhile,

$$\|X_{j-1}\| = \|H^{j-1}_{\Theta}(Z)\| \;\leq\; \|Z\|\prod_{k=1}^{j-1} L_{\sigma_k}\,\|W^k\|_2. \tag{11}$$

Combining equation 8, equation 10, and equation 11 yields

$$\|H^{\ell}_{\Theta}(Z) - H^{\ell}_{\widetilde{\Theta}}(Z)\| \;\leq\; \|Z\|\Big(\prod_{k=1}^{\ell} L_{\sigma_k}\Big)\sum_{j=1}^{\ell}\Big(\prod_{k\neq j} B_k\Big)\,\|W^j - \widetilde{W}^{j}\|.$$

By Cauchy–Schwarz,

$$\sum_{j=1}^{\ell}\Big(\prod_{k\neq j} B_k\Big)\,\|W^j - \widetilde{W}^{j}\| \;\leq\; B^{\ell}_{\Pi}\left(\sum_{j=1}^{\ell}\frac{1}{B_j^2}\right)^{1/2}\sqrt{\mathrm{SFE}(\Theta,\widetilde{\Theta})}.$$

Therefore,

$$\|H^{\ell}_{\Theta}(Z) - H^{\ell}_{\widetilde{\Theta}}(Z)\| \;\leq\; \|Z\|\Big(\prod_{k=1}^{\ell} L_{\sigma_k}\Big)\,B^{\ell}_{\Pi}\left(\sum_{j=1}^{\ell}\frac{1}{B_j^2}\right)^{1/2}\sqrt{\mathrm{SFE}(\Theta,\widetilde{\Theta})}. \tag{12}$$

Substitute equation 12 into equation 7:

$$\|C^{\ell}_{\Theta}(Z) - C^{\ell}_{\widetilde{\Theta}}(Z)\| \;\leq\; \frac{4\,\|Z\|}{m_\ell}\Big(\prod_{k=1}^{\ell} L_{\sigma_k}\Big)\,B^{\ell}_{\Pi}\left(\sum_{j=1}^{\ell}\frac{1}{B_j^2}\right)^{1/2}\sqrt{\mathrm{SFE}(\Theta,\widetilde{\Theta})}.$$

Squaring both sides gives

$$\|C^{\ell}_{\Theta} - C^{\ell}_{\widetilde{\Theta}}\|^2 \;\leq\; \frac{16\,\|Z\|^2}{m_\ell^2}\Big(\prod_{k=1}^{\ell} L_{\sigma_k}\Big)^2 (B^{\ell}_{\Pi})^2\left(\sum_{j=1}^{\ell}\frac{1}{B_j^2}\right)\mathrm{SFE}(\Theta,\widetilde{\Theta}).$$

If $L_{\sigma_k} \leq 1$ and $B_j \leq S$ for all j, then $(B^{\ell}_{\Pi})^2\sum_{j=1}^{\ell} B_j^{-2} \leq \ell\,S^{2\ell-2}$. Therefore

$$\|C^{\ell}_{\Theta} - C^{\ell}_{\widetilde{\Theta}}\| \;\leq\; \frac{4\,\|Z\|}{m_\ell}\,\sqrt{\ell}\,S^{\ell-1}\sqrt{\mathrm{SFE}(\Theta,\widetilde{\Theta})}.$$

□

Network, Loss, and Notation. Let $Z \in \mathbb{R}^{n\times d_0}$ be the input matrix and $W_k \in \mathbb{R}^{d_{k-1}\times d_k}$ $(k = 1,\ldots,L)$ be the weight matrices. Define

$$A_k = H_{k-1}W_k \in \mathbb{R}^{n\times d_k}, \qquad H_k = \rho(A_k) \quad (\rho = \mathrm{ReLU}), \qquad U := A_L \in \mathbb{R}^{n\times d_L},$$

with $H_0 := Z$. The empirical risk is

$$\mathcal{L}(W_{1:L}) = \frac{1}{n}\sum_{i=1}^{n}\ell_i(u_i), \qquad u_i \in \mathbb{R}^{d_L}.$$

Let $\theta = \mathrm{vec}(W_1,\ldots,W_L) \in \mathbb{R}^p$ and $H_\theta := \nabla^2_\theta\mathcal{L} \in \mathbb{R}^{p\times p}$. We use $\|\cdot\|_2$ for spectral norm and $\|\cdot\|_F$ for the Frobenius norm.

