Predicting Plasticity in Deep Continual Learning: A Theoretical Perspective

Jiuqi Wang
University of Virginia
jiuqi@email.virginia.edu

Jayanth Srinivasa
Cisco Research
jasriniv@cisco.com

Claire Chen
California Institute of Technology
clairechen@caltech.edu

Shuze Daniel Liu
Purdue University
liu4869@purdue.edu

Ali Payani
Cisco Research
apayani@cisco.com

Shangtong Zhang
University of Virginia
shangtong@virginia.edu

Abstract

Deep continual learning requires models to adapt to new tasks without retraining from scratch. However, neural networks can lose their ability to adapt to new tasks after training on previous ones, a phenomenon known as loss of plasticity. There have been several explanations and diagnostics proposed for plasticity loss. Motivated by the philosophy that "all models are wrong, but some are useful", we ask: can existing diagnostics predict a neural network's plasticity? In this work, we take a practical view to interpret plasticity as task-dependent trainability, i.e., a neural network's future optimization gain on a target task. We first take a theoretical approach, showing, by constructing a few counterexamples, that some widely adopted diagnostics of plasticity, including representation rank and neural tangent kernel rank, can fail to predict the loss of trainability in both regression and classification settings. We instead propose a novel metric, called optimization readiness, which combines gradient strength and gradient reliability. We prove that optimization readiness lower bounds one-step optimization gain under standard smoothness assumptions, providing a theoretical guarantee for its predictive power. Empirically, we show that across commonly used deep continual learning settings, such as Slowly-Changing Regression and Permuted MNIST, optimization readiness more reliably ranks checkpoints by their task-dependent trainabilities than prior label-agnostic diagnostics that ignore the labels when computing their metrics, even with substantially fewer samples. We release our code at https://github.com/Sequential-Intelligence-Lab/Predicting_Plasticity.

1 Introduction

Deep continual learning describes a machine learning setting in which a neural network learns from a sequence of tasks, where the learner needs to adapt sequentially while retaining useful knowledge from earlier experience [Aljundi, 2019, Parisi et al., 2019]. Such continual adaptation is possible only if the neural network model remains trainable, i.e., it can achieve meaningful improvement on the new task via gradient-based optimization. Unfortunately, continual adaptation cannot be achieved simply by continuing gradient-based training on newly arriving data, because neural networks can suffer from loss of plasticity [Dohare et al., 2024]. Loss of plasticity is the phenomenon

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

in which a previously trained model loses its ability to adapt to new tasks, often performing worse than a randomly initialized network after the same amount of training on the new task. In severe cases, optimization may stagnate, leaving the learning curve flat and posing a serious challenge to continual adaptation [Abbas et al., 2023].

Several diagnostics and mechanisms are proposed for explaining plasticity loss. For instance, the effective rank of the representations [Dohare et al., 2024], the rank of the empirical neural tangent kernel [Lyle et al., 2025, Tang et al., 2025], the directions of curvature [Lewandowski et al., 2024], and the proportion of dead/dormant neurons [Sokar et al., 2023, Ma et al., 2024]. Motivated by Box's philosophy that "all models are wrong, but some are useful," [Box, 1976, Box and Draper, 1986] we ask: can existing diagnostics predict a neural network's plasticity? To study this question, we first need to operationalize plasticity. Unfortunately, plasticity itself is operationalized differently across the literature. Dohare et al. [2024], Kumar et al. [2025] evaluate plasticity through learning speed. Wang et al. [2025] evaluate plasticity through final performance after a fixed training budget. Abbas et al. [2023], Lyle et al. [2023], Nikishin et al. [2023] evaluate plasticity through performance relative to a randomly initialized model. A consensus we perceive is that a network is plastic if additional gradient-based optimization produces a meaningful reduction in loss. Here, by meaningful reduction, we mean a reduction that is significant relative to the initial loss. We refer to this perceived consensus as trainability, and we view it as a concrete aspect of plasticity in this work. We will later mathematically formalize trainability as the k-step gain, the relative loss reduction achieved after k gradient steps on the target task, and use it as the target we wish to predict from diagnostic metrics. Note that this target is by nature task-dependent.

Recent empirical evidence suggests that some commonly used diagnostics are unreliable indicators of plasticity [Lyle et al., 2023, Lewandowski et al., 2024, Baveja et al., 2025]. However, existing falsification is largely empirical, and there remains a limited theoretical understanding of why such metrics can fail.

In this work, we show that several structural diagnostics can be misleading: they may assign a favorable value to a neural network checkpoint even when gradient descent cannot make progress on the relevant task. In particular, our theoretical counterexamples focus on label-agnostic structural metrics, such as representation rank and eNTK rank, which depend on the input distribution and checkpoint but not directly on task labels. These negative results suggest that favorable structural diagnostic values alone are insufficient for predicting future optimization progress. We therefore propose an optimization-derived metric that is directly motivated by the one-step descent dynamics of stochastic gradient descent.

Specifically, our paper makes the following contributions:

1. We prove by construction of counterexamples that some widely-adopted diagnostics of plasticity fail to predict the loss of task-dependent trainability in both regression and binary classification settings. In particular, we show that label-agnostic metrics, such as the effective rank, the 99% energy rank of the representation matrix, and the eNTK, can fail even in simple constructions. These results show that favorable structural diagnostic values alone do not guarantee future optimization progress.

2. Motivated by these negative results, we propose Optimization Readiness (OR), a lightweight optimization-derived metric designed to predict task-dependent trainability. We prove that OR lower-bounds the one-step optimization gain under standard smoothness assumptions, providing a theoretical guarantee of its predictive power.

3. Empirically, we evaluate OR on Slowly-Changing Regression and Permuted MNIST, two standard testbeds of loss of plasticity [Dohare et al., 2024], and demonstrate that it accurately predicts the relative ordering of checkpoints by their task-dependent trainability. In our ablation study, we additionally show that OR is data-efficient, achieving high pairwise ranking accuracy even with 10%, 1%, and 0.1% of the validation data in Slowly-Changing Regression and 10% and 1% of the Permuted MNIST validation set. The superiority of label-dependent metrics also carry over to out-of-distribution tasks, where we demonstrate that OR and other label-dependent diagnostics rank task-dependent trainabilities significantly more reliably than their label-agnostics counterparts on a different task family from training.

2 Preliminaries and Problem Setup

We study whether a diagnostic computed at a checkpoint can predict its *trainability* on a newly observed supervised task. A continual supervised-learning problem is a sequence of tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots$, where each task $\mathcal{T} = (\mathcal{X}, \mathcal{Y}, P_{\mathcal{XY}})$ consists of an input space, a label space, and a joint distribution over $\mathcal{X} \times \mathcal{Y}$. Tasks arrive sequentially, and samples within each task are drawn i.i.d. from the corresponding distribution. A learner is a neural network $f_\theta : \mathcal{X} \to \mathcal{Z}$, and all tasks share a pointwise loss $\ell : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$. The population loss on task $\mathcal{T}$ is $\mathcal{L}(\theta; \mathcal{T}) \doteq \mathbb{E}_{(x,y)\sim P_{\mathcal{XY}}}[\ell(f_\theta(x), y)]$.

Trainability target. For a checkpoint θ, our target quantity is the relative loss reduction obtained by continuing gradient-based optimization on $\mathcal{T}$. Let $g(\theta; \mathcal{T}) \doteq \nabla_\theta \mathcal{L}(\theta; \mathcal{T})$ denote the population gradient. For a mini-batch $\mathcal{B} \doteq \{(x_i, y_i)\}_{i=1}^m$, define $\widehat{\mathcal{L}}_{\mathcal{B}}(\theta; \mathcal{T}) \doteq \frac{1}{m}\sum_{i=1}^m \ell(f_\theta(x_i), y_i)$ as the empirical loss and $\widehat{g}_{\mathcal{B}}(\theta; \mathcal{T}) \doteq \nabla_\theta \widehat{\mathcal{L}}_{\mathcal{B}}(\theta; \mathcal{T})$. Starting from $\theta_0 = \theta$, stochastic gradient descent with learning rate $\eta > 0$ follows

$$\theta_{t+1} = \theta_t - \eta \widehat{g}_{\mathcal{B}_t}(\theta_t; \mathcal{T}), \tag{1}$$

for $t = 0, \ldots, k-1$, where $\mathcal{B}_0, \ldots, \mathcal{B}_{k-1}$ are sampled i.i.d. from $P_{\mathcal{XY}}$. We define the k-step gain as

Definition 2.1 (mini-batch k-step gain)**.**

$$\mathcal{G}^{(k)}(\theta; \mathcal{T}) \doteq \begin{cases} \dfrac{\mathcal{L}(\theta; \mathcal{T}) - \mathbb{E}_{\mathcal{B}_0,\ldots,\mathcal{B}_{k-1}}[\mathcal{L}(\theta_k; \mathcal{T})]}{\mathcal{L}(\theta; \mathcal{T})}, & \mathcal{L}(\theta; \mathcal{T}) > 0, \\ 0, & \mathcal{L}(\theta; \mathcal{T}) = 0. \end{cases}$$

Thus, $\mathcal{G}^{(k)}$ measures the expected relative improvement after k gradient steps, and we use it as an operational proxy for plasticity. When the neural network loses its plasticity, we expect the k-step gain to be small across different k's. Normalization in $\mathcal{G}^{(k)}$ is important because the absolute reduction in loss can be misleading across checkpoints with different initial losses. For example, a checkpoint close to convergence may have little room for absolute improvement, and therefore may exhibit a smaller absolute loss reduction than a randomly initialized checkpoint. This does not necessarily mean that it is less trainable — after the same number of optimization steps, it may reduce its remaining loss by a comparable, or even larger, proportion. Normalizing by the initial loss, therefore, makes the gain comparable across checkpoints at different stages of optimization.

Matrix-rank diagnostics. Several existing plasticity diagnostics are based on the spectrum of representations, neural tangent kernels (NTK, Jacot et al. [2018]), or curvature matrices. For a matrix $Z \in \mathbb{R}^{m\times n}$, let $\sigma_1 \geq \cdots \geq \sigma_q \geq 0$, $q = \min\{m, n\}$, denote its singular values. If $Z \neq 0$, define $p_i = \sigma_i / \sum_j \sigma_j$. The effective rank is $\mathrm{EffRank}(Z) \doteq \exp\left(-\sum_{i:p_i>0} p_i \log p_i\right)$, with the convention $\mathrm{EffRank}(0) = 0$. For $\tau \in (0, 1)$, the τ-energy rank is $\mathrm{EnRank}\tau(Z) \doteq \min\left\{k : \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{j=1}^r \sigma_j^2} \geq \tau\right\}$, where $r \doteq \mathrm{rank}(Z)$. Again, we assume $\mathrm{EnRank}_\tau(0) = 0$. In this paper, we use $\tau = 0.99$. Let $\phi_\theta(x) \in \mathbb{R}^d$ be the penultimate-layer representation of f_θ. For inputs $X \doteq (x_1, \ldots, x_m)$, let $\Phi_\theta(X) \in \mathbb{R}^{m\times d}$ be the representation matrix whose i-th row is $\phi_\theta(x_i)$. We consider $\mathrm{EffRank}(\Phi_\theta(X))$ and $\mathrm{EnRank}_{0.99}(\Phi_\theta(X))$ as representation-rank diagnostics.

For scalar-output networks, the empirical neural tangent kernel (eNTK) on X is $\widehat{K}(X; \theta)_{ij} \doteq \langle \nabla_\theta f_\theta(x_i), \nabla_\theta f_\theta(x_j) \rangle$. We use $\mathrm{EnRank}_{0.99}\left(\widehat{K}(X; \theta)\right)$ as the eNTK-rank diagnostic. We also compare against a curvature diagnostic, where for $H(\theta; \mathcal{T}) \doteq \nabla_\theta^2 \mathcal{L}(\theta; \mathcal{T})$, we measure $\mathrm{EnRank}_{0.99}(H(\theta; \mathcal{T}))$, or an empirical approximation thereof.

Active-neuron diagnostic. For a network with L hidden layers, let $a_i(x; \theta) \in \mathbb{R}^{w_i}$ denote the activation vector of hidden layer i. Given inputs $X \doteq \{x_1, \ldots, x_m\}$, define the relative activation score of neuron j in layer i as $s_{i,j}(X; \theta) \doteq \frac{\frac{1}{m}\sum_{x\in X} |a_i(x;\theta)_j|}{\frac{1}{m}\sum_{x\in X}\frac{1}{w_i}\sum_{r=1}^{w_i} |a_i(x;\theta)_r| + \varepsilon}$, where $\varepsilon > 0$ is a small constant to prevent division by zero. For a threshold $\tau_{\mathrm{act}} \geq 0$, the active-neuron fraction is $A_{\tau_{\mathrm{act}}}(X; \theta) \doteq \frac{\sum_{i=1}^L \sum_{j=1}^{w_i} \mathbb{I}(s_{i,j}(X;\theta) \geq \tau_{\mathrm{act}})}{\sum_{i=1}^L w_i}$, where $\mathbb{I}(\cdot)$ is the indicator function.

