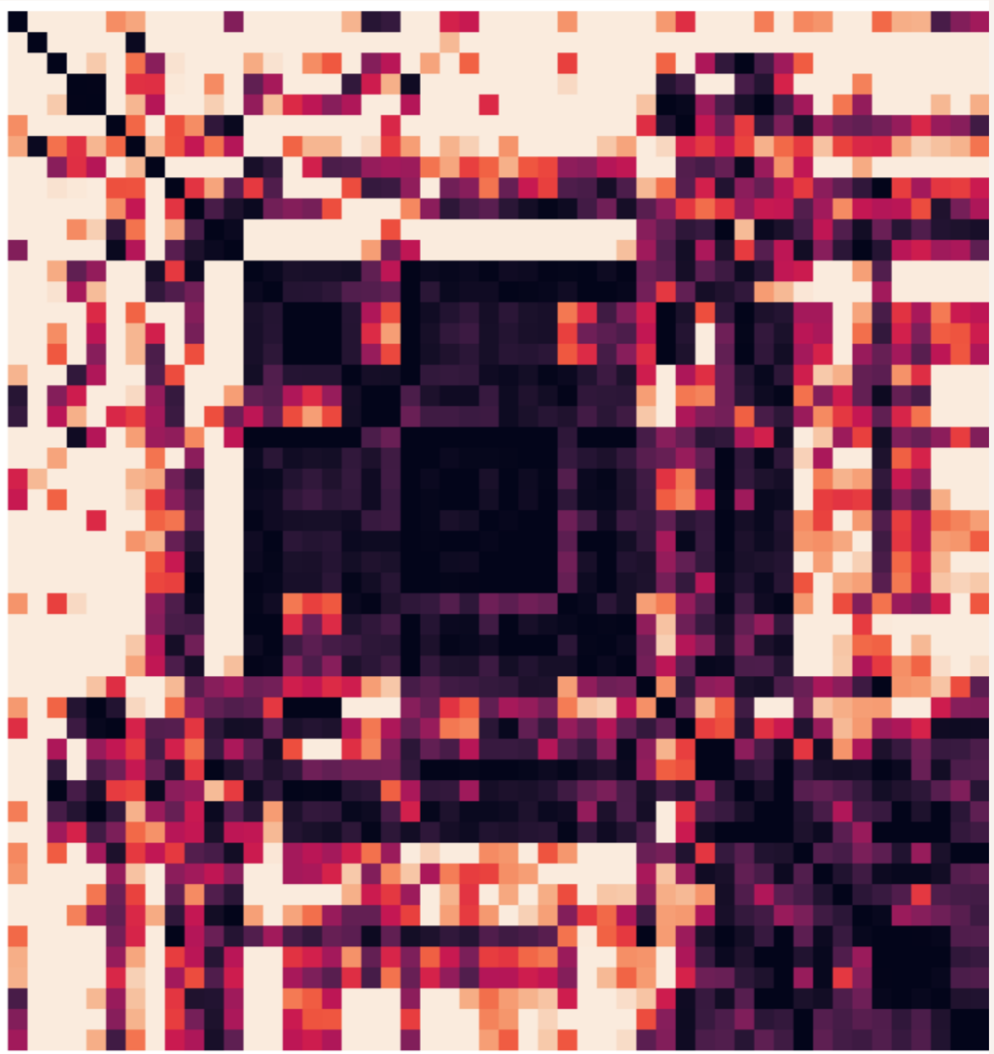
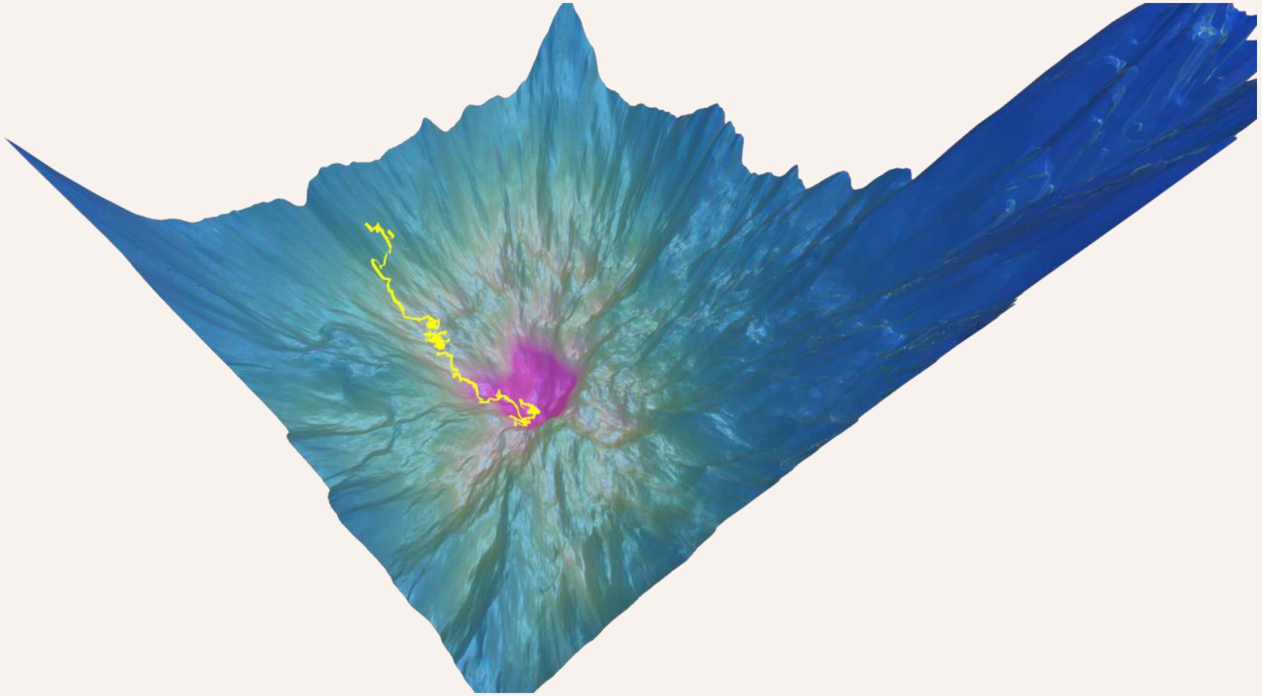
On Linear Mode Connectivity up to Permutation of Hidden Neurons in Neural Networks

WHEN DOES MODEL AVERAGING WORK?



Loss surface of Neural Networks



Source: *Loss landscape explorer: Explore real loss landscapes of deep learning optimization processes*.
https://losslandscape.com/explorer

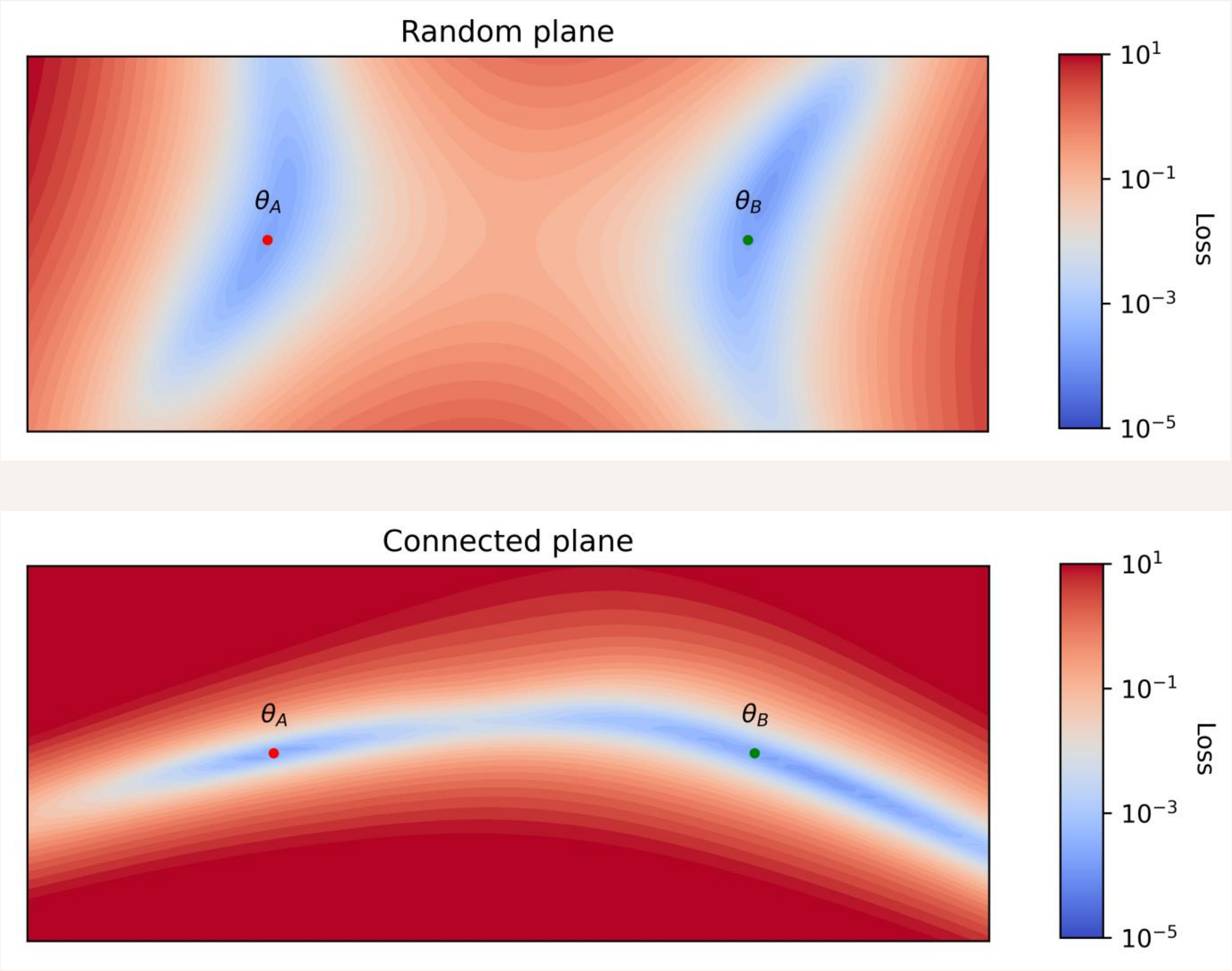
A high dimensional nonconvex mapping from parameters to the empirical loss