We denote the maximum eigen values as $\lambda_{\max}$

Deviation From Isometry (DfI). For a matrix W, $\mathrm{DfI}(W) := \|W^\top W - I\|_F^2$. For each layer, set

$$\nu_k := 1 + \sqrt{\mathrm{DfI}(W_k)}, \qquad \alpha_k := \sqrt{\nu_k}.$$

First, let us examine the lemmas required for the proof of Theorem 2.

Lemma 1 (DfI controls the spectral norm)**.** *For each layer k, $\|W_k\|_2^2 \le \nu_k$ and $\|W_k\|_2 \le \alpha_k$.*

Proof. $\|W\|_2^2 = \lambda_{\max}(W^\top W) \le \lambda_{\max}(W^\top W - I) + 1 \le \|W^\top W - I\|_2 + 1 \le \|W^\top W - I\|_F + 1 = 1 + \sqrt{\mathrm{DfI}(W)}$. □

Lemma 2 (Covariance/spectral growth through layers)**.** *Let $\Sigma_{H_k} = \frac{1}{n} H_k^\top H_k$. Then*

$$\lambda_{\max}(\Sigma_{H_k}) \;\le\; \lambda_{\max}(\Sigma_{H_{k-1}})\,\|W_k\|_2^2 \;\le\; \lambda_{\max}(\Sigma_{H_{k-1}})\,\nu_k.$$

Consequently, with $\Sigma_{H_0} = \Sigma_Z$,

$$\lambda_{\max}(\Sigma_{H_k}) \;\le\; \lambda_{\max}(\Sigma_Z) \prod_{j=1}^{k} \nu_j.$$

In particular, under $\Sigma_Z \approx I$ we have $\lambda_{\max}(\Sigma_{H_k}) \le \prod_{j=1}^{k} \nu_j$.

Proof. ReLU is 1-Lipschitz (applied elementwise), hence $\|H_k\|_2 \le \|A_k\|_2 \le \|H_{k-1}\|_2 \|W_k\|_2$. Therefore $\frac{1}{n}\|H_k\|_2^2 \le \frac{1}{n}\|H_{k-1}\|_2^2 \|W_k\|_2^2$, i.e., $\lambda_{\max}(\Sigma_{H_k}) \le \lambda_{\max}(\Sigma_{H_{k-1}}) \|W_k\|_2^2$. Apply Lemma 1. □

Lemma 3 (Block-Jacobian bound)**.** *Let $J \in \mathbb{R}^{(nd_L)\times p}$ be the Jacobian of $\mathrm{vec}(U)$ w.r.t. θ, and $J = [J_1\ J_2\ \cdots\ J_L]$ the block-columns corresponding to $\mathrm{vec}(W_k)$. Then*

$$\frac{1}{n}\|J_k\|_2^2 \;\le\; \Big(\prod_{j=1}^{k-1} \nu_j\Big)\Big(\prod_{j=k+1}^{L} \nu_j\Big) = \prod_{j\ne k} \nu_j.$$

Consequently,

$$\frac{1}{n}\|J\|_2^2 \;\le\; \sum_{k=1}^{L} \frac{1}{n}\|J_k\|_2^2 \;\le\; \sum_{k=1}^{L} \prod_{j\ne k} \nu_j.$$

Proof. Consider a perturbation ΔW_k. Without loss of generality, we may assume $\|\Delta W_k\|_F = 1$, since the operator norm is defined by the supremum over unit perturbations. With fixed ReLU gates (op. norm ≤ 1), the output perturbation over all n samples satisfies

$$\Delta U \;=\; H_{k-1}\,\Delta W_k\,B_{k+1:L}, \quad B_{k+1:L} := \underbrace{D_k W_{k+1} D_{k+1} \cdots D_{L-1}}_{\text{diag gates, } \|\cdot\|_2 \le 1} W_L.$$

Thus $\|\Delta U\|_F \le \|H_{k-1}\|_2\,\|\Delta W_k\|_F\,\|B_{k+1:L}\|_2$, so the operator norm of the linear map $\Delta W_k \mapsto \Delta U$ is at most $\|H_{k-1}\|_2 \|B_{k+1:L}\|_2$. Hence $\|J_k\|_2 \le \|H_{k-1}\|_2 \|B_{k+1:L}\|_2$ and

$$\frac{1}{n}\|J_k\|_2^2 \;\le\; \frac{1}{n}\|H_{k-1}\|_2^2\,\|B_{k+1:L}\|_2^2 \;=\; \lambda_{\max}(\Sigma_{H_{k-1}})\,\|B_{k+1:L}\|_2^2.$$

