DO NEURAL NETWORKS LOSE PLASTICITY IN A GRADUALLY CHANGING WORLD?

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning has become a trending topic in machine learning. Recent studies have discovered an interesting phenomenon called loss of plasticity, referring to neural networks gradually losing the ability to learn new tasks. However, existing plasticity research largely relies on benchmarks with abrupt task transitions, without examining whether the abruptness itself contributes to the observed plasticity loss. In this paper, we investigate the role of transition abruptness by simulating gradually changing environments through input/output interpolation and task sampling. We perform theoretical and empirical analysis, showing that the severity of plasticity loss is closely tied to the abruptness of task transitions, and can be substantially reduced when the environment changes gradually.

1 INTRODUCTION

Continual learning has become a trending topic in machine learning research, where a system incrementally learns from non-stationary distributions (Ring, 1997; Ruvolo & Eaton, 2013; Shin et al., 2017; Kirkpatrick et al., 2017; Abbas et al., 2023). In supervised learning, a system is trained to perform isolated tasks (Thrun, 1998). However, training a system from scratch is onerous and time-consuming, especially for large neural networks (Dean et al., 2012). It would be more computationally and economically efficient to incrementally adapt them to new tasks.

Loss of plasticity is an intriguing phenomenon recently discovered in continual learning, referring to neural networks gradually losing the ability to learn new tasks (Dohare et al., 2021; 2024; Lyle et al., 2023; Zilly et al., 2021; Abbas et al., 2023; Lewandowski et al., 2025a). Proposed mitigation techniques include regularizing the model weights (Zilly et al., 2021; Lewandowski et al., 2024; 2025a; Kumar et al., 2023), using alternative activation functions (Berariu et al., 2021; Lee et al., 2023; Abbas et al., 2023), controlling the loss landscape sharpness (Lyle et al., 2023), resetting less-used neurons (Dohare et al., 2024; Sokar et al., 2023), and selectively forgetting memorized noise (Shin et al., 2024).

A common thread across these studies is the use of benchmarks with abrupt task transitions; e.g., random label assignment, random pixel permutation (Dohare et al., 2024; Lyle et al., 2023; Lewandowski et al., 2024; 2025a). These benchmarks have shaped our understanding of plasticity loss, but they mostly assume abrupt task transitions. Such a design choice has not been carefully examined. This leaves open an important question of how the abruptness of task transitions affects plasticity.

This distinction matters because real-world non-stationarity is rarely abrupt. Consider the evolution of human language. Researchers show that the sense of a word may change from time to time, but this happens gradually (Kilgarriff, 1997; Mitra et al., 2015; Laurino et al., 2023). For instance, the word "awful" originally meant "full of awe", but gradually underwent pejoration to its modern sense of "terrible or appalling" (Hamilton et al., 2016). This shift did not happen overnight; rather, both senses coexisted over an extended period before the newer meaning became dominant.

In this paper, we investigate the hypothesis that gradually changing environment preserves plasticity. We propose to perform input/output interpolation or task sampling to simulate a gradually changing environment from abrupt non-stationarities. We vary the degree of transition abruptness while keeping all other aspects of the benchmark fixed. Our empirical results across four benchmarks covering both trainability and generalizability (which are two aspects of plasticity) reveal that plasticity loss is highly sensitive to transition abruptness and that neural networks preserve plasticity magnitudes

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

longer than those under abrupt task changes. In addition, we provide theoretical analysis based on the intuition that abrupt task changes cause abrupt shifts in the loss landscape, trapping parameters in poor local optima. In contrast, gradual transitions allow the landscape to evolve smoothly, guiding parameters toward better solutions for each new task.

Our findings have two practical implications. First, they suggest that future plasticity benchmarks should explicitly control for transition abruptness, as current benchmarks may overemphasize the severity of plasticity loss relative to realistic conditions. Second, for scenarios where task transitions are genuinely abrupt (e.g., robots transferred to new environments), our interpolation and task sampling protocols offer simple yet effective techniques that effectively mitigate plasticity loss.

2 Problem Formulation

The problem of loss of plasticity has recently been observed in continual learning when the data distribution in each sequentially coming task is changed rapidly and non-stationarily. In the continual learning literature, the concept of loss of plasticity can refer to either reduced ability to learn new from new data (loss of trainability) (Dohare et al., 2021; Lyle et al., 2022; Elsayed & Mahmood, 2024; Ma et al., 2024), or reduced ability to generalize to unseen data (loss of generalizability) (Dohare et al., 2024; Ash & Adams, 2020; Lee et al., 2024; Zilly et al., 2021). Our work investigates the loss of plasticity problem from the perspective of both trainability and generalizability.

Let $\mathcal{D} = \{(\boldsymbol{x}_m, y_m)\}_{m=1}^{M}$ be a training dataset where $\boldsymbol{x}_m \in \mathbb{R}^d$ is an input sample (e.g., an image with d pixels) and $y_m \in \mathbb{Y}$ be the corresponding label. We denote a deep neural network by a function f_θ. The learning algorithm optimizes the parameters by minimizing the loss of each task, given by $\text{minimize}_\theta \, \mathbb{E}_{(x,y)\sim p_t}[\ell(f_\theta(\boldsymbol{x}), y)]$. There are different settings in continual learning regarding the availability of samples in previous tasks. We adopt the common setting in the loss of plasticity research, where we assume the machine learning model is able to iterate over a task's samples multiple times in a mini-batch fashion, although the task boundary is not informed (Li et al., 2014; Lewandowski et al., 2024; Lyle et al., 2022).

To illustrate this process, we introduce two popular examples in the previous literature that simulate a continually changing environment.

Random Image Labeling. In this environment, the label of an image is randomly selected from $0, \cdots, c$ where c is the number of classes in the dataset. In other words, the dataset of the tth task is $\mathcal{D}^{(t)} = \{(\boldsymbol{x}_m, y_m^{(t)})\}_{m=1}^{M}$, where $\boldsymbol{x}_m$ comes from the original image dataset, and $y_m^{(t)} \in \{0, \cdots, c\}$ is randomly sampled. Notice that the label remains fixed within a given task, but is reassigned for each new task.

Random Pixel Permuting. This environment permutes the image pixels of an image data sample. For each task, we sample a random permutation $\pi^{(t)} : \{1, \cdots, d\} \to \{1, \cdots, d\}$, where d is the feature dimension (i.e., the number of pixels). The permutation $\pi^{(t)}$ is applied to all images for the tth task, which yields the dataset $\mathcal{D}^{(t)} = \{(\boldsymbol{x}_m^{(t)}, y_m)\}_{m=1}^{M}$ for $\boldsymbol{x}_{m,i}^{(t)} = \boldsymbol{x}_{m,\pi^{(t)}(i)}$.

To further investigate loss of plasticity in the language models, we propose two text generation environments.

Random Seq2Seq. This environment utilizes synthetic text to evaluate the model's ability to learn arbitrary sequence mappings. The dataset for the tth task is denoted as $\mathcal{D}^{(t)} = \{(\boldsymbol{x}_m^{(t)}, \boldsymbol{y}_m^{(t)})\}_{m=1}^{M}$. Both the input $\boldsymbol{x}_m^{(t)}$ (of sequence length p) and the target $\boldsymbol{y}_m^{(t)}$ (of sequence length q) consist of synthetic "words," where each word is a random character string of fixed length L sampled from a standard alphabet.

Bigram Cipher. To evaluate generalization plasticity, we must include certain learnable patterns between input and output, as opposed to the Random Seq2Seq task. To this end, we propose the Bigram Cipher environment with a dataset $\mathcal{D}^{(t)} = \{(\boldsymbol{x}_m^{(t)}, \boldsymbol{y}_m^{(t)})\}_{m=1}^{M}$. For each task t, we sample a random permutation $\pi^{(t)} : \mathcal{V} \to \{0, \cdots, |\mathcal{V}| - 1\}$ over the vocabulary $\mathcal{V}$. The output tokens $\boldsymbol{y}_m^{(t)}$ is determined by the modular sum of the permuted values of the current ($\boldsymbol{x}_{m,i}^{(t)}$) and previous ($\boldsymbol{x}_{m,i}^{(t-1)}$) input tokens, where i is the ith non-whitespace letter of $\boldsymbol{x}_m$. Given an input sequence

of tokens $\boldsymbol{x}_m^{(t)} = (\boldsymbol{x}_i)_{i=1}^{Lp}$, the target sequence of tokens $y_m^{(t)}$ is determined by the sequence of modular sum of the permuted values of the current character $\boldsymbol{x}_{m,i}^{(t)}$ and the previous character $\boldsymbol{x}_{m,i}^{(t-1)}$: $\pi^{(t)^{-1}}(\boldsymbol{y}_{m,i}^{(t)}) = (\pi^{(t)}(\boldsymbol{x}_{m,i}^{(t)}) + \pi^{(t)}(\boldsymbol{x}_{m,(i-1)}^{(t)}))\ (\mathrm{mod}\ |\mathcal{V}|)$.

3 A Gradually Changing Environment

While prior work has provided fundamental insights into the phenomenon of plasticity loss, it has primarily focused on settings with abrupt task transitions. However, natural environments and human cognition often evolve in a more gradual, continuous manner over time (Mayr, 1970; Sterelny, 2011). Consequently, the discrete task boundaries used in existing benchmarks may not fully capture the dynamics of these incremental changes, leaving it an open question whether plasticity is similarly compromised in a gradually changing world.

In our work, we develop several methods to simulate a gradually changing environment. Consider the tth task represented by the dataset $\mathcal{D}^{(t)}$ and its subsequent task with dataset $\mathcal{D}^{(t+1)}$. The simulation of a gradually changing environment can be accomplished by different ways depending on the task nature.