3 Theoretical Falsifications of Structural Plasticity Metrics

In this section, we present theoretical counterexamples to some previously adopted diagnostic metrics of plasticity. We will show, by construction, that the neural network cannot make any progress at all at a particular checkpoint in some tasks, even though those metrics would assign a favorable value to that checkpoint. In particular, the structural metrics we study in this section do not use task labels directly, and they exhibit a similar failure mode across the same checkpoint-task construction. We provide theoretical falsifications for both regression and binary classification tasks.

Shared Construction. Fix $n \geq 4$ and $M \geq n$. For binary classification, we additionally assume that n is even. Let $\bar{X} \doteq I_n$ and $\bar{x}_i \doteq e_i$ for $i = 1, \ldots, n$, and consider the two-layer ReLU network $f_\theta(x) \doteq v^\top \mathrm{ReLU}(Wx)$, where $\theta \doteq (v, W)$, $v \in \mathbb{R}^M$, and $W \in \mathbb{R}^{M \times n}$. Then, for any checkpoint θ, define the representation matrix on the full support by $\Phi_\theta(\bar{X})_{i,j} \doteq \mathrm{ReLU}(w_j^\top \bar{x}_i) = \mathrm{ReLU}(W_{j,i}), \quad \Phi_\theta(\bar{X}) \in \mathbb{R}^{n \times M}$. Given a label vector $\bar{Y} \doteq [\bar{y}_1 \quad \cdots \quad \bar{y}_n]^\top$ and a differentiable pointwise loss $\ell(z, y)$, let $\bar{\mathcal{T}}(\bar{Y}, \ell)$ denote the full-support task that samples uniformly from $\{(\bar{x}_i, \bar{y}_i)\}_{i=1}^n$. Then, its population loss is $\mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) = \frac{1}{n} \sum_{i=1}^n \ell(f_\theta(\bar{x}_i), \bar{y}_i)$. We analyze the full-support gradient descent $\theta_{s+1} = \theta_s - \eta \nabla_\theta \mathcal{L}(\theta_s; \bar{\mathcal{T}}), \quad \theta_0 = \theta$, and the corresponding full-support k-step gain

$$\mathcal{G}_{\mathrm{full}}^{(k)}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) \doteq \begin{cases} \dfrac{\mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) - \mathcal{L}(\theta_k; \bar{\mathcal{T}}(\bar{Y}, \ell))}{\mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell))}, & \mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) > 0, \\ 0, & \mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) = 0. \end{cases}$$

This is a favorable setting for the learner because the update uses the exact population gradient rather than a noisy mini-batch gradient. However, a zero full-support gradient does not necessarily imply that individual-example gradients vanish, as they may cancel in the full-support average. Thus, zero full-support gain does not in general imply zero mini-batch gain.

The following three lemmas are the only common ingredients needed for both counterexamples.

Lemma 3.1 (one step update)**.** *For a task $\bar{\mathcal{T}}(\bar{Y}, \ell)$, define $q(\theta) \doteq [q_1(\theta), \ldots, q_n(\theta)]^\top$, where $q_i(\theta) \doteq \frac{\partial}{\partial z} \ell(z, \bar{y}_i)\big|_{z = f_\theta(\bar{x}_i)}$. If $\theta = (0, W)$, then $\nabla_v \mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) = \frac{1}{n} \Phi_\theta(\bar{X})^\top q(\theta)$ and $\nabla_W \mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) = 0$. Consequently, if $\Phi_\theta(\bar{X})^\top q(\theta) = 0$, then full-support gradient descent leaves θ unchanged for every learning rate $\eta > 0$, and hence $\mathcal{G}_{\mathrm{full}}^{(k)}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) = 0$ for all $k \geq 1$. Moreover, one full-support gradient step from $\theta_0 \doteq (0, W)$ satisfies $v_1 = -\frac{\eta}{n} \Phi_{\theta_0}(\bar{X})^\top q(\theta_0)$ and $W_1 = W$, and therefore $f_{\theta_1}(\bar{X}) = -\eta G(W) q(\theta_0)$, where $G(W) \doteq \frac{1}{n} \Phi_{\theta_0}(\bar{X}) \Phi_{\theta_0}(\bar{X})^\top$ is the Gram matrix.*

We give the proof in Appendix B.1.

Lemma 3.2 (rank metrics from a positive semi-definite activation matrix)**.** *Let $B \in \mathbb{R}^{n \times n}$ be symmetric positive semidefinite with nonnegative entries. Fix $\alpha > 0$ and suppose $v = 0$ and $\Phi_\theta(\bar{X}) = \alpha \begin{bmatrix} B & 0_{n \times (M-n)} \end{bmatrix}$. If the nonzero eigenvalues of B are $b_1 \geq b_2 \geq \cdots \geq b_r > 0$, where r is the rank of B, then the nonzero singular values of $\Phi_\theta(\bar{X})$ are $\alpha b_1, \ldots, \alpha b_r$, and the nonzero singular values of the eNTK $\widehat{K}(\bar{X}; \theta)$ on $\bar{X}$ are $\alpha^2 b_1^2, \ldots, \alpha^2 b_r^2$.*

The proof is in Appendix B.2.

Lemma 3.3 (random ReLU Gram concentration)**.** *Let $W_{\mathrm{rand}} \in \mathbb{R}^{M \times n}$ have i.i.d. entries sampled from*

$$\begin{cases} +\sqrt{2/n}, & \text{with probability } 1/2, \\ -\sqrt{2/n}, & \text{with probability } 1/2. \end{cases}$$

Let $\theta_{\mathrm{rand}} \doteq (0, W_{\mathrm{rand}})$, and define $G_{\mathrm{rand}} \doteq \frac{1}{n} \Phi_{\theta_{\mathrm{rand}}}(\bar{X}) \Phi_{\theta_{\mathrm{rand}}}(\bar{X})^\top$. Using $\mathbf{1} \in \mathbb{R}^n$ to denote the all-one vector, we have $\bar{G} \doteq \mathbb{E}[G_{\mathrm{rand}}] = \lambda(J_n + I_n)$, where $\lambda \doteq \frac{M}{2n^2}$ and $J_n = \mathbf{1}_n \mathbf{1}_n^\top$. Moreover, there is a universal constant $C > 0$, such that for every $\delta \in (0, 1)$ and every $\rho > 1$, if $M \geq C n^2 \rho \log\left(\frac{2n}{\delta}\right)$, then, with probability at least $1 - \delta$, $\left\| G_{\mathrm{rand}} - \bar{G} \right\|_2 \leq \frac{\lambda}{\sqrt{\rho}}$.

We provide the proof in Appendix B.3.

Regression Counterexample. We now construct a counterexample for the regression task, showing that there exists a checkpoint-task combination in which the model does not learn, even though the metric values are near-optimal. In addition, we present a random initialization of the hidden-layer weights that guarantees learning with high probability.

Theorem 3.4. *Let* $\bar{Y}_{\mathrm{reg}} \doteq e_{n-1} - e_n$ *and* $\ell_{\mathrm{sq}}(z, y) \doteq \frac{1}{2}(z-y)^2$. *We use* $\bar{\mathcal{T}}_{\mathrm{reg}} \doteq \bar{\mathcal{T}}(\bar{Y}_{\mathrm{reg}}, \ell_{\mathrm{sq}})$ *as a shorthand. There exists a checkpoint* θ_{pre} *such that the following hold.*

(i) The checkpoint has high representation and eNTK rank. In particular,

$$\begin{cases} \mathrm{EnRank}_{0.99}\big(\Phi_{\theta_{\mathrm{pre}}}(\bar{X})\big) & = \lceil 0.99(n-2) \rceil\,; \\ \mathrm{EffRank}\big(\Phi_{\theta_{\mathrm{pre}}}(\bar{X})\big) & = n-2; \\ \mathrm{EnRank}_{0.99}\Big(\widehat{K}(\bar{X}; \theta_{\mathrm{pre}})\Big) & = \lceil 0.99(n-2) \rceil\,. \end{cases}$$

(ii) Yet, the checkpoint is completely stuck under both full-support gradient descent and mini-batch SGD: $\mathcal{L}(\theta_{\mathrm{pre}}; \bar{\mathcal{T}}_{\mathrm{reg}}) = \frac{1}{n}$ *and* $\mathcal{G}^{(k)}_{\mathrm{full}}(\theta_{\mathrm{pre}}; \bar{\mathcal{T}}_{\mathrm{reg}}) = \mathcal{G}^{(k)}(\theta_{\mathrm{pre}}; \bar{\mathcal{T}}_{\mathrm{reg}}) = 0$ *for all* $k \geq 1$, $\eta > 0$, *and mini-batch sizes* $m \geq 1$.

(iii) In contrast, let $\theta_{\mathrm{rand}} \doteq (0, W_{\mathrm{rand}})$ *be the random initialization in Lemma 3.3. Setting* $\eta = \frac{2n^2}{M}$, *there exists a universal constant* $C > 0$, *such that for every* $\delta \in (0,1)$ *and* $M_{\mathrm{gap}} > 1$, *if* $M \geq Cn^2 M_{\mathrm{gap}} \log\left(\frac{2n}{\delta}\right)$, *then, with probability at least* $1-\delta$, $\mathcal{G}^{(1)}_{\mathrm{full}}(\theta_{\mathrm{rand}}; \bar{\mathcal{T}}_{\mathrm{reg}}) \geq 1 - \frac{1}{M_{\mathrm{gap}}}$.

The proof is in Appendix B.4.

Binary Classification Counterexample Here, we present the counterexample for the binary classification task, mirroring the goal of the regression counterexample.

Theorem 3.5. *Assume* $n \geq 4$ *is even. Let*

$$(\bar{Y}_{\mathrm{cls}})_i = \begin{cases} 1, & i = 1, \ldots, n/2, \\ 0, & i = n/2+1, \ldots, n, \end{cases}$$

and define the centered sign vector $\bar{S}_{\mathrm{cls}} \doteq 2\bar{Y}_{\mathrm{cls}} - \mathbf{1}_n \in \{-1, +1\}^n$. *Let the binary logistic loss under the* $0/1$*-label convention be* $\ell_{\log}(z, y) \doteq \log(1 + \exp(z)) - yz$, *for* $y \in \{0, 1\}$. *Equivalently, we have* $\ell_{\log}(z, y) = \log\left(1 + \exp\left(-(2y-1)z\right)\right)$. *Define* $\bar{\mathcal{T}}_{\mathrm{cls}} \doteq \bar{\mathcal{T}}(\bar{Y}_{\mathrm{cls}}, \ell_{\log})$ *as a shorthand. There exists a checkpoint* θ_{pre}, *such that the following hold.*

(i) The checkpoint has high representation and eNTK rank. In particular,

$$\begin{cases} \mathrm{EnRank}_{0.99}\big(\Phi_{\theta_{\mathrm{pre}}}(\bar{X})\big) & = \lceil 0.99(n+2) - 3 \rceil\,; \\ \mathrm{EffRank}\big(\Phi_{\theta_{\mathrm{pre}}}(\bar{X})\big) & = \frac{n}{2^{2/n}}; \\ \mathrm{EnRank}_{0.99}\Big(\widehat{K}(\bar{X}; \theta_{\mathrm{pre}})\Big) & = \lceil 0.99(n+14) - 15 \rceil\,. \end{cases}$$

(ii) Yet, the checkpoint is completely stuck under a full-support gradient descent, where $\mathcal{L}(\theta_{\mathrm{pre}}; \bar{\mathcal{T}}_{\mathrm{cls}}) = \log 2$ *and* $\mathcal{G}^{(k)}_{\mathrm{full}}(\theta_{\mathrm{pre}}; \bar{\mathcal{T}}_{\mathrm{cls}}) = 0$ *for all* $k \geq 1$ *and* $\eta > 0$.

(iii) In contrast, let $\theta_{\mathrm{rand}} \doteq (0, W_{\mathrm{rand}})$ *be the random initialization in Lemma 3.3. For* $M_{\mathrm{gap}} > 1$, *define* $\gamma_{M_{\mathrm{gap}}} \doteq \log\left(\frac{2M_{\mathrm{gap}}}{\log 2}\right)$ *and* $\eta \doteq \frac{4\gamma_{M_{\mathrm{gap}}} n^2}{M}$. *There exists a universal constant* $C > 0$, *such that for every* $\delta \in (0,1)$, *if* $M \geq Cn^2 \gamma^2_{M_{\mathrm{gap}}} M^2_{\mathrm{gap}} \log\left(\frac{2n}{\delta}\right)$, *then, with probability at least* $1-\delta$, *it holds that* $\mathcal{G}^{(1)}_{\mathrm{full}}(\theta_{\mathrm{rand}}; \bar{\mathcal{T}}_{\mathrm{cls}}) \geq 1 - \frac{1}{M_{\mathrm{gap}}}$.

