Appendix

A Reproducibility Statement

We used the FFCV-SSL package by Bordes et al. built on Leclerc et al.'s FFCV package to ensure full reproducibility in terms of the SGD noise, see subsection A.1 for more details.

A.1 Data Loader Reproducibility

The data ordering affects stochastic gradient based methods and hence connectivity. To ensure our runs can be reproduced we pick a trainer seed for initialization, data loader seed that determines ordering and an augmentation seed used for random augmentations. We include two sample CIFAR-10 data loaders in Figure 4.

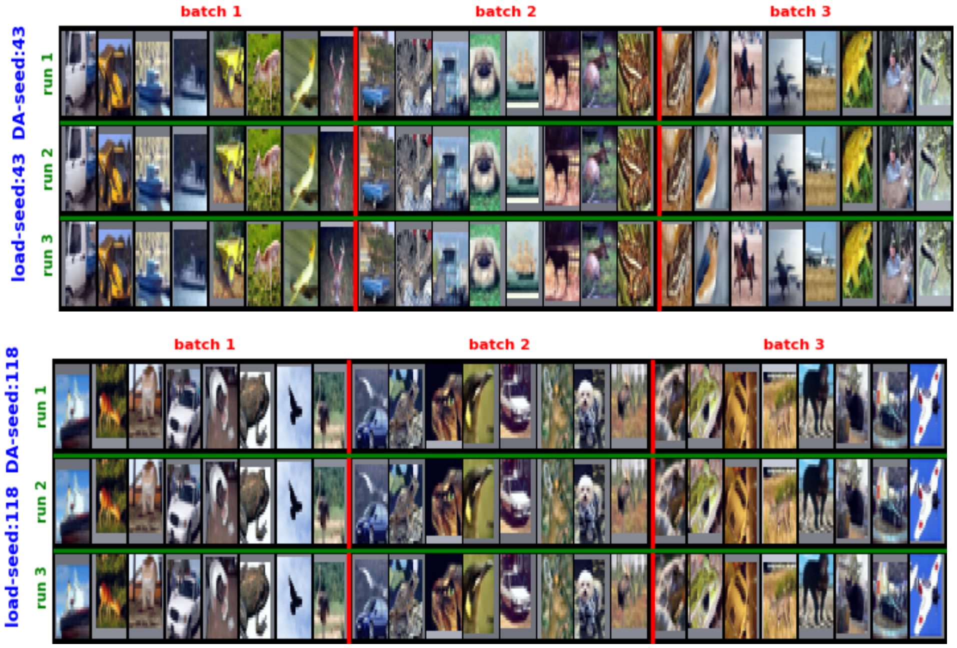

Figure 4: Each data loader is initialized three times with loader seed (used for data ordering) and data augmentation seed (used for random augmentations) set to 43 (top) and 118 (bottom). Augmentations used: random translate and horizontal flip

B Related Work

This appendix provides a brief overview of the relevant literature.

Mode-Connectivity: Garipov et al.; Draxler et al. demonstrated that optima trained from different initializations can be connected with simple parametric curves, e.g., polygonal chains or Bezier curves, without incurring a significant increase in the loss along this path. Benton et al. showed that these paths can be extended to probabilistic volumes of low loss. Simsek et al. formalized these volumes for over-parameterized networks as the *Global Minima Manifold* and provided explicit descriptions of its dimensions.

Linear Mode-Connectivity: A parallel line of work, explore *linear* mode-connectivity (LMC), where a linear path of near-constant error exists between the two optima. Frankle et al. show that two fully-connected networks trained from the same initialization but with different SGD noise, i.e. data order and augmentations, are stable to the noise and converge to linearly connected minima. Their results extend to more complex vision algorithms as well if the two networks are trained with the same SGD noise for a while and then More recently Zhou et al. coined Layerwise Linear Feature Connectivity (LLFC), a stronger setting where the feature maps of every layer also exhibit LMC.

Permutation Invariance: The permutation symmetry of the hidden neurons and how the learned parameters interact with each other after accounting for the permutation has also emerged as a notable avenue of inquiry. Entezari et al.'s initial conjecture argued that in most cases SGD converges to the same basin up-to permutation and showed the emergence of LMC for wide and shallow architectures after accounting for the permutation invariance. Ainsworth et al. proposed a general weight matching algorithm to align models trained from different initializations that supported Entezari et al.'s conjecture on ResNets as well. Benzing et al. show that two models exhibit linear mode-connectivity at initialization when merged with the permutation found later in training.

C Further Training Strategies

This appendix presents some ablation studies regarding the training techniques and optimization.

