THE SPECTRAL LAZY REGIME: SPECTRAL NORMS AND GENERALIZATION IN CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Loss of plasticity, the progressive degradation of a neural network's ability to learn, has been observed in several continual learning settings. This phenomenon can be split into two related ideas: loss of trainability, when a network loses its ability to reduce the training loss, and loss of generalization, when a network can still be trained but its generalization abilities are impaired. We focus on generalization and propose a mechanism for when this can arise: entering the *spectral lazy regime*. This learning regime is signaled by large increases in the spectral norm of the parameter matrices. When this occurs, the learning dynamics more closely resemble training in the lazy (kernel) regime—representation learning is inhibited, leading to poor generalization. We support this finding by establishing a novel theoretical connection between large spectral norms and the lazy regime. We empirically validate that training a network with constraints on the spectral norm of its weight matrices avoids loss of generalization. Through careful investigations into other norm constraints, we conclude that other choices can be useful but only the spectral norm leads to the radius sensitivity curve being invariant to the width of the matrix. Furthermore, to enable finer control over the spectral norm, we propose two new parameterization of weight matrices, the *singular value parameterization* and the *spectral norm parameterization*. We show that these approaches naturally reduce the spectral norm growth compared to standard parameter matrices and can mitigate loss of generalization.

1 INTRODUCTION

Loss of plasticity has been identified as a key challenge to training deep neural networks in continual learning settings (Dohare et al., 2021; Lyle et al., 2023; Ash & Adams, 2019; Igl et al., 2020). This phenomena refers to the gradual reduction of a neural network's capacity to learn new tasks after training on others.

We can distinguish between two types of plasticity loss: *loss of trainability* (a neural network's failure to converge to a low training error) and *loss of generalization* (the inability to generalize well despite achieving low error on the training examples). However, as Dohare et al. (2021) note, understanding the precise relationship and boundary between these two types of loss remains an open question.

In this work, we investigate the relationship between these two modes of plasticity degradation. We first highlight an experimental setting where neural networks lose the ability to generalize while remaining fully trainable. Figure 1 illustrates this via learning curves on sequential supervised tasks, including a synthetic regression task (fitting randomly generated neural network targets) and a split-TinyImageNet sequence (learning varying subsets of classes). For the baseline network, training error reliably reaches similar minimums in every task, confirming no loss of trainability. However, the test performance tells a different story. Although the network successfully minimizes test error within any given task, its overall generalization drastically deteriorates across the sequence of tasks.

We explain this phenomenon through a novel perspective on the learning dynamics: the network enters a *spectral lazy regime*. This transition is characterized by a large increase in the spectral norm of the parameter matrices (see Figure 2). Once this occurs, the network behaves similarly to the lazy

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.

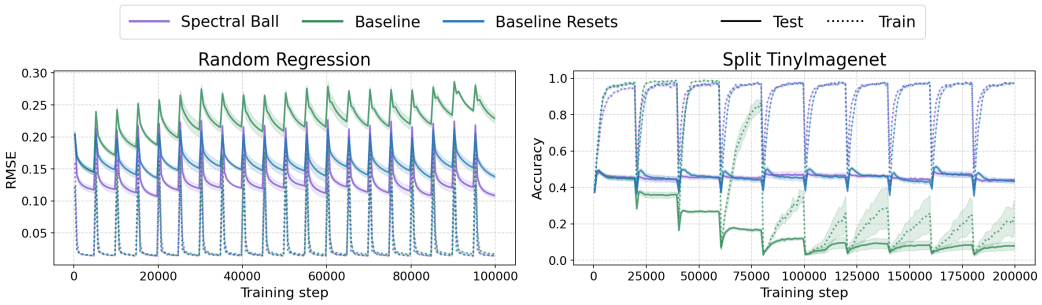

Figure 1: Learning curves for the unconstrained baseline, the baseline reset at every task and adding a spectral norm constraint. The baseline suffers from loss of generalization and the spectral ball norm constraint eliminates this issue.

(kernel) regime (Jacot et al., 2018; Chizat et al., 2019; Geiger et al., 2020), where representations remain fixed and feature learning is nonexistent. This lack of representation learning directly causes the observed drop in generalization. Despite this, trainability is preserved; the network's overparameterization allows it to minimize training error using learning dynamics equivalent to that of a linear model with one feature per learnable parameter.

More specifically, we extend the analysis of Chizat et al. (2019) to show that, for MLPs with smooth ReLU-like activations, scaling the top singular value by c gives a $O(c^{-L})$ bound on the difference between the linearized dynamics and the true dynamics of training a depth L MLP. Previously, Chizat et al. (2019) had shown that inducing the lazy regime was possible by increasing the output scale (a constant multiplying the output of a network), which is equivalent to scaling all the weights by a large constant for homogeneous networks. Here, we relax this requirement by only utilizing a single large singular value in each parameter matrix.

To complement these findings, we conduct an empirical investigation on the role of the spectral norm compared to other norms. Several works have proposed to constraint different of parameter matrices. These include bounding the spectral norm with SpectralNorm (Yoshida & Miyato, 2017) or spectral cap (Newhouse et al., 2025), normalizing parameter matrices' rows by their ℓ_2-norm (Lee et al., 2025; Karras et al., 2022) and maintaining approximately orthogonal matrices with SRIP (Bansal et al., 2018). Some of these have been proposed specifically to address forms of plasticity loss such as weight clipping (Elsayed et al., 2024), Parseval regularization (Chung et al., 2024), spectral regularization (Lewandowski et al., 2024a) and L2-init (Kumar et al., 2023). These methods can be interpreted as placing constraints on the parameters under a specific matrix norm.

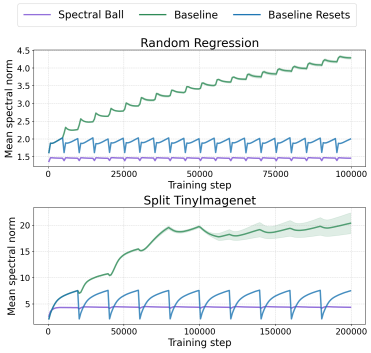

Figure 2: Average spectral norm throughout training. There is a marked increase for the baseline.

We find that constraining parameter matrices to norm balls for different choices of norms can be effective for mitigating loss of generalization if the spectral norm is suitably bounded. A careful analysis using width scaling and norm sensitivity curves identifies the spectral norm as the proper choice to constrain.

Given the importance of the spectral norm, we propose two new parameterizations of the weight matrices that allow direct control over its magnitude: the *spectral norm parameterization* (SNP) and the *singular value parameterization* (SVP). In addition to the spectral norm, the *singular value parameterization* (SVP) explicitly represents all the singular values, which enables more flexible constraints on the entire spectrum. We find that these parameterizations naturally lead to less spectral norm growth and are useful when the optimal spectral norm magnitude are unknown, as in practice.

To summarize the contributions:

- We first identify an experimental setting where loss of generalization is present without loss of trainability and posit this is due to the network entering the lazy regime.
- We derive a novel theoretical connection between large spectral norms of parameters and the lazy (kernel) regime, which we term the *spectral lazy regime*. Furthermore, we prove a bound revealing that updates to matrices are biased to increase the spectral norm.
- We empirically show that bounding spectral norms prevents loss of generalization and investigate advantages of the spectral norm over other norms.
- Finally, we introduce the *spectral norm parameterization* (SNP) and *singular value parameterization* (SVP) as reparameterizations of weight matrices allowing better learning of the spectral norm through gradient-based optimizers.

2 The Spectral Lazy Regime

In this section, we identify a novel mechanism that drives neural networks into the lazy regime. Classically, the lazy regime is induced either by sending the width of hidden layers to infinity – often utilizing specific parameterizations like the neural tangent parameterization (Jacot et al., 2018) – or by uniformly scaling up all weights within a network (Chizat et al., 2019). In this section, we demonstrate that a large spectral norm alone is sufficient to induce lazy training dynamics, even if this magnitude stems from only a single large singular value in each parameter matrix. Specifically, we establish that when the parameter matrices exhibit a large spectral norm, the evolution of the gradient flow for the true model matches that of its linearized counterpart.

Our analysis directly mirrors the qualitative realities of continual learning. To explain why neural networks inevitably fall into this trap, we establish theoretically that standard, unconstrained training naturally exerts an upward pressure on the spectral norm. Because of this inherent growth, a model facing a sequence of tasks eventually initializes with a large spectral norm. This accumulated scale acts as a structural pathology, forcing the network into the lazy regime and ultimately stripping it of its ability to adapt its representations to new data.

To study these dynamics, we focus on MLPs utilizing smooth, ReLU-like activations such as GELU (Hendrycks & Gimpel, 2016) or Swish (Ramachandran et al., 2017). This approach enables a more fine-grained analysis than Chizat et al. (2019), who approached the lazy regime by scaling the output of generic differentiable functions. Because output scaling only translates to weight scaling in strictly homogeneous networks, applying their exact framework to smooth variants introduces a theoretical mismatch. We bridge this gap by analyzing the layer-wise properties of the network, rooting our theoretical guarantees in the large parameter magnitudes observed in practice rather than relying on structural approximations. The following formalizes this network structure and lays the groundwork for analyzing its gradient flow.

2.1 Setup and Notation

Network Formulation To understand the propagation of these large singular values, we consider an L-layer MLP without biases, defined as:

$$f(W; x) = W_L\, \sigma\big(W_{L-1}\, \sigma(\cdots \sigma(W_1 x) \cdots)\big),$$

where the network parameters are given by $W = (W_1, \ldots, W_L)$ with $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$, input dimension $n_0 = d$, and output dimension $n_L = k$. The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is applied coordinate-wise. We are given a finite training set

$$\mathcal{D} \;=\; \{(x_n, y_n)\}_{n=1}^{N}, \qquad x_n \in \mathbb{R}^d,\; y_n \in \mathbb{R}^k,\; \|x_n\|_2 \le R_x.$$

Tracking the forward pass on each example, we define the inputs as $a_0(x_n) := x_n$, the pre-activations as $z_i(x_n) := W_i a_{i-1}(x_n)$, and the post-activations as $a_i(x_n) := \sigma(z_i(x_n))$ for layers $i = 1, \ldots, L-1$, culminating in the output $f(W; x_n) = z_L(x_n) = W_L a_{L-1}(x_n)$.

Large Spectral Norm Initialization To model a network that has already trained on several tasks, we assume its parameter matrices have developed a dominant singular subspace with large singular values. Fixing a per-layer rank $k_i \in \{1, \ldots, \min(n_i, n_{i-1})\}$, we choose orthonormal matrices $U_i \in \mathbb{R}^{n_i \times k_i}$ and $V_i \in \mathbb{R}^{n_{i-1} \times k_i}$. We then define the initial weight matrix as:

$$W_i\big|_{t=0} = \eta\, U_i V_i^\top + R_i, \qquad \eta \gg 1, \tag{1}$$

where η is the dominant scale factor and $R_i \in \mathbb{R}^{n_i \times n_{i-1}}$ is a residual component bounded by $\|R_i\|_{\text{op}} = O(1)$, whose dominant singular vectors are orthogonal to the columns of U_i and V_i. While our primary analysis treats the case of a single large singular value, this framework naturally extends to higher-dimensional top singular subspaces, including full-rank formulations akin to output scaling (Chizat et al., 2019).

For this high-spectral norm initialization to reliably lock the network into the lazy regime, the architecture and objective function must satisfy three basic assumptions:

Assumption 1 (Smooth ReLU-like activation)**.** The activation σ has continuous second derivative, with σ' bounded and σ'' bounded. Moreover, $\sigma(z) = z \cdot \mathbf{1}\{z > 0\} + \xi(z)$ where $|\xi(z)|, |\xi'(z)|, |\xi''(z)|$ decay rapidly as $|z| \to \infty$: specifically, $|\xi''(z)| \le C_\sigma e^{-|z|/C_\sigma}$ for some constant $C_\sigma > 0$. As examples, GELU, swish, and softplus satisfy this assumption.

Assumption 2 (Smooth loss)**.** The loss $\ell : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}_{\ge 0}$ is C^2 in its first argument. For each example, define the *loss residual*

$$r_t(x_n) := \nabla_u \ell(h(W_t; x_n), y_n) \in \mathbb{R}^k.$$

There exist constants $G, M > 0$, independent of η, such that

$$\sup_{t\in[0,T]} \max_n \|r_t(x_n)\|_2 \le G, \qquad \sup_{u\in\mathbb{R}^k} \left\|\nabla_u^2 \ell(u, y)\right\|_{\text{op}} \le M \quad \text{for all } y.$$

For example, the squared loss $\ell(u, y) = \frac{1}{2}\|u - y\|_2^2$ satisfies this with $r_t(x_n) = h(W_t; x_n) - y_n$, $G = \left(\sum_{n=1}^N \|y_n\|_2^2\right)^{1/2}$, and $M = 1$.