Using Lemma 2, $\lambda_{\max}(\Sigma_{H_{k-1}}) \le \prod_{j=1}^{k-1} \nu_j$. Also $\|B_{k+1:L}\|_2 \le \prod_{j=k+1}^{L} \|W_j\|_2 \le \prod_{j=k+1}^{L} \alpha_j$, thus $\|B_{k+1:L}\|_2^2 \le \prod_{j=k+1}^{L} \nu_j$. Multiplying the two bounds yields the claim. Finally, since J is a horizontal concatenation of blocks, $\|J\|_2^2 \le \sum_k \|J_k\|_2^2$. □

Lemma 4 (Gauss–Newton part)**.** *For each sample, $\nabla_\theta^2 \ell_i = J_i^\top (\nabla_u^2 \ell_i) J_i + R_i$ with some remainder R_i. By (A2), $\|\nabla_u^2 \ell_i\|_2 \le \beta$, hence*

$$\left\|\frac{1}{n}\sum_{i=1}^{n} J_i^\top (\nabla_u^2 \ell_i) J_i\right\|_2 \;\le\; \frac{\beta}{n}\,\|J\|_2^2 \;\le\; \beta \sum_{k=1}^{L} \prod_{j\ne k} \nu_j,$$

where the last inequality uses Lemma 3.

Lemma 5 (Remainder term). *Let $R := \frac{1}{n}\sum_{i=1}^{n} R_i$. Under (A2) and (A3),*

$$\|R\|_2 \;\le\; 2\gamma \sum_{1\le k<\ell\le L} \prod_{j\notin\{k,\ell\}} \nu_j.$$

Proof. Fix the ReLU gates locally (piecewise linear region) and $\sum_k \|\Delta W_k\|_F = 1$, since the operator norm is defined by the supremum over unit perturbations. Then the network output U is *multilinear* in $\{W_k\}_{k=1}^{L}$. For $k<\ell$, the mixed second derivative block maps $(\Delta W_k, \Delta W_\ell)$ to

$$H_{k-1}\,\Delta W_k\, C_{k+1:\ell-1}\,\Delta W_\ell\, B_{\ell+1:L},$$

where $C_{k+1:\ell-1}$ is the product of intermediate gated weights, and $B_{\ell+1:L}$ the tail product as in Lemma 3. By submultiplicativity,

$$\|H_{k-1}\Delta W_k C_{k+1:\ell-1}\Delta W_\ell B_{\ell+1:L}\|_F \le \|H_{k-1}\|_2\,\|\Delta W_k\|_F\,\|C_{k+1:\ell-1}\|_2\,\|\Delta W_\ell\|_F\,\|B_{\ell+1:L}\|_2.$$

Using Lemma 2 and Lemma 1,

$$\|H_{k-1}\|_2 \le \sqrt{n}\prod_{j=1}^{k-1}\alpha_j, \quad \|C_{k+1:\ell-1}\|_2 \le \prod_{j=k+1}^{\ell-1}\alpha_j, \quad \|B_{\ell+1:L}\|_2 \le \prod_{j=\ell+1}^{L}\alpha_j.$$

Therefore, after dividing by n (from the prefactor $1/n$ in $\mathcal{L}$) and summing the symmetric contribution (ℓ,k), the bilinear remainder contributes at most

$$\frac{2}{n}\sum_{k<\ell}\|H_{k-1}\|_2\|C_{k+1:\ell-1}\|_2\|B_{\ell+1:L}\|_2\,\|\Delta W_k\|_F\|\Delta W_\ell\|_F \;\le\; 2\sum_{k<\ell}\Big(\prod_{j\notin\{k,\ell\}}\alpha_j^2\Big)\,\|\Delta W_k\|_F\|\Delta W_\ell\|_F.$$

Finally, by $2ab \le a^2+b^2$ and $\sum_k \|\Delta W_k\|_F^2 = 1$ (unit parameter direction), the operator norm of the second-derivative map is bounded by $\sum_{k<\ell}\prod_{j\notin\{k,\ell\}}\nu_j$. Multiplying by $\|\nabla_u \ell_i\|_2 \le \gamma$ and averaging over i gives the claim. □