$$L: \theta \to \mathbb{R}^+$$

Image: 2D slice with weight updates projected onto it

From isolated minima to mode connectivity

Mode connectivity is the phenomenon where solutions obtained through variants of gradient descent are **connected** by simple curves in the weight space along which the loss remains low



Why is this surprising at all?

IT HAS TO DO WITH
IDENTIFIABILITY

Example 1: Matrix decomposition

- Given a similarity matrix S, say we want X such that $S = X^T X$
- Any X we estimate is only identifiable up to an orthogonal transform Q

  i.e., if X is a solution, then $\tilde{X} = QX$ is also a solution

  $$\tilde{X}^T \tilde{X} = (QX)^T (QX) = X^T Q^T Q X = X^T X = S$$

  as Q is an orthonormal matrix

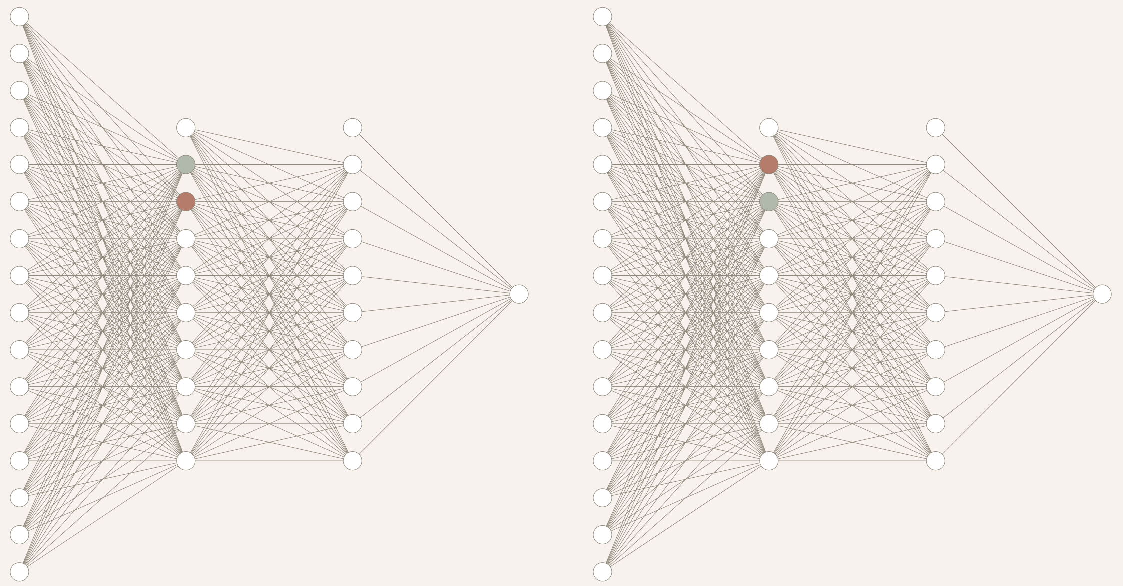
- Here two equally good solutions are a continuous transformation apart

Example 2: Neural Networks

- Predominant symmetry is Permutation whose elements are discrete

$$\pi(\theta) = \left\{P_i W_i P_{i-1}^T, P_i b_i \,\middle|\, P_0 = P_k = \mathbb{I}\,, 1 \leq\ i \leq\ k\right\}$$

- Hidden neurons can be permuted without changing the function
- Other symmetries like scaling, etc., under specific activations or weights



Mode connectivity to linear mode connectivity

- Is the parameter-wise average of two networks a good network?
- Typically, no!

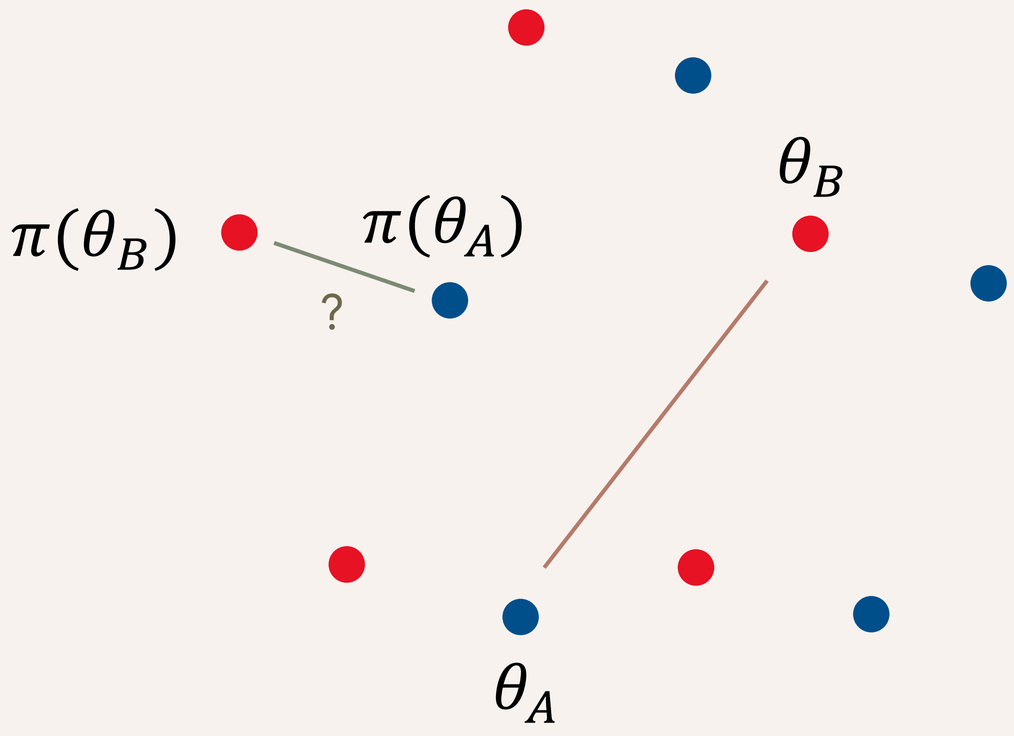
- Consider linear networks, $f_{\theta_A} = W_2^A W_1^A;\ f_{\theta_B} = W_2^B W_1^B$
- Interpolations between them will be of the form,

$$\alpha^2 W_2^A W_1^A + \alpha(1-\alpha)\left(W_2^A W_1^B + W_2^B W_1^A\right) + (1-\alpha)^2\, W_2^B W_1^B$$