Assumption 3 (Pre-activation size)**.** The pre-activations are bounded away from zero over the dataset: there exists $m > 0$, independent of η, such that

$$\min_{n\in\{1,\dots,N\}} \min_{i\in\{1,\dots,L-1\},\, j\in\{1,\dots,n_i\}} |\tilde{z}_{i,j}(x_n)| \ \ge\ m.$$

This last assumption is optional. When it holds, it gives a stronger laziness guarantee with better agreement between the linearized dynamics and the true dynamics. The assumption is used to bound the size of the second derivative of the activation function when applied to pre-activations and has the net effect of bounding the size of the Hessian of the network.

Centering and Gradient Flow Dynamics While parameter matrices can accumulate large spectral norms, the network's output magnitude remains bounded. To reflect this, and matching the protocol of Chizat et al. (2019), we center the network at its initialization W_0: for each x_n,

$$h(W; x_n) := f(W; x_n) - f(W_0; x_n), \qquad h(W_0; x_n) = 0. \tag{2}$$

This centering operation preserves the exact gradient ($\nabla_W h \equiv \nabla_W f$) and Hessian ($\nabla_W^2 h \equiv \nabla_W^2 f$) landscapes. We note that this condition may be substituted with different settings on the weights (see Appendix A.3).

We define the empirical objective $\mathcal{L}(W) := \frac{1}{N}\sum_{n=1}^N \ell(h(W; x_n), y_n)$ and observe its gradient flow. Crucially, we introduce a time-rescaling factor $\alpha = \eta^{-2(L-1)}$ to ensure that the outputs $h(W_t; x_n)$ evolve at most at unit speed, entirely independent of the weight scale (see Chizat et al. (2019)). The true network evolves according to:

$$\dot{W}_t = -\alpha\, \nabla_W \mathcal{L}(W_t) = -\frac{\alpha}{N} \sum_{n=1}^N J_t(x_n)^\top r_t(x_n), \tag{3}$$

where $J_t(x_n) := \nabla_W h(W_t; x_n)$ is the per-example Jacobian and $r_t(x_n)$ is from Assumption 2. We compare this to the linearized objective $\bar{\mathcal{L}}(W) := \frac{1}{N}\sum_{n=1}^N \ell(\bar{h}(W; x_n), y_n)$ based on the first-order

Taylor expansion $\bar{h}(W; x_n) := \langle J_0(x_n), W - W_0 \rangle$, which follows the flow:

$$\dot{\bar{W}}_t = -\frac{\alpha}{N} \sum_{n=1}^{N} J_0(x_n)^\top \bar{r}_t(x_n) \text{ where } \bar{r}_t(x_n) := \nabla_u \ell(\bar{h}(\bar{W}_t; x_n), y_n) \text{ and } \bar{W}_0 = W_0 \,. \quad (4)$$

2.2 Equivalence to Linearized Dynamics

With the structural setup defined, we formalize the network's transition into the lazy regime.

Theorem 1 (Lazy training). *Under Assumptions 1–3, fix any time horizon $T < \infty$. There exists $\eta_0 = \eta_0(T, L, m, G, M, R, R_x, N, \sigma, \mathcal{D})$ (dependence on problem-dependent constants) such that for all $\eta \geq \eta_0$, the following bounds hold, where the rate in parentheses is the one obtained under Assumption 3 (Pre-activation size), denoted (A3):*

$$\sup_{t \in [0,T]} \|W_t - W_0\|_F = O(\eta^{-(L-1)}) \quad (5)$$

$$\sup_{t \in [0,T]} \max_{n \in \{1,\dots,N\}} \left\|h(W_t; x_n) - \bar{h}(\bar{W}_t; x_n)\right\|_2 = O(\eta^{-1}) \quad \big(O(\eta^{-L}) \textit{ with (A3)}\big), \quad (6)$$

$$\sup_{t \in [0,T]} \left\|W_t - \bar{W}_t\right\|_F = O(\eta^{-L}) \quad \big(O(\eta^{-(2L-1)}) \textit{ with (A3)}\big). \quad (7)$$

This theorem should be interpreted as a qualitative result. When the spectral norm is sufficiently large, we should expect the learned network to match its linearization at initialization. This means that the training evolves as in the lazy regime, where the NTK hardly changes. Since the NTK is closely related to the learned representation and inductive bias (Jacot et al., 2018), a frozen NTK would indicate no representation learning. This result does not give precise finite bounds on how large η should be to induce lazy behaviour due to problem-dependent constants folded into the asymptotic rate but indicates this is the expected behaviour when weights grow. In a later section, we experiment with deliberately inducing the lazy regime by increasing the spectral norm to study its effect on continual training.

2.3 Updates tend to increase spectral norm

We now present a complementary result showing that, when updating weight matrices, there is a tendency to increase the spectral norm due to a mis-alignment between updates and the weight matrix's top singular vectors. This result supports the idea that training neural networks naturally increases the spectral norm, pushing them further towards the spectral lazy regime.

Theorem 2 (Spectral norm increase). *Consider a dense parameter matrix W and a perturbation G (which we interpret as the update) with $W' = W + G$ be the updated values. Let u_i and v_i denote the i-th left and right singular vectors respectively. Define $\beta = u_i^\top G v_i$ and $\beta_i^\perp = (I - u_i u_i^\top)(G v_i)$, representing the components of $G v_i$ aligned with u_i and in the orthogonal component of u_i respectively.*

Then, the updated spectral norm is lower bounded:

$$\sigma_1(W') \geq \max_i \sqrt{(\sigma_i(W) + \beta_i)^2 + ||\beta_i^\perp||^2} \geq \sqrt{(\sigma_1(W) + \beta_1)^2 + ||\beta_1^\perp||^2}$$

where $\sigma_i(W)$ denotes the i-th singular value and $\sigma_1(W)$ is the spectral norm of W.

To interpret the bound, if we focus on the second inequality for simplicity, it is most relevant if the top singular value is unique and well-separated from the others. Theorem 2 shows that the spectral norm is influenced by two terms. Only the first term, $(\sigma_1(W) + \beta)^2$, can reduce the spectral norm if β is negative. The second term $||B^\perp||^2$ is always positive, so it can only increase the bound. This second term represents the component of the gradient that is not aligned with u_1. As such, if there is any noise in the update, we can expect an orthogonal component to exist, which contributes to pushing up the spectral norm (see Appendix D for the proof and more explanations).

3 How to control the spectral norm?

Our findings showed that controlling the spectral norm is key to preventing entering the spectral lazy regime and suffering from loss of generalization. We outline several methods of accomplish this.

3.1 Direct norm constraints

One approach is to constrain the maximum spectral norm throughout training. Since we cannot directly access the spectral norm of a matrix and computing the SVD frequently is too expensive, we turn to an alternative. While several options exist (Yoshida & Miyato, 2017; Bjorck et al., 2021; Lewandowski et al., 2023), we choose the spectral cap (Newhouse et al., 2025) method as it is the only computationally efficient method to project the matrix to the spectral ball of a specified radius. This algorithm pairs well with the Muon optimizer (Jordan et al., 2024) since it can be viewed as steepest descent under the spectral norm, with constant-sized updates in spectral norm. Spectral cap can only guarantee a bound when we know a bound on the update size in spectral norm.

Constraining the spectral norm with other norms.
As an alternative, we can opt to bound a different norm for each parameter matrix. Since other norms can upper bound the spectral norm (with a certain dimension-dependent factor), this is also effective. In particular, for a norm $||.||_\Delta$ we would have $||W||_{sp} \leq f_\Delta(d)||W||_\Delta$ using its scaling factor f_Δ (see Table 1). We consider the following three alternative norms:

Norm type	Scale
Frobenius	1
Max	d
Oblique	$\sqrt{d}$

Table 1: Scale to upper bound spectral norm by each type for a $d \times d$ matrix.

Oblique norm. This constraint ensures each row of the parameter matrix lies within an ℓ_2-norm ball of radius R. It corresponds to constraining the $\ell_{2\to\infty}$ operator norm, defined as the the maximum ℓ_2 norm over the rows of the matrix. A similar constraint was used in previous works to normalize the input weights of each neuron (Lee et al., 2025; Karras et al., 2024). These works set the radius R to be 1 and only allowed row norms of exactly 1 (this corresponds to constraining the parameter matrix to the oblique manifold). We allow the realized norms to be less than R during training. Preliminary experiments showed that this ball constraint was better than the manifold version. More details of the differences in Appendix C.3.

Max norm. This constraint clips every entry of the weight matrix so their absolute value is bounded. This is equivalent to bounding the $\ell_{1\to\infty}$ operator norm of the matrix and was previously used to preserve plasticity under the name *weight clipping* (Elsayed et al., 2024).

Frobenius norm. This constraint bounds the Frobenius norm of each parameter matrix. This corresponds to a bound on the ℓ_2-norm on the flattened matrix, equivalent to classic ℓ_2-regularization (Hastie, 2009) and closely related to $L2$-init regularizer (Kumar et al., 2023).

3.2 Spectral Norm and Singular Value Parameterizations

While constraints on the spectral norm can limit the maximum value, this does not address growth within the constraint set. More specifically, Theorem 2 points at an issue: the spectral norm tends to increase in standard matrices due to an update's orthogonal components to the top singular vector. This result is due to the inherent geometry of the space of matrices. To avoid this problem, we can design different parameterizations which explicitly separate the spectral norm from the matrix-structured weights. We consider two such parameterizations.

Spectral Norm Parameterization (SNP). First, we consider the following reparameterization of a matrix W: $W = s\widetilde{W}$ where s is a scalar and $\widetilde{W}$ is constrained to the spectral ball of unit norm.

Given the tendency of increasing spectral norms, we expect $\widetilde{W}$ to achieve its upper bound and have a spectral norm of 1 throughout training. In that case, s indeed represents the spectral norm of the combined $W = s\widetilde{W}$ matrix.
Under this parameterization, updates to $\widetilde{W}$ cannot increase its spectral norm past 1, eliminating the enlargening effect of orthogonal update components from Theorem 2. The only way to increase the spectral norm is through s which gets updated with standard gradient-based optimizers. This separation means that the spectral norm should only increase when the gradient truly indicates it, without incidental increases as a side-effect.

Singular Value Parameterization (SVP). Second, we consider a parameterization akin to the singular value decomposition, enabling control over the entire singular value spectrum. It retains the previous benefit of explicitly parameterizing the spectral norm and may have other advantages. For example, it may be beneficial to ensure that the minimum singular value is not too small to ensure

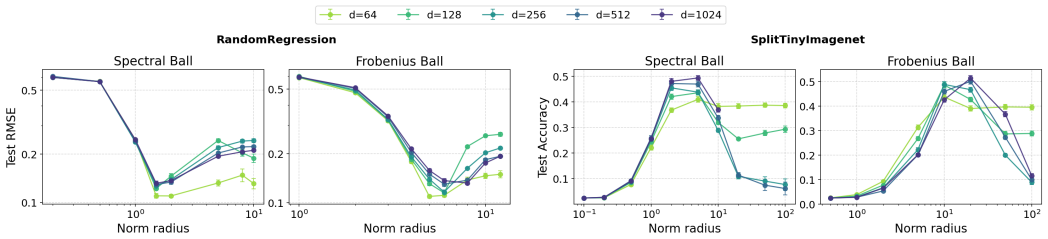

Figure 3: Sensitivity curves for the radius using spectral norm and Frobenius norm balls. Rescaling is not needed for Frobenius norm since it bounds the spectral norm with a scale factor of 1. We see that the curves' overall shapes align with different widths.

proper gradient propagation in all directions (Lewandowski et al., 2024a; Chung et al., 2024; Kastner et al., 2026) or avoid rank deficiency (Joudaki et al., 2025; Kumar et al., 2020).

With SVP, each standard weight matrix W is expressed as the product of three matrices ASB. We optimize directly on these matrices (A, S, B) keeping Stiefel manifold (orthogonal) constraints on A and B with S being a vector of nonnegative scaling factors (additional details for manifold optimization in Appendix C.3). S holds the singular values of the product $W = ASB$ and ASB is a valid SVD of W although it may not be unique. This allows direct control on the singular values and avoids the spectral norm increases as does SNP (more details in Appendix C.2).

4 Experiments

4.1 Experimental setup

Datasets. We consider two classes of datasets. We used randomized mappings as a synthetic task where we can carefully study the phenomenon at hand and also consider a setting with more natural data using a sequence of TinyImageNet tasks.
For the random mapping dataset, we define a regression task from $\mathbb{R}^{10} \to \mathbb{R}^{10}$. This is done by generating inputs from the sphere of RMSNorm equal to 1 (ℓ_2-norm is equal to $\sqrt{10}$). A random three layer norm-preserving neural network is applied to the inputs. We generate 1000 training examples and 500 test examples. On each task, the random mapping is re-generated while the training and test inputs are maintained.
The TinyImageNet setup (similar to Joudaki et al. (2025)) splits the dataset into 5 groups of 40 classes (20000 training examples). These are trained on in a sequence. This is repeated on a new random split of classes for a total of 10 tasks, each for 20k updates. For both datasets, the model is unaware of task changes.

Network architectures and optimizers. For the random regression experiments, we consider a standard MLP and a Resnet-style network using residual connections and LayerNorm (Ba et al., 2016). For the image dataset, we use a small convnet. We run each task for 5k/20k updates with minibatches of size 128 for regression/image classification. This is enough steps to achieve a low error on the first task. We mainly consider the Muon optimizer paired the Lion optimizer for vector-valued parameters. Experiments with Adam can be found in the appendix. We apply the parameter constraints to all layers. Generally, we run 5 seeds for every combination of hyperparameters. Other experimental details can be found in Appendix C.