Proof of Theorem 2

Proof. Combine Lemma 4 and Lemma 5 and use $\|A+B\|_2 \le \|A\|_2 + \|B\|_2$. □

Corollary 1 (Near-interpolation or small-gradient regime). *If the training gradients at the outputs are small so that $\gamma \approx 0$, then*

$$\big\|\nabla_\theta^2 \mathcal{L}(W_{1:L})\big\|_2 \;\lesssim\; \beta\sum_{k=1}^{L}\prod_{j\ne k}\big(1+\sqrt{\mathrm{DfI}(W_j)}\big).$$

Next, we present the lemma required for the proof of Theorem 3.

Lemma 6 (A basic spectral lemma from DfI). *Let $\varepsilon = \sqrt{\mathrm{DfI}(W)}$. Then*

$$\|W^\top W - I\|_2 \;\le\; \|W^\top W - I\|_F \;=\; \varepsilon,$$

hence every eigenvalue μ of $W^\top W = S^2$ satisfies $1-\varepsilon \le \mu \le 1+\varepsilon$. Equivalently,

$$\sqrt{1-\varepsilon}\,I \;\preceq\; S \;\preceq\; \sqrt{1+\varepsilon}\,I.$$

Proof. By the definition of ε, we have

$$\|W^\top W - I\|_2 \;\le\; \|W^\top W - I\|_F \;=\; \varepsilon.$$

Therefore, all eigenvalues μ of $W^\top W$ lie within the interval

$$1-\varepsilon \;\le\; \mu \;\le\; 1+\varepsilon.$$

Since $W^\top W = S^2$ with $S \succeq 0$, this is equivalent to the spectral bound

$$\sqrt{1-\varepsilon}\,I \;\preceq\; S \;\preceq\; \sqrt{1+\varepsilon}\,I.$$

□

Using this Lemma, we provide proof of Theorem 3 below:

Proof of Theorem 3

Proof. Let $Z \in \mathbb{R}^{n\times a}$ be the input matrix and $W \in \mathbb{R}^{a\times b}$ a weight matrix. The resulting feature matrix is $\Phi = ZW \in \mathbb{R}^{n\times b}$ and the empirical covariances are

$$\Sigma_Z = \tfrac{1}{n}Z^\top Z \in \mathbb{R}^{a\times a}, \qquad \Sigma_\Phi = \tfrac{1}{n}\Phi^\top \Phi = W^\top \Sigma_Z W \in \mathbb{R}^{b\times b}.$$

Let $W = QS$ denote the right polar decomposition of W, where $Q \in \mathbb{R}^{a\times b}$ has orthonormal columns ($Q^\top Q = I_b$) and $S = (W^\top W)^{1/2} \in \mathbb{R}^{b\times b}$ is positive definite. Then

$$\Sigma_\Phi \;=\; W^\top \Sigma_Z W \;=\; S\,(Q^\top \Sigma_Z Q)\,S.$$

Write $M := Q^\top \Sigma_Z Q \succeq 0$, and let its positive eigenvalues be $\eta_1 \geq \cdots \geq \eta_d > 0$, where $d = \operatorname{rank}(M) \leq \min\{b, \operatorname{rank}(\Sigma_Z)\}$. Let $\sigma_1(\Phi) \geq \cdots \geq \sigma_d(\Phi) > 0$ denote the nonzero singular values of Φ.

For any $x \in \mathbb{R}^b$ with $\|x\| = 1$,

$$x^\top \Sigma_\Phi x \;=\; x^\top SMSx \;=\; (Sx)^\top M(Sx).$$

Let $y = Sx$. Then $x^\top \Sigma_\Phi x = y^\top My$ and, by Lemma 6,

$$\|y\|_2^2 = \|Sx\|_2^2 \in [\,1-\varepsilon,\, 1+\varepsilon\,].$$

Therefore

$$(1-\varepsilon)\,\lambda^+_{\min}(M) \;\leq\; x^\top \Sigma_\Phi x \;\leq\; (1+\varepsilon)\,\lambda_{\max}(M),$$

where $\lambda^+_{\min}(M) = \eta_d$ denotes the smallest positive eigenvalue of M (the lower bound is interpreted on the subspace where $My \neq 0$). Taking the maximum over unit x yields

$$\lambda_{\max}(\Sigma_\Phi) \;\leq\; (1+\varepsilon)\,\eta_1,$$

and taking the minimum Rayleigh quotient over the orthogonal complement of $\ker(\Sigma_\Phi)$ yields

$$\lambda^+_{\min}(\Sigma_\Phi) \;\geq\; (1-\varepsilon)\,\eta_d.$$

Since $\sigma_{\max}(\Phi)^2 \;=\; n\,\lambda_{\max}(\Sigma_\Phi)$ and $\big(\sigma^+_{\min}(\Phi)\big)^2 \;=\; n\,\lambda^+_{\min}(\Sigma_\Phi)$, below inequality holds with $d = \operatorname{rank}(M)$.