We give our proof in Appendix B.5.

Our constructions are intentionally stylized as they are not meant to model the typical trajectory of practical continual learning systems, but to isolate a failure mode showing that favorable structural diagnostic values alone do not guarantee optimization progress on a target task.

The two theorems show that high representation rank and high eNTK rank do not by themselves imply trainability on a target task. In both cases, the failure is caused by a label-dependent null

direction, where the checkpoint representation has large rank on the inputs, but the task gradient is exactly orthogonal to the directions that the output layer can use at initialization. This observation motivates an optimization-derived diagnostic that depends directly on the gradient signal and its reliability, rather than only on structural properties of the checkpoint.

4 Optimization Readiness

In this section, we propose Optimization Readiness (OR) as a predictive metric of trainability. OR is the product of *gradient strength* and *gradient reliability*, where the former measures if there is a gradient signal at all, and the latter indicates the signal-to-noise ratio. Unlike diagnostics motivated primarily by structural or geometric properties of the checkpoint, OR is derived from the expected descent dynamics of stochastic gradient descent. This connection allows us to prove a lower bound on one-step gain under a smoothness assumption and an appropriate learning rate.

Let $\mathcal{T} \doteq (\mathcal{X}, \mathcal{Y}, P_{\mathcal{XY}})$ be an arbitrary supervised learning task and $\mathcal{B}$ be a mini-batch of size m, where each input-label pair from $\mathcal{B}$ is drawn i.i.d. from $P_{\mathcal{XY}}$. We formally state the assumptions and define OR as follows.

Assumption 4.1 (unbiasedness)**.** For any checkpoint θ, it holds that $\mathbb{E}_{\mathcal{B}}[\widehat{g}_{\mathcal{B}}(\theta; \mathcal{T})] = g(\theta; \mathcal{T})$.

Assumption 4.2 (smoothness)**.** The population loss $\mathcal{L}(\theta; \mathcal{T})$ is differentiable with respect to θ and β-smooth, such that for all θ_1, θ_2, it holds that $\|\nabla_\theta \mathcal{L}(\theta_1; \mathcal{T}) - \nabla_\theta \mathcal{L}(\theta_2; \mathcal{T})\|_2 \leq \beta \|\theta_1 - \theta_2\|_2$ for some $\beta > 0$.

The Euclidean norm of the gradient vector is a straightforward way to monitor the magnitude of the gradient signal. When the gradient norm becomes zero, gradient descent stops updating the parameters. In addition, gradient norm alone can be misleading because it does not distinguish checkpoints at different stages of convergence. Hence, we normalize the squared gradient norm by the population loss and define gradient strength as

Definition 4.3 (gradient strength)**.** $S(\theta; \mathcal{T}) \doteq \frac{\|g(\theta;\mathcal{T})\|_2^2}{\mathcal{L}(\theta;\mathcal{T})}$.

The noise in gradient estimation can also dictate the progress of gradient descent, as discussed in Baveja et al. [2025]. Gradient descent may fail to reduce population loss if the gradient signal is overwhelmed by noise, even when the gradient is strong. In light of this observation, we define gradient reliability as

Definition 4.4 (gradient reliability)**.** $R(\theta; \mathcal{T}) \doteq \frac{\|g(\theta;\mathcal{T})\|_2^2}{\|g(\theta;\mathcal{T})\|_2^2 + V_{\mathcal{B}}(\theta;\mathcal{T})}$, where $V_{\mathcal{B}}(\theta; \mathcal{T}) \doteq \mathbb{E}_{\mathcal{B}}\left[\|\widehat{g}_{\mathcal{B}}(\theta; \mathcal{T}) - g(\theta; \mathcal{T})\|_2^2\right]$.

Notably, if Assumption 4.1 holds, then $R(\theta; \mathcal{T}) = \frac{\|g(\theta;\mathcal{T})\|_2^2}{\|g(\theta;\mathcal{T})\|_2^2 + \left(\mathbb{E}_{\mathcal{B}}\left[\|\widehat{g}_{\mathcal{B}}(\theta;\mathcal{T})\|_2^2\right] - \|g(\theta;\mathcal{T})\|_2^2\right)} = \frac{\|g(\theta;\mathcal{T})\|_2^2}{\mathbb{E}_{\mathcal{B}}\left[\|\widehat{g}_{\mathcal{B}}(\theta;\mathcal{T})\|_2^2\right]}$. We assume the mini-batch size to estimate $R(\theta; \mathcal{T})$ is consistent with the gradient descent procedure. We therefore define

Definition 4.5 (Optimization Readiness)**.**

$$\mathrm{OR}(\theta; \mathcal{T}) \doteq \begin{cases} S(\theta; \mathcal{T}) R(\theta; \mathcal{T}), & \mathcal{L}(\theta; \mathcal{T}) > 0 \text{ and } \mathbb{E}_{\mathcal{B}}\left[\|\widehat{g}_{\mathcal{B}}(\theta; \mathcal{T})\|_2^2\right] > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 4.6 (Optimization Readiness Lower-Bounds One-Step Gain)**.** *Let $\mathcal{T} \doteq (\mathcal{X}, \mathcal{Y}, P_{\mathcal{XY}})$ be any supervised learning task and θ be an arbitrary parameterization of the neural network. Let $\theta_0 \doteq \theta$ and θ_1 be the parameters after one step of mini-batch gradient descent as in (1) with $\mathcal{B}_0 \doteq \{(x_1, y_1), \ldots, (x_m, y_m)\}$, where $(x_i, y_i) \overset{\text{i.i.d.}}{\sim} P_{\mathcal{XY}}$ for $i = 1, \ldots, m$. Suppose Assumption 4.1 and 4.2 hold. If $0 < \eta < R(\theta; \mathcal{T})/\beta$, then it holds that $\mathcal{G}^{(1)}(\theta; \mathcal{T}) \geq \frac{\alpha(1-\alpha)}{\beta} \mathrm{OR}(\theta; \mathcal{T})$, where $\alpha = \eta\beta / R(\theta; \mathcal{T})$ and β is the smoothness constant.*

The proof can be found in Appendix B.6.

Table 1: Dataset generation statistics for Slowly-Changing Regression and Permuted MNIST.

	Slowly-Changing Regression	Permuted MNIST
Training tasks per sequence	1,000	800
Training samples per sequence	1,000,000	8,000,000
Validation samples per task	10,000	10,000
Training sequences	20	20
Validation tasks	30	10
Task shift mechanism	Bit flipping every 1,000 samples	Random pixel permutation per task

5 Experiments

In this section, we empirically verify if the metrics of plasticity can reliably predict k-step gains. We use Slowly-Changing Regression (SCR) for continual regression and Permuted MNIST (P-MNIST) for continual classification as testbeds [Dohare et al., 2024]. For SCR, the features are binary vectors of dimension $u + v$, where the first u dimensions are slowly-changing bits, and the remaining v dimensions are random bits. Suppose each task has N data points, the v random bits are randomly sampled for each data point, while the u slowly changing bits stay constant for N steps. After N steps, one of the u bits is uniformly randomly selected and flipped from 1 to 0 or from 0 to 1, thereby creating a slowly shifting distribution in the input space across tasks. The labels are generated by a linear threshold unit (LTU) network [Dohare et al., 2024] with its weights randomly initialized to be $+1$ or -1 and fixed across all tasks. Regarding P-MNIST, we generate a random permutation of the pixels for each task and apply it to all images within that task, while keeping the original labels. We flatten the images row-wise to form the feature vectors.

We compare OR against representative structural diagnostics, including representation-rank metrics, eNTK rank, and active-neuron fraction. We also include Hessian rank as a task-dependent curvature baseline, allowing us to test whether OR remains useful beyond purely structural input-checkpoint diagnostics.

Data Generation. We generate separate training and validation tasks for both SCR and P-MNIST. The training tasks are designed to induce loss of plasticity through continual adaptation to non-stationary data streams, while the validation tasks are used to evaluate the predictive power of plasticity metrics on held-out stationary tasks. Detailed dataset statistics are summarized in Table 1.

Checkpoint Generation. We initialize and train a multi-layer perceptron (MLP) with ReLU activation on each training sequence of SCR and P-MNIST. The purpose of training is to obtain a set of checkpoints with varying trainabilities for later testing of the metrics. The MLP for SCR has 2 hidden layers of 5 units, whereas the MLP for P-MNIST has 3 hidden layers of 100 units. We employ mean-squared error for SCR and cross-entropy loss for P-MNIST without regularization. We use Adam [Kingma and Ba, 2015] as the optimizer with batch size $m = 1$. The learning rates are set to $\eta = 0.01$ and $\eta = 0.003$ for SCR and P-MNIST, respectively. Figure 1 shows the learning curves for both tasks. There is a clear sign of plasticity loss in both scenarios. As we save a checkpoint every 5 tasks, we collect a wide range of checkpoints with varying levels of trainability.

k-Step Gain and Metric Estimation. Given a validation task $\mathcal{T}_{\mathrm{val}}$ and a checkpoint θ, we estimate the k-step gain by running stochastic gradient descent for k steps from θ on mini-batches sampled from the validation set, and then measuring the resulting relative reduction in empirical loss. We repeat this procedure over multiple independent optimization rollouts to approximate the expectation in Definition 2.1.

We estimate OR using empirical counterparts of its two components. The population gradient is approximated by the gradient of the full validation set, while the expected mini-batch squared gradient norm is estimated using independently sampled mini-batches from the validation set. Together, these quantities give empirical estimates of gradient strength, gradient reliability, and OR.

For the baseline diagnostics, we compute representation-rank metrics, active-neuron fraction, eNTK rank, and Hessian-rank metrics on the same validation data whenever applicable. For SCR, the network is small enough to compute the exact Hessian and eNTK rank. For P-MNIST, we omit the eNTK rank because the vector-valued output makes the corresponding empirical kernel less directly comparable to the scalar-output case, and we approximate the Hessian rank using the Gram-matrix

Table 2: Pairwise ranking accuracy (mean $\pm$ standard error) across tasks.

Task	Metric	$k = 1$	$k = 10$	$k = 100$
Slowly-Changing Regression	OR	**0.987 $\pm$ 0.001**	**0.988 $\pm$ 0.001**	**0.977 $\pm$ 0.002**
	Repr Effective Rank	0.698 $\pm$ 0.001	0.700 $\pm$ 0.001	0.704 $\pm$ 0.001
	Repr 99% Energy Rank	0.689 $\pm$ 0.002	0.690 $\pm$ 0.002	0.694 $\pm$ 0.002
	eNTK 99% Energy Rank	0.765 $\pm$ 0.002	0.768 $\pm$ 0.002	0.779 $\pm$ 0.003
	Hessian 99% Energy Rank	0.734 $\pm$ 0.003	0.737 $\pm$ 0.003	0.746 $\pm$ 0.005
	Active Neuron Fraction	0.671 $\pm$ 0.007	0.673 $\pm$ 0.007	0.679 $\pm$ 0.007
Permuted MNIST	OR	**0.906 $\pm$ 0.005**	**0.905 $\pm$ 0.005**	**0.899 $\pm$ 0.004**
	Repr Effective Rank	0.858 $\pm$ 0.002	0.864 $\pm$ 0.003	0.873 $\pm$ 0.002
	Repr 99% Energy Rank	0.828 $\pm$ 0.003	0.834 $\pm$ 0.003	0.844 $\pm$ 0.003
	Hessian 99% Energy Rank	0.821 $\pm$ 0.003	0.753 $\pm$ 0.007	0.717 $\pm$ 0.007
	Active Neuron Fraction	0.346 $\pm$ 0.003	0.377 $\pm$ 0.003	0.391 $\pm$ 0.003

estimator of Lewandowski et al. [2024]. Full estimator definitions and implementation details are provided in Appendix C.1.

Qualitative Analysis. We first conduct a qualitative analysis to examine the relationship between the metric values and their corresponding k-step gains. Figure 2 displays the metric values against the ground-truth k-step gains. We observe an approximately linear relationship between the 1-step gain and OR, whereas the other metrics do not exhibit a clear positive correlation with trainability. Even for $k = 100$, OR maintains a strong positive relationship with the k-step gain, suggesting that it remains informative beyond the one-step setting considered in our theory. However, we note that some points lie on the x-axis for OR in P-MNIST. We attribute these to false positives, where low-trainability checkpoints receive high metric values. Nonetheless, this is not surprising, as we do not guarantee that the assumptions in our theory hold in these empirical tests, and OR's correlation with trainability is still considerably clearer than the others.