4.2 Does bounding the spectral norm prevent loss of generalization?

We first examine the effect of training with constraints on the spectral norm of the weight matrices.
In Figure 4, we see that none of the radius values leads to loss of trainability as the train error on the first and last tasks are similar. For test error, it is critical to choose a bounding radius that is not too large, as those suffer from loss of generalization. Small radii ($r \leq 1$) cannot learn at all as indicated by high training loss. The training error is low on all tasks when the radius is sufficiently large.

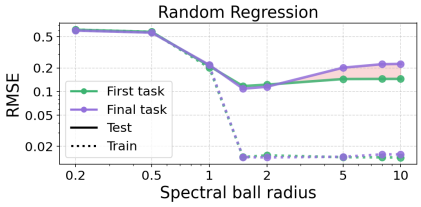

Figure 4: Sensitivity curve for the spectral ball radius. Larger radii suffer from loss of generalization (shaded area).

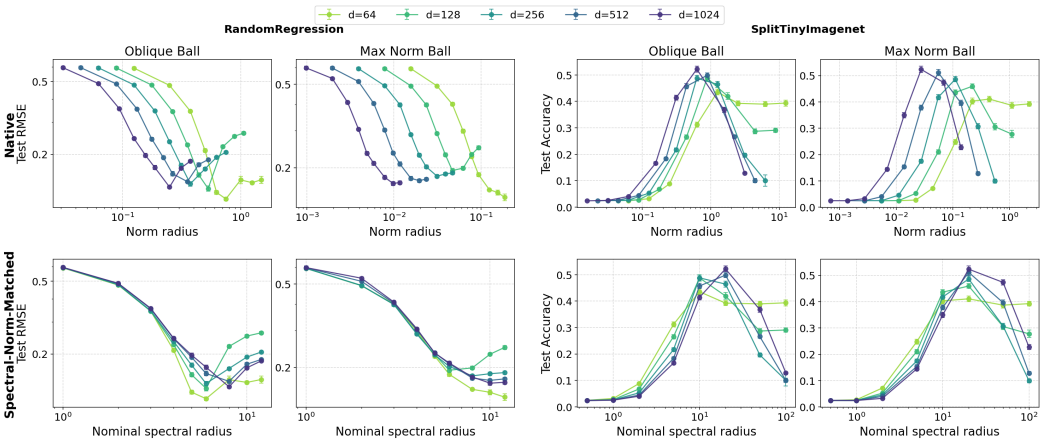

Figure 5: Sensitivity curves showing the final test performance at the end of the task sequence for oblique norm and max norm constraints. The x-axis denotes the norm value. Each constraint type is bounded according to a different norm. (Top row) The x-axis denotes the value of the norm for that constraint type. (Bottom row) The x-axis denotes the rescaled radius after converting to spectral norm. With rescaling, the curves align better, suggesting the spectral norm is the correct dimension-independent choice. See text for rescaling details.

4.3 Is the spectral norm the correct norm?

A confounding factor is that the spectral norm can be upper bounded by other matrix norms as outlined in §3.1. It is possible that bounding a different norm would be sufficient instead. To distinguish between the norms, we can utilize the the fact that they differ by a scale $f_\Delta(d)$. Thus, by taking a single radius for a particular norm and varying the widths of the parameter layers, we can observe if the performance changes. The correct choice of norm should be invariant to the change of width whereas incorrect choices will have variable results. In other words, the radius sensitivity curve for the correct norm should stay the same shape across widths.

The max norm and oblique norm have a dimension-dependent bound to the spectral norm (see Table 1) so we can expect a difference. Inspecting Figure 5 (top row), we see that the curves shift with the dimension. Once we rescale by dividing by the appropriate scale factor, we get a bound on the spectral norm. With this rescaling, Figure 5 (bottom row) shows the sensitivity curves align much better. This provides evidence that the spectral norm is the correct metric to measure and bound. For spectral and Frobenius norms, the curves already align without rescaling (see Figure 3).

Inspecting the realized spectral norms, we also find that using the spectral ball leads to smaller realized spectral norms compared to other ball constraints. Curiously, for some datasets the optimal realized spectral norm is very similar across different norm choices, suggesting an intrinsic spectral norm size for the target function. See Appendix B.1 and B.2 for more discussion.

4.4 Learning the spectral norm through gradient descent

Through the proposed weight matrix reparameterizations, SVP and SNP, the spectral norm is directly represented. This makes it amenable to updates with standard optimizers and avoids the issue that Theorem 2 exhibits. As such, we expect that this would lead to smaller spectral norms. Figure 6 (top row) verifies this claim and, indeed, we find that using SNP or SVP leads to stabilization at a smaller spectral norm than the standard unconstrained matrix. Moreover, when adding a maximum bound, we find that SNP and SVP are less sensitive to this radius when compared to spectral ball constraints. On SplitTinyImageNet, the results are better at all radii.

4.5 Constraining the singular spectrum

In addition to representing the top singular value, the SVP allows more fine-grained control over the entire spectrum. We can directly impose any constraints we wish on all the singular values. This

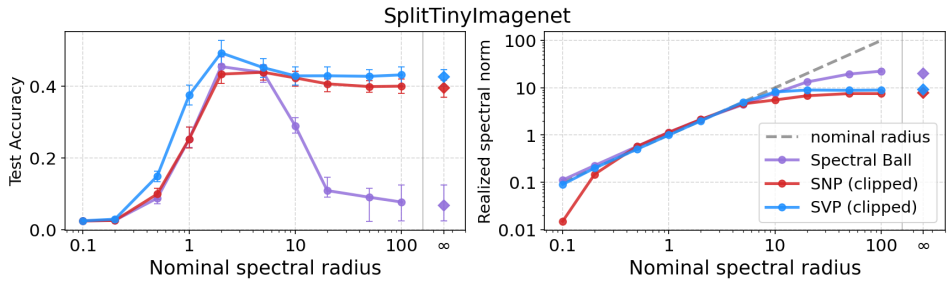

Figure 6: (Left) Performance vs. the constraint radius (∞ denotes no bound). SNP and SVP can offer better results and are less sensitive to the bound radius compared to the standard parameterization with spectral ball constraint. (Right) Comparing realized vs. bound radius for spectral ball and proposed parameterizations.. We see that both SNP and SVP stabilize at a lower spectral norm when there is no bound.

includes certain constraints that may be expensive to do with the standard parameterization, such as bounding the condition number (ratio of largest to lowest singular values) or the nuclear norm. Table 2 summarizes the results. Overall, using additional constraints on the singular values is beneficial. In particular, normalizing the singular values to have mean 1 and variance 1 performs well in both datasets. This constraint is difficult to implement on standard matrices without forming expensive SVDs.

Constraint type	RandomReg RMSE	SplitTinyImg Accuracy
None (base SVP)	0.194	0.423
Frobenius ball	**0.145**	0.489
Normalize mean/var	*0.149*	**0.501**
Condition-number bound	0.162	0.421
Nuclear-norm ball	0.173	*0.498*
Clip min value	0.190	0.411

Table 2: Final test performance per SVP constraint type. Adding constraints generally improves performance.

4.6 Additional Investigations

Learning rate sensitivity analysis. The μP parameterization offers the promise of learning rate transfer (Yang et al., 2022) across widths too, greatly simplifying scaling to larger models without retuning hyperparameters. We investigate this transfer in conjunction with the norm constraints. We find that, with the spectral ball constraint we can also obtain joint hyperparameter transfer of both the optimal learning rate and optimal spectral radius across widths. See Figure 9 in the appendix.

Weight decay and cautious weight decay. Decoupled weight decay (Loshchilov & Hutter, 2017) is a standard method for limiting the magnitude of weights. When combined with Muon, it also produces an implicit constraint on the maximum spectral norm equal to $\frac{1}{\lambda}$ where λ is the regularization coefficient. We find that this can also be an effective method, similar to spectral ball projection. Cautious weight decay (Chen et al., 2026) is a variant that does not produce a hard constraint but we find it to be just as effective and less sensitive to λ. See appendix B.7 for details.

Inducing the lazy regime at initialization. We find that training in the lazy regime can reduce loss of plasticity (for trainability and generalization) but at the cost of worst overall performance on each task. This is sensible as the lazy regime has simplified training dynamics similar to optimizing a linear function, where it is known that plasticity is preserved (Lewandowski et al., 2024b). See appendix B.4.

5 Conclusion

In this work, we focused on loss of generalization in continual learning and linked the phenomenon to large spectral norms with a theoretical result: the spectral lazy regime. Deeper investigations into the spectral norm identified its key role in this effect and the proposed parameterizations SNP and SVP offer a simple solution to controlling the spectral norm. This work suggests the importance of the singular value spectrum and further investigations into its properties, beyond the maximal singular value, could lead to important insights about the learning dynamics of neural networks.

AI USE STATEMENT

In this work, we used generative AI tools for helping prove and refine mathematical claims. We have not used generative AI tools for generating data, research methodology, translation, or interpreting results. Additionally, we used generative AI tools for creating figures, searching the existing literature, brainstorming and assisting coding. We have reviewed all AI-assisted work. The code was tested for correctness, all mathematical proofs and claims were inspected. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on lifelong learning agents*, pp. 620–636. PMLR, 2023.

Noah Amsel, David Persson, Christopher Musco, and Robert M Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm. *arXiv preprint arXiv:2505.16932*, 2025.

Alexandre Araujo, Aaron Havens, Blaise Delattre, Alexandre Allauzen, and Bin Hu. A unified algebraic perspective on lipschitz neural networks. *arXiv preprint arXiv:2303.03169*, 2023.

Jordan T Ash and Ryan P Adams. On the difficulty of warm-starting neural network training. 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31, 2018.

Jeremy Bernstein. Modular manifolds. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250926. https://thinkingmachines.ai/blog/modular-manifolds/.

Nils Bjorck, Carla P Gomes, and Kilian Q Weinberger. Towards deeper deep reinforcement learning with spectral normalization. *Advances in neural information processing systems*, 34:8242–8255, 2021.

N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(42):1455–1459, 2014. URL `https://www.manopt.org`.

Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023. doi: 10.1017/9781009166164. URL `https://www.nicolasboumal.net/book`.

Lizhang Chen, Bo Liu, Kaizhao Liang, and Qiang Liu. Lion secretly solves a constrained optimization: As lyapunov predicts. In *International Conference on Learning Representations*, volume 2024, pp. 35404–35439, 2024.

Lizhang Chen, Jonathan Li, Kaizhao Liang, Baiyu Su, Cong Xie, Chen Liang, Ni Lao, and Qiang Liu. Cautious weight decay. In *International Conference on Learning Representations*, volume 2026, pp. 126743–126774, 2026.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

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

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pp. 854–863. PMLR, 2017.

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

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

Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020 (11):113301, 2020.

Trevor Hastie. The elements of statistical learning: data mining, inference, and prediction, 2009.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. *arXiv preprint arXiv:2006.05826*, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

Amir Joudaki, Giulia Lanzillotta, Mohammad Samragh Razlighi, Iman Mirzadeh, Keivan Alizadeh, Thomas Hofmann, Mehrdad Farajtabar, and Fartash Faghri. Barriers for learning in an evolving world: Mathematical understanding of loss of plasticity. *arXiv preprint arXiv:2510.00304*, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24174–24184, 2024.

Tyler Kastner, Nimrod De La Vega, and Amir-massoud Farahmand. Singularclip: Preventing spectral collapse to maintain plasticity in continual and reinforcement learning. *arXiv preprint arXiv:2608.18319*, 2026.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. *arXiv preprint arXiv:2010.14498*, 2020.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. *arXiv preprint arXiv:2308.11958*, 2023.

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

Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C Machado. Directions of curvature as an explanation for loss of plasticity. *arXiv preprint arXiv:2312.00246*, 2023.

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

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

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

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. *arXiv preprint arXiv:2204.09560*, 2022.

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

Laker Newhouse, R Preston Hess, Franz Cesista, Andrii Zahorodnii, Jeremy Bernstein, and Phillip Isola. Training transformers with enforced lipschitz constants. *arXiv preprint arXiv:2507.13338*, 2025.

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

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Seyed Roozbeh Razavi Rohani, Khashayar Khajavi, Wesley Chung, Mo Chen, and Sharan Vaswani. Preserving plasticity in continual learning with adaptive linearity injection. *arXiv preprint arXiv:2505.09486*, 2025.

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

Shuo Xie and Zhiyuan Li. Implicit bias of adamw: ℓ_∞ norm constrained optimization. *arXiv preprint arXiv:2404.04454*, 2024.