$$\sqrt{n}\,\sqrt{1-\varepsilon}\,\sqrt{\eta_d} \;\;\leq\;\; \sigma^+_{\min}(\Phi) \;\;\leq\;\; \sigma_{\max}(\Phi) \;\;\leq\;\; \sqrt{n}\,\sqrt{1+\varepsilon}\,\sqrt{\eta_1}.$$

Here $\sigma^+_{\min}(\Phi)$ denotes the smallest positive singular value of Φ (defined only when $d \geq 1$).

Therefore, if $d \geq 1$, then

$$\rho_\Phi := \frac{\sigma_{\max}(\Phi)}{\sigma^+_{\min}(\Phi)} \;\;\leq\;\; \sqrt{\frac{1+\varepsilon}{1-\varepsilon}} \;\cdot\; \sqrt{\frac{\eta_1}{\eta_d}}. \tag{13}$$

Consider the worst-case allocation of the nonzero singular values that maximizes the cumulative ratio $\sum_{i=1}^k \sigma_i / \sum_{i=1}^d \sigma_i$ given a fixed condition number bound ρ_Φ: the top k singular values all equal $\sigma_{\max}$ and the remaining $d-k$ equal $\sigma^+_{\min}$. Then

$$\frac{\sum_{i=1}^k \sigma_i(\Phi)}{\sum_{i=1}^d \sigma_i(\Phi)} \;\leq\; \frac{k\,\sigma_{\max}}{k\,\sigma_{\max} + (d-k)\,\sigma^+_{\min}} \;=\; \frac{k\,\rho_\Phi}{k\,\rho_\Phi + (d-k)}. \tag{14}$$

To achieve a coverage level of $1-\delta$ with k singular values, it is necessary that

$$\frac{k\,\rho_\Phi}{k\,\rho_\Phi + (d-k)} \;\geq\; 1-\delta \quad\Longrightarrow\quad k \;\geq\; \frac{(1-\delta)\,d}{\delta\,\rho_\Phi + (1-\delta)}.$$

Taking the ceiling and substituting the bound on ρ_Φ from (13), establishes the left inequality in (3)

When $\Sigma_Z = I$, we have $M = Q^\top IQ = I$, so $\eta_1 = \cdots = \eta_b = 1$ and $d = b$. Then,

$$\sqrt{n}\sqrt{1-\varepsilon} \leq \sigma_i(\Phi) \leq \sqrt{n}\sqrt{1+\varepsilon} \qquad (\forall i),$$

Which leads to

$$\rho_\Phi \leq \sqrt{\tfrac{1+\varepsilon}{1-\varepsilon}}$$

Substituting ρ_Φ in first inequality of (3) leads to the right inequality (3). Without whitening, the achievable flattening is limited by the compressed input spectrum $M = Q^\top \Sigma_Z Q$.

□

Next, we present the proof for Theorem 4.

Proof of Theorem 4

Proof. By isotropy and positive homogeneity, there exists a constant $c_\sigma > 0$ such that $\mathbb{E}_z[\|\sigma(\langle z, w_j\rangle)\|] = c_\sigma \|w_j\|$ for all j. Hence $s_j = \|w_j\|/\big(\frac{1}{b}\sum_k \|w_k\|\big)$. Let $u_j = \|w_j\|^2 = [W^\top W]_{jj}$. Since

$$\mathrm{DfI}(W) = \|W^\top W - I\|_F^2 = \sum_{j=1}^{b}(u_j - 1)^2 + 2\sum_{i<j}\langle w_i, w_j\rangle^2 \;\geq\; \sum_{j=1}^{b}(u_j-1)^2,$$

we obtain $|u_j - 1| \leq \sqrt{\sum_k (u_k-1)^2} \leq \delta$, i.e., $1-\delta \leq \|w_j\|^2 \leq 1+\delta$ for all j. The same bounds imply $\sqrt{1-\delta} \leq \bar{r} := \frac{1}{b}\sum_k \|w_k\| \leq \sqrt{1+\delta}$. Therefore

$$\frac{\sqrt{1-\delta}}{\sqrt{1+\delta}} \leq s_j = \frac{\|w_j\|}{\bar{r}} \leq \frac{\sqrt{1+\delta}}{\sqrt{1-\delta}}.$$