Output Interpolation. For the Random Image Labeling environments (Dohare et al., 2021; Lewandowski et al., 2024; Lyle et al., 2022), the inputs are the same for $\mathcal{D}^{(t)}$ and $\mathcal{D}^{(t+1)}$, but the outputs become new random labels. We perform label smoothing (Müller et al., 2019) for task interpolation. Let $\alpha_t \in [0, 1]$ be an interpolation coefficient that starts from 0 and increases uniformly by a constant step size s. We have $\boldsymbol{y}_m^{(\alpha_t)} = (1 - 2\alpha_t)\boldsymbol{y}_m^{(t)} + 2\alpha_t \boldsymbol{u}$ for $\alpha_t \in [0, \frac{1}{2}]$, where $\boldsymbol{y}_m^{(t)}$ is a one-hot vector for target $\boldsymbol{y}_m^{(t)}$ and $\boldsymbol{u}$ is a uniform distribution over the discrete target categories; for $\alpha_t \in [\frac{1}{2}, 1]$, we have $\boldsymbol{y}_m^{(\alpha_t)} = (2\alpha_t - 1)\boldsymbol{y}_m^{(t+1)} + (2 - 2\alpha_t)\boldsymbol{u}$ and $\alpha_{t+1} = \alpha_t + s$. In other words, we first anneal $\mathcal{D}^{(t)}$ to a uniform distribution and then anneal it to $\mathcal{D}^{(t+1)}$, offering a smooth transition from the tth task to the $(t + 1)$th task.

Input Interpolation. For the Random Image Permuting environment, the input pixel locations are shuffled in a new task, while the output remains the same. In this setting, we propose to gradually interpolate the input (i.e., pixel values) by direct averaging, i.e., $\boldsymbol{x}_m^{(\alpha)} = (1 - \alpha)\boldsymbol{x}_m^{(t)} + \alpha \boldsymbol{x}_m^{(t+1)}$, where $\boldsymbol{x}_m^{(t)}$ and $\boldsymbol{x}_m^{(t+1)}$ denote the m-th sample under the previous and new permutations, respectively. As before, α is an interpolation coefficient governed by the step size s, enabling a smooth transition from the previous input structure to the new one.

It is important to note that input and output interpolations depend on the nature of the environment. In particular, they require the mth samples in $\mathcal{D}^{(t)}$ and $\mathcal{D}^{(t+1)}$ to have certain correspondence. This requirement can be reflected in certain real-world scenarios, such as changes in word meanings, where the input word is fixed while its associated meaning changes. However, such a correspondence may be unrealistic for other contexts, such as robots transitioning to a new environment, where the relationship between input and output is less predicable. To address this, we propose a general task interpolation method by annealed sampling.

Task Sampling. For more general task transitions, we sample a subset $\mathcal{D}^{(t)'}$ of size $\lceil (1 - \alpha)M \rceil$ from $\mathcal{D}^{(t)}$ and a subset $\mathcal{D}^{(t+1)'}$ of size $\lfloor \alpha M \rfloor$ from $\mathcal{D}^{(t+1)}$ to construct a new intermediate dataset $\mathcal{D}^{\alpha} = \mathcal{D}^{(t)'} \cup \mathcal{D}^{(t+1)'}$. Similar to input/output interpolation, $\alpha \in [0, 1]$ is a coefficient that starts by 0 and increments by a certain step size.

Theoretical Analysis. To motivate our formal analysis, consider the geometric effect of environmental shifts on the loss landscape. Intuitively, an abrupt task change causes abrupt change of the loss landscape, and thus the parameters may be trapped in a poor local minimum in the new landscape. On the other hand, if the task changes gradually, the loss landscape also evolves gradually. With proper gradient descent, the parameters may remain in the catchment basin corresponding to the original local minimum (which is optimized for the first task).

To model a gradually changing environment, we decompose an abrupt task shift into a sequence of small, near-identity linear transformations $\boldsymbol{W}_t$ (micro-tasks). We start by making two standard assumptions from optimization theory: smoothness and local strong convexity. Smoothness (Def. 3.1)

controls how quickly the gradient can change, preventing the landscape from exhibiting arbitrarily sharp curvature. Local strong convexity (Def. 3.2) ensures that each local minimum sits inside a basin with sufficient curvature, so that gradient descent can reliably converge to it. These two properties allow us to track how the basin geometry evolves for each small interpolation step on the loss function. To show that GD converges to a specific minimizer, namely the one corresponding to the minimizer of the previous task, we require a local condition around it; local strong convexity is a standard choice. We assume it only near minimizers and allow the loss to be non-convex elsewhere.

Definition 3.1 (Smoothness)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. We say f is β-smooth ($\beta > 0$), if

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq \beta \|\boldsymbol{x} - \boldsymbol{y}\| \tag{1}$$

for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$.

Definition 3.2 (Locally Strongly Convex)**.** For a (local) minimizer $\boldsymbol{x}_f^*$ of a differentiable function f, we say f is (r, μ)-strongly convex on a convex set $\mathbb{D}_{\boldsymbol{x}_f^*} \supseteq \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}_f^*\| < r\}$, if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_f^*) + \langle \nabla f(\boldsymbol{x}_f^*), \boldsymbol{x}_f - \boldsymbol{x}_f^* \rangle + \frac{\mu}{2} \|x - \boldsymbol{x}_f^*\|^2 \tag{2}$$

for any $\boldsymbol{x} \in \mathbb{D}_{\boldsymbol{x}_f^*}$. Here, $r, \mu > 0$.

Our first lemma establishes a basic building block that if the loss landscape is smooth and locally strongly convex in a basin, then GD with a sufficiently small step size will converge to the basin's minimizer without escaping it.

Lemma 3.3 (GD Convergence in a Local Basin)**.** *Consider gradient descent (GD) starting from any point in an (r, μ)-locally strongly convex domain $\mathbb{D}_{\boldsymbol{x}_f^*}$ of a β-smooth function f, for some $\beta \geq \mu > 0$. Let $(\boldsymbol{x}_k)_{k=1}^N$ be a sequence generated by GD. If the step size satisfies $\eta \leq \min(\frac{1}{\beta}, \frac{r - \|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|}{\beta \|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|})$, we have (1) $\boldsymbol{x}_k$ remains within the locally strongly convex region $\mathbb{D}_{\boldsymbol{x}_f^*}$ in every step k of GD, and that (2) $\boldsymbol{x}_k$ converges to $\boldsymbol{x}_f^*$ as $k \to +\infty$. See proof in Appendix A.1.*

The proof follows that for GD convergence in general, except that we need to check the radius of the locally strongly convex region.

We then take a step further from proving convergence for a fixed loss function to a changing loss landscape. In our framework, a full task transition (e.g., switching from one random labeling to another) is not applied abruptly; instead, it is decomposed into a sequence of small interpolation steps. We model each such interpolation step (i.e., a *micro-task*), as a near-identity linear transformation. The following lemma shows that a single micro-task preserves the essential basin structure.

Lemma 3.4 (Basin Property Preservation)**.** *Consider a gradually changing environment where, after one micro-task change, the loss function evolves from f to $g(\boldsymbol{x}) = f(\boldsymbol{W}\boldsymbol{x})$, with $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ full-rank. Assume $\boldsymbol{W}$ is near-identity in the sense that*

$$\|I - \boldsymbol{W}\| \leq \frac{\epsilon}{1+\epsilon}, \quad \frac{1}{1+\epsilon} \leq \sigma_i(\boldsymbol{W}) \leq \frac{1}{1-\epsilon}, \quad \forall\, i, \tag{3}$$

for some $\epsilon > 0$. If f is β-smooth and (r, μ)-strongly convex in a local basin $\mathbb{D}_{\boldsymbol{x}_f^*}$, then g is also smooth and locally strongly convex in $\mathbb{D}_{\boldsymbol{x}_g^*} = \{\, \boldsymbol{x} : \boldsymbol{W}\boldsymbol{x} \in \mathbb{D}_{\boldsymbol{x}_f^*} \,\}$, with $\beta' = \beta(1+\epsilon)^2$, $\mu' = \mu(1+\epsilon)^{-2}$, and $r' = r(1-\epsilon)$. See proof in Appendix A.2.

We can then verify that the GD iterates optimizing for the old loss function f is actually lies inside the new locally strongly convex basin in g.

Lemma 3.5 (Basin Containment)**.** *Under on the assumptions in Lemma 3.4, the new locally convex set $\mathbb{D}_{\boldsymbol{x}_g^*}$ contains $\boldsymbol{x}$ if $\mathbb{D}_{\boldsymbol{x}_f^*}$ contains $\boldsymbol{W}\boldsymbol{x}$. See proof in Appendix A.3.*

With all three building blocks in place, we can now demonstrate that GD-optimized parameters would converge to a new local optimum of g corresponding to the previous one, i.e., $\boldsymbol{x}_g^* = \boldsymbol{W}^{-1}\boldsymbol{x}_f^*$.

Lemma 3.6 (Convergence After Single Micro-task Transition)**.** *For any $0 < c < r$ and learning rate $\eta \leq \min(\frac{1}{\beta}, \frac{r - \|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|}{\beta \|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|})$, the GD algorithm converges to the new minimizer $\boldsymbol{x}_g^* = \boldsymbol{W}^{-1}\boldsymbol{x}_f^*$, as long as (1) $\frac{1}{1+\epsilon} \leq \sigma_i(\boldsymbol{W}) \leq \frac{1}{1-\epsilon}$ with $\epsilon = \frac{r-c}{r-c+\|\boldsymbol{x}_f^*\|}$ for all $i \in [d]$; and (2) the **current point** $\boldsymbol{x}$ is close to the previous minimizer x_f^* satisfying $\|\boldsymbol{x} - \boldsymbol{x}_f^*\| \leq c(1-\epsilon)$. See proof in Appendix A.4.*

Finally, we formally state our main theorem by extending Lemma 3.6 from a single micro-task to a sequence of micro-tasks via induction. The key observation is that after each micro-task, GD converges to the new basin's minimizer, restoring the preconditions needed for the next micro-task. The theorem implies that plasticity is preserved across arbitrarily large task changes—provided each interpolation step is sufficiently small.