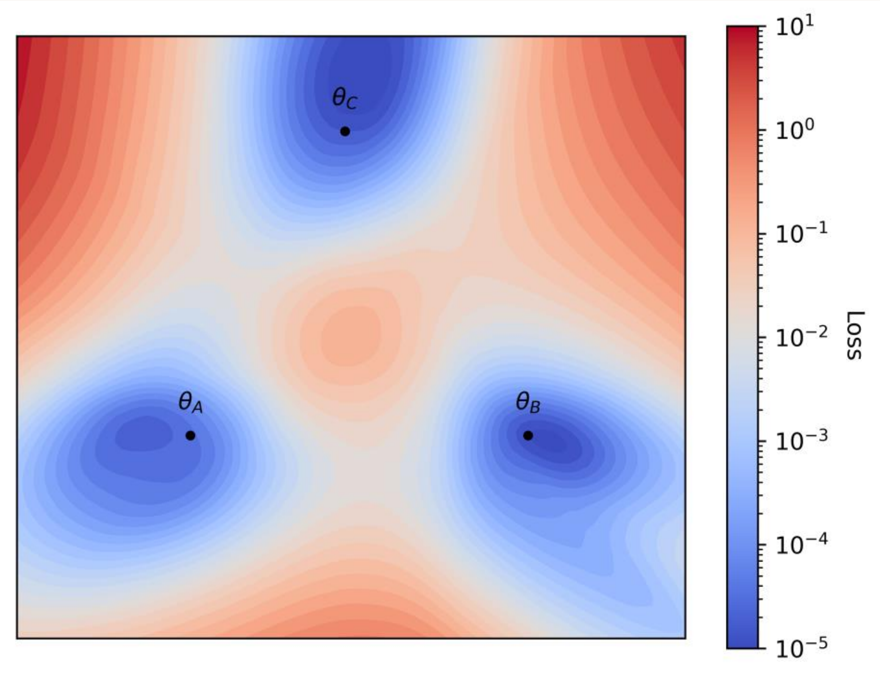
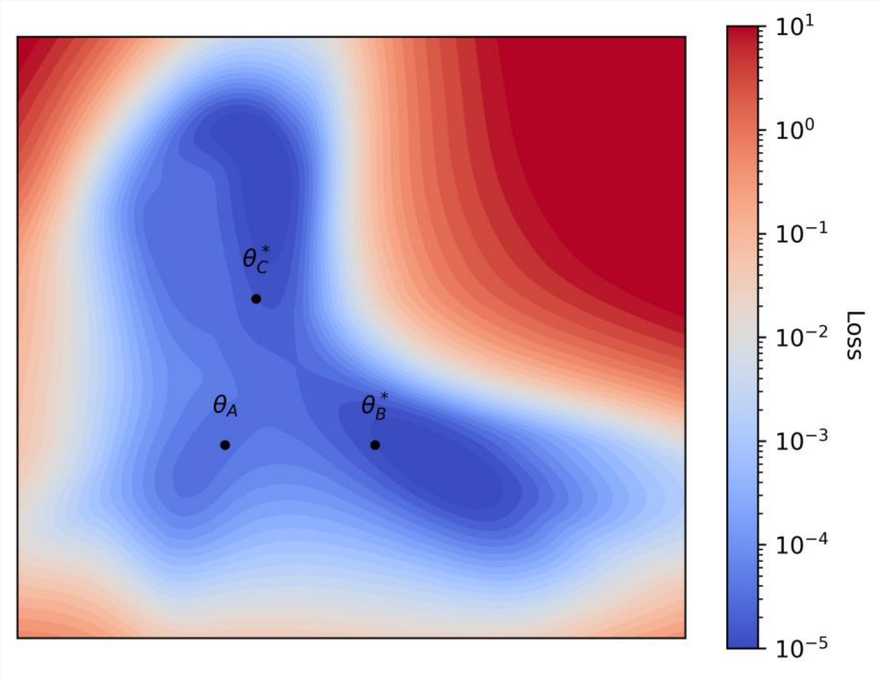
Linear mode connectivity up to permutation

Two solutions θ_A and θ_B are said to be linearly mode connected up to permutation if there exists some $\theta_A^* \in [\theta_A]$ and $\theta_B^* \in [\theta_A]$ such that θ_A^* and θ_B^* are linearly mode connected

Here, $[\theta]$ represents an equivalence class with the relation $\theta_i \sim \theta_j$ if and only if there exists a permutation transform π such that $\theta_j = \pi(\theta_i)$



Most trained networks are LMC up to permutation

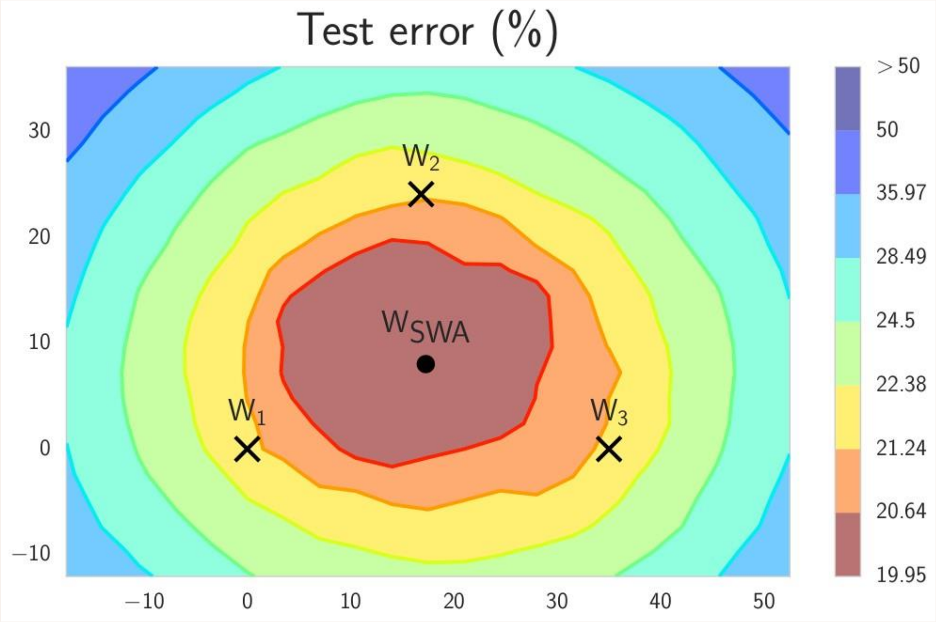
Loss plane through three trained networks weights	Two networks are reparameterized to be aligned
	

Research questions

1. For LMC, is reparameterization for permutation…
   - necessary?
   - sufficient?

2. LMC ⇔ Good average networks
   - How can we explain success of weight averaging methods like SWA that don't perform explicit permutation alignment?

Source: Izmailov et al. (2018)



Experiment setup

Architectures:

- 2-layer fully connected ReLU networks
- 4-layer fully connected ReLU networks with layernorm

Data:

- Moons
- MNIST
- CIFAR-10

Model zoo: Moons (left) and MNIST (right)

Width	Loss		Accuracy	
	train	test	train	test
2	0.264 ± 0.088	0.284 ± 0.084	0.879 ± 0.074	0.859 ± 0.077
4	0.186 ± 0.087	0.206 ± 0.092	0.920 ± 0.041	0.903 ± 0.049
8	0.105 ± 0.095	0.124 ± 0.105	0.957 ± 0.045	0.941 ± 0.057
16	0.019 ± 0.038	0.028 ± 0.043	0.996 ± 0.017	0.991 ± 0.024
32	0.005 ± 0.001	0.012 ± 0.002	1.0 ± 0.0	0.996 ± 0.001
64	0.003 ± 0.001	0.009 ± 0.002	1.0 ± 0.0	0.997 ± 0.001
128	0.002 ± 0.0	0.006 ± 0.001	1.0 ± 0.0	0.998 ± 0.001
256	0.002 ± 0.0	0.005 ± 0.001	1.0 ± 0.0	0.999 ± 0.001
512	0.001 ± 0.0	0.004 ± 0.0	1.0 ± 0.0	0.998 ± 0.001

Width	Loss		Accuracy	
	train	test	train	test
2	2.301 ± 0.0	2.301 ± 0.0	0.112 ± 0.0	0.113 ± 0.002
4	0.660 ± 0.047	0.690 ± 0.056	0.803 ± 0.017	0.798 ± 0.018
8	0.172 ± 0.007	0.225 ± 0.010	0.950 ± 0.002	0.938 ± 0.003
16	0.040 ± 0.002	0.134 ± 0.008	0.989 ± 0.001	0.966 ± 0.002
32	0.003 ± 0.0	0.120 ± 0.010	0.999 ± 0.0	0.978 ± 0.002
64	0.001 ± 0.0	0.108 ± 0.008	1.0 ± 0.0	0.983 ± 0.001
128	0.0 ± 0.0	0.102 ± 0.009	1.0 ± 0.0	0.985 ± 0.001
256	0.0 ± 0.0	0.102 ± 0.009	1.0 ± 0.0	0.985 ± 0.001
512	0.0 ± 0.0	0.111 ± 0.009	1.0 ± 0.0	0.985 ± 0.001