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

Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

Appendices

Appendix A Other works **14**

A.1 Additional background 14

A.2 Constraint sets 14

A.3 Alternative centering mechanisms 14

Appendix B Additional experiments **15**

B.1 Performance vs. observed spectral norms 15

B.2 Comparing observed spectral norms and their bounds 15

B.3 Learning rate sensitivity and transfer 15

B.4 Lazy linear learning and maintaining plasticity 15

B.5 Methods for bounding the spectral norm 17

B.6 Muon vs. Adam 17

B.7 Weight decay and cautious weight decay 17

Appendix C Experimental details **21**

C.1 General settings 21

C.2 SNP and SVP 21

C.3 Steepest descent on manifolds 21

Appendix D Proof of Theorem 2 (Spectral norm increase) **22**

Appendix E Proof of Theorem 1 **23**

E.1 Conventions 23

E.2 Sizes of the network and its derivatives at W_0 23

E.3 Local stability of the size estimates 24

E.4 Bound on weight motion under the rescaled flow 25

E.5 Linearization comparison 26

A OTHER WORKS

A.1 ADDITIONAL BACKGROUND

Loss of plasticity in continual learning. While loss of plasticity has been observed in various settings, its causes are not yet well understood. Many metrics and explanations have been proposed that are linked to loss of plasticity such as the emergence of dead units or cloned units, increasing parameter norms, Hessian curvature rank decreasing, empirical NTK rank decreases, increasing distances to initialization, parameter matrices having imbalanced singular value spectrums and poor gradient propagation. Many methods have been proposed to try to mitigate loss of plasticity in different settings. Shrink-and-perturb (Ash & Adams, 2019), l2-regularization towards init (Kumar et al., 2023), LayerNorm (Ba et al., 2016; Lyle et al., 2022), CReLU (Abbas et al., 2023), AdaLin (Rohani et al., 2025), Fourier features (Lewandowski et al., 2024b), full or partial weight resets (Nikishin et al., 2022), Parseval regularization(Chung et al., 2024), spectral regularization (Lewandowski et al., 2024a), ReDO (Sokar et al., 2023), Continual Backprop (Dohare et al., 2021) among others. While some of these prior works have discussed trainability vs. generalization at a high level, as far as we are aware, mechanisms to explain loss of generalization when trainability is unimpaired have not been identified.

The lazy and rich regimes of neural network training. Deep neural networks have been favoured in recent times due to their strong representation learning and generalization capabilities. Theorists have sought to understand the process by which neural networks learn representations. Two distinct regimes of training have been uncovered: The lazy (kernel) regime, where neural networks can fit labels but where no feature-learning occurs and the rich regime, where neural networks can successfully develop useful representations. Neural networks trained in the lazy regime consistently underperform networks in the rich regime due to their impaired representation-learning ability. One way to elicit the lazy regime is to consider increasing the width of a hidden layer, with appropriate scaling factors (Jacot et al., 2018). This was shown to be an example of a more general recipe: Increasing the scale of the outputs of a network (with appropriate modifications), including scaling up all the weights of a neural network (Chizat et al., 2019), would be sufficient to enter the lazy regime. The lazy regime is characterized by having large changes in the output function being accompanied by negligible changes in the gradient. In other words, the overall function behaves like a linear function throughout training. The input-output gradient and by extension, the neural tangent kernel, remains fixed all throughout also.

A.2 CONSTRAINT SETS

Stiefel manifold. This choice restricts the parameters to lie on the Stiefel manifold (the space of semi-orthogonal matrices). This is implemented by using Newton-Schulz iterations (Amsel et al., 2025) to orthogonalize the matrix at regular intervals. This constraint has also been used to improve plasticity, using regularization to approximately maintain the constraint (Chung et al., 2024; Cisse et al., 2017).

More about the oblique manifold: We make some improvements to the constraint implementation compared to these two works. SimbaV2 (Lee et al., 2025) did not consider the tangent space of the manifold in the gradient calculations, which would bias the update direction towards an incorrect direction. Karras et al. (2024) uses a reparamterization of the weights as $w/||w||_2$ on each row-weight vector along with renormalizing the weights after every step. The net effect is that the gradient w.r.t. w will be projected to its tangent space, but they do not consider reprojecting the final updates and also do not use vector transport on the momentum vector towards the current tangent space. Neither used vector transport on the momentum, which is done here.

A.3 ALTERNATIVE CENTERING MECHANISMS

Concerning the centering assumption for spectral lazy regime: Instead of subtracting a frozen copy of the initial network, we can impose an anti-symmetric structure (Chizat et al., 2019) on the final two layers. This initialization divides hidden activations from the last hidden layer into two groups. The second group has identical weights to the first group and thus identical outputs. The final layer divides the weights from the last parameter layer into two parts, with one group having weights

equal to the negative of the other. The net effect is the contribution from each pair of hidden units to the output is canceled out and the output is zero for any input to the network.

B Additional Experiments

B.1 Performance vs. observed spectral norms

We can also check the realized spectral norms at the end of training for each norm constraint. Interestingly, from Fig. 7 we see that as we vary the constraint radius, the performance-vs.-realized-spectral-norm curves closely coincide for Split-TinyImagenet. For Random Regression, the realized spectral norm are slightly different across norm choice but spectral ball leading to slightly lower spectral norms and is tied for the lowest error. Again, this could suggest that the spectral norm ball is better choice since the spectral norm is linked to the Lipchitz constant of the network: Lower spectral norms suggest a simpler, more robust solution (Cisse et al., 2017; Araujo et al., 2023).

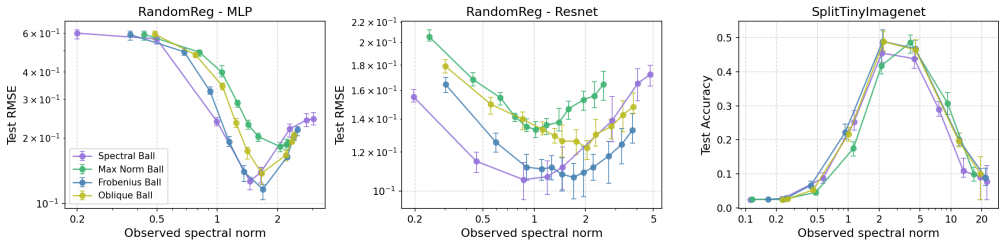

Figure 7: Test performance vs. realized spectral norm at the end of the tasks for RandomReg (MLP and Resnet) and SplitTinyImagenet.

B.2 Comparing observed spectral norms and their bounds

How close are the realized spectral norms to their bound?
For different constraint balls and their radii, we can inspect the realized spectral norm after training and compare to the imposed bound. Even if the maximum allowed spectral norm can be the same by using a different norm and rescaling it, the shape of the constraint set will still be different. This is similar to how, on $\mathbb{R}^3$, the ℓ_∞-norm ball is a sphere and ℓ_2-norm ball is a sphere (including their interiors) even though each can be rescaled to envelop the other.

Looking at Fig. 8, we find that spectral ball constraints leads to realized spectral norms close to the bound whenever the radius is small enough. When the radius is sufficiently loose, the realized spectral norm is smaller than radius, which is expected as the constraint may no longer be active. For the other norm balls, we find a substantial gap between the realized spectral norms and the maximum possible bound. This suggests that the shape of the ball delimited by the spectral norm is a better match than the others. When bounding using the other norms, the parameters do not reach the largest spectral norm allowed under the constraint, even if those are favourable solutions.

B.3 Learning rate sensitivity and transfer

See figure 9. We investigate whether we can also have learning rate transfer across widths. We find that, indeed, the sensitivity surface (2-dimensional since we look at both radius and learning rate simultaneously) is qualitatively similar across widths. The best hyperparameter combinations are in similar areas.

B.4 Lazy linear learning and maintaining plasticity

While the lazy regime is generally undesirable given its negative effects on representation learning, we discuss how the simplified learning dynamics can be beneficial for learning future tasks. In fact, networks trained in the lazy regime may perfectly preserve plasticity by sacrificing representation learning capabilities.

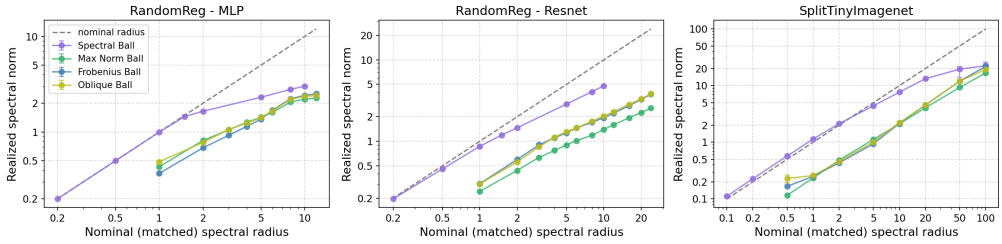

Figure 8: The realized spectral norm vs. the radius of the bound of at the end of training. All points are below the diagonal due to the bound. For spectral ball, the constraint is close to tight for smaller radii while there is always a larger gap for other normed balls.

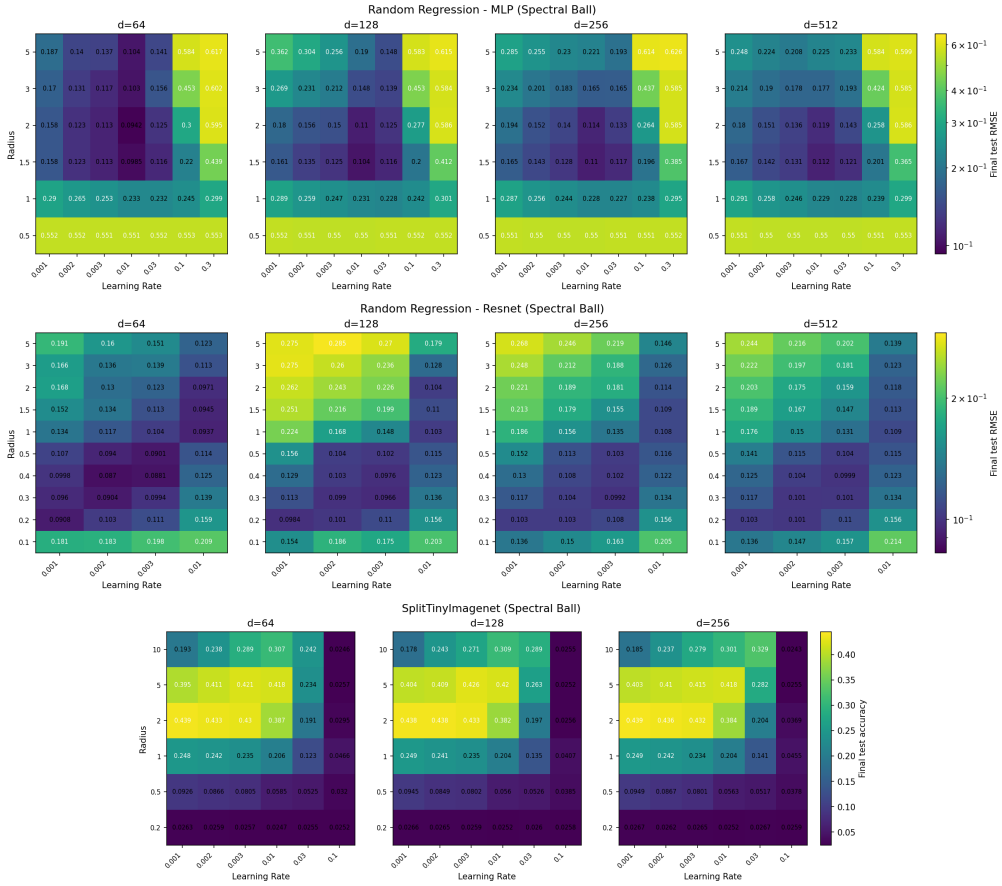

Figure 9: Plots of joint sensitivity of learning rate and radius for spectral ball constraint for different datasets. The overall pattern is very similar across widths, indicating that the best radius and learning rate would transfer across widths.

To understand this idea, we first discuss a scenario where plasticity can be fully preserved, both in terms of trainability and generalization. We can consider a sequence of linear regression tasks, with changing input/target pairs on each task. Since the loss is convex on each task, gradient descent can find a global minimum no matter the initial weights. This has previously been formalized: Linear functions trained on strongly-convex losses do not lose trainability (see (Lewandowski et al., 2024b)). Loss of generalization is also not a concern since this is determined by the fixed features given, which the learned parameters cannot influence. Thus, we can say that plasticity has been fully preserved for linear regression.

In the lazy regime, the neural network weights closely track those from training a linear model with fixed features for each input x given by the gradient $\nabla_w f(w; x)|_{w=w_0}$ at initialization. Due to these simplified learning dynamics, the behaviour is similar to that of the linear regression scenario, where plasticity was perfectly maintained.

Other works showed evidence that linear neural networks, a network without nonlinear activation functions, also preserve plasticity more effectively and adding sources of linearity in a nonlinear network can also improve their plasticity (Abbas et al., 2023; Lewandowski et al., 2024b; Rohani et al., 2025). It's important to note that this type of linearity is different that the one discussed by lazy regime. In these works, the network is made to resemble a *linear neural network*. These networks still define a nonconvex (nonlinear) optimization problem in their parameters but they define a linear function of the inputs. On the other hand, in the lazy regime, the optimization problem is *linear* in the weights but the function is still nonlinear in the inputs (with features defined by gradient of the output w.r.t. weights).