Theorem 3.7. *Let f_0 be β_0-smooth and (r_0, μ_0)-strongly convex. Let $\boldsymbol{x}_0^*$ be a local minimizer of f_0. We assume, under the gradually changing environment, the sequence of loss functions has the form of $f_t(\boldsymbol{x}_t) = f_{t-1}(\boldsymbol{W}_t \boldsymbol{x}_{t-1})$ for $t \geq 1$, where $\boldsymbol{W}_t$ satisfies $\|\sigma_i(\boldsymbol{W}_t) - 1\| \leq \frac{r-c+\|\boldsymbol{x}_{t-1}^*\|}{\|\boldsymbol{x}_{t-1}^*\|}, \forall i$, and $\|\boldsymbol{I} - \boldsymbol{W}_t\| \leq \frac{r-c}{2(r-c)+\|\boldsymbol{x}_{t-1}^*\|}$, for some $c \in (0, r)$. Consider $(\boldsymbol{x}_{t,k})_{k\in[N]}$ being a sequence generated by the GD algorithm while minimizing f_t w.r.t. $\boldsymbol{x}_t$, a learning rate $\eta \leq \min(\frac{1}{\beta_t}, \frac{r_t - \|\boldsymbol{x}_{t,k} - \boldsymbol{x}_{t-1}^*\|}{\beta_t \|\boldsymbol{x}_{t,k} - \boldsymbol{x}_{t-1}^*\|})$ of the tth micro-task guaranties that $\boldsymbol{x}_{t,k}$ converges to $\boldsymbol{x}_t^* = \boldsymbol{W}_t^{-1} \boldsymbol{x}_{t-1}^*$. See proof in Appendix A.5.*

In summary, our analysis provides an explanation on why a gradually changing environment can retain plasticity, as the parameters learned for the previous task can be optimized to the corresponding optimum for the new task.

4 Experiments

In this section, we empirically examine whether a gradually changing environment preserves plasticity in continual learning. We compare the following two scenarios: (1) A gradually changing environment, and (2) An abruptly changing environment with and without loss-of-plasticity mitigation methods. In particular, we consider widely adopted and recent methods including L2 regularization (Krogh & Hertz, 1991), Shrink&Perturb (Ash & Adams, 2020), Spectral regularization (Lewandowski et al., 2025a), and ReDO (Sokar et al., 2023).

We adopt the same benchmarks (Random Image Labeling and Random Pixel Permuting) widely used in prior plasticity research (Dohare et al., 2021; 2024; Lyle et al., 2023; Lewandowski et al., 2024; 2025a) to ensure that any difference in observed plasticity is attributable to the transition abruptness rather than task complexity. We also introduce two language tasks (Random Seq2Seq and Bigram Cipher) to extend the study of plasticity to sequence-to-sequence models, a setting largely unexplored in prior work despite the growing prominence of language models.

As mentioned in §2, the loss of plasticity may happen in the sense of both trainability (Dohare et al., 2021; Abbas et al., 2023) and generalizability (Ash & Adams, 2020; Lee et al., 2024); they are evaluated in §4.1 and §4.2, respectively. Subsection 4.3 provides additional in-depth analyses.

Unless stated otherwise, all results are averaged over 5 random seeds, and shaded regions indicate the minimum and maximum across seeds. Gradual methods use $N = 50$ micro-tasks per task transition ($s = 0.02$) by default. To ensure a fair comparison, abrupt and gradual methods receive the same training budget for each task (one abrupt change on the dataset): when a transition is divided into N micro-tasks, each micro-task is trained for $1/N$ of the per-task epoch budget. A gradual transition therefore uses the same number of gradient updates and training samples as an abrupt one. All methods are evaluated at the end of each task.

4.1 Evaluation of Trainability

Trainability refers to whether the model parameters can be optimized towards new tasks, and it is a fundamental requirement for continual learning (Lyle et al., 2023; Lewandowski et al., 2025a). Loss of trainability refers to a progressive decline in a model's ability to learn new tasks. Extensive research on this phenomenon has examined the conditions under which neural networks fail to retain trainability in non-stationary environments (Lyle et al., 2022; Elsayed & Mahmood, 2024; Dohare et al., 2024; Ma et al., 2024).

Tasks and Models. We experiment with the Random Image Labeling and Random Seq2seq environments (task formulations presented in §2). In particular, we consider three image benchmark datasets with increasing scales: MNIST (LeCun et al., 1998), CIFAR-10 (He et al., 2016), and Tiny-ImageNet (Le & Yang, 2015). Most previous plasticity studies have used selected datasets for their evaluation of trainability, but we include all three in our work.

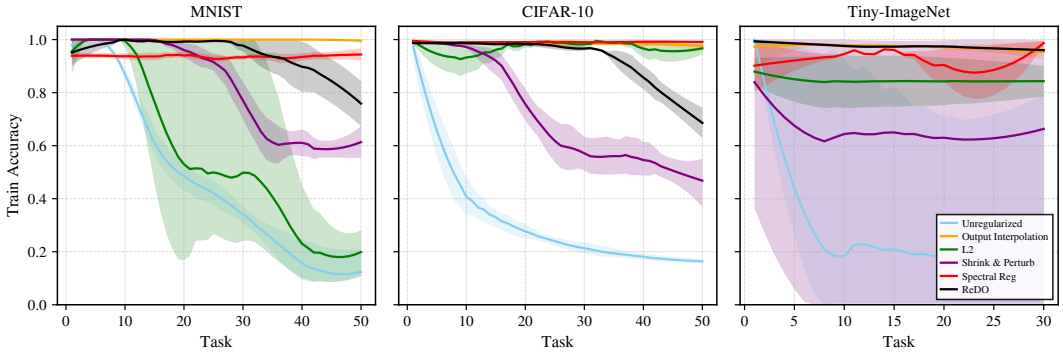


Figure 1: **Trainability for Random Image Labeling on MNIST (4-layer MLP), CIFAR-10, and Tiny-ImageNet (ResNet-18).** Baselines face abrupt label changes, while output interpolation changes labels gradually. Output interpolation maintains near-perfect accuracy, matching or exceeding the best baseline.

For MNIST, we utilize a 4-layer Multi-Layer Perceptron (MLP) with 256 units per layer. Each hidden layer consists of a linear transformation followed by Layer Normalization and a ReLU nonlinearity. For the more complex CIFAR-10 and Tiny-ImageNet tasks, we employ an off-the-shelf ResNet-18 (He et al., 2016) architecture comprising four cascaded residual blocks with batch normalization and skip connections, totaling approximately 11 million parameters.

For Random Seq2Seq, we adopt a pre-trained T5-small model (Raffel et al., 2020) but only keep the first Transformer block. Note that Transformers are powerful models; however, our Random Seq2Seq is synthetic and relatively simple. Therefore, we restrict the Transformer capacity to study the phenomenon of loss of plasticity in our research environment.

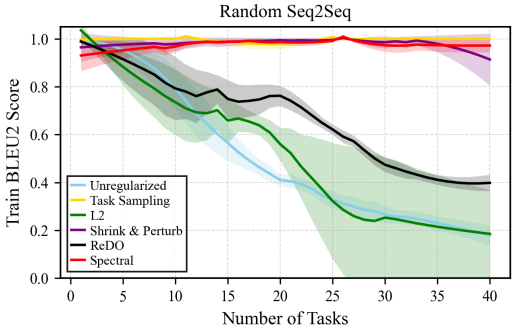


Figure 2: **Trainability for random Seq2Seq task on synthetic text using T5-small.** Without any mitigators, T5-small has diminishing train BLEU2 score, suggesting loss of trainability. Task sampling is the most effective among all baselines in preserving plasticity.

Results. Figs. 1 and 2 present the trainability results for aforementioned benchmarks.

We observe loss of trainability in all the tasks. For example, in an unregularized MLP on the Random MNIST benchmark, the training accuracy collapses from 100% to random guessing after just 50 tasks. This is consistent with findings in the previous work that neural networks lose plasticity with abruptly changing data distribution (Dohare et al., 2021; 2024; Lyle et al., 2022; Elsayed & Mahmood, 2024; Ma et al., 2024).

We compare standard mitigation strategies designed for abrupt changes against our proposed simulation of a gradually changing environment.

Among the mitigation strategies, performance varies significantly by the architecture and benchmarks. Shrink&Perturb (Ash & Adams, 2020) and Spectral Regularization (Lewandowski et al., 2025a) excel only in the Seq2Seq language task, showing high volatility in vision benchmarks, particularly on complex datasets like Tiny-Imagenet. This disparity aligns with the findings by Bai et al., who suggest that Transformer models are inherently more robust to perturbations than CNNs (Bai et al., 2021). L2 regularizated ResNet-18 sustains near perfect training accuracy for both larger vision benchmarks. However, the performance of L2 regularized MLP drops as quickly as an unregularized network, possibly due to dormant neurons and lack of skip connections. ReDO provides moderate remedy for plasticity loss in vision benchmarks via neuronal recycling; however, its performance on language tasks is less satisfactory. Previous work suggests that ReDO is ill-suited for Transformers,

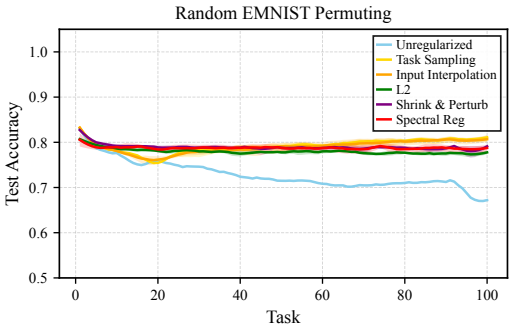


Figure 3: **Generalizability on Random Pixel Permuting tasks with EMNIST and a 4-layer MLP model.** Plasticity loss is milder in this setting, likely because the permutation task is easier to solve. Input interpolation and task sampling end with the highest test accuracy among all methods.