Next, we visualize the evolution of the k-step gains and the metric values throughout continual training in Figure 3, where the top three rows correspond to the k-step gains and the bottom row shows the metric trajectories. Among all metrics, the trajectory of OR aligns most closely with the evolution of the k-step gains, particularly in SCR. Interestingly, trainability is not maximized near random initialization. Instead, the k-step gains initially increase rapidly before gradually declining. We conjecture that this phenomenon may be related to forward transfer [Lopez-Paz and Ranzato, 2017], where previously learned tasks temporarily facilitate adaptation to similar future tasks before the effects of plasticity loss begin to dominate.

Quantitative Analysis. We next quantitatively evaluate whether each metric preserves the ordering of checkpoints by trainability. Let $\omega(\theta; \mathcal{T})$ denote the value of a diagnostic metric at checkpoint θ on task $\mathcal{T}$. For two checkpoints θ_1 and θ_2, a metric ranks them correctly if its ordering agrees with the ordering induced by the k-step gain, i.e., $(\omega(\theta_1; \mathcal{T}) - \omega(\theta_2; \mathcal{T}))\big(\mathcal{G}^{(k)}(\theta_1; \mathcal{T}) - \mathcal{G}^{(k)}(\theta_2; \mathcal{T})\big) > 0$. We therefore measure pairwise ranking accuracy, defined as the fraction of checkpoint pairs whose metric ordering agrees with their ordering under the estimated ground-truth k-step gain:

$$\frac{\sum_{(\theta_1,\theta_2)\in\tilde{\Theta}\times\tilde{\Theta}} \mathbb{I}(\Delta_\omega(\theta_1,\theta_2;\mathcal{T})\Delta_{\mathcal{G}^{(k)}}(\theta_1,\theta_2;\mathcal{T}) > 0)\mathbb{I}(\Delta_{\mathcal{G}^{(k)}}(\theta_1,\theta_2;\mathcal{T}) \neq 0)}{\sum_{(\theta_1,\theta_2)\in\tilde{\Theta}\times\tilde{\Theta}} \mathbb{I}(\Delta_{\mathcal{G}^{(k)}}(\theta_1,\theta_2;\mathcal{T}) \neq 0)},$$

where $\tilde{\Theta}$ is the set of recorded checkpoints, $\Delta_\omega(\theta_1, \theta_2; \mathcal{T}) \doteq \omega(\theta_1; \mathcal{T}) - \omega(\theta_2; \mathcal{T})$, and $\Delta_{\mathcal{G}^{(k)}}(\theta_1, \theta_2; \mathcal{T}) \doteq \mathcal{G}^{(k)}(\theta_1; \mathcal{T}) - \mathcal{G}^{(k)}(\theta_2; \mathcal{T})$. This quantity is closely related to Kendall-style rank correlation measures [Kendall, 1938], but has a direct probabilistic interpretation as the probability that a metric correctly ranks two checkpoints by trainability.

We collect 201 checkpoints per SCR run and 161 per P-MNIST run. Table 2 reports the pairwise ranking accuracies across tasks and horizons. OR achieves the highest ranking accuracy for both tasks and all values of k, with the largest margin appearing in SCR.

Subsampling Ablation. The preceding results estimate all metrics using the full validation set. However, access to large held-out datasets may be unrealistic in practical continual learning scenarios. We therefore evaluate whether OR remains predictive when estimated from substantially fewer samples. For SCR, we estimate OR using 10%, 1%, and 0.1% of the validation set, corresponding

to 1,000, 100, and 10 data points, respectively. For P-MNIST, we use 10% and 1% of the validation set, corresponding to 1,000 and 100 data points. In all subsampling experiments, we estimate OR using only 16 mini-batches of size $m = 1$.

Figure 4 summarizes the results. In SCR, OR remains the most accurate metric even when estimated from only 0.1% of the validation data, suggesting that the full-data configuration used above is conservative. In P-MNIST, the accuracy of OR decreases under subsampling, especially for larger k, and can fall below representation-rank metrics computed using the full validation set. Nevertheless, it remains the strongest predictor among the metrics compared under the same data budget. This observation also suggests that OR can remain informative even when estimated using mini-batches smaller than those used in the k-step gain rollouts.

Out-of-domain Generalization. So far, we have validated the checkpoints' trainability within the same domain. A natural question is: Do the metrics predict the future trainability of checkpoints on task B after they are trained on task A? To answer this question, we employ the Fashion-MNIST dataset [Xiao et al., 2017], which has the same input and output dimensions as P-MNIST. We construct 10 validation tasks, each with 1,000 random samples from the dataset. Then, we apply the same permutation and flattening scheme as P-MNIST to process the images. Without further training, we use the checkpoints generated from learning P-MNIST directly on the permuted Fashion-MNIST validation set. We measure whether each metric computed on the target tasks preserves checkpoint ordering by estimated one-step gain, using the pairwise ranking accuracy defined above. Furthermore, we also include individual components of OR, such as the unnormalized squared gradient norm $\|g(\theta; \mathcal{T}_{\text{val}})\|_2^2$, gradient strength ($S$) alone, and gradient reliability (R) alone in the analysis. Table 3 summarizes the results. Overall, label-dependent diagnostics considerably outperform label-agnostics metrics. Among them, OR has a slight edge. This result is consistent with our theoretical insights.

Table 3: Mean pairwise ranking accuracy for one-step gain on permuted Fashion-MNIST.

Metric	Pairwise ranking accuracy ($k = 1$)
Optimization Readiness (OR)	**0.971 ± 0.001**
Squared Gradient Norm ($\|g(\theta; \mathcal{T}_{\text{val}})\|_2^2$)	0.965 ± 0.001
Gradient Strength (S)	<u>0.969</u> ± 0.001
Gradient Reliability (R)	0.950 ± 0.001
Repr Effective Rank	0.889 ± 0.001
Repr 99% Energy Rank	0.859 ± 0.001
Hessian 99% Energy Rank	0.852 ± 0.001
Active Neuron Fraction	0.534 ± 0.008

6 Related Work

Explanations and Diagnostics of Loss of Plasticity. Understanding and diagnosing loss of plasticity has been an active area of research in both continual supervised and reinforcement learning settings [Dohare et al., 2024, Lyle et al., 2025]. A prominent line of work studies the structural properties of neural representations. In particular, representation collapse has been proposed as a key mechanism underlying plasticity loss, where metrics such as the effective rank and the 99% energy rank of the representation are used to explain and detect degraded trainability [Dohare et al., 2023, 2024]. Another widely studied class of diagnostics focuses on spectral properties of the neural tangent kernel (NTK), including its rank, sharpness, and condition number [Lyle et al., 2025, Tang et al., 2025], where ill-conditioned NTKs are associated with optimization difficulty. Loss of curvature directions, reflected by a collapse of the Hessian rank, has also been proposed as an explanation for plasticity loss [Lewandowski et al., 2024, Prakash et al., 2026]. More recently, Wang et al. [2025] study finite-time Lyapunov exponents (FTLEs), arguing that both excessively small and excessively large FTLEs can impair plasticity. On a more microscopic level, dormant or dead neurons are also used as indicators of plasticity loss [Sokar et al., 2023, Ma et al., 2024].

While these works provide valuable explanations and empirical correlations, many existing diagnostics are motivated primarily through structural or geometric properties of the checkpoint, rather than through a direct characterization of future optimization progress. It therefore remains unclear

whether such diagnostics can reliably predict future trainability. Our work studies this predictive question directly by evaluating diagnostics against future optimization gain and by proposing an optimization-derived metric with theoretical and empirical support.

Empirical Falsifications and Limitations of Existing Diagnostics. Several works have demonstrated that existing diagnostics cannot fully explain trainability in neural networks. Lyle et al. [2023] show that metrics such as weight norm, weight rank, dead units, and feature rank can exhibit inconsistent correlations with plasticity across settings. Lewandowski et al. [2024] construct empirical counterexamples showing that update norm, neuron dormancy, representation rank, and weight norm can all be misleading indicators of plasticity. Likewise, Baveja et al. [2025] demonstrate that individual indicators such as Hessian rank, unit sign entropy, and gradient norm cannot fully characterize trainability on their own.

These works suggest that existing diagnostics are incomplete, but their analysis is primarily empirical. Our work complements this line of research by providing theoretical counterexamples that explicitly characterize failure modes of several widely used diagnostics.

7 Discussion and Conclusion

In this work, we study plasticity in deep continual learning from the perspective of future optimization progress. We propose k-step gain as an operational proxy for trainability and use it to evaluate whether existing diagnostics can predict how much a checkpoint will improve after further optimization on a target task.

Our theoretical results show that high structural diversity alone does not guarantee trainability. In particular, we construct regression and binary classification tasks on which metrics based on representation rank and eNTK rank assign favorable values to a checkpoint, even though full-support gradient descent cannot reduce the loss from that checkpoint. On the same tasks, we also show that randomly initialized weights are likely to make substantial one-step progress, highlighting a concrete gap between structural diagnostic values and actual optimization gain.

Motivated by this gap, we propose optimization readiness, an optimization-derived metric combining gradient strength and gradient reliability. Under a smoothness assumption and an appropriate learning rate, we prove that the optimization readiness lower bounds the one-step gain. Empirically, we find that OR reliably ranks checkpoints by their 1-, 10-, and 100-step gains on both Slowly-Changing Regression and Permuted MNIST, outperforming several commonly used diagnostics. Our subsampling ablation further suggests that OR can remain informative even when estimated with limited validation data and a small number of Monte Carlo samples.

Our study deliberately focuses on simple yet standard continual learning testbeds in order to isolate the predictive behavior of plasticity diagnostics. This controlled setting allows us to connect theory and empirical evaluation cleanly, but it also leaves several important directions for future work. First, our counterexamples use ReLU networks with one hidden layer. Extending the analysis to deeper networks and other activation functions would strengthen the theoretical scope. Second, our empirical evaluation focuses on supervised continual learning. Thus, testing whether the same observations hold in continual reinforcement learning and larger-scale architectures and problems is an important next step. Overall, our results suggest that diagnostics of plasticity should be evaluated not only as retrospective explanations of observed training behavior, but also as prospective predictors of future optimization gain.

Acknowledgments and Disclosure of Funding

This work is supported in part by the US National Science Foundation under the awards III-2128019, SLES-2331904, and CAREER-2442098, the Commonwealth Cyber Initiative's Central Virginia Node under the award VV-1Q26-001, VV-1Q27-007, and VV-1Q27005, 4VA at UVA, a Cisco Faculty Research Award, and a Google Research Award.

References

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C. Machado. Loss of plasticity in continual deep reinforcement learning. In *Proceedings of the Conference on Lifelong Learning Agents*, 2023.

Rahaf Aljundi. *Continual Learning in Neural Networks*. PhD thesis, KU Leuven, 2019.

Gunbir Singh Baveja, Alex Lewandowski, and Mark Schmidt. A unified noise-curvature view of loss of trainability. In *the NeurIPS Workshop on Optimization for Machine Learning*, 2025.

George E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 1976.

George E P Box and Norman R Draper. *Empirical model-building and response surfaces*. John Wiley & Sons, Inc., 1986.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Maintaining plasticity in deep continual learning. *arXiv Preprint*, 2023.

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*, 2024.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 1938.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. In *Proceedings of the Conference on Lifelong Learning Agents*, 2025.

Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C. Machado. Directions of curvature as an explanation for loss of plasticity. *arXiv Preprint*, 2024.

David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *Proceedings of the International Conference on Machine Learning*, 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. In *Proceedings of the Conference on Lifelong Learning Agents*, 2025.

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 *Proceedings of the International Conference on Learning Representations*, 2024.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and Andre Barreto. Deep reinforcement learning with plasticity injection. In *Advances in Neural Information Processing Systems*, 2023.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

Arjun Prakash, Naicheng He, Kaicheng Guo, Saket Tiwari, Tyrone Serapio, Ruo Yu Tao, Amy Greenwald, and George Konidaris. Spectral collapse drives loss of plasticity in deep continual learning. In *Proceedings of the International Conference on Machine Learning*, 2026.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2023.

Hongyao Tang, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, and Glen Berseth. Mitigating plasticity loss in continual reinforcement learning by reducing churn. In *Proceedings of the International Conference on Machine Learning*, 2025.

Haoyu Wang, Wei P Dai, Jiawei Zhang, Jialun Ma, Mingyi Huang, and Yuguo Yu. The dual nature of plasticity loss in deep continual learning: Dissection and mitigation. In *Advances in Neural Information Processing Systems*, 2025.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv Preprint*, 2017.