We can try to induce the lazy regime at initialization by scaling up the weights. We can choose how many top singular values to scale up (the spike dim) and by how much to scale it up (spike scale). Fig. 10 and 11 show the results. Overall, the performance is worse as you increase the scaling factor. But we can sometimes observe less loss of plasticity by comparing the performance on the first and last tasks, especially at larger scales. Sometimes an intermediate value can achieve fairly good performance such as scale=2 on the SplitTinyImagenet dataset. There, we can clearly see plasticity loss being mitigated.

B.5 Methods for bounding the spectral norm

There are several methods that can be used to bound the spectral norm. In the main text, we use spectral cap (Newhouse et al., 2025) primarily.

We compare the spectral cap, SpectralNorm rescaling, SNP with clipping and SVP with clipping as described in previous sections.

For SpectralNorm rescaling, we consider a variant of SpectralNorm (Yoshida & Miyato, 2017). In our implementation, we estimate the spectral norm σ_{max} of the weight matrix W using power iteration and divide the parameter matrix by σ_{max}/B whenever σ_{max} exceeds B. The net effect is to rescale all singular values downwards so the maximum is at most B. The original spectral norm maintains an estimate $\hat{\sigma}$ of the spectral norm of the matrix W, then during the forward pass, it computes $\widetilde{W} = W/\hat{\sigma}$ and applies $\widetilde{W}$ to the activations. When the estimate is accurate, $\widetilde{W}$ has spectral norm bounded by 1. But during the backward pass, since we update W and not $\widetilde{W}$, the gradient w.r.t W will get rescaled by $\hat{\sigma}$. Note that this means the true spectral norm of W is not bounded under the original method, but is bounded with our variant.

For our proposed parameterizations SNP and SVP, we can directly clip the top singular value.

B.6 Muon vs. Adam

We repeat the experiments using Adam using different constraints. Overall, we find very similar results as shown in Fig. 12.

We note that, for Adam, spectral cap is not guaranteed to bound the spectral norm because the Adam update can have an update of unbounded spectral norm size. This can occur if the second moment estimates are very small in comparison to the momentum. In the worst case, it can cause divergence of the spectral cap iteration. As a heuristic, we simply use a larger (10x) step size in the spectral cap solver to try to have a conservative (upper bound) on the size of Adam updates. In practice, this was able to avoid divergence.

B.7 Weight decay and cautious weight decay

Weight decay is the traditional method for mitigating large parameter norms. We investigate its effectiveness in these settings. We focus on decoupled weight decay (Loshchilov & Hutter, 2017) as it is the standard method and also experiment with cautious weight decay (Chen et al., 2026), a recently proposed variant which only applies weight decay when the update matches its direction.

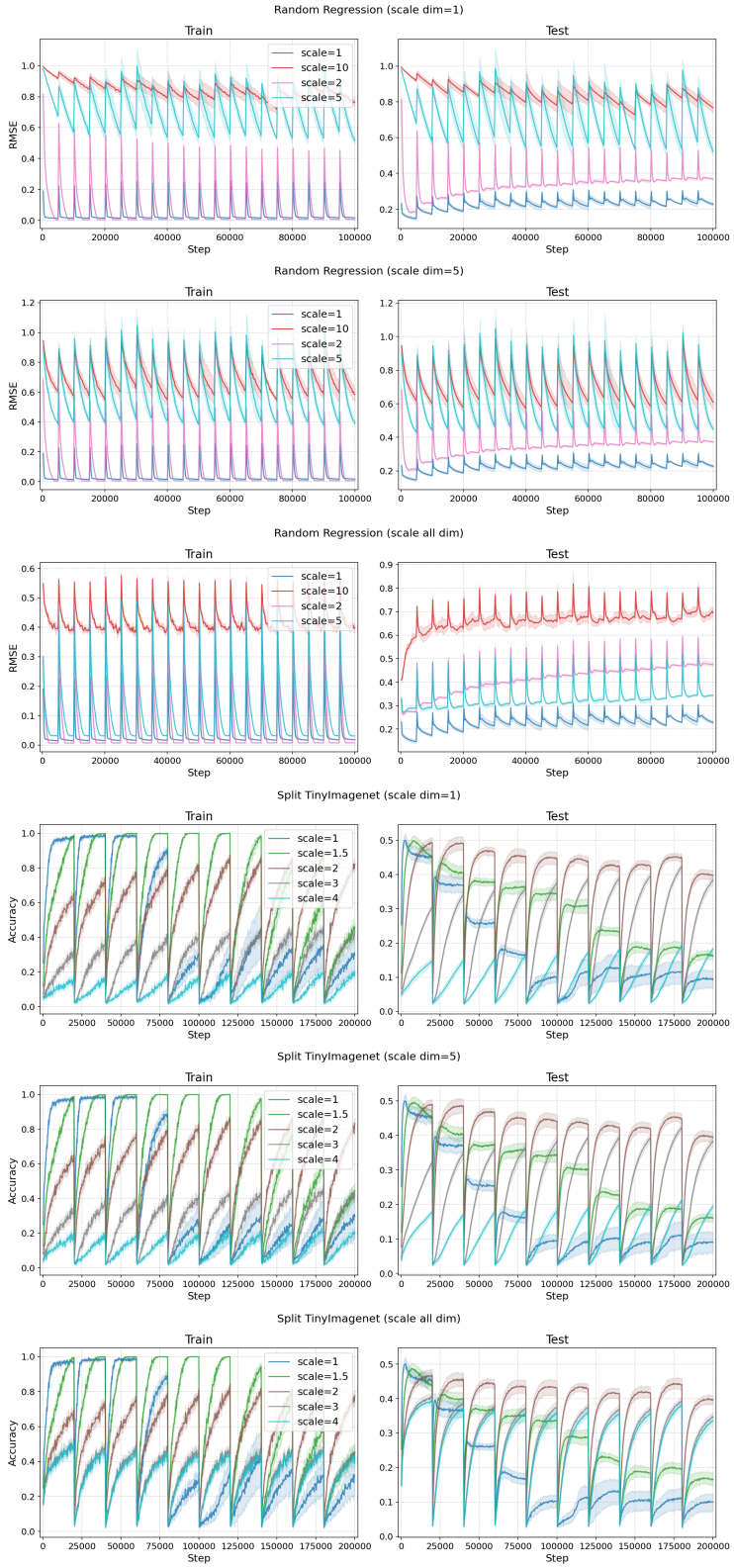

Figure 10: Learning curves with different number of top singular values being multiplied to scale them up. The scaling factor is indicated in the legend.

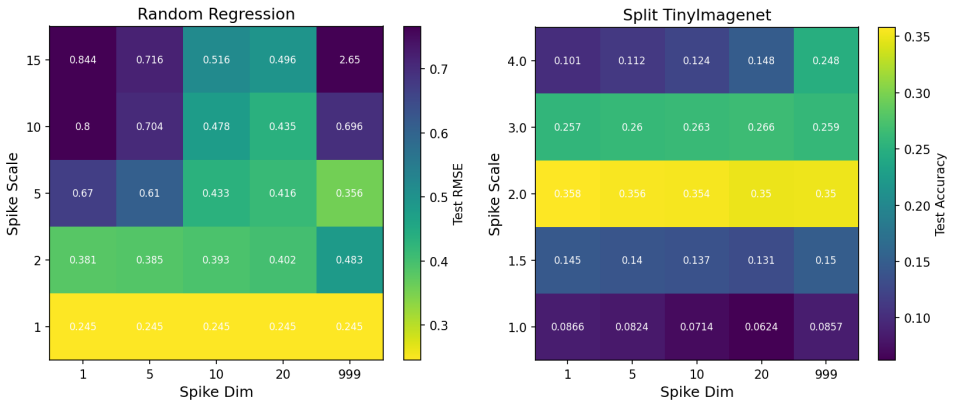

Figure 11: Summarizing results through a heatmap across different number of singular values being scaled (spike dim) and the scaling factor

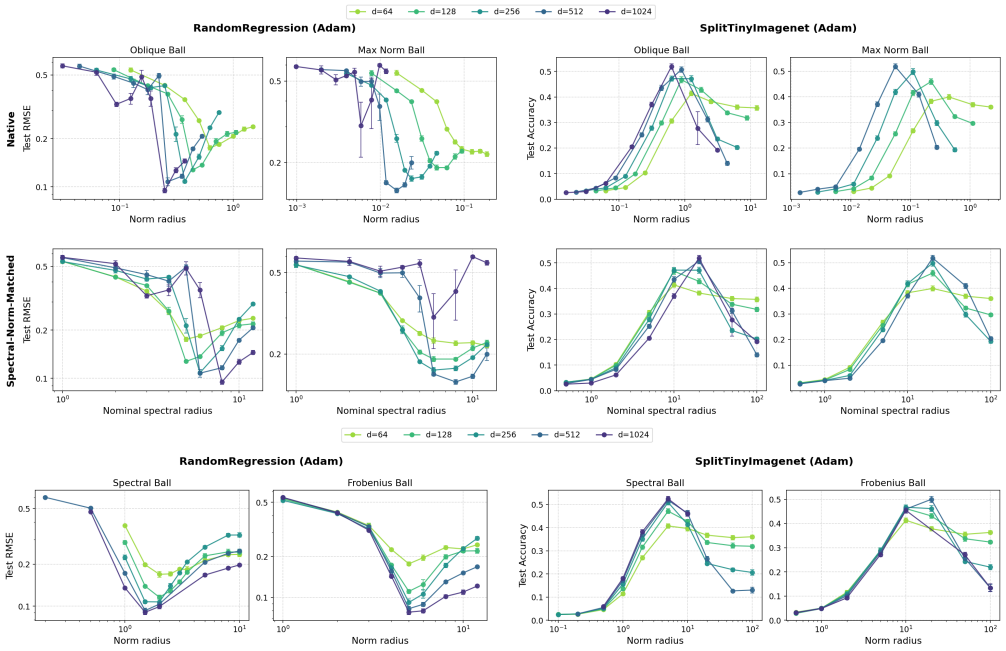

Figure 12: (Same as Fig. 5 and Fig. 3 but using Adam instead of Muon.) We plot the norm radius for the native norm and the spectral-norm-rescaled version. Overall, we find a similar trend as Muon, where the rescaled version aligns the curves better.

Weight decay implies a constraint set but its effect is more subtle: The set depends on the optimizer used (Chen et al., 2024). For example, Muon with weight decay leads to a bound on the spectral norm. Lion with weight decay leads to a bound on the ℓ_∞ (max) norm. Adam does not strictly lead to a hard constraint but we can view Adam as an approximation to sign descent and thus view AdamW as bounding the ℓ_∞ (max) norm (Xie & Li, 2024).

We see from Fig. 13 that weight decay has similarly sensitivity to the and performs best when the spectral norm is about approximately the same as the best for the explicit constraints.

From Fig. 14, we also find that cautious weight decay has less sensitivity to the regularization strength than standard weight decay, with comparable performance at its optimum. Its behaviour when the regularization coefficient is very large differs qualitatively from standard weight decay.

While standard weight decay prevents any learning, overly constraining the spectral norm, cautious weight decay allows larger norms.

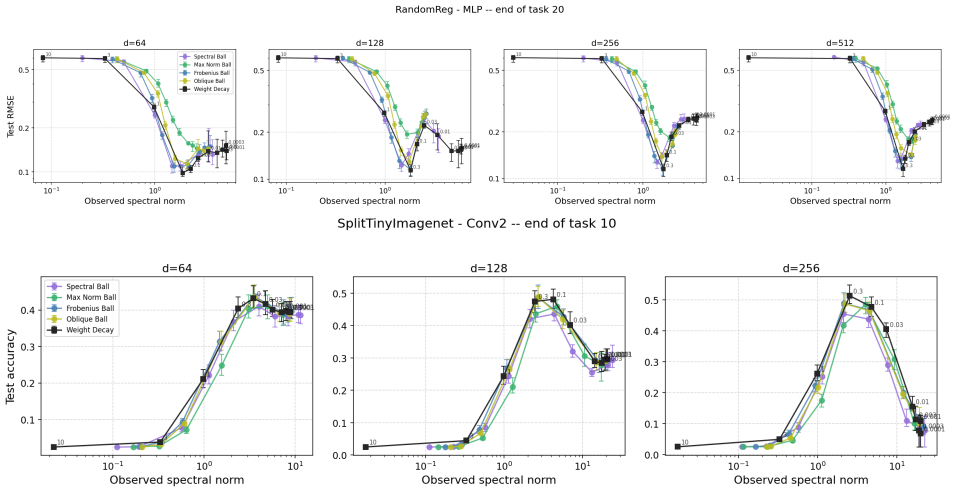

Figure 13: Performance vs. observed spectral norm. These plots include weight decay. It overall has a very similar sensitivity curve as the other constraints.