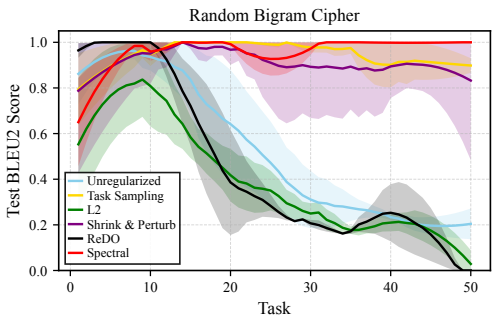


Figure 4: **Generalizability evaluated by test BLEU2 score on Bigram Cipher tasks on customized T5-small model.** Task sampling and Spectral Regularization maintain strong generalizability, while the unregularized baseline, L2, and ReDO degrade significantly.

as the smooth nature of GELU/SiLU activations and the centering effects of Layer Normalization prevent the "dying unit" phenomenon that is common in ReLU-based CNNs (Jain et al., 2026). This phenomenon occurs when a neuron consistently outputs zero for all inputs, causing its gradient to vanish and preventing any further weight updates.

Our work investigates how a gradually changing environment affect plasticity. We find that a gradually changing environment—either simulated by output interpolation or task sampling—largely preserves the plasticity of neural networks. Output interpolation effectively preserves trainability in vision benchmarks, while task sampling is effective for language domains, matching or exceeding the performance of competing strategies. Admittedly, our simulation of gradually changing environments is not perfect, as we observe minor plasticity loss after task 35 for Random MNIST. That said, our approach still outperforms the best baseline from prior work. These results suggest that simulating a gradually changing environment is a simple yet theoretically grounded mitigation for the loss of trainability.

In summary, mitigation methods relying on resetting, regularization, or noise injection offer only partial mitigation; their strict global constraints on the optimization landscape lead to high instability. Further, these methods necessitate exhaustive hyperparameter tuning for each distinct environment; without precise calibration, their constraints often impede optimization. In contrast, our simulation of a gradually changing environment provides a robust, model-agnostic solution that preserves plasticity without requiring extensive hyperparameter search.

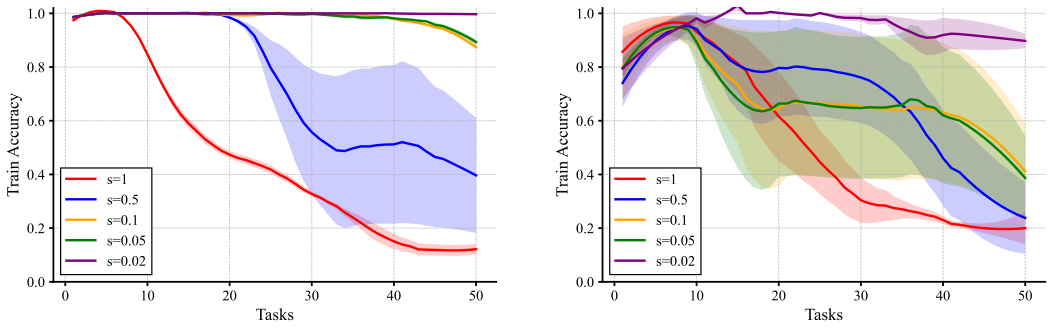
4.2 Evaluation of Generalizability

Loss of generalizability refers to a model's declined ability to apply learned knowledge to unseen data. Since trainability is a prerequisite for generalization, prior work has largely focused on the former (Dohare et al., 2021; Lyle et al., 2022; Sokar et al., 2023; Lewandowski et al., 2024; 2025a). However, because high training accuracy can be achieved through memorization, it is critical to evaluate generalizability to confirm that the model has learned underlying patterns rather than merely overfitting the training set.

Tasks and Models. We investigate with the Random Pixel Permuting and Random Bigram Cipher environments. These permutation-based formulations, as detailed in §2, are particularly well-suited for evaluating generalizability because they can be applied directly to the test splits of the benchmarks.

Results. Figs. **??** and 4 demonstrate the generalizability results for the benchmarks described above.

For Random Pixel Permuting, we observe only a mild plasticity loss (test accuracy drops around 10%) for the unregularized MLP on the Random EMNIST benchmark with 100 tasks. Such a mild plasticity loss is consistent with previous work (Kumar et al., 2023; Lewandowski et al., 2025a), and



(a) Random MNIST Output Interpolation

(b) Bigram Cipher Task Sampling

Figure 5: **The effect of granularity of the interpolation step size on plasticity preservation for both trainability and generalizability task.** Abrupt task change ($s = 1$) causes severe plasticity loss in both settings. As the interpolation step size decreases, plasticity is progressively better preserved, with the finest granularity ($s = 0.02$) maintaining stable performance throughout. This confirms a monotonic relationship between transition smoothness and plasticity retention.

is likely because the nature of the task allows the model to reuse previously learned image features, thereby facilitating adaptation. In this task, we observe all previously introduced mitigation methods are able to preserve plasticity to a large extent (Fig. **??**). Nevertheless, our task sampling and input interpolation (yellow and orange curves) remain the two most competitive methods at the 100th task.

In the Random Bigram Cipher task (Fig. 4), we observe a drastic loss of generalizability for an unregularized T5-small model, with the test BLEU2 score falling to 20% after 50 tasks, reflecting high complexity of the task. Consistent with our Random Seq2Seq findings, ReDO and L2 regularization fail to retain generalizability in the language domain. Similarly, Spectral Regularization and Shrink&Perturb successfully preserve generalizability, mirroring their strong performance on the Random Seq2Seq task.

Overall, our proposed simulation of a gradually changing environment preserves generalizability in both the vision task and the complex language task, matching the performance of the best-performing mitigation methods designed for abrupt task changes. Experiment details are shown in Appendix B.

4.3 In-Depth Analyses

Granularity of Interpolation and Sampling. To investigate the relationship between task transition smoothness and plasticity, we adopt two distinct experimental setups: the classic MNIST model with a 4-layer MLP model due to limited resources, and the Bigram Cipher setup with a T5-small model for generalizability. We adjust the interpolation coefficient α (as mentioned in §2) from 0 to 1 with a step size ranging from $\{0.02, 0.05, 0.1, 0.5\}$. As seen in Figs. 5a and 5b, abrupt task change (step size $= 1$) causes significant accuracy drop after 20 tasks for Random MNIST and 12 tasks for Bigram Cipher, respectively, while coarse-grained task change (step size $= 0.5$) mitigate these declines but are still not satisfactory. In comparison, step size 0.02 sustains near-perfect accuracy throughout the experiments. We have also investigated the relationship between interpolation granularity and four neural network properties including maximum hessian eigenvalue, relative representation change, dormant unit fraction, and distance from initialization (shown in Appendix C). Overall, this analysis verifies that a gradually changing environment does preserve plasticity, and that a finer-grained task transition preserves more plasticity.

Loss Landscape Geometry. To empirically characterize why gradually changing environments retain plasticity, we analyze from the loss landscape perspective. Previous work has shown that the maximum eigenvalue of the weight matrices can be interpreted as the sharpness of the loss landscape (Dinh et al., 2017; Lyle et al., 2023; Lewandowski et al., 2024). We evaluate the loss landscape geometry on a trainability task (Random MNIST) and a generalizability task (Bigram Cipher) with this metric. As shown in Fig. 6, top-performing methods including Spectral Regularization and Shrink&Perturb have relatively small eigenvalues. Notably, gradual task transitions (through interpolation and task

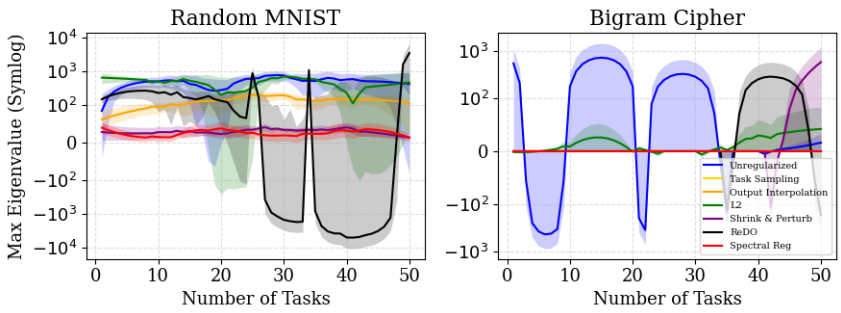


Figure 6: **Maximum eigenvalue across tasks for Random MNIST and Bigram Cipher.** Interpolation and task sampling keep the eigenvalue low and stable, whereas the abrupt and unregularized runs show large task-to-task fluctuations, consistent with a sharper, less stable loss landscape under abrupt transitions.

sampling) naturally maintain low and stable eigenvalues without explicit regularization penalties, suggesting a smooth landscape for optimization.