□

Corollary 2 (Absence of τ-dormant neurons)**.** *Fix $\tau \in (0,1)$. If $\mathrm{DfI}(W) \leq \big(\frac{1-\tau^2}{1+\tau^2}\big)^2$, then $s_j \geq \tau$ for all j.*

Note that Sokar et al. (2023) measured neurons with a dormancy score of 0.025 or lower as dormant. In this threshold, based on our theoretical analysis, $\mathrm{DfI}(W) < 0.9975$ can eliminate dormant neurons from the network.

A.2 Implementation Details of FIRE

Our orthogonalization strategy builds on the Newton–Schulz (NS) iteration, which has also been adopted in recent works such as Muon. However, the exact recurrence used in Muon differs substantially from ours. Muon employs a tuned quintic polynomial of the form

$$\varphi(x) = ax + bx^3 + cx^5,$$

with optimized coefficients such as $(a, b, c) = (3.4445, -4.7750, 2.0315)$, chosen to accelerate convergence so that only a few iterations are needed in practice. However, this sacrifices accuracy for speed, since the singular values do not converge to 1, but oscillate near it. Since our interest is accuracy rather than speed, we adopt the standard coefficients $(a, b, c) = (2, -1.5, 0.5)$, which correspond to a well-known rectangular variant of NS:

$$X_{k+1} \;=\; 2X_k \;-\; 1.5X_k(X_k^\top X_k) \;+\; 0.5X_k(X_k^\top X_k)^2.$$

Although this more slowly increases small singular values than Muon's tuned version, it is known to be stable even when the matrix is not square and thus provides a more reliable approximation for arbitrary weight shapes.

Having established the specific form of NS iteration we adopt, we now describe how it is applied in practice to different modules of the network.

Linear layers. For fully-connected weights $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we first normalize and then apply NS iteration to approximate an orthogonal matrix. Since orthogonalization alone changes the scale of the outputs, we multiply the result by

$$\text{scale} = \sqrt{\tfrac{d_{\text{out}}}{d_{\text{in}}}}.$$

This factor is motivated by the Modular Duality framework (Bernstein & Newhouse, 2025), which shows that taking the ratio of output to input dimension is sufficient to preserve stable signal variance. In short, the orthogonalization ensures the weights are well-conditioned, and the scaling factor restores the right magnitude.

Convolutional layers. For convolutional filters $W \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k_h \times k_w}$, we apply the same procedure slice by slice over the spatial indices. Here the scaling factor additionally accounts for the size of the kernel:

$$\text{scale} = \frac{\sqrt{C_{\text{out}}/C_{\text{in}}}}{k_h k_w}.$$

Intuitively, the larger the kernel, the more input values contribute to each output, so we divide by the kernel area to prevent the output variance from exploding.

Note that for each spatial location (i, j), the slice $W[:, :, i, j] \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$ is orthogonalized independently by applying the Newton–Schulz iteration.

Attention modules. In Vision Transformers (ViTs), we restrict orthogonalization to the query (Q) and key (K) projections. Empirically, applying it to the feedforward MLP layers or the output projections does not provide clear benefits and may even reduce performance. Because the dot-product $QK^\top$ is the part most sensitive to poor conditioning, orthogonalizing Q and K helps improve the stability of similarity scores while leaving the value (V), output (O), and MLP weights unchanged.

A.3 Baseline Methods

Shrink & Perturb. Shrink & Perturb (S&P) is a Reset-based method that shrinks weight parameters and injects noise (Ash & Adams, 2020). This method has proven particularly beneficial for warm-start training. Following the setup in prior work Lee et al. (2024a), we control both the noise level and the shrinkage strength using a single hyperparameter. Formally, letting θ denote the learnable parameters, θ_0 the initial parameters, and λ the S&P coefficient, the update rule is: $\theta \leftarrow (1 - \lambda)\theta + \lambda\theta_0$.