Reparameterization method: Weight matching

- Too expensive to be exact, so rely on heuristics
- Minimize the distance of parameters in the Euclidean norm

$$\arg\min_P \left[\left\| W_1^A - P\, W_1^B \right\|_F^2 + \left\| W_2^A - W_2^B P^T \right\|_F^2 \right]$$

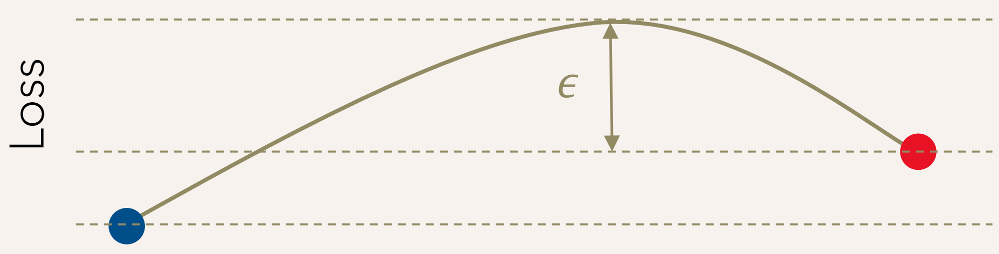
$$= \arg\max_P : \left\langle P, \mathrm{W}_1^{\mathrm{A}} \left(\mathrm{W}_1^{\mathrm{B}}\right)^{\mathrm{T}} + \left(\mathrm{W}_2^{\mathrm{A}}\right)^{T} \mathrm{W}_2^{\mathrm{B}} \right\rangle_{\mathbf{F}}$$