5 Related Work

Plasticity represents a neural system's adaptability to new environments, a foundational concept in neuroscience (Citri & Malenka, 2008; Rodriguez et al., 2009) that has recently gained attention in deep learning (Dohare et al., 2021; 2024; Lyle et al., 2022; Sokar et al., 2023). The loss of plasticity manifests in two forms: reduced trainability (Dohare et al., 2021; 2024; Lyle et al., 2022; Elsayed & Mahmood, 2024; Lewandowski et al., 2024; 2025b;a; Ma et al., 2024) and reduced generalizability (Dohare et al., 2024; Ash & Adams, 2020; Zilly et al., 2021; Lewandowski et al., 2025a; Lee et al., 2024; Lillo & Cheney, 2026).

Several underlying causes have been identified, each motivating corresponding mitigation strategies. *Dormant neurons*, i.e., inactive units that accumulate during training and reduce effective capacity, can be addressed by periodically resetting less-used neurons (Sokar et al., 2023; Dohare et al., 2024; Frati et al., 2024). *Optimizer instability* arises because standard optimizers are designed for stationary distributions; remedies include optimizer resets (Nikishin et al., 2022; Asadi et al., 2023) and layer normalization (Lyle et al., 2023). *Loss of curvature directions*, characterized by Hessian rank collapse, restricts future optimization (Lewandowski et al., 2024) and has motivated various regularization approaches: L2 (Kumar et al., 2023; Lyle et al., 2022; Dohare et al., 2024), Wasserstein (Lewandowski et al., 2024), and Spectral (Lewandowski et al., 2025a) regularization. *Memorization* leads to loss of generalization as weight magnitudes grow; Shrink&Perturb (Ash & Adams, 2020) addresses this by scaling down weights and injecting noise. Other approaches include alternative activation functions (Abbas et al., 2023; Lillo & Cheney, 2026; Lewandowski et al., 2025b), smaller batch sizes (Obando Ceron et al., 2023), orthogonal regularization (Chung et al., 2024), and churn reduction (Tang et al., 2025).

6 Conclusion

Prior plasticity research has largely focused on an artificial setup of abrupt task change, which may not fully reflect real-world continual learning scenarios. In this paper, we emphasize that real-world environments often evolve gradually. We motivate this intuition with a theoretical analysis and show empirically that plasticity can be largely preserved in a gradually changing environment. Even for abruptly changing setups, loss of plasticity can be effectively mitigated by simulating gradual change from task to task through input/output interpolation and task sampling.

AI USE STATEMENT

In this work, we used generative AI tools mainly to assist with writing and code review. We did not use them to formulate the research ideas, develop the theory, or generate data. We did not use them to formulate the research ideas, develop the theory, or generate data. All AI-assisted text and code were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

The environments and the proposed interpolation and task-sampling methods are defined in §2 and §3. The assumptions of our theoretical results are stated with each lemma and theorem, and complete proofs are given in Appendix A. Datasets, models, and selected hyperparameters are reported in Appendix B. Our code to reproduce the main results is available at `https://github.com/anoniclr1900-oss/plasticity-grad.git`.

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 2nd Conference on Lifelong Learning Agents*, pp. 620–636, 2023. URL https://proceedings.mlr.press/v232/abbas23a.html.

Kavosh Asadi, Rasool Fakoor, and Shoham Sabach. Resetting the optimizer in deep RL: An empirical study. In *Proceedings of the 37th Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=AnFUgNC3Yc.

Jordan Ash and Ryan P Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, pp. 3884–3894, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/288cd2567953f06e460a33951f55daaf-Paper.pdf.

Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than CNNs? In *Advances in Neural Information Processing Systems*, pp. 26831–26843, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/e19347e1c3ca0c0b97de5fb3b690855a-Paper.pdf.

Heinz H Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond lipschitz gradient continuity: First-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017. URL https://doi.org/10.1287/moor.2016.0817.

Tudor Berariu, Wojciech Czarnecki, Soham De, Jörg Bornschein, Samuel L. Smith, Razvan Pascanu, and Claudia Clopath. A study on the plasticity of neural networks. *arXiv preprint arXiv:2106.00042*, 2021. URL url={https://arxiv.org/abs/2106.00042}.

Ang Bian, Wei Li, Hangjie Yuan, Chengrong Yu, Mang Wang, Zixiang Zhao, Aojun Lu, Pengliang Ji, and Tao Feng. Make continual learning stronger via c-flat. In *Proceedings of the Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Dokew2u49m.

Wesley Chung, Lynn Cherif, David Meger, and Doina Precup. Parseval regularization for continual reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 127937–127967, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e6df4efa20adf8ef9acb80e94072a429-Paper-Conference.pdf.

Ami Citri and Robert C Malenka. Synaptic plasticity: Multiple forms, functions, and mechanisms. *Neuropsychopharmacology*, 33(1):18–41, 2008. URL https://doi.org/10.1038/sj.npp.1301559.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *Proceedings of the 2017 International Joint Conference on Neural Networks*, 2017. URL https://doi.org/10.1109/IJCNN.2017.7966217.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Ng. Large scale distributed deep networks. In *Advances in neural information processing systems*, volume 25, 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/6aca97005c68f1206823815f66102863-Paper.pdf.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1019–1028, 2017. URL https://proceedings.mlr.press/v70/dinh17b.html.

Shibhansh Dohare, Richard S Sutton, and A Rupam Mahmood. Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv preprint arXiv:2108.06325*, 2021. URL https://arxiv.org/abs/2108.06325.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024. URL https://doi.org/10.1038/s41586-024-07711-7.

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

Lapo Frati, Neil Traft, Jeff Clune, and Nick Cheney. Reset it and forget it: Relearning last-layer weights improves continual and transfer learning. In *Proceedings of the 27th European Conference on Artificial Intelligence*, pp. 2998–3005, 2024. URL http://dx.doi.org/10.3233/FAIA240840.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. URL https://www.sciencedirect.com/science/article/pii/S1364661399012942.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2015. URL https://arxiv.org/abs/1312.6211.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1489–1501, 2016. URL https://aclanthology.org/P16-1141/.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.

Nitin Jain, Georg Martius, and Marin Vlastelica. Revive and recouple: Mitigating plasticity loss in transformer architectures. *Online Manuscript*, 2026. URL https://openreview.net/forum?id=ahcb5auRmy.

Adam Kilgarriff. I don't believe in word senses. *Computers and the Humanities*, 31:91–113, 1997. URL https://doi.org/10.1023/A:1000583911091.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*, 2015. URL https://doi.org/10.48550/arXiv.1412.6980.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *National Academy of Sciences*, 114(13):3521–3526, 2017. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, 1991. URL https://proceedings.neurips.cc/paper_files/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, 2023. URL https://lifelong-ml.cc/Conferences/2024/acceptedpapersandvideos/conf-2024-26.

Julieta Laurino, Simon De Deyne, Alvaro Cabana, and Laura Kaczer. The pandemic in words: Tracking fast semantic changes via a large-scale word association task. *Open Mind*, 7:221–239, 2023. ISSN 2470-2986. URL https://doi.org/10.1162/opmi_a_00081.

Ya Le and Xiaofeng Yang. Tiny imagenet visual recognition challenge, 2015. URL https://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf. CS231N course project, Stanford University.

Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

Hojoon Lee, Hanseul Cho, Hyunseung Kim, Daehoon Gwak, Joonkee Kim, Jaegul Choo, Se-Young Yun, and Chulhee Yun. PLASTIC: Improving input and label plasticity for sample efficient reinforcement learning. In *Proceedings of the 37th Conference on Advances in Neural Information Processing Systems*, pp. 62270–62295, 2023. URL https://openreview.net/forum?id=c5WOU7p4ES.

Hojoon Lee, Hyeonseo Cho, Hyunseung Kim, Donghu Kim, Dugki Min, Jaegul Choo, and Clare Lyle. Slow and steady wins the race: Maintaining plasticity with hare and tortoise networks. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 26416–26438, 2024. URL https://proceedings.mlr.press/v235/lee24d.html.

Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C. Machado. Directions of curvature as an explanation for loss of plasticity. *arXiv preprint arXiv:2312.00246*, 2024. URL https://arxiv.org/abs/2312.00246.

Alex Lewandowski, Michał Bortkiewicz, Saurabh Kumar, András György, Dale Schuurmans, Mateusz Ostaszewski, and Marlos C. Machado. Learning continually by spectral regularization. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=Hcb2cgPbMg.

Alex Lewandowski, Dale Schuurmans, and Marlos C. Machado. Plastic learning with deep fourier features. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=NIkfix2eDQ.

Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 661–670, 2014. URL https://doi.org/10.1145/2623330.2623612.

Lute Lillo and Nick Cheney. Activation function design sustains plasticity in continual learning. In *Proceedings of the 14th International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XZf6wObHX4.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. In *Proceedings of International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ZkC8wKoLbQ7.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 23190–23211, 2023. URL https://proceedings.mlr.press/v202/lyle23b.html.

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 3rd Conference on Lifelong Learning Agents*, 2024. URL https://lifelong-ml.cc/Conferences/2024/acceptedpapersandvideos/conf-2024-9.

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 Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=0aR1s9YxoL.

Ernst Mayr. *Populations, Species, and Evolution: An Abridgment of Animal Species and Evolution*. Harvard University Press, 1970. URL https://archive.org/details/populationsspeci0000unse/page/2/mode/2up.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24: 109–165, 1989. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Martial Mermillod, Aurélia Bugaiska, and Patrick BONIN. The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, Volume 4 - 2013, 2013. URL https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2013.00504.

Sunny Mitra, Ritwik Mitra, Suman Kalyan Maity, Martin Riedl, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. An automatic approach to identify word sense changes in text media across timescales. *Natural Language Engineering*, 21(5):773–798, 2015. URL https://doi.org/10.1017/S135132491500011X.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f1748d6b0fd9d439f71450117eba2725-Paper.pdf.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 16828–16847, 2022. URL https://proceedings.mlr.press/v162/nikishin22a.html.