DASH. Direction-Aware SHrinking (DASH) is a Reset-based method that selectively shrinks network weights according to their directional alignment with the loss gradient, measured by cosine similarity. This method suppresses parameters that contribute to noise memorization while preserving weights that encode task-relevant features. This method enhances training efficiency and preserves model plasticity, leading to improved generalization under stationary data distributions. In our experiments, we applied DASH when new data was added.

Parseval Regularization. Parseval Reg. introduces a regularization term that enforces the weight matrices of neural network layers to remain approximately orthogonal (Chung et al., 2024). Formally, letting W denote a weight matrix, I identity matrix, and $\|\cdot\|_F$ the Frobenius norm. The loss term is $\lambda\|WW^\top - sI\|_F^2$, where $s > 0$ is a scaling factor and λ is the regularization strength. It penalizes the deviation of $WW^\top$ from a scaled identity matrix, encouraging the rows of each weight matrix W to be orthogonal and have controlled norms. This constraint keeps the singular values of W close to a constant, preventing gradient explosion or vanishing and leading to more stable and efficient optimization. We used $s = 1$ in all experiments and only swept λ.

A.4 Detailed settings

For all experiments in this paper, we report the results with 3 runs.

A.4.1 Continual Visual Learning

In this section, we describe the detailed settings for conducting continual visual learning. We note that most of the hyperparameters we used were adopted from Lee et al. (2024a).

Table 1: Detailed settings in continual visual learning.

Parameter	**Value**
Optimizer	Adam (Kingma & Ba, 2014)
Learning Rate	$1e{-3}$
Learning Rate Scheduler	Warmup
Gradient norm clipping	0.5
Batch Size	256
Epochs per Chunk	100
Data Augmentation	False

For the Warmup scheduler, the learning rate is increased linearly from 0 to the target learning rate during the first 10% of training on each dataset. In other words, in the case of Table 1, the learning rate is gradually raised from 0 to $1e{-3}$ over the first 10 epochs of each data chunk.

In the warm-start scenario described in Section 4.1, we trained for 1000 epochs before new data arrived and for 100 epochs after its arrival, in order to balance the total number of gradient updates before and after the introduction of new data.

A.4.2 Continual Pretraining of LLMs

Table 2: Detailed settings in continual pretraining of LLMs.

Parameter	**Value**
Optimizer	AdamW (Loshchilov & Hutter, 2017)
Weight Decay	$1e{-1}$
Learning Rate	$6e{-4}$
Minimum Learning Rate	$6e{-5}$
Learning Rate Scheduler	Warmup + Linearly Decaying
Gradient norm clipping	1.0
Batch Size	480

We used implementation and hyperparameters of nanoGPT from Karpathy (2023).

During first phase, the learning rate linearly increases from 0 to the target learning rate ($6e{-4}$) during 2,000 steps, then annealed to minimum learning rate ($6e{-5}$) until the end of the phase. In the second phase, the learning rate linearly increases from 0 to the target learning rate ($6e{-4}$) during 10% of training iterations of second phase. Then, it decreases linearly to minimum learning rate ($6e{-5}$) until the end of the phase.

A.4.3 Reinforcement Learning

For S&P method, we apply S&P to the encoder and Reset to the fully connected layers (D'Oro et al., 2022) for discrete control, and S&P with $\lambda = 0.8$ to whole parameters for continuous control tasks. We perform a single intervention (Full Reset, S&P, FIRE) at the midpoint of learning. We followed the hyperparameter configurations used in prior work (Sokar et al., 2023).

Table 3: Hyperparameters used in the ALE environment with DQN algorithm.

Parameter	**Value**
Optimizer	Adam (Kingma & Ba, 2014)
Optimizer: ϵ	$1.5\text{e}-4$
Optimizer: Learning rate	$6.25\text{e}-4$
Minimum ϵ for training	0.01
Evaluation ϵ	0.001
Discount factor γ	0.99
Replay buffer size	10^6
Minibatch size	32
Initial collect steps	20000
Training iterations	10
Training environment steps per iteration	$250K$
Updates per environment step (Replay Ratio)	1
Target network update period	2000
Loss function	Huber Loss (Huber, 1992)

For continuous tasks, the hyperparameter setting is followed by Lee et al. (2024b).

Table 4: Hyperparameters used in HumanoidBench environments with SimBa.