A Math Background

Theorem A.1. *(Matrix Bernstein inequality) Let $M_1, \ldots, M_k$ be independent random symmetric matrices such that $M_i \in \mathbb{R}^{n\times n}$, $\mathbb{E}[M_i] = 0$, and $\|M_i\| \le C$ almost surely for all $i \in \{1, \ldots, k\}$. Then, for all $t \ge 0$, it holds that*

$$\Pr\left(\left\|\sum_{i=1}^{k} M_i\right\| \ge t\right) \le 2n \exp\left(\frac{-t^2}{2(\sigma^2 + Ct/3)}\right),$$

where $\sigma^2 \doteq \left\|\sum_{i=1}^{k} \mathbb{E}[M_i^2]\right\|$.

B Proof of Theoretical Results

B.1 Proof of Lemma 3.1

Proof. For any x, we have

$$\nabla_v f_\theta(x) = \mathrm{ReLU}(Wx)$$

and

$$\nabla_{w_j} f_\theta(x) = v_j \mathbb{I}\big(w_j^\top x > 0\big)x.$$

Therefore,

$$\nabla_v \mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) = \frac{1}{n}\sum_{i=1}^{n} q_i(\theta)\mathrm{ReLU}(W\bar{x}_i) = \frac{1}{n}\Phi_\theta(\bar{X})^\top q(\theta),$$

and, for each hidden unit j,

$$\nabla_{w_j} \mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{Y}, \ell)) = \frac{1}{n}\sum_{i=1}^{n} q_i(\theta) v_j \mathbb{I}\big(w_j^\top \bar{x}_i > 0\big)\bar{x}_i.$$

If $v = 0$, then every first-layer gradient is zero. If additionally $\Phi_\theta(\bar{X})^\top q(\theta) = 0$, then the output-layer gradient is also zero, so $\nabla_\theta \mathcal{L}(\theta; \bar{\mathcal{T}}(\bar{\mathcal{Y}}, \ell)) = 0$. Hence gradient descent leaves θ unchanged for all time, which implies zero full-support gain.

For the one-step update formula, when $\theta_0 = (0, W)$, the first-layer gradient is zero and

$$\begin{cases} v_1 & = -\frac{\eta}{n}\Phi_{\theta_0}(\bar{X})^\top q(\theta_0), \\ W_1 & = W. \end{cases}$$

Thus, we have

$$f_{\theta_1}(\bar{X}) = \Phi_{\theta_0}(\bar{X})v_1 = -\frac{\eta}{n}\Phi_{\theta_0}(\bar{X})\Phi_{\theta_0}(\bar{X})^\top q(\theta_0) = -\eta G(W) q(\theta_0).$$

□

B.2 Proof of Lemma 3.2

Proof. Since B is symmetric positive semidefinite, its singular values are exactly its eigenvalues. Appending zero columns and multiplying by α gives nonzero singular values

$$\alpha b_1, \ldots, \alpha b_r$$

for $\Phi_\theta(\bar{X})$. Since $v = 0$, the first-layer portion of $\nabla_\theta f_\theta(\bar{x}_i)$ is zero for every i. Hence, the eNTK is generated only by output-layer gradients, and we have

$$\widehat{K}(\bar{X}; \theta) = \Phi_\theta(\bar{X})\Phi_\theta(\bar{X})^\top = \alpha^2 B^2.$$

The nonzero eigenvalues, and thus the nonzero singular values, of this positive semidefinite matrix are

$$\alpha^2 b_1^2, \ldots, \alpha^2 b_r^2.$$

□

B.3 Proof of Lemma 3.3

Proof. Define $a \doteq \sqrt{2/n}$ for simplicity. For each hidden unit j, define $h_j \in \mathbb{R}^n$ by

$$(h_j)_i \doteq \mathrm{ReLU}((W_{\mathrm{rand}})_{j,i}).$$

Then $(h_j)_i = a$ with probability $1/2$ and 0 with probability $1/2$, independently over i. Therefore, it holds that

$$G_{\mathrm{rand}} = \frac{1}{n} \sum_{j=1}^{M} h_j h_j^\top.$$

For $i \neq k$, we have

$$\mathbb{E}[(h_j)_i (h_j)_k] = \mathbb{E}[(h_j)_i]\mathbb{E}[(h_j)_k] = \frac{a^2}{4} = \frac{1}{2n},$$

while

$$\mathbb{E}\big[(h_j)_i^2\big] = \frac{a^2}{2} = \frac{1}{n}.$$

Thus, we have

$$\mathbb{E}\big[h_j h_j^\top\big] = \frac{1}{2n}(J_n + I_n)$$

and

$$\bar{G} \doteq \mathbb{E}[G_{\mathrm{rand}}] = \frac{M}{2n^2}(J_n + I_n) = \lambda(J_n + I_n).$$

It remains to prove concentration. Let $A_j \doteq \frac{1}{n} h_j h_j^\top$ and $Q_j \doteq A_j - \mathbb{E}[A_j]$. Then, $G_{\mathrm{rand}} - \bar{G} = \sum_{j=1}^{M} Q_j$, and the Q_j's are independent, zero-mean, symmetric matrices. Since $\|h_j\|_2^2 \leq na^2 = 2$, it holds that

$$\|A_j\|_2 = \frac{1}{n}\|h_j\|_2^2 \leq \frac{2}{n}.$$

By Jensen's inequality, $\|\mathbb{E}[A_j]\|_2 \leq 2/n$, and hence $\|Q_j\|_2 \leq \frac{4}{n}$. Moreover, we have

$$\big\|\mathbb{E}\big[Q_j^2\big]\big\|_2 \leq \big\|\mathbb{E}\big[A_j^2\big]\big\|_2 \leq \mathbb{E}[\|A_j\|_2^2] \leq \frac{4}{n^2}.$$

Therefore, the matrix-variance parameter satisfies

$$\left\| \sum_{j=1}^{M} \mathbb{E}\big[Q_j^2\big] \right\|_2 \leq \frac{4M}{n^2}.$$

By Matrix Bernstein (Theorem A.1), there is a universal constant $C_0 > 0$, such that with probability at least $1 - \delta$,

$$\big\|G_{\mathrm{rand}} - \bar{G}\big\|_2 \leq C_0 \left(\frac{\sqrt{M \log(2n/\delta)}}{n} + \frac{\log(2n/\delta)}{n} \right).$$

Let $z \doteq \log(2n/\delta)$. Since $\lambda = M/(2n^2)$, the last display gives

$$\frac{\left\|G_{\text{rand}} - \bar{G}\right\|_2}{\lambda} \leq 2C_0 \frac{n\sqrt{z}}{\sqrt{M}} + 2C_0 \frac{nz}{M}.$$

If $M \geq Cn^2\rho z$, then, after increasing the universal constant C if necessary, the right-hand side is at most $1/\sqrt{\rho}$. Consequently, we have

$$\left\|G_{\text{rand}} - \bar{G}\right\|_2 \leq \frac{\lambda}{\sqrt{\rho}},$$

as claimed. □

B.4 Proof of Theorem 3.4

Proof. Let

$$B_{\text{reg}} \doteq \begin{bmatrix} I_{n-2} & 0_{(n-2)\times 2} \\ 0_{2\times(n-2)} & 0_{2\times 2} \end{bmatrix}$$

and $\alpha \doteq \frac{1}{2}$. Choose $v_{\text{pre}} = 0$ and choose W_{pre}, such that

$$\Phi_{\theta_{\text{pre}}}(\bar{X}) = \alpha \begin{bmatrix} B_{\text{reg}} & 0_{n\times(M-n)} \end{bmatrix}.$$

This is realized, for example, by setting

$$(W_{\text{pre}})_{j,i} = \begin{cases} \alpha, & j = i \leq n-2, \\ -1, & \text{otherwise.} \end{cases}$$

The nonzero eigenvalues of B_{reg} are all equal to 1, with multiplicity $n-2$. Lemma 3.2 therefore implies that the nonzero singular values of $\Phi_{\theta_{\text{pre}}}(\bar{X})$ are

$$\underbrace{\alpha, \ldots, \alpha}_{n-2 \text{ times}},$$

and the nonzero singular values of $\widehat{K}(\bar{X}; \theta_{\text{pre}})$ are

$$\underbrace{\alpha^2, \ldots, \alpha^2}_{n-2 \text{ times}}.$$

Thus, we have

$$\begin{cases} \text{EnRank}_{0.99}\left(\Phi_{\theta_{\text{pre}}}(\bar{X})\right) & = \lceil 0.99(n-2) \rceil; \\ \text{EnRank}_{0.99}\left(\widehat{K}(\bar{X}; \theta_{\text{pre}})\right) & = \lceil 0.99(n-2) \rceil. \end{cases}$$

The normalized nonzero singular values of $\Phi_{\theta_{\text{pre}}}(\bar{X})$ are all $1/(n-2)$, so

$$\text{EffRank}\left(\Phi_{\theta_{\text{pre}}}(\bar{X})\right) = \exp\left(-\sum_{i=1}^{n-2} \frac{1}{n-2} \log \frac{1}{n-2}\right) = n-2.$$

This proves part (i) of Theorem 3.4.

For the squared loss, at a zero-output checkpoint, we have

$$q(\theta_{\text{pre}}) = -\bar{Y}_{\text{reg}}.$$

Since $B_{\text{reg}}\bar{Y}_{\text{reg}} = 0$, we have

$$\Phi_{\theta_{\text{pre}}}(\bar{X})^\top q(\theta_{\text{pre}}) = -\alpha \begin{bmatrix} B_{\text{reg}} \\ 0 \end{bmatrix} \bar{Y}_{\text{reg}} = 0.$$

Lemma 3.1 implies that full-support gradient descent leaves θ_{pre} unchanged for every learning rate. In fact, every individual-example gradient also vanishes. For $i \leq n-2$, both the prediction and

label are zero, so the squared-loss derivative is zero. For $i \in \{n-1, n\}$, all hidden preactivations are strictly negative and hence the hidden activations and their derivatives are zero. Therefore,

$$\nabla_\theta \ell_{\mathrm{sq}}\big(f_{\theta_{\mathrm{pre}}}(\bar{x}_i), \bar{y}_i\big) = 0 \qquad \text{for every } i = 1, \dots, n.$$

Every mini-batch gradient at θ_{pre} is consequently zero, so mini-batch SGD stays at θ_{pre} for every sequence of batches, batch size, learning rate, and number of steps. In addition, it holds that

$$\mathcal{L}(\theta_{\mathrm{pre}}; \bar{\mathcal{T}}_{\mathrm{reg}}) = \frac{1}{2n}\big\|\bar{Y}_{\mathrm{reg}}\big\|_2^2 = \frac{1}{n}.$$

Hence, both $\mathcal{G}^{(k)}_{\mathrm{full}}(\theta_{\mathrm{pre}}; \bar{\mathcal{T}}_{\mathrm{reg}})$ and the mini-batch gain $\mathcal{G}^{(k)}(\theta_{\mathrm{pre}}; \bar{\mathcal{T}}_{\mathrm{reg}})$ in Definition 2.1 are zero for all $k \geq 1$, proving part (ii) of Theorem 3.4.