Johan Obando Ceron, Marc Bellemare, and Pablo Samuel Castro. Small batch deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 26003–26024, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/528388f1ad3a481249a97cbb698d2fe6-Paper-Conference.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Mark B. Ring. Child: A first step towards continual learning. *Machine Learning*, 28(1):77–104, 1997. URL https://doi.org/10.1023/A:1007331723572.

JJ Rodriguez, M Olabarria, A Chvatal, and A Verkhratsky. Astroglia in dementia and alzheimer's disease. *Cell Death & Differentiation*, 16(3):378–385, 2009. URL https://doi.org/10.1038/cdd.2008.172.

Paul Ruvolo and Eric Eaton. ELLA: An efficient lifelong learning algorithm. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 507–515, 2013. URL https://proceedings.mlr.press/v28/ruvolo13.html.

Baekrok Shin, Junsoo Oh, Hanseul Cho, and Chulhee Yun. DASH: Warm-starting neural network training in stationary settings without loss of plasticity. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=IdQuUYMA1t.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/0efbe98067c6c73dba1250d2beaa81f9-Paper.pdf.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32145–32168, 2023. URL https://proceedings.mlr.press/v202/sokar23a.html.

Kim Sterelny. From hominins to humans: How sapiens became behaviourally modern. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1566):809–822, 2011. URL https://doi.org/10.1098/rstb.2010.0301.

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 42nd International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EkoFXfSauv.

Sebastian Thrun. *Lifelong Learning Algorithms*. Springer US, 1998. URL https://doi.org/10.1007/978-1-4615-5529-2_8.

Yichen Wu, Long-Kai Huang, Renzhen Wang, Deyu Meng, and Ying Wei. Meta continual learning revisited: Implicitly enhancing online hessian approximation via variance reduction. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TpD2aG1h0D.

Julian Zilly, Alessandro Achille, Andrea Censi, and Emilio Frazzoli. On plasticity, invariance, and mutually frozen weights in sequential task learning. In *Advances in Neural Information Processing Systems*, pp. 12386–12399, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/6738fc33dd0b3906cd3626397cd247a7-Paper.pdf.

A PROOFS OF LEMMAS AND THEOREM

For completeness, we restate the descent lemma (Bauschke et al., 2017) below.

Lemma A.1 (Descent Lemma)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. If f is β-smooth, then for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\beta}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{4}$$

Proof. Let $\boldsymbol{h} = \boldsymbol{y} - \boldsymbol{x}$ and define the scalar function $\phi(t) = f(\boldsymbol{x} + t\boldsymbol{h})$ for $t \in [0, 1]$. By the fundamental theorem of calculus,

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) = \int_0^1 \nabla f(\boldsymbol{x} + t\boldsymbol{h})^\top \boldsymbol{h}\, dt. \tag{5}$$

Subtracting and adding $\nabla f(\boldsymbol{x})^\top \boldsymbol{h}$ inside the integral gives

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) - \nabla f(\boldsymbol{x})^\top \boldsymbol{h} = \int_0^1 [\nabla f(\boldsymbol{x} + t\boldsymbol{h}) - \nabla f(\boldsymbol{x})]^\top \boldsymbol{h}\, dt. \tag{6}$$

Since f is β-smooth, we have $\|\nabla f(\boldsymbol{x} + t\boldsymbol{h}) - \nabla f(\boldsymbol{x})\| \leq \beta t\|\boldsymbol{h}\|$. Applying the Cauchy–Schwarz inequality and integrating over $t \in [0, 1]$ yields

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) - \nabla f(\boldsymbol{x})^\top \boldsymbol{h} \leq \int_0^1 \beta t\|\boldsymbol{h}\|^2\, dt = \frac{\beta}{2}\|\boldsymbol{h}\|^2. \tag{7}$$

Rearranging the terms gives the desired result. □

A.1 PROOF OF LEMMA 3.3

Proof. Since $\boldsymbol{x}_f^*$ is a local minimizer, $\nabla f(\boldsymbol{x}_f^*) = 0$.

By β-smoothness (Def. 3.1),

$$\|\nabla f(\boldsymbol{x}_k)\| = \|\nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_f^*)\| \leq \beta\|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|. \tag{8}$$

Part (1): Iterates remain inside $\mathbb{D}_{\boldsymbol{x}_f^*}$.

From the GD update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta\nabla f(\boldsymbol{x}_k)$ and equation 8,

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_f^*\| \leq \|\boldsymbol{x}_k - \boldsymbol{x}_f^*\| + \eta\beta\|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|. \tag{9}$$

If $\eta \leq \frac{r - \|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|}{\beta\|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|}$ is the binding constraint, then $\eta\beta\|\boldsymbol{x}_k - \boldsymbol{x}_f^*\| \leq r - \|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|$, so equation 9 gives $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_f^*\| \leq r$.

If instead $\eta \leq 1/\beta$, then $1/\beta \leq \frac{r - \|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|}{\beta\|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|}$, which implies $\|\boldsymbol{x}_k - \boldsymbol{x}_f^*\| \leq r/2$, so equation 9 gives $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_f^*\| \leq 2\|\boldsymbol{x}_k - \boldsymbol{x}_f^*\| \leq r$.

In both cases $\boldsymbol{x}_{k+1} \in \mathbb{D}_{\boldsymbol{x}_f^*}$, and the claim follows by induction.

Part (2): Geometric convergence.

By the descent lemma (Lemma A.1) with $\boldsymbol{y} = \boldsymbol{x}_k - \eta\nabla f(\boldsymbol{x}_k)$ and $\eta \leq 1/\beta$:

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) - \frac{\eta}{2}\|\nabla f(\boldsymbol{x}_k)\|^2. \tag{10}$$

From local strong convexity (Def. 3.2) and $\nabla f(\boldsymbol{x}_f^*) = 0$, we have $f(\boldsymbol{x}_k) - f(\boldsymbol{x}_f^*) \geq \frac{\mu}{2}\|\boldsymbol{x}_k - \boldsymbol{x}_f^*\|^2$, which combined with equation 8 yields $\|\nabla f(\boldsymbol{x}_k)\|^2 \geq 2\mu\big(f(\boldsymbol{x}_k) - f(\boldsymbol{x}_f^*)\big)$.

By substituting we have:

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_f^*) \leq (1 - \eta\mu)\big(f(\boldsymbol{x}_k) - f(\boldsymbol{x}_f^*)\big). \tag{11}$$

Since $\eta\mu \leq \mu/\beta \leq 1$, the function values converge geometrically, which implies geometric decay and $\boldsymbol{x}_k \to \boldsymbol{x}_f^*$ as $k \to \infty$. □

A.2 Proof of Lemma 3.4

Proof. The gradient of g is $\nabla g(\boldsymbol{x}) = \boldsymbol{W}^\top \nabla f(\boldsymbol{W}\boldsymbol{x})$. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,

$$\|\nabla g(\boldsymbol{x}) - \nabla g(\boldsymbol{y})\| = \|\boldsymbol{W}^\top(\nabla f(\boldsymbol{W}\boldsymbol{x}) - \nabla f(\boldsymbol{W}\boldsymbol{y}))\| \le \|\boldsymbol{W}\|^2 \cdot \beta\|\boldsymbol{x} - \boldsymbol{y}\| \le \beta(1+\epsilon)^2\|\boldsymbol{x} - \boldsymbol{y}\|, \tag{12}$$

so g is $\beta' = \beta(1+\epsilon)^2$-smooth.

Next, since f is μ-strongly convex,

$$f(\boldsymbol{W}\boldsymbol{x}) \ge f(\boldsymbol{W}\boldsymbol{y}) + \nabla f(\boldsymbol{W}\boldsymbol{y})^\top(\boldsymbol{W}\boldsymbol{x} - \boldsymbol{W}\boldsymbol{y}) + \tfrac{\mu}{2}\|\boldsymbol{W}(\boldsymbol{x} - \boldsymbol{y})\|^2. \tag{13}$$

By the chain rule, $\nabla g(\boldsymbol{y}) = \boldsymbol{W}^\top \nabla f(\boldsymbol{W}\boldsymbol{y})$, and using $\|\boldsymbol{W}(\boldsymbol{x} - \boldsymbol{y})\| \ge \sigma_{\min}(\boldsymbol{W})\|\boldsymbol{x} - \boldsymbol{y}\| \ge \frac{1}{1+\epsilon}\|\boldsymbol{x} - \boldsymbol{y}\|$, we have

$$g(\boldsymbol{x}) \ge g(\boldsymbol{y}) + \nabla g(\boldsymbol{y})^\top(\boldsymbol{x} - \boldsymbol{y}) + \tfrac{\mu}{2}\left(\tfrac{1}{1+\epsilon}\right)^2\|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{14}$$

Thus g is (r', μ')-locally strongly convex with $\mu' = \mu(1+\epsilon)^{-2}$.