- Can be solved efficiently as linear assignment problem for 2-layer networks

- NP-hard for deeper networks, so greedy iterative layer-wise reparameterization

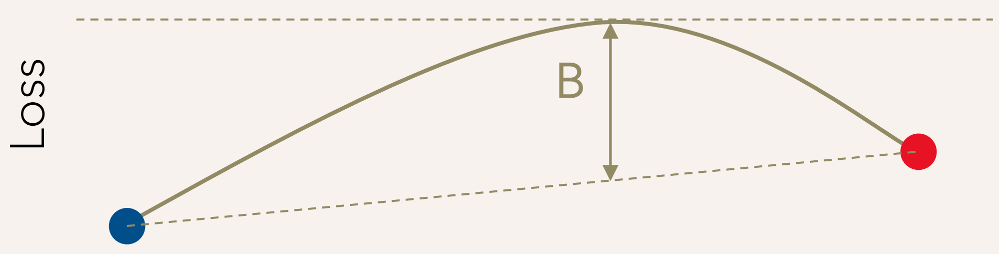
Measure of linear mode connectivity

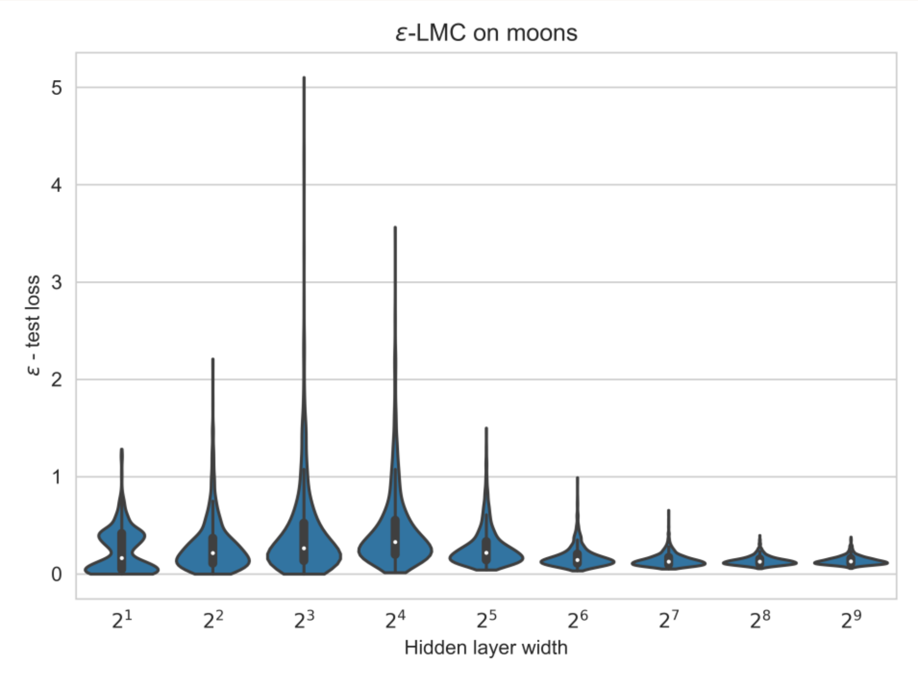
- ϵ-mode connected

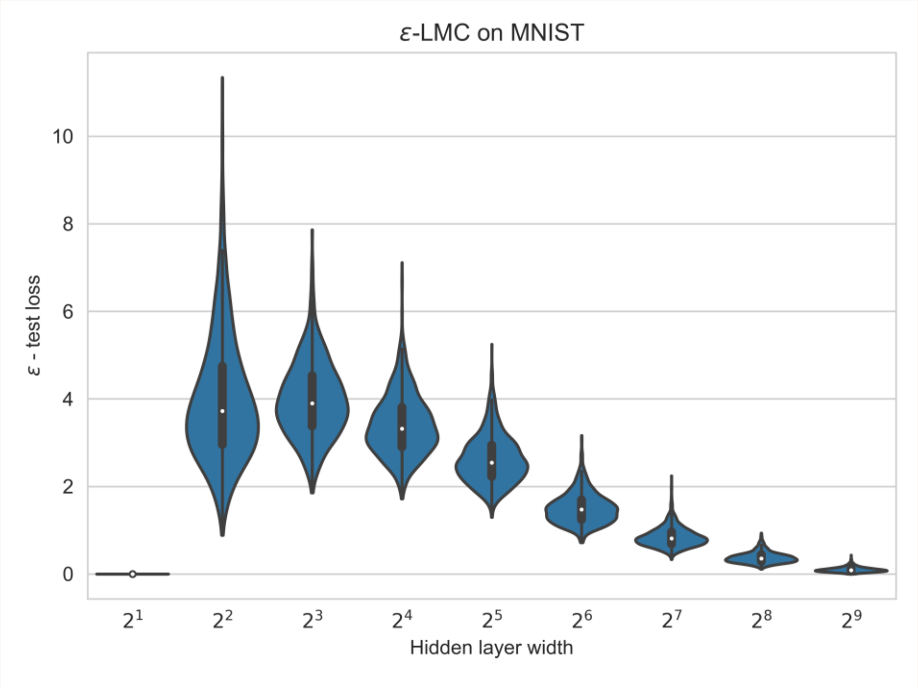


- We will use this

- Loss barrier

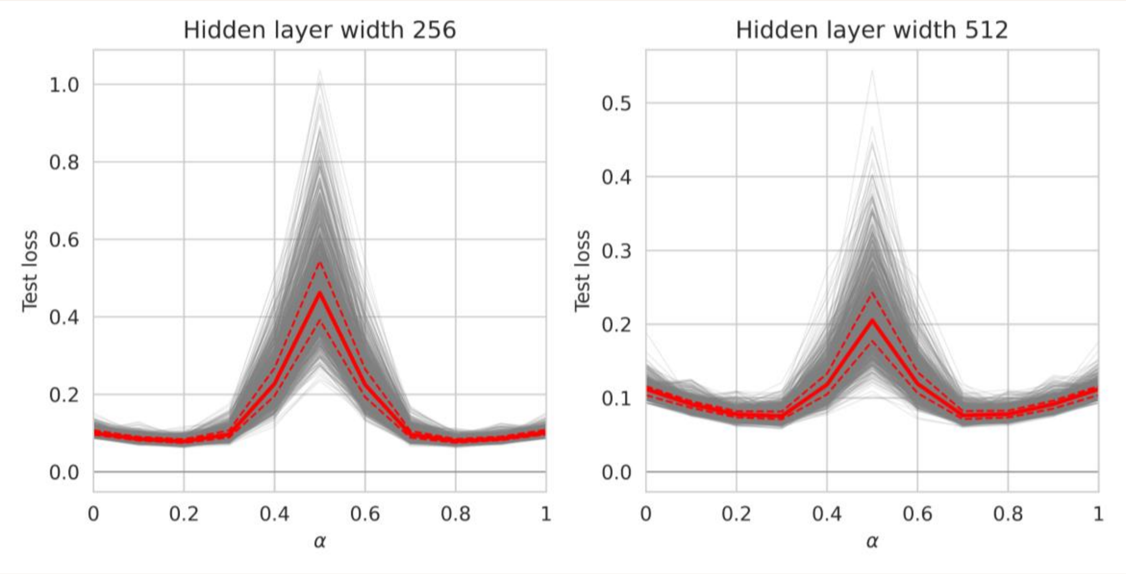




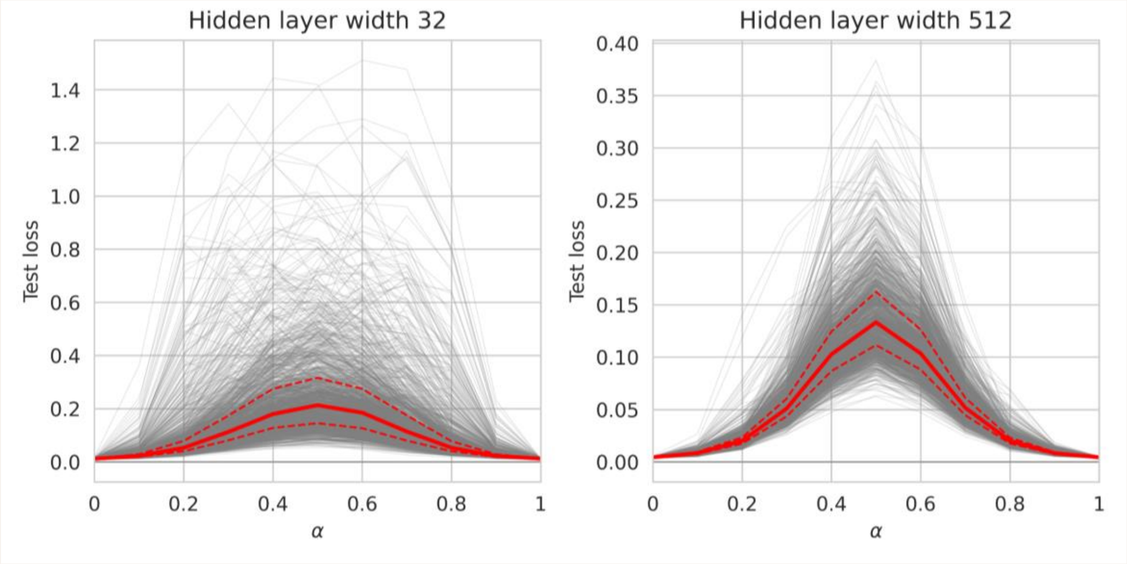


Interpolation: naïve networks

Networks are more linearly mode connected with increasing hidden layer width even without reparameterization