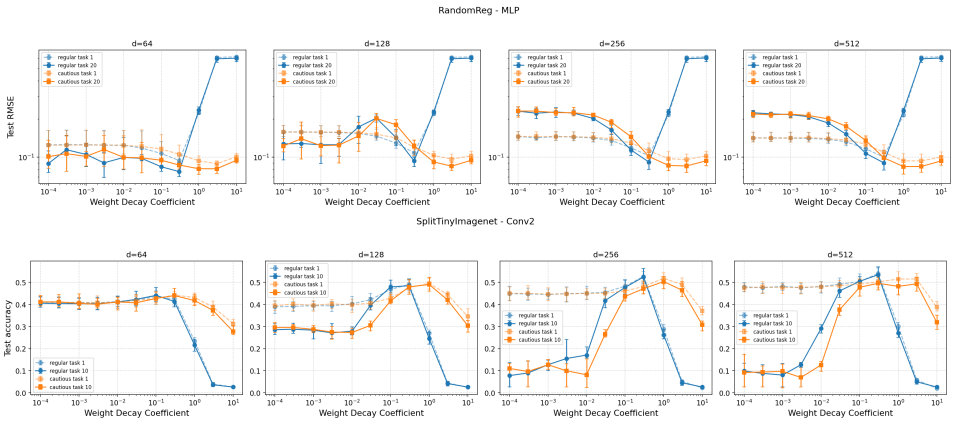

Figure 14: Performance vs. weight decay coefficient for standard weight decay and cautious weight decay. Cautious weight decay is less sensitive to the regularization strength and produces similar performance.

C EXPERIMENTAL DETAILS

C.1 GENERAL SETTINGS

For the learning rates, we do a cursory sweep using for each optimizer/environment/architecture combination. For example, Muon/Random Regression/MLP. For each combination, we choose a constant learning rate where the unconstrained network is able to learn the first task well and fit the training examples. This is kept fixed across all experiments with constraints imposed on the parameters to keep the learning progress comparable. The

Random Regression dataset.
To generate the random norm-preserving neural network we do the following: The input vector with RMSNorm of 1 is multiplied by three layers of random orthogonal matrices and a rotated MaxMin nonlinearity. The outputs are then also on the sphere of RMSNorm equal to 1. We also tried a variant where we re-generated the train inputs for each task but the results were very similar.

Network architectures. For Random Regression, the MLP consists of 4 dense layers. The Resnet used also consists of four layers. Each residual branch has a layernorm followed by dense and the activation function. This is similar to the PreAct Resnet architecture. For SplitTinyImagenet, the network is a convnet consisting of three 3x3 convolutional layers of (32, 64, 128) channels, each followed by max pooling. Then a global average pooling layer is used be before ending with two dense layers.

C.2 SNP AND SVP

Additional details on SVP

With this parameterization, we can directly impose constraints on the singular values, through S. For example, we can clip entries of S to $[0, \sigma_{max}]$ to bound the spectral norm. Other, more complex constraints are also possible such as normalization, nuclear norm and condition number capping. These would not be easily accomplished on the standard parameterization of W.

For convnets, we implement SVP on the kernel of each convolutional layer. This is done by reshaping the kernel parameters from $(n_{out}, n_{in}, height, width)$ into a 2-dimensional matrix of shape $(n_{in}, n_{out} \times height \times width)$ which is factorized as ASB. Under this scheme, we treat the the second dimension as a flattened input dimension to the kernel and the first dimension being the output dimension. A similar approach was used with orthogonal matrices in CNNs (Bansal et al., 2018).

C.3 STEEPEST DESCENT ON MANIFOLDS

To optimize the A and B matrices on the Stiefel manifold, we take into account the smooth manifold structure to stabilize gradient-based optimization. Optimization on manifolds has a rich history with many algorithms (Boumal, 2023; Boumal et al., 2014). We consider a simple procedure consisting of four steps.

Project gradients to tangent space. Since the Stiefel manifold defines a tangent space at every point, we project the (Euclidean) gradient produced by standard autodiff to the tangent space to get the Riemannian gradient. This avoids including components of the gradient pointing away from the manifold, which would only serve as additional noise.

Vector transport of momentum. Since the momentum vectors are accumulated on previous tangent spaces, it would be incorrect to combine the current (projected) gradient with the previous momentum vector. To account for this, we need to transport the previous momentum vector to the current tangent space. This could best be done using parallel transport add cite but this is an expensive operations. Instead, it is common to use vector transport operations as an approximation. To do this, we simply project the previous momentum vector to the current tangent space before adding it with the (projected) gradient vector at every step.

Re-projecting updates to the tangent space. For optimizers such as Adam and Muon, we apply the preconditioner or orthogonalization on the momentum vector, which may cause the result to lave the tangent space. So, we then re-project the result back to the tangent space.

Parameter projection back to the manifold. Despite keeping updates within the tangent space, due to discrete taking discrete steps, the parameters can slowly drift away from the manifold. To account for this, we periodically project back to the Stiefel manifold (every 5 updates in our experiments). This is done by using the Newton-Schulz orthogonalization method (with PolarExpress coefficients) (Jordan et al., 2024; Amsel et al., 2025).

We note that this procedure can be applied to any smooth manifolds. In particular, we make use of it for the Stiefel manifold in SVP but also the Oblique manifold in the experiments. While previous papers such as (Lee et al., 2025; Karras et al., 2024; Bansal et al., 2018; Bernstein, 2025) used some of these components, they did not use every step of this procedure.

D Proof of Theorem 2 (Spectral norm increase)

Proof. We let $\|.\|$ denote the ℓ_2 norm on vectors. By definition of the spectral norm, $\sigma_1(W') = \max_{||v||=1} ||W'v|| \geq ||W'v_i||$ for any singular vector v_i. Then,

$$\begin{aligned} W'v_i &= (W+G)v_i \\ &= Wv_i + Gv_i \\ &= u_i\sigma_i + Gv_i \\ &= u_i\sigma_i + u_iu_i^\top(Gv_i) + (I - u_iu_i^\top)(Gv_i) \\ &= u_i(\sigma_i + u_i^\top Gv_i) + (I - u_iu_i^\top)(Gv_i) \end{aligned}$$

Taking norms and using the fact that both terms are orthogonal to each other,

$$\begin{aligned} ||W'v_i||^2 &= ||(\sigma_i + u_i^\top Gv_i)||^2 + ||(I - u_iu_i^\top)(Gv_i)||^2 \\ &= (\sigma_i + u_i^\top Gv_i)^2 + ||(I - u_iu_i^\top)(Gv_i)||^2 \\ ||W'v_i|| &= \sqrt{(\sigma_i(W) + \beta_i)^2 + ||\beta_i^\perp||^2} \end{aligned}$$

Finally, since $\sigma_1(W') \geq ||W'v_i||$ for any i, it also holds for the maximum, giving us our desired bound.

□

Additional explanations

The $\max_i$ operator in the bound covers the case when there may be several singular values that are equal to the spectral norm or near the maximal value. In that case, the bound can take into account the update's alignment with any of them. If there are multiple singular vectors with equal values, we can also freely replace u_i and v_i by any vector in the subspace associated to the singular value.

In particular, we can apply this bound to standard SGD. Letting the parameter matrix W compute the transformation $y = Wx$ for inputs x and outputs y, we let δ be the derivative of the loss w.r.t. y. Then the gradient descent update is $G = -d\mathcal{L}/dW = -\alpha\delta x^\top$. In this case, we have that $\beta_1 = -\alpha(u_1^\top\delta)(x^\top v_1)$. From this, we glean that the alignment $x^\top v_1$ of the input and the top singular vector contributes one factor to the size of β_1 and $\beta_1^\perp$. The more important difference is how δ aligns with u_1, where the orthogonal component contributes to the second term of the bound, increasing the spectral norm.

E PROOF OF THEOREM 1

We prove Theorem 1 in the setting of Section 2.1: the network f, the dataset $\mathcal{D}$, the initialization equation 1, the centered output h, and the flows equation 3 and equation 4 are as defined there, and so are Assumptions 1–3. Assumption 3 is used only where stated.

E.1 CONVENTIONS

Per-example quantities. Quantities attached to a single example are written as functions of the input, e.g. $z_i(x_n)$, $J_t(x_n)$, $r_t(x_n)$. Coordinates are a second subscript, so $z_{i,j}(x)$ is the j-th coordinate of $z_i(x)$. In proofs where a single example is fixed, the argument is suppressed, and this is stated at the start of the proof. Along the two flows we abbreviate the outputs as $h_t(x) := h(W_t; x)$ and $\bar{h}_t(x) := \bar{h}(\bar{W}_t; x)$, and write $J_0(x_n) := \nabla_W h(W_0; x_n)$ for the Jacobian at initialization.

Asymptotic notation. $A = O(\eta^p)$ means $\|A\|_{\mathrm{op}}$ or $\|A\|_F$ is bounded by $C\eta^p$, with C depending on L, the widths, $\|R_i\|_{\mathrm{op}}$, R_x, G, M, T, N, and the activation constants, but not on η. We write $\|A\|_{\mathrm{op}} := \|A\|_{2\to 2}$ and $\|A\|_F := (\sum_{ij} A_{ij}^2)^{1/2}$. Throughout, $L \geq 2$ and $\eta \geq 1$.

E.2 SIZES OF THE NETWORK AND ITS DERIVATIVES AT W_0

Lemma 3 (Scaling at initialization)**.** *Under Assumptions 1–2, with W_0 as in equation 1, the following hold* uniformly over $n \in \{1, \dots, N\}$*:*

(i) $\|f(W_0; x_n)\| = O(\eta^L)$*;*

(ii) $\|\nabla_W h(W_0; x_n)\|_{\mathrm{F}} = O(\eta^{L-1})$*;*

(iii) Under Assumption 3: $\left\|\nabla_W^2 h(W_0; x_n)\right\|_{\mathrm{op}} = O(\eta^{L-2})$.
Without Assumption 3: $\left\|\nabla_W^2 h(W_0; x_n)\right\|_{\mathrm{op}} = O(\eta^{2L-3})$.

The implicit constants depend on R_x (the data radius) and, in part (iii) under Assumption 3, on m (the pre-activation margin), but not on n.

Proof. Fix any $n \in \{1, \dots, N\}$. For the duration of this proof the example is fixed and its argument is suppressed: we write z_i, a_i, D_i, B_i for $z_i(x_n), a_i(x_n), D_i(x_n), B_i(x_n)$, and $z_{i,j}$ for the j-th coordinate of z_i. Every quantity is evaluated at $W = W_0$. Constants depend on L, widths, $\|R_i\|_{\mathrm{op}}$, R_x, and activation constants (and on m where indicated), but not on η or n (since the bounds below depend on x_n only through $\|x_n\|_2 \leq R_x$). Let $D_i := \mathrm{diag}(\sigma'(z_i))$ and $D_i^{(2)} := \mathrm{diag}(\sigma''(z_i))$ at W_0. Define the backward maps $B_i := W_L D_{L-1} W_{L-1} \cdots W_{i+1} D_i \in \mathbb{R}^{k \times n_i}$ for $i < L$, $B_L := I_k$.

(i) Output size. By induction using $W_i = \eta U_i V_i^\top + R_i$, each forward map multiplies by a matrix of operator norm $O(\eta)$. Also, $\|\sigma'\|_\infty < \infty$, so the activation function only increases the size by at most a constant factor (independent of η) This means $\|a_i\|_2 = O(\eta^i)$ and $\|f(W_0; x_n)\|_2 = O(\eta^L)$ uniformly in n.

Leading-order expansion. The following estimate holds for any weights of the form $W_i = \eta\, U_i V_i^\top + \widetilde{R}_i$ with $\max_i \left\|\widetilde{R}_i\right\|_{\mathrm{op}} \leq \rho$, not only at W_0 (where $\widetilde{R}_i = R_i$); it is reused in Lemma 4. Writing $z_i(W; x_n)$ for the pre-activation under W, there is a constant C_ρ, depending on ρ, L, widths, R_x and σ but not on η or n, such that for all $\eta \geq 1$ and $i = 1, \dots, L-1$,

$$\left\|z_i(W; x_n) - \eta^i\, \tilde{z}_i(x_n)\right\|_\infty \;\leq\; C_\rho\, \eta^{i-1}. \tag{8}$$

For $i = 1$, $z_1 - \eta\, \tilde{z}_1 = \widetilde{R}_1 x_n$ has norm at most ρR_x. For the inductive step, write $\sigma(z) = z_+ + \xi(z)$, where ξ is bounded by Assumption 1. Since $u \mapsto u_+$ is 1-Lipschitz and positively homogeneous, $(\eta^i \tilde{z}_i)_+ = \eta^i \tilde{a}_i$ and so $\left\|a_i - \eta^i \tilde{a}_i\right\|_2 \leq \left\|z_i - \eta^i \tilde{z}_i\right\|_2 + \sqrt{n_i}\, \|\xi\|_\infty = O(\eta^{i-1})$. Then

$$z_{i+1} - \eta^{i+1} \tilde{z}_{i+1} \;=\; \eta\, U_{i+1} V_{i+1}^\top \big(a_i - \eta^i \tilde{a}_i\big) + \widetilde{R}_{i+1}\, a_i \;=\; O(\eta^i),$$

using $\|a_i\|_2 = O(\eta^i)$ and $\left\|\widetilde{R}_{i+1}\right\|_{\text{op}} \le \rho$. Since $\tilde{z}_i(x_n)$ depends only on U_i, V_i, x_n, the leading term in equation 8 is the same for every such W.