Finally, for any $\boldsymbol{x}$ satisfying $\|\boldsymbol{x} - \boldsymbol{x}_g^*\| \le r' = r/\sigma_{\max}(\boldsymbol{W})$, we have $\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{x}_f^*\| \le \|\boldsymbol{W}\| \cdot \|\boldsymbol{x} - \boldsymbol{x}_g^*\| \le \sigma_{\max}(\boldsymbol{W})r' = r$, which implies $\boldsymbol{W}\boldsymbol{x} \in \mathbb{D}_{\boldsymbol{x}_f^*}$. Hence $\boldsymbol{x} \in \mathbb{D}_{\boldsymbol{x}_g^*}$, completing the proof. □

A.3 Proof of Lemma 3.5

Proof. For any $\boldsymbol{x} \in \mathbb{R}^d$ satisfying $\|\boldsymbol{x} - \boldsymbol{x}_g^*\| \le r(1-\epsilon)$, we have

$$\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{x}_f^*\| = \|\boldsymbol{W}(\boldsymbol{x} - \boldsymbol{x}_g^*)\| \le \|\boldsymbol{W}\|\,\|\boldsymbol{x} - \boldsymbol{x}_g^*\| \le \tfrac{1}{1-\epsilon}\, r(1-\epsilon) = r. \tag{15}$$

Hence $\boldsymbol{W}\boldsymbol{x} \in \mathbb{D}_{\boldsymbol{x}_f^*}$, which implies $\boldsymbol{x} \in \mathbb{D}_{\boldsymbol{x}_g^*}$. Therefore, the ball of radius $r(1-\epsilon)$ around $\boldsymbol{x}_g^*$ is fully contained within the preimage of the strongly convex basin $\mathbb{D}_{\boldsymbol{x}_f^*}$. □

A.4 Proof of Lemma 3.6

Proof. Since $g(\boldsymbol{x}) = f(\boldsymbol{W}\boldsymbol{x})$, we have $\nabla g(\boldsymbol{x}) = \boldsymbol{W}^\top \nabla f(\boldsymbol{W}\boldsymbol{x})$. Setting $\nabla g(\boldsymbol{x}_g^*) = 0$ and using full rank of $\boldsymbol{W}$ gives $\nabla f(\boldsymbol{W}\boldsymbol{x}_g^*) = 0$, hence $\boldsymbol{W}\boldsymbol{x}_g^* = \boldsymbol{x}_f^*$ and $\boldsymbol{x}_g^* = \boldsymbol{W}^{-1}\boldsymbol{x}_f^*$.

The near-identity assumption $\|\boldsymbol{I} - \boldsymbol{W}\| \le \frac{\epsilon}{1+\epsilon} < 1$ ensures that the Neumann series $\boldsymbol{W}^{-1} = (\boldsymbol{I} - (\boldsymbol{I} - \boldsymbol{W}))^{-1} = \sum_{k=0}^{\infty}(\boldsymbol{I} - \boldsymbol{W})^k$ converges, giving

$$\|\boldsymbol{W}^{-1} - \boldsymbol{I}\| = \Big\|\sum_{k=1}^{\infty}(\boldsymbol{I} - \boldsymbol{W})^k\Big\| \le \frac{\|\boldsymbol{I} - \boldsymbol{W}\|}{1 - \|\boldsymbol{I} - \boldsymbol{W}\|} \le \frac{\epsilon/(1+\epsilon)}{1/(1+\epsilon)} = \epsilon. \tag{16}$$

The shift between the two minimizers is therefore bounded by

$$\|\boldsymbol{x}_f^* - \boldsymbol{x}_g^*\| = \|(\boldsymbol{I} - \boldsymbol{W}^{-1})\boldsymbol{x}_f^*\| \le \|\boldsymbol{I} - \boldsymbol{W}^{-1}\|\,\|\boldsymbol{x}_f^*\| \le \epsilon\|\boldsymbol{x}_f^*\|. \tag{17}$$

Substituting $\epsilon = \frac{r-c}{r-c+\|\boldsymbol{x}_f^*\|}$ we have:

$$\epsilon\|\boldsymbol{x}_f^*\| = \frac{(r-c)\|\boldsymbol{x}_f^*\|}{r-c+\|\boldsymbol{x}_f^*\|} = (r-c)\left(1 - \frac{\|\boldsymbol{x}_f^*\|}{r-c+\|\boldsymbol{x}_f^*\|}\right) = (r-c)(1-\epsilon). \tag{18}$$

If $\|\boldsymbol{x} - \boldsymbol{x}_f^*\| \le c(1-\epsilon)$, then by the triangle inequality,

$$\|\boldsymbol{x} - \boldsymbol{x}_g^*\| \le \|\boldsymbol{x} - \boldsymbol{x}_f^*\| + \|\boldsymbol{x}_f^* - \boldsymbol{x}_g^*\| \le c(1-\epsilon) + (r-c)(1-\epsilon) = r(1-\epsilon), \tag{19}$$

which implies $\boldsymbol{x} \in \mathbb{D}_{\boldsymbol{x}_g^*}$ by Lemma 3.5.

Since g is β'-smooth and (r', μ')-locally strongly convex in $\mathbb{D}_{\boldsymbol{x}_g^*}$ by Lemma 3.4 (with $\beta' = \beta(1+\epsilon)^2$, $\mu' = \mu(1+\epsilon)^{-2}$, $r' = r(1-\epsilon)$), Lemma 3.3 ensures that $\boldsymbol{x}_k \to \boldsymbol{x}_g^*$ for a step size $\eta \le \min\left(\frac{1}{\beta'}, \frac{r' - \|\boldsymbol{x}_k - \boldsymbol{x}_g^*\|}{\beta'\|\boldsymbol{x}_k - \boldsymbol{x}_g^*\|}\right)$.

Note that the effective smoothness β may vary slightly as the environment evolves. □

A.5 Proof of Theorem 3.7

Proof. We begin with $\boldsymbol{x}_0^*$, a benign minimizer of f_0, and initialize GD within its locally strongly convex and β-smooth basin $\mathbb{D}_{\boldsymbol{x}_0^*}$. When the loss transitions from f_0 to $f_1(\boldsymbol{x}) = f_0(\boldsymbol{W}_1\boldsymbol{x})$, the spectral constraint on $\boldsymbol{W}_1$ ensures the minimizer shift is bounded:

$$\|\boldsymbol{x}_0^* - \boldsymbol{x}_1^*\| = \|\boldsymbol{x}_0^* - \boldsymbol{W}_1^{-1}\boldsymbol{x}_0^*\| \leq \|\boldsymbol{I} - \boldsymbol{W}_1^{-1}\|\,\|\boldsymbol{x}_0^*\| \leq \epsilon\|\boldsymbol{x}_0^*\| \leq (r-c)(1-\epsilon). \tag{20}$$

During optimization, Lemma 3.3 guarantees that the GD iterates $\boldsymbol{x}_k$ contract toward $\boldsymbol{x}_0^*$, eventually satisfying $\|\boldsymbol{x}_k - \boldsymbol{x}_0^*\| \leq c(1-\epsilon)$. By Lemma 3.6, this implies $\boldsymbol{x}_k \in \mathbb{D}_{\boldsymbol{x}_1^*}$, ensuring convergence to the new minimizer $\boldsymbol{x}_1^*$.

Assuming the result holds for $\boldsymbol{x}_i^*$, the same argument applies recursively to the transition $f_i \mapsto f_{i+1}$ under the corresponding $\boldsymbol{W}_{i+1}$ satisfying the same spectral condition. By induction, GD converges to each successive shifted minimizer $\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_t^*$. □

B Experiment Details

B.1 Tasks and Models

B.1.1 Trainability Evaluations

Random Image Labeling. To simulate continual learning, the Random Image Labeling environment randomly relabels the images as a new task. It has been studied the most in loss of plasticity research Lyle et al. (2023; 2024); Lewandowski et al. (2024); Kumar et al. (2023); Tang et al. (2025) and is also adopted in our evaluation. In particular, we consider three image benchmark datasets with increasing scales: MNIST (LeCun et al., 1998), CIFAR-10 (He et al., 2016), and Tiny-ImageNet (Le & Yang, 2015). Most previous plasticity studies have used selected datasets for their evaluation of trainability, but we include all three in our work.

For MNIST, we utilize a 4-layer Multi-Layer Perceptron (MLP) with 256 units per layer. Each hidden layer consists of a linear transformation followed by Layer Normalization and a ReLU nonlinearity. For the more complex CIFAR-10 and Tiny-ImageNet tasks, we employ an off-the-shelf ResNet-18 (He et al., 2016) architecture comprising four cascaded residual blocks with batch normalization and skip connections, totaling approximately 11 million parameters.

Random Seq2Seq. To enrich the evaluation of trainability preservation, we propose a Random Seq2Seq environment, following the spirit of Random Image Labeling. The Random Seq2Seq environment presents a similar challenge of memorizing arbitrary mappings but in the text domain.

We adopt a pre-trained T5-small model (Raffel et al., 2020) but only keep the first Transformer block. Note that Transformers are powerful models; however, our Random Seq2Seq is synthetic and relatively simple. Therefore, we restrict the Transformer capacity to study the phenomenon of loss of plasticity in our research environment.

B.1.2 Generalizability Evaluations

Random Pixel Permuting: This environment random permutes the input image pixels for each task (Goodfellow et al., 2015). It is a standard benchmark for studying generalization in supervised continual learning (Dohare et al., 2024; Kumar et al., 2023; Lewandowski et al., 2025a), a protocol we similarly adopt for our evaluation. We evaluate this setting using the EMNIST dataset (Cohen et al., 2017), employing the same 4-layer MLP model utilized for the Random MNIST Labeling task.

Bigram Cipher: To diversify the evaluation of generalization, we propose a Random Bigram Cipher task. This is a synthetic sequence-to-sequence task where the output sequence is generated by applying a cipher to each bigram of the input sequence. We evaluate this task using the same T5-small architecture employed in the Random Seq2Seq task.

B.2 General Setups

All experiments were optimized using Adam (Kingma & Ba, 2015). The learning rate was selected via a sweep over $\{0.005, 0.001, 0.0005, 0.0001\}$; a learning rate of 0.001 performed best for MNIST,

Tiny-ImageNet, and the Bigram Cipher task, while 0.0001 was optimal for the remaining experiments. All results are averaged over 5 random seeds, and shaded regions in figures indicate the indicate the minimum and maximum across seeds.