Moons



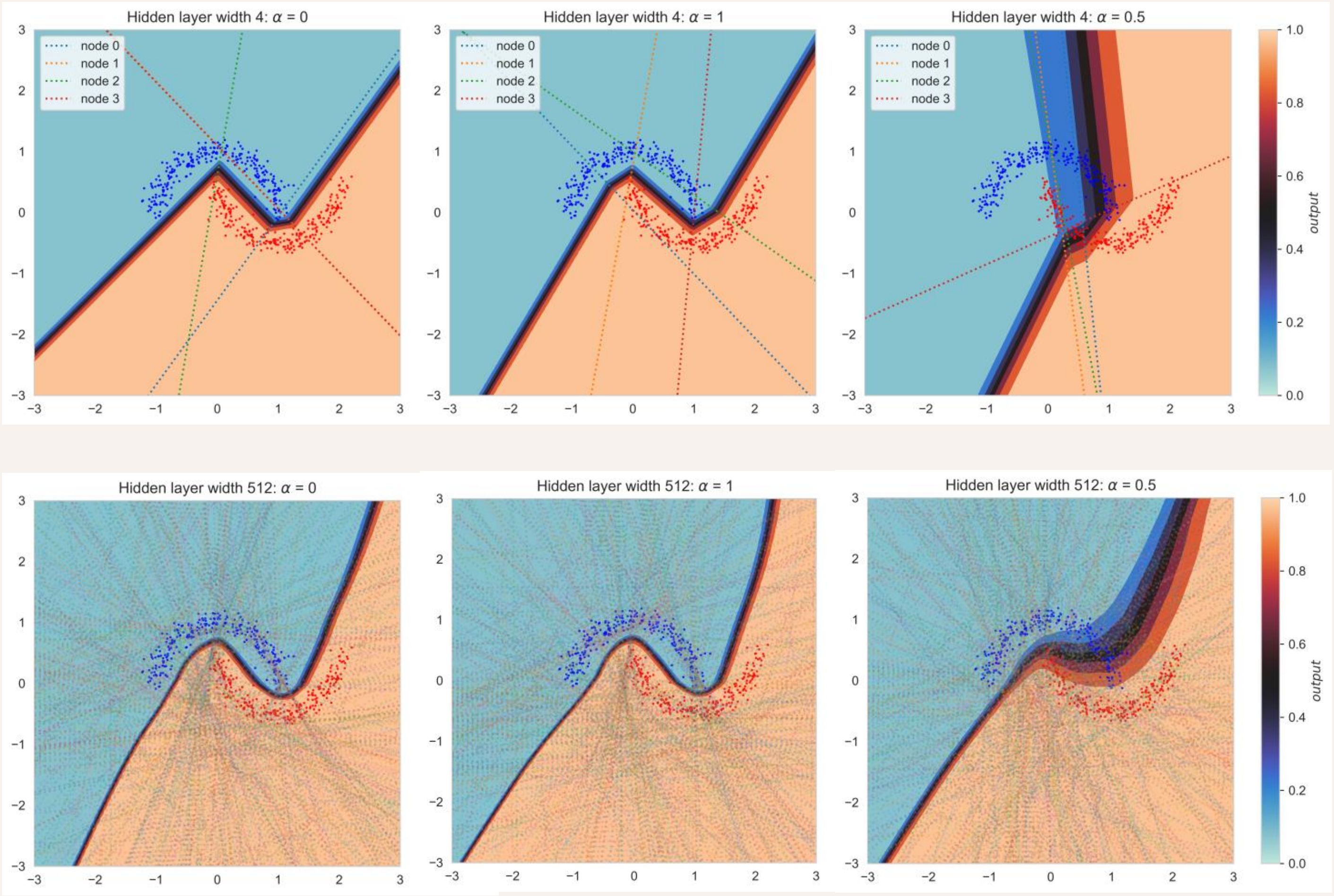
MNIST

Naïve interpolation: a closer look

Average of two trained networks is worse, but not by much

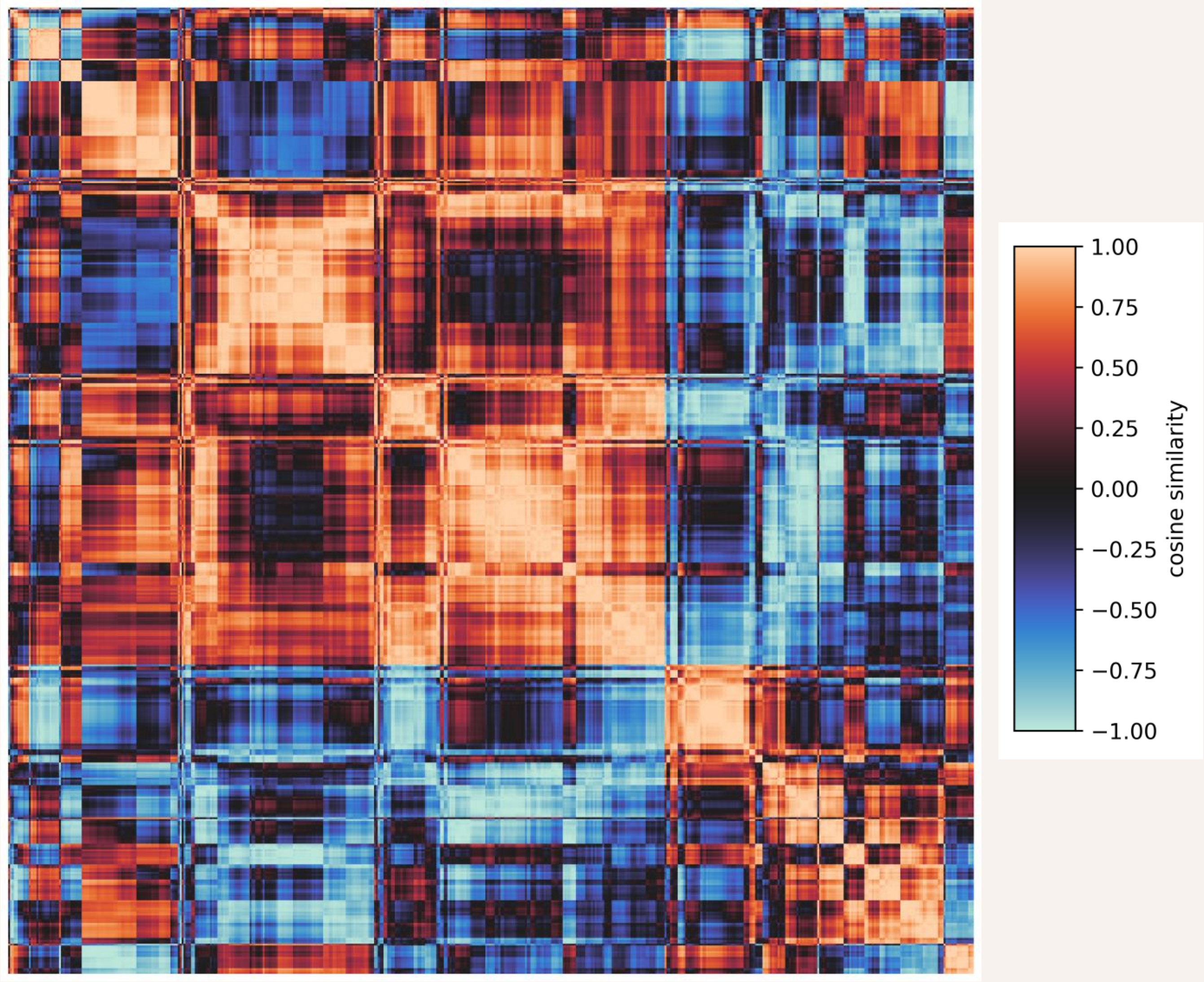
Linear regions and decision boundaries

In wide networks, multiple hidden units compute the same features



Feature similarity

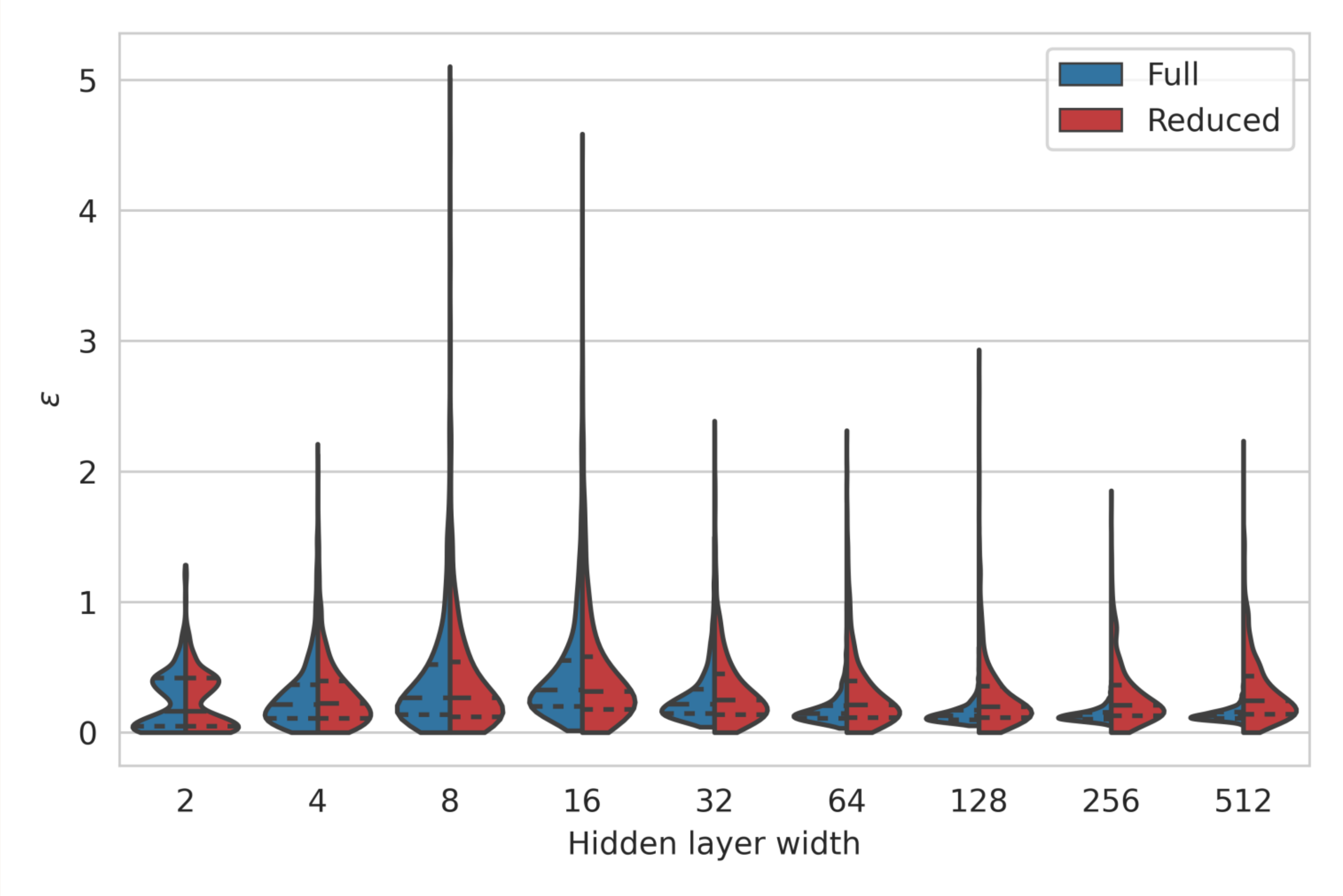
Redundancies in features lead to lowering loss barrier in wide networks, as hidden units are more likely to be aligned

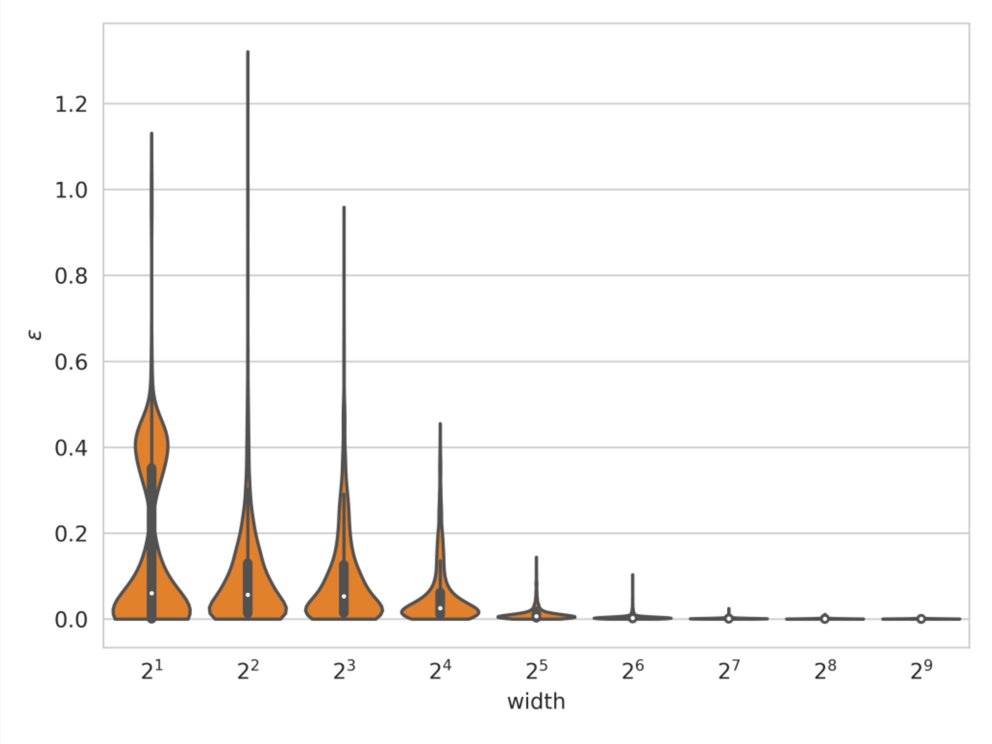


So, what if we prune the networks?