(ii) Gradient size. Differentiating the forward pass along $W(t) = W_0 + t\Delta W$ at $t = 0$: $\dot{z}_i = \Delta W_i a_{i-1} + W_i \dot{a}_{i-1}$, $\dot{a}_i = D_i \dot{z}_i$, $\dot{a}_0 = 0$. Unrolling gives $\dot{z}_L = \sum_i B_i \Delta W_i a_{i-1}$, so $\|\nabla_{W_i} h(\cdot; x_n)\|_F = \|B_i\|_F \|a_{i-1}\|_2$. The gradient is bounded by taking the supremum over ΔW_i with $||\Delta W_i||_F = 1$. We can rewrite the expression using the vectorization technique $B_i \Delta W_i a_{i-1} = (a_{i-1}^\top \otimes B_i)(vec(\Delta W_i))$. Then, we have that the gradient is equal to $||(a_{i-1}^\top \otimes B_i)||_F = ||a_{i-1}||_2 ||B_i||_F$. The supremum attains this bound when ΔW_i is aligned with the tensor product.

The upper bound $\|B_i\|_{\text{op}} = O(\eta^{L-i})$ follows from submultiplicativity and using $\|W_j\|_{\text{op}} = O(\eta)$, $\|D_j\|_{\text{op}} \le \|\sigma'\|_\infty$. Combined with $\|a_{i-1}\|_2 = O(\eta^{i-1})$ from part (i), this gives $\|\nabla_{W_i} h(\cdot; x_n)\|_F = O(\eta^{L-1})$ for all i, and summing gives $\|\nabla_W h(\cdot; x_n)\|_F = O(\eta^{L-1})$, uniformly in n.

(iii) Hessian size. We use the second-order chain rule along $W(t) = W_0 + t\Delta W$ with $\|\Delta W\|_F = 1$. Since $\ddot{W}_i = 0$ (the path is affine):

$$\ddot{z}_i = 2\,\Delta W_i\,\dot{a}_{i-1} + W_i\,\ddot{a}_{i-1}, \qquad \ddot{a}_i = \underbrace{D_i^{(2)}\,(\dot{z}_i)^{\odot 2}}_{\text{curvature}} + \underbrace{D_i\,\ddot{z}_i}_{\text{recursive}}, \tag{9}$$

with $\dot{a}_0 = \ddot{a}_0 = 0$. Note $\ddot{z}_1 = 0$ (since $\dot{a}_0 = \ddot{a}_0 = 0$, and $z_1 = W_1 x_n$ is affine in t).

Extending the previous gradient bound to any layer, the first-order recursion gives $\|\dot{z}_i\|_2, \|\dot{a}_i\|_2 \le C^* \eta^{i-1}$ for some constant C^* independent of n by induction using $\|\Delta W_i\|_{\text{op}} \le 1$, $\|W_i\|_{\text{op}} = O(\eta)$, and $\|x_n\|_2 \le R_x$.

The Hessian operator norm is $\left\|\nabla^2 h(\cdot; x_n)\right\|_{\text{op}} = \sup_{\|\Delta W\|_F = 1} \|\ddot{z}_L\|_2$.

Case 1: Under Assumption 3. By equation 8 at W_0 (with $\rho = \max_i \|R_i\|_{\text{op}}$) and Assumption 3, $|z_{i,j}| \ge \eta^i m - C_\rho\, \eta^{i-1} \ge \eta^i m/2$ for all i, j (uniformly in n) whenever $\eta \ge \max\{1, 2C_\rho/m\}$. Then Assumption 1 gives

$$\left\|D_i^{(2)}\right\|_{\text{op}} = \max_j |\sigma''(z_{i,j})| \le C_\sigma e^{-\eta^i m/(2C_\sigma)} = o(\eta^{-N'})\ \forall N'.$$

The last equality follows since exponential decay is faster than any polynomial decay. Hence, the curvature term $\left\|D_i^{(2)}(\dot{z}_i)^{\odot 2}\right\|_2 \le \left\|D_i^{(2)}\right\|_{\text{op}} \|\dot{z}_i\|_2^2 = o(\eta^{-N'}) \cdot O(\eta^{2(i-1)})$ is negligible, uniformly in n. Let $\mathcal{A}_i := \|\ddot{a}_i\|_2$. From equation 9, $\mathcal{A}_i \le o(\eta^{-N'}) + \|\sigma'\|_\infty (2C^* \eta^{i-2} + O(\eta)\mathcal{A}_{i-1}) = O(\eta^{i-2}) + O(\eta)\mathcal{A}_{i-1}$.

With $\mathcal{A}_1 = o(\eta^{-N'})$, induction gives $\mathcal{A}_i = O(\eta^{i-2})$ uniformly in n. Therefore

$$\|\ddot{z}_L\|_2 \le 2\,\|\Delta W_L\|_{\text{op}}\,\|\dot{a}_{L-1}\|_2 + \|W_L\|_{\text{op}}\,\mathcal{A}_{L-1} \le 2C^* \eta^{L-2} + O(\eta) \cdot O(\eta^{L-3}) = O(\eta^{L-2}).$$

Case 2: Without Assumption 3. In the worst case $|z_{i,j}| = O(1)$ for some i, j, giving $\left\|D_i^{(2)}\right\|_{\text{op}} = O(1)$. The curvature term at layer i is then $\left\|D_i^{(2)}(\dot{z}_i)^{\odot 2}\right\|_2 = O(\eta^{2(i-1)})$. This contribution propagates through the remaining $L - i$ layers via the recursive term, picking up a factor of $O(\eta)$ at each step:

$$\text{curvature at layer } i\text{, propagated to output} = O(\eta^{2(i-1)}) \cdot O(\eta^{L-i}) = O(\eta^{L+i-2}).$$

Summing over layers and taking the worst case $i = L - 1$ gives $O(\eta^{2L-3})$. (The terms $2\Delta W_i \dot{a}_{i-1}$ inside the recursive term contribute $O(\eta^{L-2})$ regardless of regime, as in Case 1.) □

E.3 Local stability of the size estimates

Lemma 4 (Local size and Lipschitz estimates). *For every $C_1 > 0$ there exist constants $C_2 = C_2(C_1) > 0$, $C_3 = C_3(C_1) > 0$, and a threshold $\eta_0 = \eta_0(C_1) \geq 1$ such that for all $\eta \geq \eta_0$ and every W in the ball*

$$\mathcal{B}_{C_1} := \{W : \|W - W_0\|_F \leq C_1\, \eta^{-(L-1)}\},$$

the following hold uniformly in $n \in \{1, \dots, N\}$*:*

(i) $\|\nabla_W h(W; x_n)\|_F = O(\eta^{L-1})$*, with implicit constants converging to those of* $\|\nabla_W h(W_0; x_n)\|_F$ *as* $\eta \to \infty$.

(ii) Under Assumption 3: $\left\|\nabla^2_W h(W; x_n)\right\|_{\mathrm{op}} \leq C_2\, \eta^{L-2}$.
Without Assumption 3: $\left\|\nabla^2_W h(W; x_n)\right\|_{\mathrm{op}} \leq C_2\, \eta^{2L-3}$.

(iii) Under Assumption 3:
$\|\nabla_W h(W; x_n) - \nabla_W h(W_0; x_n)\|_F \leq C_3\, \eta^{L-2}\, \|W - W_0\|_F$.
Without Assumption 3:
$\|\nabla_W h(W; x_n) - \nabla_W h(W_0; x_n)\|_F \leq C_3\, \eta^{2L-3}\, \|W - W_0\|_F$.

Moreover $C_2(C_1)$ and $C_3(C_1)$ remain bounded as C_1 varies over any bounded set, while $\eta_0(C_1)$ may grow with C_1.

Proof. For the bound in (i), we can absorb the perturbation ΔW_i into the remainder part R_i of W_i. Then, the same bounds hold as previously derived. Formally, fix $C_1 > 0$. Decompose $W_i = \eta\, U_i V_i^\top + \widetilde{R}_i$ with $\widetilde{R}_i := R_i + \Delta W_i$, where $\|\Delta W_i\|_{\mathrm{op}} \leq \|\Delta W_i\|_F \leq C_1\, \eta^{-(L-1)}$. Then

$$\left\|\widetilde{R}_i\right\|_{\mathrm{op}} \leq \|R_i\|_{\mathrm{op}} + C_1\, \eta^{-(L-1)} = \|R_i\|_{\mathrm{op}} + o(1) \quad \text{as } \eta \to \infty,$$

so $\|W_i\|_{\mathrm{op}} = O(\eta)$ is preserved on $\mathcal{B}_{C_1}$, which suffices for the upper bounds in (i).

Preservation of the pre-activation margin (under Assumption 3). The decomposition above writes every $W \in \mathcal{B}_{C_1}$ in the form of the initialization, $W_i = \eta\, U_i V_i^\top + \widetilde{R}_i$, with the same U_i, V_i. Whenever $\eta \geq \max\{1, C_1^{1/(L-1)}\}$ we have $C_1 \eta^{-(L-1)} \leq 1$, and hence $\left\|\widetilde{R}_i\right\|_{\mathrm{op}} \leq \|R_i\|_{\mathrm{op}} + 1$. The expansion equation 8 therefore applies at W with $\rho := \max_i \|R_i\|_{\mathrm{op}} + 1$, and Assumption 3 gives

$$|z_{i,j}(W; x_n)| \geq \eta^i m - C_\rho\, \eta^{i-1} \geq \tfrac{\eta^i m}{2} \qquad \text{for all } W \in \mathcal{B}_{C_1} \text{ and all } n, i, j,$$

whenever $\eta \geq \eta_0(C_1) := \max\{1, C_1^{1/(L-1)}, 2C_\rho/m\}$. This is exactly the hypothesis of Case 1 of Lemma 3(iii), so the curvature suppression used there persists on $\mathcal{B}_{C_1}$.

Conclusions (i)–(iii). With $\|W_i\|_{\mathrm{op}} = O(\eta)$ and (under Assumption 3) the pre-activation margin preserved on $\mathcal{B}_{C_1}$ uniformly in n, the analysis of Lemma 3 applies uniformly in $W \in \mathcal{B}_{C_1}$ and $n \in \{1, \dots, N\}$, with implicit constants controlled by $\|R_i\|_{\mathrm{op}} + C_1\, \eta^{-(L-1)}$, the pre-activation margin $m/2$, and R_x. This yields (i) and (ii). Part (iii) follows from the mean-value inequality

$$\|\nabla h(W; x_n) - \nabla h(W_0; x_n)\|_F \leq \sup_{W' \in [W_0, W]} \left\|\nabla^2 h(W'; x_n)\right\|_{\mathrm{op}} \cdot \|W - W_0\|_F,$$

combined with the supremum bound from (ii), uniformly in n.

The boundedness of $C_2(C_1), C_3(C_1)$ on bounded C_1-sets follows from the continuous dependence of the expansion on $\left\|\widetilde{R}_i\right\|_{\mathrm{op}}$, itself continuous in C_1. □

E.4 Bound on weight motion under the rescaled flow

Proposition 5 (Weight motion). *On the maximal interval $[0, T^*) \subseteq [0, T]$ on which $\|W_t - W_0\|_{\mathrm{F}} \leq C_1 \eta^{-(L-1)}$, the gradient flow equation 3 satisfies*

$$\|W_t - W_0\|_{\mathrm{F}} \leq C \cdot t \cdot \eta^{-(L-1)} \qquad \forall t \in [0, T^*),$$

for a constant C independent of η. For η sufficiently large, $T^* = T$.

Proof. Velocity at the centre. Lemma 3(ii) gives $\|J_0(x_n)\|_{\mathrm{op}} \le K_0\,\eta^{L-1}$ uniformly in n, for a constant K_0 independent of η and n. Hence the averaged gradient at W_0 obeys

$$\|\nabla_W \mathcal{L}(W_0)\|_F \;\le\; \frac{1}{N}\sum_{n=1}^{N} \|J_0(x_n)\|_{\mathrm{op}}\,\|r_0(x_n)\|_2 \;\le\; K_0 G\,\eta^{L-1}.$$

Setting $C_0 := K_0 G$,

$$\alpha\,\|\nabla_W \mathcal{L}(W_0)\|_F \;\le\; C_0\,\eta^{-(L-1)}.$$

Choice of ball radius. Choose

$$C_1 \;:=\; 2\,C_0\,T,$$

a finite constant depending on $T, G, L, \|R_i\|_{\mathrm{op}}, R_x, M, \sigma$ but not on η. Lemma 4 with this C_1 provides C_2, C_3 and a threshold η_0 such that the size and Lipschitz estimates hold uniformly on

$$\mathcal{B} \;:=\; \left\{W : \|W - W_0\|_F \le C_1\,\eta^{-(L-1)}\right\} \qquad \text{for all } \eta \ge \eta_0,$$

uniformly in $n \in \{1, \dots, N\}$.

Velocity inside the ball. By Lemma 4(i), for $W \in \mathcal{B}$ we have $\|J(W; x_n)\|_{\mathrm{op}} \le K(C_1)\,\eta^{L-1}$ uniformly in n. And we know $K(C_1) \to K_0$ as $\eta \to \infty$ from continuity discussed in Lemma 4. Due to continuity, by increasing η_0 if necessary, we can assume $K(C_1) \le \frac{3}{2}K_0$ for $\eta \ge \eta_0$.