It remains to prove the random-initialization claim. Let $\theta_0 = \theta_{\mathrm{rand}}$ and G_{rand} be as in Lemma 3.3, and set $\lambda = \frac{M}{2n^2}$ and $\eta = \frac{1}{\lambda} = \frac{2n^2}{M}$. By Lemma 3.1, since $q(\theta_0) = -\bar{Y}_{\mathrm{reg}}$, it holds that

$$f_{\theta_1}(\bar{X}) = \eta G_{\mathrm{rand}} \bar{Y}_{\mathrm{reg}}.$$

Furthermore, since $\bar{Y}_{\mathrm{reg}} \perp \mathbf{1}_n$, we get

$$\bar{G}\bar{Y}_{\mathrm{reg}} = \lambda \bar{Y}_{\mathrm{reg}}.$$

Therefore, we have

$$f_{\theta_1}(\bar{X}) - \bar{Y}_{\mathrm{reg}} = \eta(G_{\mathrm{rand}} - \bar{G})\bar{Y}_{\mathrm{reg}}.$$

Applying Lemma 3.3 with $\rho = M_{\mathrm{gap}}$, the stated width condition implies that, with probability at least $1 - \delta$,

$$\big\|G_{\mathrm{rand}} - \bar{G}\big\|_2 \leq \frac{\lambda}{\sqrt{M_{\mathrm{gap}}}}.$$

On this event, it holds that

$$\mathcal{L}\big(\theta_1; \bar{\mathcal{T}}_{\mathrm{reg}}\big) = \frac{1}{2n}\big\|f_{\theta_1}(\bar{X}) - \bar{Y}_{\mathrm{reg}}\big\|_2^2 \leq \frac{1}{2n}\frac{1}{M_{\mathrm{gap}}}\big\|\bar{Y}_{\mathrm{reg}}\big\|_2^2 = \frac{1}{M_{\mathrm{gap}}}\mathcal{L}\big(\theta_0; \bar{\mathcal{T}}_{\mathrm{reg}}\big).$$

As a result, we have

$$\mathcal{G}^{(1)}_{\mathrm{full}}(\theta_{\mathrm{rand}}; \bar{\mathcal{T}}_{\mathrm{reg}}) = \frac{\mathcal{L}(\theta_0; \bar{\mathcal{T}}_{\mathrm{reg}}) - \mathcal{L}(\theta_1; \bar{\mathcal{T}}_{\mathrm{reg}})}{\mathcal{L}(\theta_0; \bar{\mathcal{T}}_{\mathrm{reg}})} \geq 1 - \frac{1}{M_{\mathrm{gap}}}.$$

□

B.5 Proof of Theorem 3.5

Proof. For simplicity, we write $\bar{Y} = \bar{Y}_{\mathrm{cls}}$ and $\bar{S} = \bar{S}_{\mathrm{cls}} = 2\bar{Y} - \mathbf{1}_n$. By construction, it holds that $\bar{S}^\top \mathbf{1}_n = 0$ and $\big\|\bar{S}\big\|_2^2 = n$. Define $B_{\mathrm{cls}} \doteq I_n + \frac{1}{n}\big(\mathbf{1}_n\mathbf{1}_n^\top - \bar{S}\bar{S}^\top\big)$ and $\alpha \doteq \frac{1}{2}$. Equivalently, we have

$$(B_{\mathrm{cls}})_{i,j} = \begin{cases} 1, & i = j, \\ 0, & i \neq j \text{ and } \bar{y}_i = \bar{y}_j, \\ 2/n, & \bar{y}_i \neq \bar{y}_j. \end{cases}$$

Thus, B_{cls} has nonnegative entries. Choose $v_{\mathrm{pre}} = 0$ and choose W_{pre}, such that

$$\Phi_{\theta_{\mathrm{pre}}}(\bar{X}) = \alpha \begin{bmatrix} B_{\mathrm{cls}} & 0_{n \times (M-n)} \end{bmatrix}.$$

One concrete realization is to set, for $j \leq n$,

$$(W_{\mathrm{pre}})_{j,i} = \begin{cases} \alpha(B_{\mathrm{cls}})_{i,j}, & (B_{\mathrm{cls}})_{i,j} > 0, \\ -1, & (B_{\mathrm{cls}})_{i,j} = 0, \end{cases}$$

and to set $(W_{\mathrm{pre}})_{j,i} = -1$ for all $j > n$.

The eigenspaces of B_{cls} are

$$\begin{cases} B_{\mathrm{cls}}\bar{S} & = 0; \\ B_{\mathrm{cls}}\mathbf{1}_n & = 2\mathbf{1}_n; \\ B_{\mathrm{cls}}u & = u \end{cases}$$

for every $u \perp \operatorname{span}\{\mathbf{1}_n, \bar{S}\}$. Hence, the eigenvalues of B_{cls} are

$$2, \quad \underbrace{1, \ldots, 1}_{n-2 \text{ times}}, \quad 0.$$

By Lemma 3.2, the nonzero singular values of $\Phi_{\theta_{\text{pre}}}(\bar{X})$ are

$$2\alpha, \quad \underbrace{\alpha, \ldots, \alpha}_{n-2 \text{ times}}.$$

Therefore, for $1 \le k \le n-1$, the fraction of squared singular-value energy captured by the top k singular values is

$$\frac{(2\alpha)^2 + (k-1)\alpha^2}{(2\alpha)^2 + (n-2)\alpha^2} = \frac{k+3}{n+2}.$$

The smallest k for which this is at least 0.99 is $\lceil 0.99(n+2) - 3 \rceil$. For the effective rank, the normalized, nonzero singular values are

$$\frac{2}{n}, \quad \underbrace{\frac{1}{n}, \ldots, \frac{1}{n}}_{n-2 \text{ times}}.$$

Therefore, we have

$$\operatorname{EffRank}\big(\Phi_{\theta_{\text{pre}}}(\bar{X})\big) = \exp\left(-\frac{2}{n}\log\frac{2}{n} - (n-2)\frac{1}{n}\log\frac{1}{n}\right) = \frac{n}{2^{2/n}}.$$

Similarly, the nonzero singular values of $\widehat{K}(\bar{X}; \theta_{\text{pre}})$ are

$$4\alpha^2, \quad \underbrace{\alpha^2, \ldots, \alpha^2}_{n-2 \text{ times}}.$$

Thus, for $1 \le k \le n-1$, the fraction of squared singular-value energy captured by the top k eNTK singular values is

$$\frac{(4\alpha^2)^2 + (k-1)(\alpha^2)^2}{(4\alpha^2)^2 + (n-2)(\alpha^2)^2} = \frac{k+15}{n+14},$$

and hence

$$\operatorname{EnRank}_{0.99}\Big(\widehat{K}(\bar{X}; \theta_{\text{pre}})\Big) = \lceil 0.99(n+14) - 15 \rceil.$$

This proves part (i) of Theorem 3.5.

For the $0/1$-label logistic loss,

$$\frac{\partial}{\partial z}\ell_{\log}(z, y) = \frac{\exp(z)}{1+\exp(z)} - y.$$

At a zero-output checkpoint, this gives $q_i(\theta_{\text{pre}}) = \frac{1}{2} - \bar{y}_i = -\frac{1}{2}\bar{s}_i$, and $q(\theta_{\text{pre}}) = -\frac{1}{2}\bar{S}$. Since $B_{\text{cls}}\bar{S} = 0$, then we get

$$\Phi_{\theta_{\text{pre}}}(\bar{X})^\top q(\theta_{\text{pre}}) = -\frac{\alpha}{2}\begin{bmatrix} B_{\text{cls}} \\ 0 \end{bmatrix}\bar{S} = 0.$$

Lemma 3.1 implies that full-support gradient descent leaves θ_{pre} unchanged for every learning rate. Since $f_{\theta_{\text{pre}}}(\bar{x}_i) = 0$ for all i, we get

$$\mathcal{L}(\theta_{\text{pre}}; \bar{\mathcal{T}}_{\text{cls}}) = \frac{1}{n}\sum_{i=1}^{n}\log 2 = \log 2.$$

Consequently, we have

$$\mathcal{G}^{(k)}_{\text{full}}(\theta_{\text{pre}}; \bar{\mathcal{T}}_{\text{cls}}) = 0$$

for all $k \ge 1$ proving part (ii) of Theorem 3.5. Unlike in the regression construction, the individual-example gradients here are nonzero and cancel in the full-support average The full-support result alone does not establish zero mini-batch gain.

It remains to prove the random-initialization claim. Let $\theta_0 = \theta_{\mathrm{rand}}$ and G_{rand} be as in Lemma 3.3. Set $\lambda = \frac{M}{2n^2}$. At θ_0, the prediction is zero and $q(\theta_0) = -\frac{1}{2}\bar{S}$. By Lemma 3.1, we have

$$f_{\theta_1}(\bar{X}) = \frac{\eta}{2} G_{\mathrm{rand}} \bar{S}.$$

With

$$\eta = \frac{2\gamma_{M_{\mathrm{gap}}}}{\lambda} = \frac{4\gamma_{M_{\mathrm{gap}}} n^2}{M},$$

and using

$$\bar{G}\bar{S} = \lambda \bar{S}$$

by $\bar{S} \perp \mathbf{1}_n$, we obtain

$$f_{\theta_1}(\bar{X}) - \gamma_{M_{\mathrm{gap}}} \bar{S} = \frac{\eta}{2}(G_{\mathrm{rand}} - \bar{G})\bar{S}.$$

Next, we apply Lemma 3.3 with

$$\rho = \left(\frac{2\gamma_{M_{\mathrm{gap}}} M_{\mathrm{gap}}}{\log 2} \right)^2.$$

The stated width condition implies $M \geq C n^2 \rho \log(2n/\delta)$, after absorbing the numerical factor $(2/\log 2)^2$ into the universal constant C. Hence, with probability at least $1 - \delta$, it holds that

$$\frac{\left\| G_{\mathrm{rand}} - \bar{G} \right\|_2}{\lambda} \leq \frac{\log 2}{2\gamma_{M_{\mathrm{gap}}} M_{\mathrm{gap}}}.$$

On this event, we get

$$\left\| f_{\theta_1}(\bar{X}) - \gamma_{M_{\mathrm{gap}}} \bar{S} \right\|_2 \leq \gamma_{M_{\mathrm{gap}}} \frac{\left\| G_{\mathrm{rand}} - \bar{G} \right\|_2}{\lambda} \left\| \bar{S} \right\|_2.$$

Since $\left\| \bar{S} \right\|_2 = \sqrt{n}$, we have

$$\frac{1}{n} \sum_{i=1}^{n} \left| \bar{s}_i f_{\theta_1}(\bar{x}_i) - \gamma_{M_{\mathrm{gap}}} \right| = \frac{1}{n} \left\| f_{\theta_1}(\bar{X}) - \gamma_{M_{\mathrm{gap}}} \bar{S} \right\|_1 \leq \gamma_{M_{\mathrm{gap}}} \frac{\left\| G_{\mathrm{rand}} - \bar{G} \right\|_2}{\lambda} \leq \frac{\log 2}{2M_{\mathrm{gap}}}.$$

Let $\varphi(u) \doteq \log(1 + \exp(-u))$. Since $|\varphi'(u)| \leq 1$, it holds that φ is 1-Lipschitz. Also, for $y \in \{0, 1\}$ and $s = 2y - 1$, we have

$$\ell_{\log}(z, y) = \varphi(sz).$$

Therefore,

$$\mathcal{L}(\theta_1; \bar{\mathcal{T}}_{\mathrm{cls}}) = \frac{1}{n} \sum_{i=1}^{n} \varphi(\bar{s}_i f_{\theta_1}(\bar{x}_i)) \leq \varphi(\gamma_{M_{\mathrm{gap}}}) + \frac{\log 2}{2M_{\mathrm{gap}}}.$$

By the definition

$$\gamma_{M_{\mathrm{gap}}} = \log\left(\frac{2M_{\mathrm{gap}}}{\log 2} \right),$$

we have

$$\varphi(\gamma_{M_{\mathrm{gap}}}) = \log\left(1 + \exp\left(-\gamma_{M_{\mathrm{gap}}}\right)\right) \leq \exp\left(-\gamma_{M_{\mathrm{gap}}}\right) = \frac{\log 2}{2M_{\mathrm{gap}}}.$$

Thus, we have

$$\mathcal{L}(\theta_1; \bar{\mathcal{T}}_{\mathrm{cls}}) \leq \frac{\log 2}{M_{\mathrm{gap}}}.$$

Since $\mathcal{L}(\theta_0; \bar{\mathcal{T}}_{\mathrm{cls}}) = \log 2$, it follows that

$$\mathcal{G}^{(1)}_{\mathrm{full}}(\theta_{\mathrm{rand}}; \bar{\mathcal{T}}_{\mathrm{cls}}) = \frac{\mathcal{L}(\theta_0; \bar{\mathcal{T}}_{\mathrm{cls}}) - \mathcal{L}(\theta_1; \bar{\mathcal{T}}_{\mathrm{cls}})}{\mathcal{L}(\theta_0; \bar{\mathcal{T}}_{\mathrm{cls}})} \geq 1 - \frac{1}{M_{\mathrm{gap}}}.$$

□

B.6 Proof of Theorem 4.6

Proof. We have by Assumption 4.2

$$\mathcal{L}(\theta_1;\mathcal{T}) \le \mathcal{L}(\theta;\mathcal{T}) - \eta\,\langle g(\theta;\mathcal{T}),\, \widehat{g}_{\mathcal{B}_0}(\theta;\mathcal{T})\rangle + \frac{\beta\eta^2}{2}\|\widehat{g}_{\mathcal{B}_0}(\theta;\mathcal{T})\|^2.$$