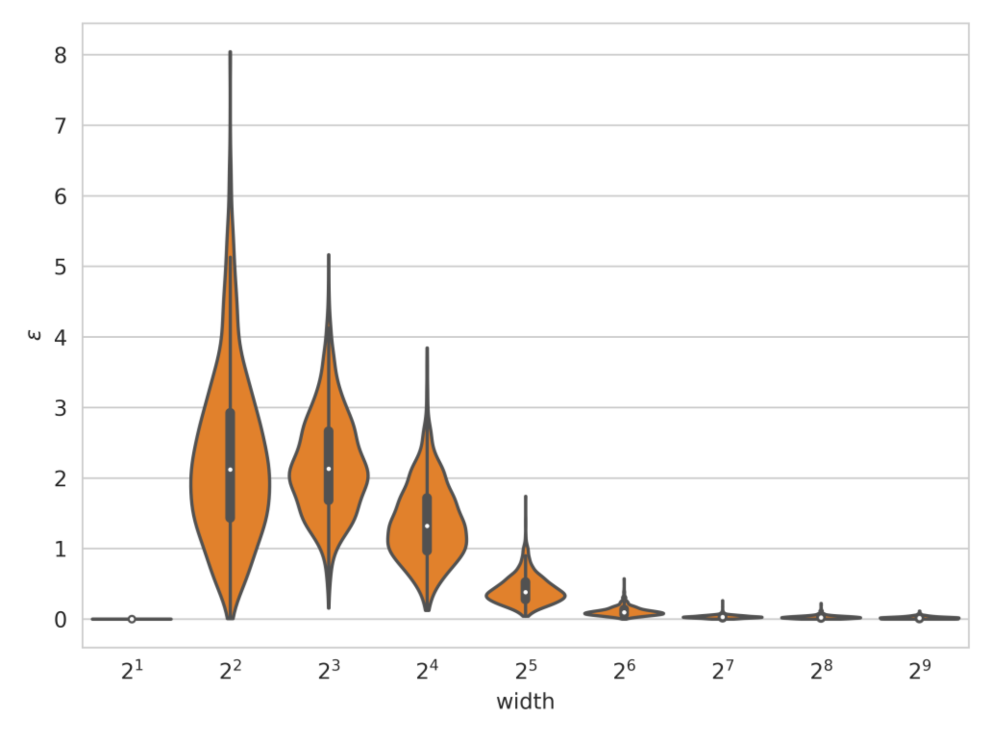
Configuration	Width	Test Loss	Test Accuracy
Full	2	0.285 ± 0.084	0.859 ± 0.077
Reduced	2 ± 0.0	0.285 ± 0.084	0.859 ± 0.077
Full	4	0.206 ± 0.092	0.903 ± 0.049
Reduced	3.720 ± 0.492	0.220 ± 0.111	0.897 ± 0.056
Full	8	0.124 ± 0.105	0.941 ± 0.057
Reduced	7.180 ± 0.865	0.148 ± 0.126	0.934 ± 0.062
Full	16	0.028 ± 0.043	0.991 ± 0.024
Reduced	13.040 ± 1.216	0.057 ± 0.086	0.979 ± 0.037
Full	32	0.012 ± 0.002	0.996 ± 0.001
Reduced	23.280 ± 1.887	0.045 ± 0.088	0.986 ± 0.026
Full	64	0.009 ± 0.002	0.997 ± 0.001
Reduced	39.260 ± 1.598	0.050 ± 0.111	0.985 ± 0.028
Full	128	0.006 ± 0.001	0.998 ± 0.001
Reduced	73.620 ± 1.340	0.047 ± 0.122	0.987 ± 0.023
Full	256	0.005 ± 0.001	0.999 ± 0.001
Reduced	141.020 ± 2.005	0.048 ± 0.148	0.987 ± 0.027
Full	512	0.004 ± 0.0	0.998 ± 0.001
Reduced	271.940 ± 2.176	0.027 ± 0.052	0.990 ± 0.019

Removing feature redundancies removes LMC upon naïve interpolation





Moons

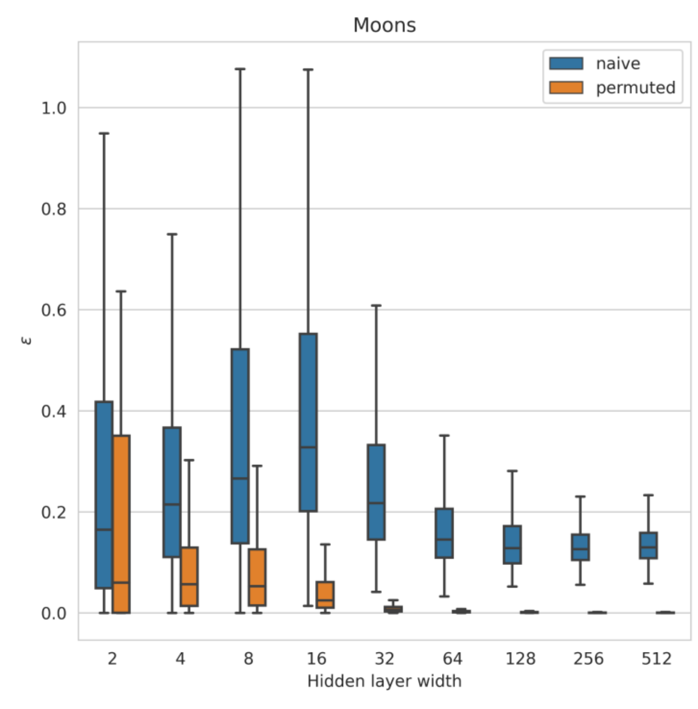


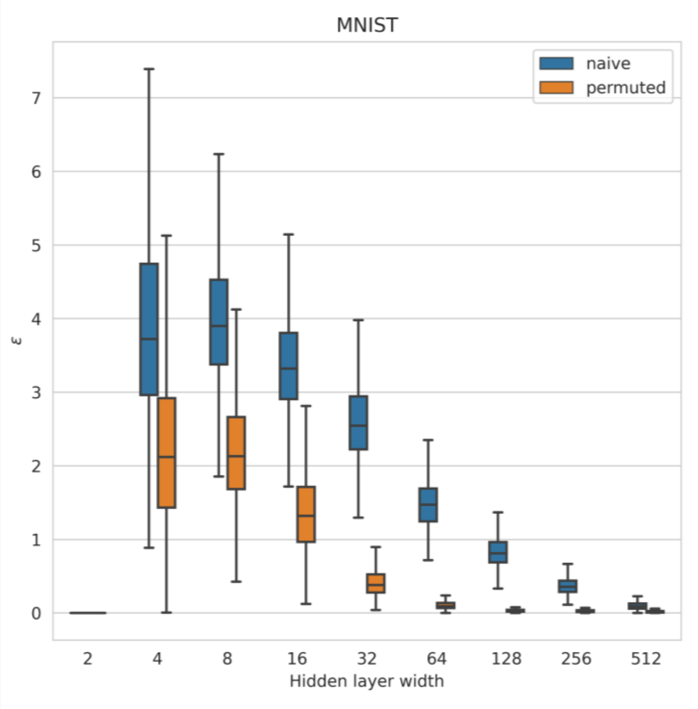
MNIST

Interpolation: reparameterized networks

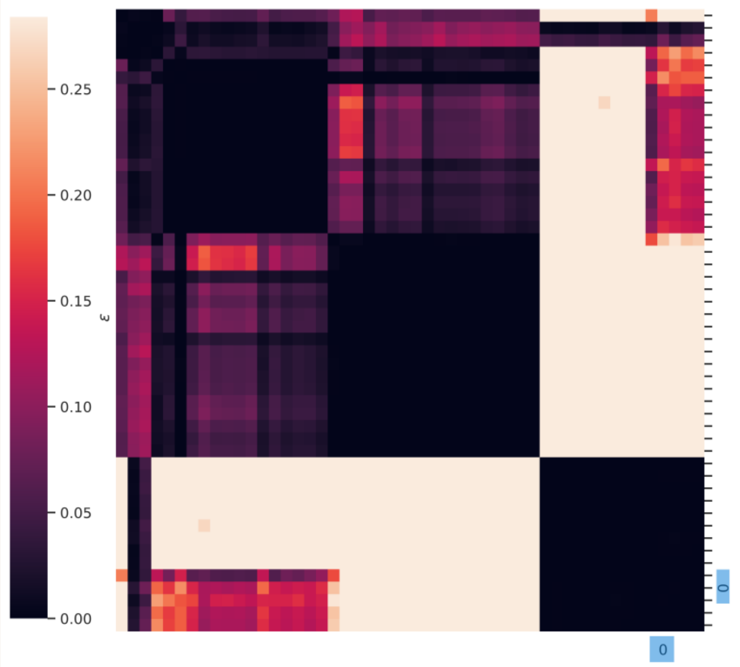
Networks are lot more linearly mode connected upon reparameterization