Although we examined the effect of other regularization techniques, our preliminary experiments proved these three dimensions to be the most important for LMC across different architectures. For example, varying the batch size, turning off momentum, adding a weight decay term or cosine learning rate scheduler doesn't have a significant impact on the behavior of the previous settings. We found that gradient clipping can also be used to preserve LMC.

C.1 ADAM on MLPs

Since ADAM already breaks LMC in deeper linear models, we find it trivial that it also doesn't preserve LMC in MLPs. Still, for completeness we provide the performance-aware barrier for MNIST and CiFAR-10 in Figure 5.

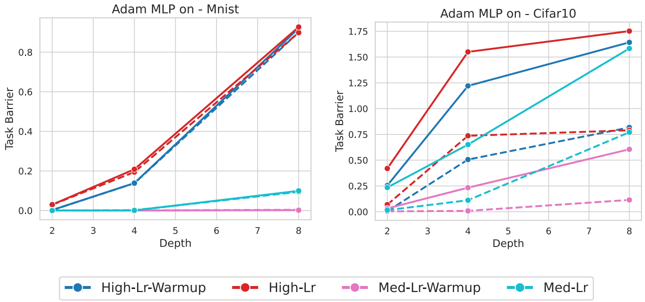

Figure 5: Task barrier for MLPs trained with ADAM on MNIST (left) and CiFAR-10 (right)

D Architectural Correspondence

Similar to Neyshabur, we study shallow convolutions and establish their MLP counterpart based on the Toeplitz representation of the underlying convolutional layer. For simplicity, we set stride equal to the kernel size and do not use 0-padding. A convolutional layer operating on a $C_i \times H \times W$ input with kernel size (k_1, k_2) and C_o filters has $C_o \times C_i \times k_1 \times k_2$ parameters. Its locally connected counterpart uses different kernels to compute each target pixel, hence it has $C_o \times H' \times W' \times C_i \times k_1 \times k_2$ parameters, where H', W' is the output spatial dimension of the resulting feature map. Both of them can be embedded in a $C_o \cdot H' \cdot W' \times C_i \cdot H \cdot W$ linear layer. Table 3 shows the total number of parameters for a 2-Layer network, where the first layer is either a convolution, locally connected or linear layer whose weights can be represented with the same Toeplitz matrix. Note that ViTs could also be considered in this framework thanks to Cordonnier et al., however we leave it to future work to study the exact correspondence.

ViT-like MLP To study the effect of attention on LMC, we consider the simplest setting of a ViT. We don't modify the patch embeddings, normalizations and the classifier layer but simplify the encoder. We remove skip connections and the MLP part (last two linear layers) from the transformer

Table 3: Parameter Count (M) for a 2-Layer Network where the first layer is either a CNN/LC-CNN/Linear equivalent to kernel size=4, padding=0

CNN	0.09
LC-CNN	0.48
MLP	25.26

encoder block and only use one block. We use patch size of 4 to establish similarity to the (LC)-CNN case. 8 heads each of dimension 48. The resulting architecture has ~ 1.08M parameters.

E Data

In Figure 3, we gradually increase the complexity of the task by changing the dataset:

1. *MNIST* $\rightarrow$ *CiFAR-10* input dimensions (both spatial and number of channels) increase from $(28, 28, 1)$ to $(32, 32, 3)$ while keeping the number of target labels the same. These two datasets also have similar number of samples (60,000 and 50,000).
2. *CiFAR-10* $\rightarrow$ *CiFAR-100* number of samples and image resolution stay constant while the number of labels increase by a factor of 10, from 10 to 100.
3. *CiFAR-100* $\rightarrow$ *Tiny-ImageNet* image resolution, number of labels and number of samples double.

We limit this analysis to 2-4-8-Layer MLPs trained using SGD with high (0.1) or medium (0.01) learning rate. Since we are interested in the most simple settings, we don't use any data augmentation, which hurts generalization. Moreover, MLPs are known for their subpar performance on large scale image classification tasks. See Table 4 for a comparison of the test accuracies across these four datasets. We propose Equation 2 to account for this performance gap. This modification allows us to view error barrier as a ratio of the lost performance.

Table 4: Test accuracies (%) reached on varying datasets by L-Layer MLPs trained using SGD with high (0.1) or medium (0.01)

	2-Layer		4-Layer		8-Layer	
	High	Med	High	Med	High	Med
MNIST	98.32	98.34	98.41	98.19	98.41	97.14
CiFAR-10	*	54.24	58.75	55.51	57.44	54.45
CiFAR-100	*	26.01	14.26	27.16	25.33	20.30
TinyImageNet	*	7.62	1.68	8.27	5.80	5.79