Then on $\mathcal{B}$,

$$\left\|\dot{W}_t\right\|_F \;\le\; \frac{\alpha}{N}\sum_{n=1}^{N} \|J_t(x_n)\|_{\mathrm{op}}\,\|r_t(x_n)\|_2 \;\le\; \tfrac{3}{2}\,C_0\,\eta^{-(L-1)}.$$

The second inequality uses $\|r_t(x_n)\|_2 \le G$ from Assumption 2.

Bootstrap argument on the interval of time where the bound is valid. Let $T^* := \sup\{t \in [0, T] : W_s \in \mathcal{B} \text{ for all } s \in [0, t]\}$. On $[0, T^*)$, integrating the velocity bound gives

$$\|W_t - W_0\|_F \;\le\; \tfrac{3}{2}\,C_0\,t\,\eta^{-(L-1)} \;\le\; \tfrac{3}{2}\,C_0\,T\,\eta^{-(L-1)} \;=\; \tfrac{3}{4}\,C_1\,\eta^{-(L-1)}.$$

The strict inequality $\frac{3}{4}C_1 < C_1$ and continuity of $t \mapsto \|W_t - W_0\|_F$ imply that W_{T^*} lies in the interior of the ball $\mathcal{B}$. Then, if $T^* < T$, we can find $T^* + \delta$ for a small $\delta > 0$ where $W_{T^*+\delta} \in \mathcal{B}$ also, contradicting T^* being the maximal time. Thus, $T^* = T$.

Concluding, the bound $\|W_t - W_0\|_F \le \frac{3}{2}\,C_0\,t\,\eta^{-(L-1)}$ holds on $[0, T]$ proving the claim with $C = \frac{3}{2}C_0$. □

This proves equation 5, which holds with or without Assumption 3.

E.5 Linearization comparison

We work on $[0, T]$ with $T^* = T$ from Proposition 5. Let $J_t(x_n) := \nabla_W h(W_t; x_n)$ and $J_0(x_n) := \nabla_W h(W_0; x_n)$ for each n.

Lemma 6 (Jacobian drift)**.** *Under the respective bounds in Lemma 4(iii) and Proposition 5, uniformly in $n \in \{1, \dots, N\}$:*

$$\textit{Under Assumption 3:} \quad \|J_t(x_n) - J_0(x_n)\|_F \le C_3\eta^{L-2} \cdot Ct \cdot \eta^{-(L-1)} = O(\eta^{-1}),$$
$$\textit{Without Assumption 3:} \quad \|J_t(x_n) - J_0(x_n)\|_F \le C_3\eta^{2L-3} \cdot Ct \cdot \eta^{-(L-1)} = O(\eta^{L-2}).$$

Lemma 7 (Output deviation)**.** *For $t \in [0, T]$ and uniformly in $n \in \{1, \dots, N\}$:*

$$\textit{Under Assumption 3:} \quad \left\|h(W_t; x_n) - \bar{h}(\bar{W}_t; x_n)\right\|_2 \le C\,\eta^{-L},$$
$$\textit{Without Assumption 3:} \quad \left\|h(W_t; x_n) - \bar{h}(\bar{W}_t; x_n)\right\|_2 \le C\,\eta^{-1}.$$

Proof. For each n, let $\Delta_t(x_n) := h_t(x_n) - \bar{h}_t(x_n)$.

Bounding $\Delta_t(x_n)$ via coupled Grönwall. The deviation evolves as

$$\begin{aligned}
\dot{\Delta}_t(x_n) &= J_t(x_n)\dot{W}_t - J_0(x_n)\dot{\bar{W}}_t && \text{(chain rule; } \bar{h}(\cdot; x_n) \text{ is linear)} \\
&= -\frac{\alpha}{N}\sum_{m=1}^{N}\left[J_t(x_n)J_t(x_m)^\top r_t(x_m) - J_0(x_n)J_0(x_m)^\top \bar{r}_t(x_m)\right] && \text{(by equation 3, equation 4)} \\
&= -\frac{\alpha}{N}\sum_{m=1}^{N} J_0(x_n)J_0(x_m)^\top H_t(x_m)\Delta_t(x_m) && \text{(see below)} \\
&\quad -\frac{\alpha}{N}\sum_{m=1}^{N}\big(J_t(x_n)J_t(x_m)^\top - J_0(x_n)J_0(x_m)^\top\big) r_t(x_m),
\end{aligned}$$

where the last step adds and subtracts $J_0(x_n)J_0(x_m)^\top r_t(x_m)$ and uses

$$r_t(x_m) - \bar{r}_t(x_m) = H_t(x_m)\,\Delta_t(x_m), \qquad H_t(x_m) := \int_0^1 \nabla_u^2 \ell\big(s\,h_t(x_m) + (1-s)\,\bar{h}_t(x_m),\, y_m\big)\,ds.$$

This identity follows from the fundamental theorem of calculus along the segment from $\bar{h}_t(x_m)$ to $h_t(x_m)$. Let $\gamma(s) := \nabla_u \ell\big(s\,h_t(x_m) + (1-s)\,\bar{h}_t(x_m),\, y_m\big)$ for $s \in [0,1]$, so that $\gamma(0) = \bar{r}_t(x_m)$ and $\gamma(1) = r_t(x_m)$. Since ℓ is C^2 in its first argument, the chain rule gives $\gamma'(s) = \nabla_u^2 \ell\big(s\,h_t(x_m) + (1-s)\,\bar{h}_t(x_m),\, y_m\big)\,\Delta_t(x_m)$, and hence $r_t(x_m) - \bar{r}_t(x_m) = \int_0^1 \gamma'(s)\,ds = H_t(x_m)\,\Delta_t(x_m)$, the factor $\Delta_t(x_m)$ being independent of s. Each Hessian in the integral has operator norm at most M by Assumption 2, so $\|H_t(x_m)\|_{\mathrm{op}} \le M$.

Homogeneous coefficient. For the homogeneous part, using $\|J_0(x_n)\|_{\mathrm{op}}, \|J_0(x_m)\|_{\mathrm{op}} \le K_0\eta^{L-1}$:

$$\frac{\alpha}{N}\sum_{m=1}^{N}\big\|J_0(x_n)J_0(x_m)^\top H_t(x_m)\big\|_{\mathrm{op}}\,\|\Delta_t(x_m)\|_2 \;\le\; \alpha K_0^2\eta^{2(L-1)}M\cdot\frac{1}{N}\sum_{m=1}^{N}\|\Delta_t(x_m)\|_2 \;\le\; K_0^2 M\cdot\Delta_t^{\max},$$

where $\Delta_t^{\max} := \max_m \|\Delta_t(x_m)\|_2$ (since the average is bounded by the max).

Inhomogeneous term. By Lemma 6 (uniform in n, m):

$$\begin{aligned}
&\big\|J_t(x_n)J_t(x_m)^\top - J_0(x_n)J_0(x_m)^\top\big\|_{\mathrm{op}} \\
&\quad\le \|J_t(x_n) - J_0(x_n)\|_{\mathrm{op}}\,\|J_t(x_m)\|_{\mathrm{op}} + \|J_0(x_n)\|_{\mathrm{op}}\,\|J_t(x_m) - J_0(x_m)\|_{\mathrm{op}}.
\end{aligned}$$

Under Assumption 3: $O(\eta^{-1})\cdot O(\eta^{L-1}) = O(\eta^{L-2})$. Without: $O(\eta^{L-2})\cdot O(\eta^{L-1}) = O(\eta^{2L-3})$. Hence, recalling that $\alpha = O(\eta^{-2(L-1)})$,

$$\begin{aligned}
&\frac{\alpha}{N}\sum_{m=1}^{N}\big\|J_t(x_n)J_t(x_m)^\top - J_0(x_n)J_0(x_m)^\top\big\|_{\mathrm{op}}\; G \\
&\qquad\le \begin{cases} \alpha\cdot O(\eta^{L-2})\cdot G = O(\eta^{-L}) & \text{(pre-activation size assumption)} \\ \alpha\cdot O(\eta^{2L-3})\cdot G = O(\eta^{-1}) & \text{(no pre-activation size assumption)} \end{cases},
\end{aligned}$$

uniformly in n.

Uniform Grönwall. Combining, for each n: $\left\|\dot{\Delta}_t(x_n)\right\|_2 \le K_0^2 M\,\Delta_t^{\max} + \epsilon$, where $\epsilon = O(\eta^{-L})$ (resp. $O(\eta^{-1})$). Integrating from $\Delta_0(x_n) = 0$:

$$\|\Delta_t(x_n)\|_2 \le \int_0^t \left[K_0^2 M\,\Delta_s^{\max} + \epsilon\right] ds.$$

Taking $\max_n$ on the left (the right-hand side does not depend on n) yields $\Delta_t^{\max} \le K_0^2 M\int_0^t \Delta_s^{\max}\,ds + \epsilon t$, and Grönwall gives $\Delta_t^{\max} \le \epsilon T\, e^{K_0^2 M T} = O(\epsilon)$. This proves $\max_n \|\Delta_t(x_n)\|_2 = O(\eta^{-L})$ (resp. $O(\eta^{-1})$). □

This proves equation 6.

Lemma 8 (Parameter deviation). $\sup_{t\in[0,T]} \left\|W_t - \bar{W}_t\right\|_F = O(\eta^{-(2L-1)})$ *under Assumption 3, and* $O(\eta^{-L})$ *without it.*

Proof. From equation 3 and equation 4:

$$\begin{aligned}\dot{W}_t - \dot{\bar{W}}_t &= -\frac{\alpha}{N}\sum_{n=1}^{N}\left[J_t(x_n)^\top r_t(x_n) - J_0(x_n)^\top \bar{r}_t(x_n)\right] \\ &= -\frac{\alpha}{N}\sum_{n=1}^{N} J_0(x_n)^\top H_t(x_n)\Delta_t(x_n) - \frac{\alpha}{N}\sum_{n=1}^{N}\big(J_t(x_n)^\top - J_0(x_n)^\top\big) r_t(x_n),\end{aligned}$$

where we added and subtracted $J_0(x_n)^\top r_t(x_n)$ and used $r_t(x_n) - \bar{r}_t(x_n) = H_t(x_n)\Delta_t(x_n)$ as in Lemma 7.

Bounding each term in Frobenius norm using $\|J_0(x_n)\|_{\mathrm{op}} = O(\eta^{L-1})$, $\|H_t(x_n)\|_{\mathrm{op}} \le M$, $\max_n \|\Delta_t(x_n)\|_2 = O(\epsilon)$ from Lemma 7, and $\max_n \|J_t(x_n) - J_0(x_n)\|_F = O(\lambda)$ from Lemma 6, where $\lambda := \eta^{-1}$ under Assumption 3 and $\lambda := \eta^{L-2}$ without it:

$$\begin{aligned}\left\|\dot{W}_t - \dot{\bar{W}}_t\right\|_F &\le \frac{\alpha}{N}\sum_{n=1}^{N}\|J_0(x_n)\|_{\mathrm{op}}\, M\, \|\Delta_t(x_n)\|_2 + \frac{\alpha}{N}\sum_{n=1}^{N}\|J_t(x_n) - J_0(x_n)\|_F\, G \\ &\le \alpha\, O(\eta^{L-1})\, M\, O(\epsilon) + \alpha\, O(\lambda)\, G,\end{aligned}$$

where each term is uniform in n. With $\alpha = \eta^{-2(L-1)}$:

- With the pre-activation size assumption ($\epsilon = O(\eta^{-L})$, $\lambda = \eta^{-1}$): $\alpha\,\eta^{L-1}\epsilon = O(\eta^{-(2L-1)})$ and $\alpha\,\lambda = \eta^{-(2L-1)}$, so both terms are $O(\eta^{-(2L-1)})$.
- Without the pre-activation size assumption ($\epsilon = O(\eta^{-1})$, $\lambda = \eta^{L-2}$): $\alpha\,\eta^{L-1}\epsilon = O(\eta^{-L})$ and $\alpha\,\lambda = \eta^{-L}$, so both terms are $O(\eta^{-L})$.

These velocity bounds hold uniformly in $t \in [0,T]$, since those of Lemma 7 and Lemma 6 do. Since both flows start from $W_0 = \bar{W}_0$, the fundamental theorem of calculus gives, for every $t \in [0,T]$,

$$W_t - \bar{W}_t \;=\; \int_0^t \big(\dot{W}_s - \dot{\bar{W}}_s\big)\, ds.$$

Bounding the norm of the integral by the integral of the norm,

$$\left\|W_t - \bar{W}_t\right\|_F \;\le\; \int_0^t \left\|\dot{W}_s - \dot{\bar{W}}_s\right\|_F \, ds \;\le\; T \sup_{s\in[0,T]} \left\|\dot{W}_s - \dot{\bar{W}}_s\right\|_F.$$

As T is fixed and independent of η, taking the supremum over $t \in [0,T]$ gives $\sup_{t\in[0,T]} \left\|W_t - \bar{W}_t\right\|_F = O(\eta^{-(2L-1)})$ under Assumption 3 and $O(\eta^{-L})$ without it. □

This proves equation 7, completing the proof of Theorem 1.