Then, by Assumption 4.1 and Theorem 2.1.5 of Nesterov [2013], we get

$$\begin{aligned}
\mathbb{E}_{\mathcal{B}_0}[\mathcal{L}(\theta_1;\mathcal{T})] &\le \mathcal{L}(\theta;\mathcal{T}) - \eta\|g(\theta;\mathcal{T})\|^2 + \frac{\beta\eta^2}{2}\mathbb{E}_{\mathcal{B}_0}\left[\|\widehat{g}_{\mathcal{B}_0}(\theta;\mathcal{T})\|^2\right] \\
\mathcal{L}(\theta;\mathcal{T}) - \mathbb{E}_{\mathcal{B}_0}[\mathcal{L}(\theta_1;\mathcal{T})] &\ge \eta\|g(\theta;\mathcal{T})\|^2 - \frac{\beta\eta^2}{2}\mathbb{E}_{\mathcal{B}_0}\left[\|\widehat{g}_{\mathcal{B}_0}(\theta;\mathcal{T})\|^2\right] \\
&\ge \eta\|g(\theta;\mathcal{T})\|^2 - \beta\eta^2\mathbb{E}_{\mathcal{B}_0}\left[\|\widehat{g}_{\mathcal{B}_0}(\theta;\mathcal{T})\|^2\right] \\
&= \eta\|g(\theta;\mathcal{T})\|^2 - \beta\eta^2\frac{\|g(\theta;\mathcal{T})\|^2}{R(\theta;\mathcal{T})} \\
&= \eta\|g(\theta;\mathcal{T})\|^2\left(1 - \frac{\beta\eta}{R(\theta;\mathcal{T})}\right).
\end{aligned}$$

Therefore, we further have

$$\begin{aligned}
\frac{\mathcal{L}(\theta;\mathcal{T}) - \mathbb{E}_{\mathcal{B}_0}[\mathcal{L}(\theta_1;\mathcal{T})]}{\mathcal{L}(\theta;\mathcal{T})} &\ge \eta\frac{\|g(\theta;\mathcal{T})\|^2}{\mathcal{L}(\theta;\mathcal{T})}\left(1 - \frac{\beta\eta}{R(\theta;\mathcal{T})}\right) \\
\mathcal{G}^{(1)}(\theta;\mathcal{T}) &\ge \eta S(\theta;\mathcal{T})\left(1 - \frac{\beta\eta}{R(\theta;\mathcal{T})}\right).
\end{aligned}$$

Let $\eta = \alpha R(\theta;\mathcal{T})/\beta$. Since $0 < \eta < R(\theta;\mathcal{T})/\beta$, we then have $0 < \alpha < 1$. Plugging in η, we have

$$\begin{aligned}
\mathcal{G}^{(1)}(\theta;\mathcal{T}) &\ge \frac{\alpha(1-\alpha)}{\beta}S(\theta;\mathcal{T})R(\theta;\mathcal{T}) \\
&= \frac{\alpha(1-\alpha)}{\beta}\mathrm{OR}(\theta;\mathcal{T}).
\end{aligned}$$

□

C Additional Experiment Details

C.1 Estimation Details

k-Step Gain Estimation. Let $\mathcal{T}_{\mathrm{val}} \doteq (\mathcal{X}, \mathcal{Y}, P^{\mathrm{val}}_{\mathcal{X}\mathcal{Y}})$ be a validation task and let θ be the checkpoint whose trainability we wish to estimate. We denote the validation set by

$$\mathcal{D}_{\mathrm{val}} \doteq \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{val}}}, \qquad (x_i, y_i) \sim P^{\mathrm{val}}_{\mathcal{X}\mathcal{Y}}.$$

We estimate the population loss $\mathcal{L}(\theta;\mathcal{T}_{\mathrm{val}})$ using the empirical validation loss

$$\widehat{\mathcal{L}}_{\mathcal{D}_{\mathrm{val}}}(\theta;\mathcal{T}_{\mathrm{val}}) \doteq \frac{1}{N_{\mathrm{val}}}\sum_{(x_i,y_i)\in\mathcal{D}_{\mathrm{val}}} \ell(f_\theta(x_i), y_i).$$

In all experiments, we use $N_{\mathrm{val}} = 10{,}000$.

Starting from $\theta_0 = \theta$, we run stochastic gradient descent for k steps,

$$\theta_{s+1} = \theta_s - \eta\widehat{g}_{\mathcal{B}_s}(\theta_s;\mathcal{T}_{\mathrm{val}}), \qquad s = 0, \dots, k-1,$$

where each mini-batch $\mathcal{B}_s$ consists of $m = 4$ samples drawn uniformly with replacement from $\mathcal{D}_{\mathrm{val}}$. We use $\eta = 10^{-3}$ for the gain-estimation rollouts. For each checkpoint and validation task, we repeat this procedure for $\mathcal{R} = 128$ independent rollouts and obtain terminal parameters $\left\{\theta_k^{(i)}\right\}_{i=1}^{\mathcal{R}}$. The expected post-update empirical loss is estimated as

$$\widehat{\mathcal{L}}_{\mathcal{R}}(\theta_k;\mathcal{T}_{\mathrm{val}}) \doteq \frac{1}{\mathcal{R}}\sum_{i=1}^{\mathcal{R}}\widehat{\mathcal{L}}_{\mathcal{D}_{\mathrm{val}}}\left(\theta_k^{(i)};\mathcal{T}_{\mathrm{val}}\right).$$

We then estimate the k-step gain by

$$\widehat{\mathcal{G}}^{(k)}(\theta;\mathcal{T}_{\mathrm{val}}) \doteq \frac{\widehat{\mathcal{L}}_{\mathcal{D}_{\mathrm{val}}}(\theta;\mathcal{T}_{\mathrm{val}}) - \widehat{\mathcal{L}}_{\mathcal{R}}(\theta_k;\mathcal{T}_{\mathrm{val}})}{\widehat{\mathcal{L}}_{\mathcal{D}_{\mathrm{val}}}(\theta;\mathcal{T}_{\mathrm{val}})}.$$

Optimization Readiness Estimation. We estimate the population gradient $g(\theta;\mathcal{T}_{\mathrm{val}})$ using the gradient of the full validation set

$$\widehat{g}(\theta;\mathcal{T}_{\mathrm{val}}) \doteq \nabla_\theta \widehat{\mathcal{L}}_{\mathcal{D}_{\mathrm{val}}}(\theta;\mathcal{T}_{\mathrm{val}}).$$

The empirical gradient strength is

$$\widehat{S}(\theta;\mathcal{T}_{\mathrm{val}}) \doteq \frac{\|\widehat{g}(\theta;\mathcal{T}_{\mathrm{val}})\|_2^2}{\widehat{\mathcal{L}}_{\mathcal{D}_{\mathrm{val}}}(\theta;\mathcal{T}_{\mathrm{val}})}.$$

To estimate gradient reliability, we independently sample $\mathcal{R} = 128$ mini-batches $\{\mathcal{B}_i\}_{i=1}^{\mathcal{R}}$ of size $m = 4$ from $\mathcal{D}_{\mathrm{val}}$ with replacement and compute

$$\widehat{R}(\theta;\mathcal{T}_{\mathrm{val}}) \doteq \frac{\|\widehat{g}(\theta;\mathcal{T}_{\mathrm{val}})\|_2^2}{\frac{1}{\mathcal{R}}\sum_{i=1}^{\mathcal{R}} \|\widehat{g}_{\mathcal{B}_i}(\theta;\mathcal{T}_{\mathrm{val}})\|_2^2}.$$

The estimated optimization readiness is then

$$\widehat{\mathrm{OR}}(\theta;\mathcal{T}_{\mathrm{val}}) \doteq \widehat{S}(\theta;\mathcal{T}_{\mathrm{val}})\widehat{R}(\theta;\mathcal{T}_{\mathrm{val}}).$$

Baseline Metric Estimation. Let X_{val} denote the matrix of validation inputs, where the i-th row is $x_i^\top$. For both SCR and P-MNIST, we compute the representation effective rank and 99% energy rank as

$$\mathrm{EffRank}(\Phi_\theta(X_{\mathrm{val}})), \qquad \mathrm{EnRank}_{0.99}(\Phi_\theta(X_{\mathrm{val}})).$$

We also compute the active-neuron fraction on the full validation set using threshold $\tau_{\mathrm{act}} = 0.1$.

For SCR, the network has a manageable number of parameters, so we compute the eNTK rank and exact Hessian rank directly:

$$\mathrm{EnRank}_{0.99}\Big(\widehat{K}(X_{\mathrm{val}};\theta)\Big), \qquad \mathrm{EnRank}_{0.99}\Big(\nabla_\theta^2 \widehat{\mathcal{L}}_{\mathcal{D}_{\mathrm{val}}}(\theta;\mathcal{T}_{\mathrm{val}})\Big).$$

For P-MNIST, we omit the eNTK rank because $f_\theta(x)$ is vector-valued, so $\nabla_\theta f_\theta(x)$ is a Jacobian rather than a vector and the resulting empirical kernel is not directly comparable to the scalar-output case.

For the Hessian-rank metric in P-MNIST, we follow the Gram-matrix approximation used by Lewandowski et al. [2024]. Specifically, we sample $b = 128$ input-label pairs from $\mathcal{D}_{\mathrm{val}}$ and form

$$G \doteq [\nabla_\theta \ell(f_\theta(x_1), y_1) \quad \cdots \quad \nabla_\theta \ell(f_\theta(x_b), y_b)].$$

We then use

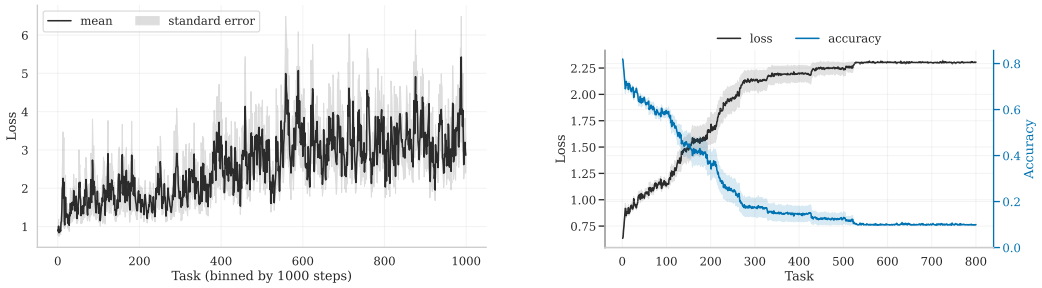
$$\mathrm{EnRank}_{0.99}(G^\top G)$$

as the approximated 99% energy rank of the Hessian.

C.2 Compute Resources

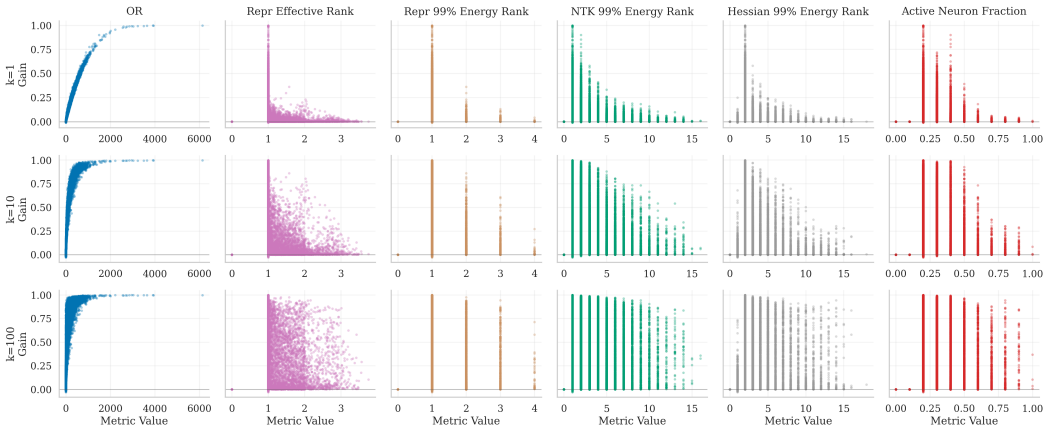
Experiments were run on a shared Slurm cluster. Since jobs were scheduled opportunistically, the exact hardware allocation varied across runs. SCR experiments were faster on CPU and took less than 3 hours per seed. P-MNIST experiments benefited from GPU training and took approximately 10.5 hours per seed on an NVIDIA RTX 4000 GPU with 8GB of GPU memory.