For our interpolation and task sampling methods, the key hyperparameter is the interpolation/sampling granularity, i.e., the number of intermediate micro-tasks per task transition. A finer granularity preserves more plasticity (see §4.3). We used 50 intermediate steps per transition across all experiments for consistency. All experiments were conducted on a computer with four NVIDIA Quadro RTX 6000 GPUs (24GB each) and an Intel Core i9-9940X CPU (14 cores).

B.3 Baseline and Hyperparameters

For all baseline methods, hyperparameters were tuned via grid search over the following candidate values:

- **L2/Spectral Regularization** Regularization strength is selected from $\{0.1,\ 0.01,\ 10^{-3},\ 10^{-4},\ 10^{-5}, 10^{-6},\ 10^{-7}\}$.
- **Shrink&Perturb.** The shrink factor is chosen from $\{0.5, 0.7, 0.9\}$, and the perturb factor from $\{0.1, 0.01, 10^{-3}, 10^{-4}, 10^{-5}\}$.
- **ReDO.** Recycling frequency is selected from $\{10, 100, 1000, 10000\}$ iterations.

The best configuration for each baseline was selected independently per benchmark.

B.4 Dataset and Training Configurations

MNIST. The dataset contains 28×28 grayscale images from 10 classes. We use $5{,}120$ training images, sampled uniformly across classes, with a batch size of 512, and train each task for 240 epochs.

CIFAR-10. The dataset contains 32×32 color images from 10 classes. We use all $50{,}000$ training images with a batch size of 250, and train each task for 120 epochs.

Tiny-ImageNet. The dataset contains 64×64 color images from 200 classes. We use all $100{,}000$ training images with a batch size of 250, and train each task for 120 epochs.

Random Seq2Seq. The dataset consists of $5{,}120$ synthetic samples generated from the English alphabet, with a word length of 5, an input sequence length of 50, and a target length of 5. We use a batch size of 64 and train each task for 900 epochs.

Bigram Cipher. Synthetic text generated from the English alphabet with word length 5, input sequence length 10, target length 10. 1280 samples for training and 512 for testing. We select a batch size of 64 and trained 200 epochs per task.

C Analysis on Task Transition Granularity

We investigate how interpolation granularity affects the loss landscape and internal network behavior by tracking three properties across different granularity levels. These properties are evaluated on the Random Image Labeling benchmark using a 4-layer MLP and the Bigram Cipher benchmark using T5-small.

Maximum Hessian Eigenvalue. The maximum eigenvalue of the weight matrices measures the sharpness of the loss landscape (Dinh et al., 2017; Lewandowski et al., 2024; Lyle et al., 2022). A higher value indicates a sharper region, which is associated with greater difficulty in optimization. We compute the maximum Hessian eigenvalue at checkpoints after each task transition, under varying numbers of interpolation steps. As shown in Figure 7, finer-grained transitions consistently produce lower maximum eigenvalues, indicating that the optimizer stays in a flatter, better-conditioned region. This aligns with our theoretical analysis that local smoothness is preserved under small task changes.

Relative Representation Change. Relative representation change captures how much hidden representations shift between consecutive tasks, serving as a proxy for the network's ability to

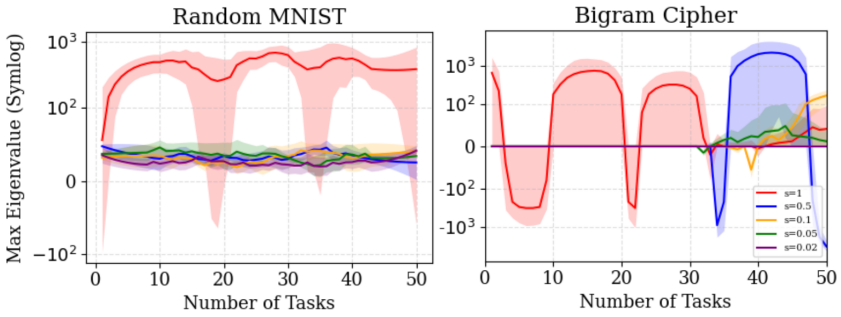


Figure 7: Finer-grained transitions consistently produce lower maximum eigenvalues, indicating that the optimizer remains in a flatter, better-conditioned region of the loss landscape. This aligns with our theoretical analysis that under small task changes, the local smoothness is preserved.

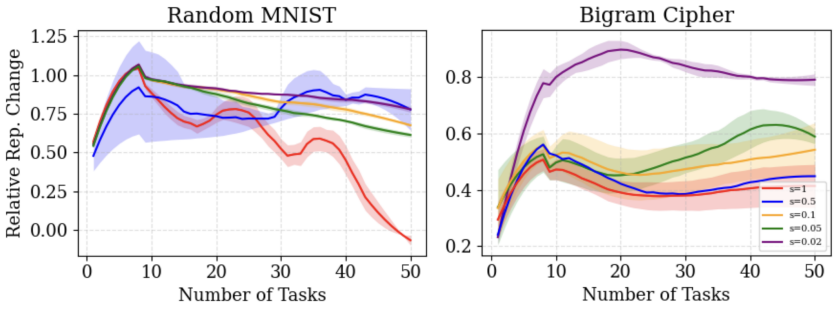


Figure 8: Smaller interpolation steps better preserve the ability to update hidden representations between consecutive tasks, indicating that smoother task changes preserve the network's ability to learn (plasticity).

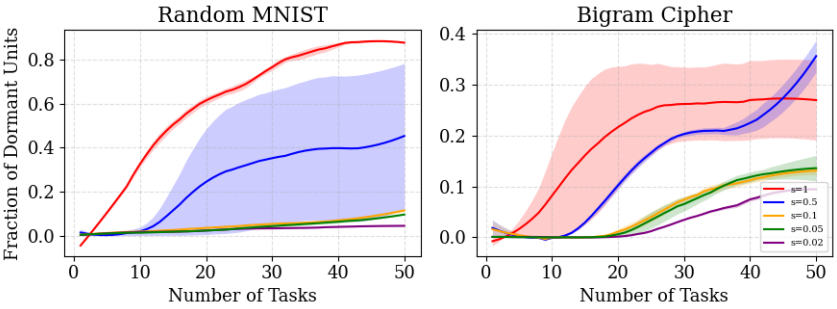


Figure 9: Finer-grained transitions consistently have a lower fraction of dormant neurons while also preserving plasticity.

learn (Lewandowski et al., 2025a). We measure the change in hidden representations between consecutive task checkpoints on a fixed probe batch:

$$\text{RelRepChange} = \mathbb{E}_x \left[\frac{\|h_{\text{curr}}(x) - h_{\text{prev}}(x)\|}{\|h_{\text{prev}}(x)\|} \right],$$

where $h(x)$ is the hidden representation of input x. As shown in Figure 8, smaller interpolation steps better preserve the ability to update hidden representations between consecutive tasks. This suggests that smoother task transitions maintain the network's capacity to adapt to new data.

Dormant Unit Fraction. A dormant neuron is one that rarely fires, and a high dormant fraction has been linked to plasticity loss (Dohare et al., 2024; Sokar et al., 2023). Following Dohare et al. (2024),

a unit is dormant if its post-ReLU activation is positive on fewer than $\tau = 1\%$ of probe inputs:

$$\text{DormantFraction} = \frac{\sum_{l=1}^{L} \sum_{i=1}^{H^l} \mathbf{1}\left[\frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\left[a_{i,n}^{l} > 0\right] \leq \tau\right]}{\sum_{l=1}^{L} H^l},$$

where l is the layer index, H^l the number of hidden units in layer l, N the number of probe inputs, and $a_{i,n}^{l}$ the post-ReLU activation of neuron i in layer l on input n. As shown in Figure 9, finer-grained transitions consistently produce fewer dormant neurons. This is consistent with prior findings that plasticity loss is closely tied to neuron dormancy, and suggests that gradual transitions help keep more of the network active and trainable.

D Limitations and Future Work

Our theoretical analysis assumes Lipschitz smoothness and local strong convexity of the loss landscape, which may not hold for all real-world tasks. That said, both are standard assumptions in optimization theory. Local strong convexity naturally arises near well-behaved minima through second-order Taylor approximation, and Lipschitz smoothness is widely adopted in convergence analysis (Bian et al., 2024; Wu et al., 2024).

For empirical analysis, we evaluated on four tasks, which may not cover all possible environments. However, we selected image benchmarks widely used in prior work and additionally proposed more challenging language tasks to broaden the evaluation. Therefore, we believe our experiments are broad and convincing.

Our study focuses on one side of continual learning. This field has traditionally focused on *catastrophic forgetting*, the degradation of performance on earlier tasks after training on later ones (McCloskey & Cohen, 1989; French, 1999), which is mitigated, for example, by regularizing important parameters toward their earlier values (Kirkpatrick et al., 2017) or by replaying earlier data (Shin et al., 2017). The tension between retaining earlier knowledge and acquiring new knowledge is known as the stability–plasticity dilemma (Mermillod et al., 2013). Following prior plasticity research (Dohare et al., 2021; 2024; Lyle et al., 2022), we study the plasticity side of this dilemma, the ability to learn the current task, and do not evaluate forgetting. Whether gradual transitions also affect how much of earlier tasks is retained remains an open question.

A natural direction for future work is therefore to study plasticity and forgetting jointly under gradual change. Another direction is to determine the transition granularity automatically for a given application, since it is currently set manually. Finally, extending our analysis to reinforcement learning is an important next step, since the data distribution there can change not only with the environment but also with the agent's own behavior (Dohare et al., 2024).