Parameter	**Value**
Optimizer	AdamW (Loshchilov & Hutter, 2017)
Optimizer: Learning rate	$1\text{e}-4$
Optimizer: Weight decay	0.01
Actor hidden dim	128
Actor num blocks	1
Critic hidden dim	512
Critic num blocks	2
Discount factor γ	0.99
Clipped Double-Q (Fujimoto et al., 2018)	True
Replay buffer size	10^6
Minibatch size	256
Initial collect steps	5000
Updates per environment step (Replay Ratio)	2
Soft target update factor τ	0.005

A.5 Hyperparameter Search Space

Table 5 presents the hyperparameter search space, and Tables 6-9 present their optimal values.

Table 5: Hyperparameter search space for all experiments.

Experiment	**Method**	**Hyperparameters**	**Search Space**
Warm-Starting	S&P	λ	$0.2, 0.4, 0.6, 0.8$
	DASH	α	$0.1, 0.3$
		λ	$0.05, 0.1, 0.3$
	Parseval Reg.	λ	$1e{-3}, 1e{-4}, 1e{-5}$
Continual Learning	S&P	λ	$0.2, 0.4, 0.6, 0.8$
	DASH	α	$0.1, 0.3$
		λ	$0.05, 0.1, 0.3$
	Parseval Reg.	λ	$1e{-3}, 1e{-4}, 1e{-5}$
Class-Incremental Learning	S&P	λ	$0.2, 0.4, 0.6, 0.8$
	DASH	α	$0.1, 0.3$
		λ	$0.05, 0.1, 0.3$
	Parseval Reg.	λ	$1e{-3}, 1e{-4}, 1e{-5}$
Continual pretraining of GPT-0.1B	S&P	λ	$0.2, 0.5, 0.8$

Table 6: Hyperparameters for Warm-Start setting.

Dataset	**Method**	**Value**
CIFAR-10	S&P	$\lambda = 0.8$
(ResNet-18)	DASH	$\alpha = 0.3$, $\lambda = 0.05$
	Parseval Reg.	$\lambda = 1e{-3}$
	FIRE	iter $= 10$
CIFAR-100	S&P	$\lambda = 0.8$
(ViT-Tiny)	DASH	$\alpha = 0.1$, $\lambda = 0.05$
	Parseval Reg.	$\lambda = 1e{-5}$
	FIRE	iter $= 10$
Tiny ImageNet	S&P	$\lambda = 0.8$
(VGG-16)	DASH	$\alpha = 0.1$, $\lambda = 0.05$
	Parseval Reg.	$\lambda = 1e{-3}$
	FIRE	iter $= 10$

Table 7: Hyperparameters for Continual Learning.

Dataset	Method	Value
CIFAR-10 (ResNet-18)	S&P	$\lambda = 0.8$
	DASH	$\alpha = 0.3$, $\lambda = 0.05$
	Parseval Reg.	$\lambda = 1\text{e}{-3}$
	FIRE	iter $= 10$
CIFAR-100 (ViT-Tiny)	S&P	$\lambda = 0.8$
	DASH	$\alpha = 0.1$, $\lambda = 0.05$
	Parseval Reg.	$\lambda = 1\text{e}{-5}$
	FIRE	iter $= 10$
Tiny ImageNet (VGG-16)	S&P	$\lambda = 0.8$
	DASH	$\alpha = 0.1$, $\lambda = 0.1$
	Parseval Reg.	$\lambda = 1\text{e}{-4}$
	FIRE	iter $= 10$

Table 8: Hyperparameters for Class-Incremental Learning.

Dataset	Method	Value
CIFAR-100 (ViT-Tiny)	S&P	$\lambda = 0.8$
	DASH	$\alpha = 0.3$, $\lambda = 0.3$
	Parseval Reg.	$\lambda = 1\text{e}{-5}$
	FIRE	iter $= 10$
Tiny ImageNet (VGG-16)	S&P	$\lambda = 0.8$
	DASH	$\alpha = 0.3$, $\lambda = 0.1$
	Parseval Reg.	$\lambda = 1\text{e}{-3}$
	FIRE	iter $= 10$

Table 9: Hyperparameters for Continual pretraining of LLMs.

Method	CKPT	Value
S&P	Best	$\lambda = 0.5$
	30k	$\lambda = 0.8$
	60k	$\lambda = 0.5$
FIRE	Best	iter $= 5$
	30k	iter $= 5$
	60k	iter $= 5$