C.3 Figures

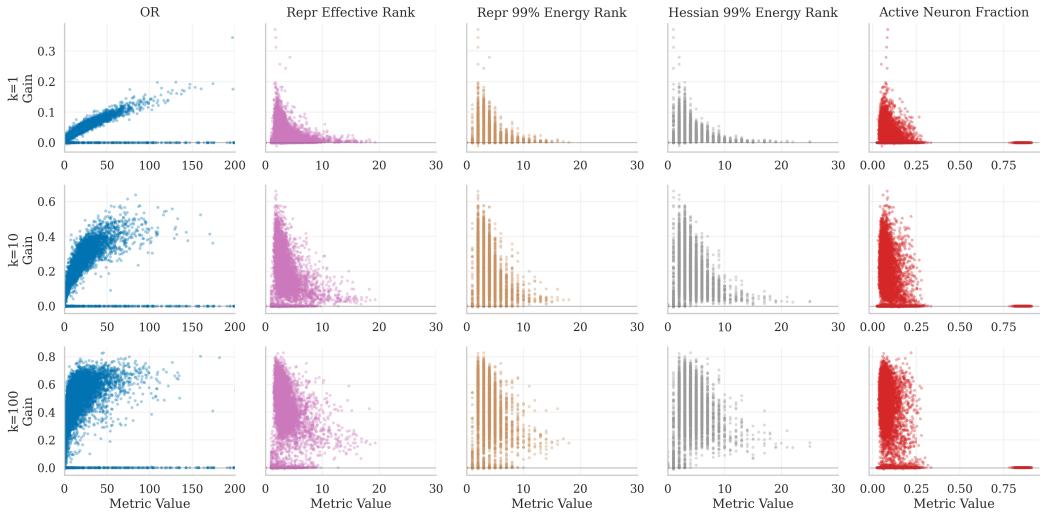


(a) Slowly-Changing Regression (b) Permuted MNIST

Figure 1: Learning curves of Slowly-Changing Regression and Permuted MNIST.

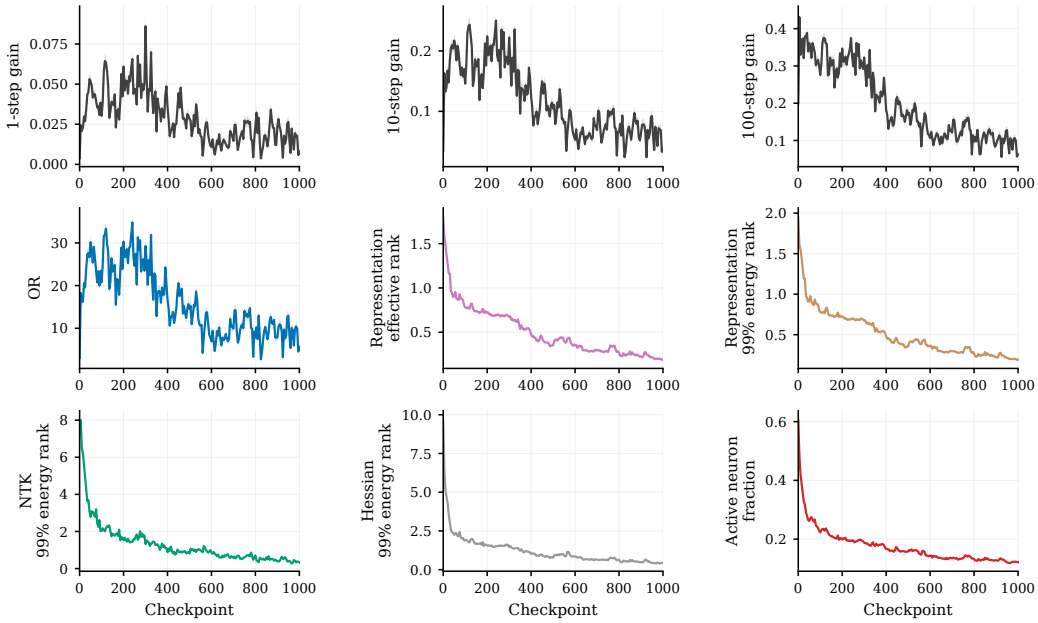


(a) Slowly-Changing Regression. Results aggregated from 20 training runs and 30 validation tasks.

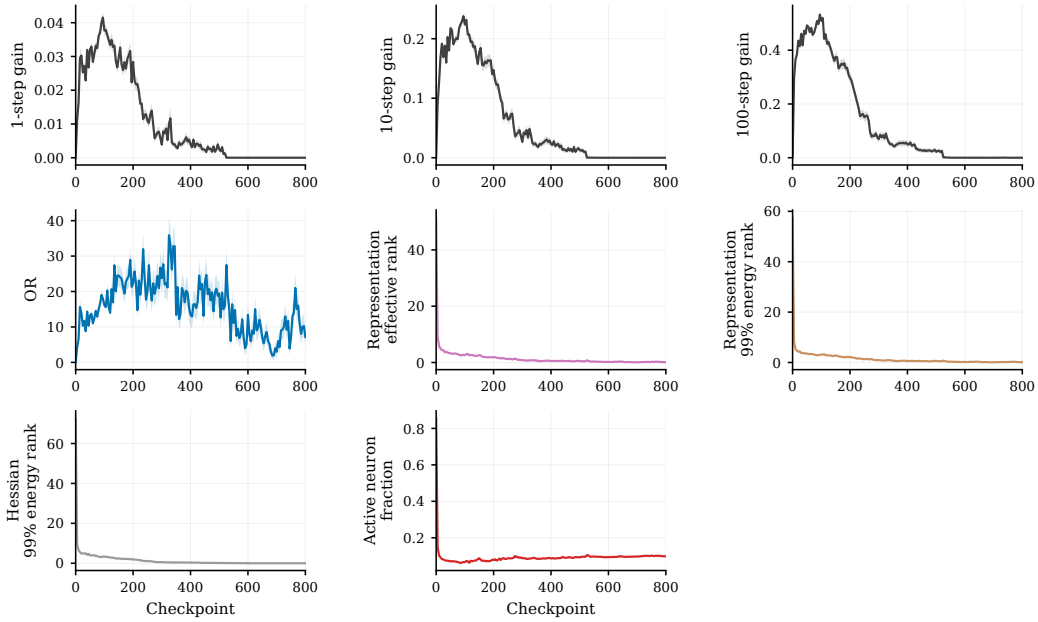


(b) Permuted MNIST. Results aggregated from 20 training runs and 10 validation tasks.

Figure 2: 1-, 10-, and 100-step gain vs. plasticity metric values for Slowly-Changing Regression and Permuted MNIST (extreme metric values clipped to account for outliers).

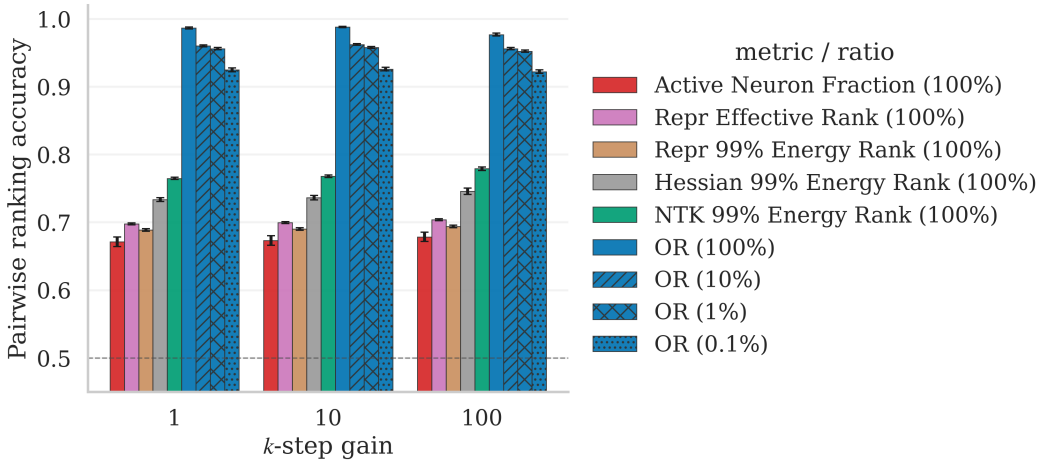


(a) Slowly-Changing Regression. Results aggregated from 20 training runs and 30 validation tasks.

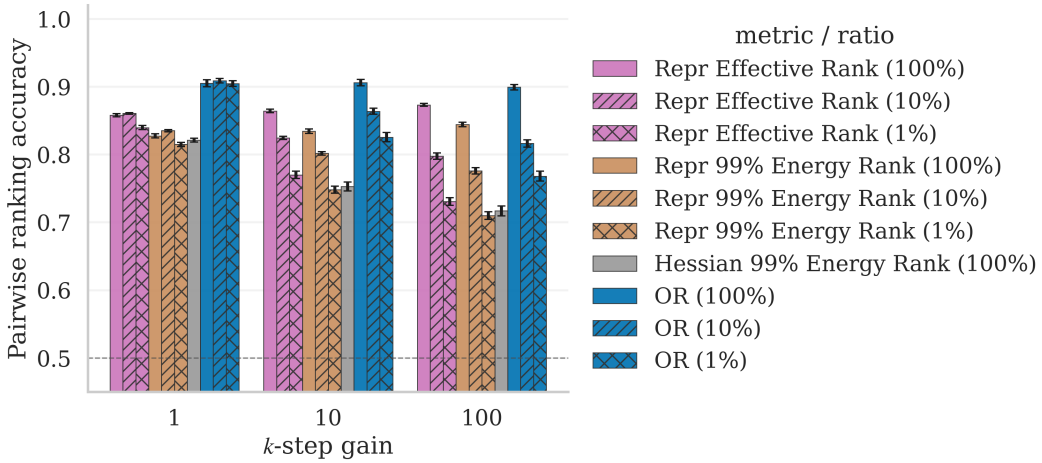


(b) Permuted MNIST. Results aggregated from 20 training runs and 10 validation tasks.

Figure 3: 1-, 10-, and 100-step gain and plasticity metric values against checkpoints for Slowly-Changing Regression and Permuted MNIST (extreme metric values clipped to account for outliers). Shaded areas denote standard errors.



(a) Slowly-Changing Regression. Results aggregated from 20 training runs and 30 validation tasks.



(b) Permuted MNIST. Results aggregated from 20 training runs and 10 validation tasks. We omit the active neuron fraction because its accuracy is all under 50%.

Figure 4: Subsampling ablation study results for Slowly-Changing Regression and Permuted MNIST. The error bars denote standard errors.

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: The abstract and introduction accurately summarize the main contributions, including defining k-step gain as an operational proxy for trainability, theoretically analyzing failure modes of existing diagnostics, proposing optimization readiness, and empirically evaluating predictive performance. The claims are scoped to the theoretical settings and supervised continual learning experiments studied in the paper.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 discussion and conclusion section describes limitations of the theoretical and empirical scope, including the use of ReLU networks with one hidden layer in the counterexamples and evaluation on controlled supervised continual learning testbeds. It also discusses future directions such as deeper networks, other activation functions, reinforcement learning, and larger-scale settings.

   Guidelines:

   - The answer [N/A] 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: The assumptions for the theoretical counterexamples and the optimization readiness guarantee are stated in the corresponding theorem and lemma statements. Complete proofs are provided in the appendix.

   Guidelines:

   - The answer [N/A] 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: The paper describes the continual learning benchmarks, model architectures, data generation procedures, checkpoint collection, metric estimation procedures, optimizer settings, and evaluation protocol. Additional estimator and implementation details are provided in the appendix.

   Guidelines:

   - The answer [N/A] 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: We release our code at https://github.com/Sequential-Intelligence-Lab/Predicting_Plasticity.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The experiments section and appendix describe the data generation procedures, validation tasks, model architectures, optimizer, learning rates, mini-batch sizes, number of checkpoints, Monte Carlo repetitions, and metric estimation details.

   Guidelines:

   - The answer [N/A] 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 paper reports variability across independent runs and validation tasks in the quantitative results. The tables report mean performance with uncertainty estimates, and the experimental section specifies the sources of randomness used to compute them.

Guidelines:

- The answer [N/A] 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 appendix reports that experiments were run on a shared Slurm cluster with opportunistic hardware allocation, along with approximate runtime per seed for SCR and P-MNIST and the representative GPU used for P-MNIST.

   Guidelines:

   - The answer [N/A] 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: The work is foundational research on diagnostics for trainability in continual learning and uses standard or synthetically generated datasets. There does not appear to be any aspect of the research that violates the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: The work is foundational research on optimization diagnostics for neural networks and is not tied to a specific deployment domain or application. There are no direct societal impacts beyond the general implications of improving machine learning training reliability.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, generative models, or other assets with substantial misuse potential.

    Guidelines:

    - The answer [N/A] 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: The paper uses standard datasets and synthetic data generation procedures and cites the corresponding sources where applicable. No restricted or proprietary assets are used.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not introduce or release a new dataset, benchmark, pretrained model, or standalone software package as part of the submission. The experiments use standard datasets and synthetic data generation procedures described in the paper.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology, theory, and experiments do not use LLMs as an important, original, or non-standard component. Any LLM use was limited to writing/editing assistance, basic code assistance, and non-authoritative checking of derivations. All proofs, code, experiments, claims, figures, and references were independently verified and finalized by the authors.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.