Comparison between naïve and reparameterized interpolations

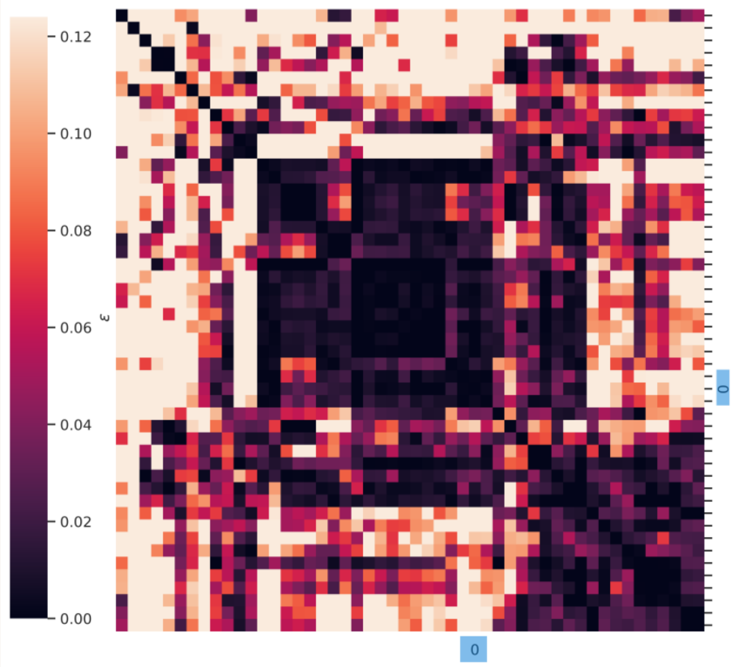




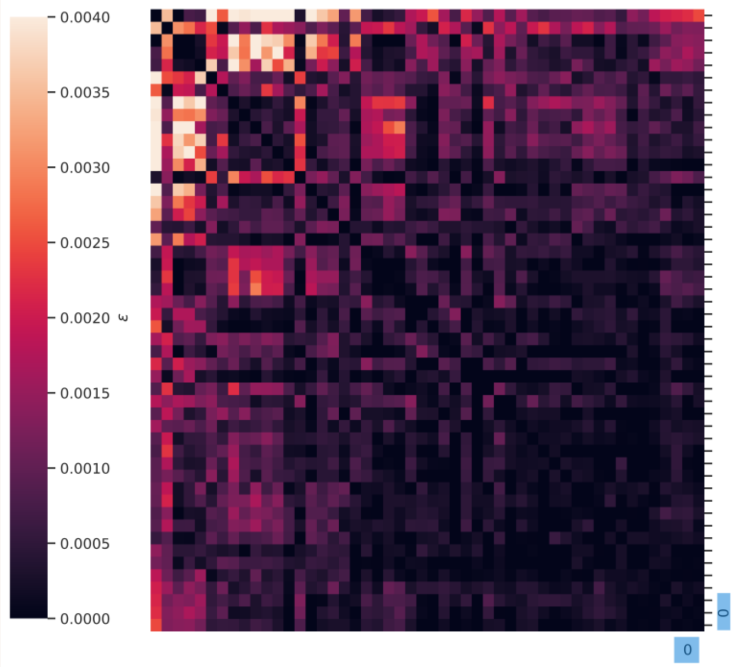
Outliers are omitted from the plots



Width: 2



Width: 8

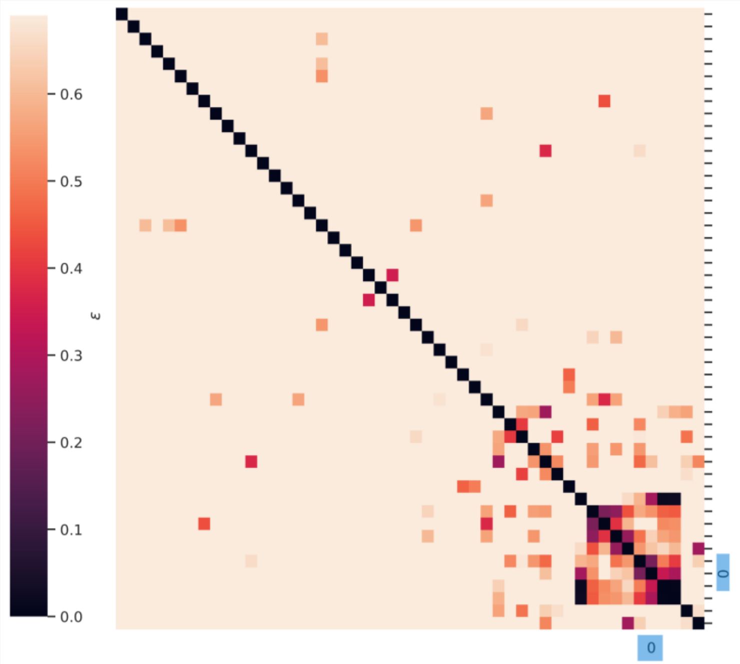


Width: 512

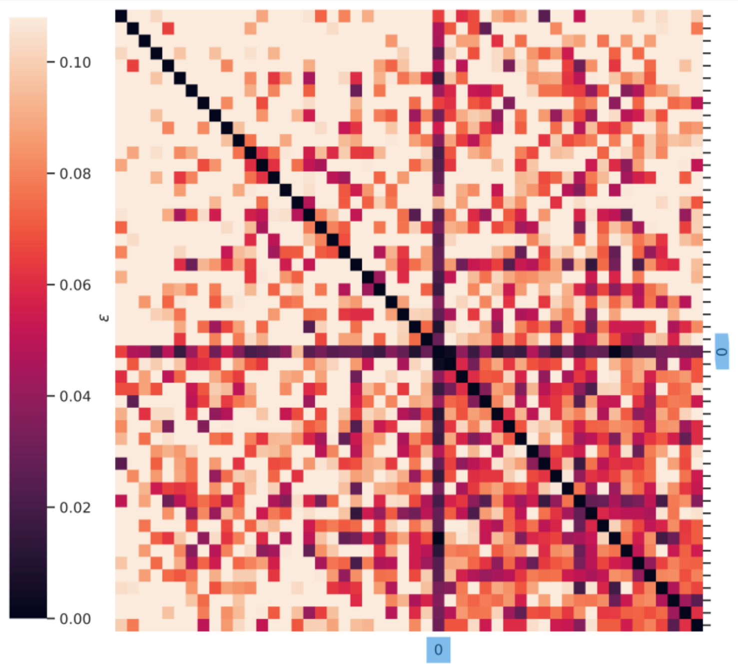
Are there clusters of Moons networks that are LMC with each other and not others?

Using ε-loss as a measure of similarity, we do hierarchical clustering of networks. Colors are scaled so mean test loss is the maximum

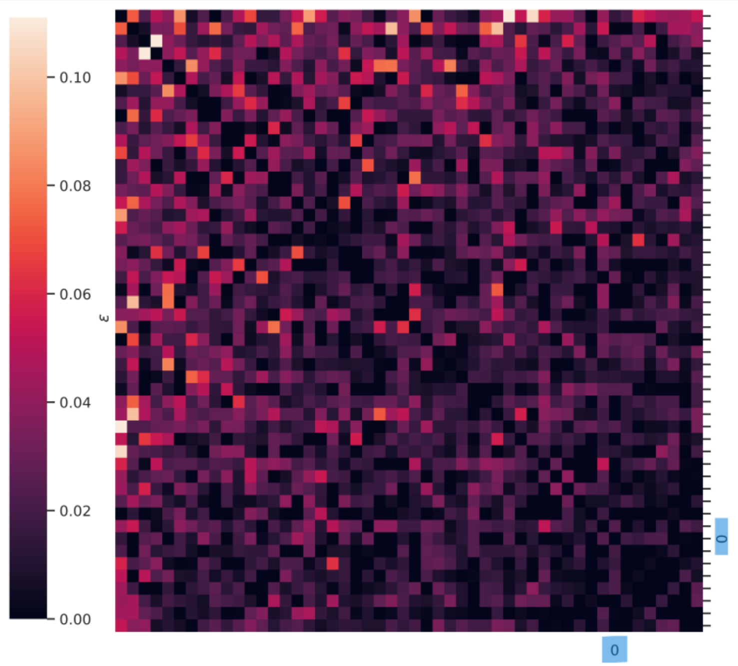
Note: ε-loss is not a true metric as triangle inequality does not hold



Width: 4



Width: 64



Width: 512

Are there clusters of MNIST networks that are LMC with each other and not others?

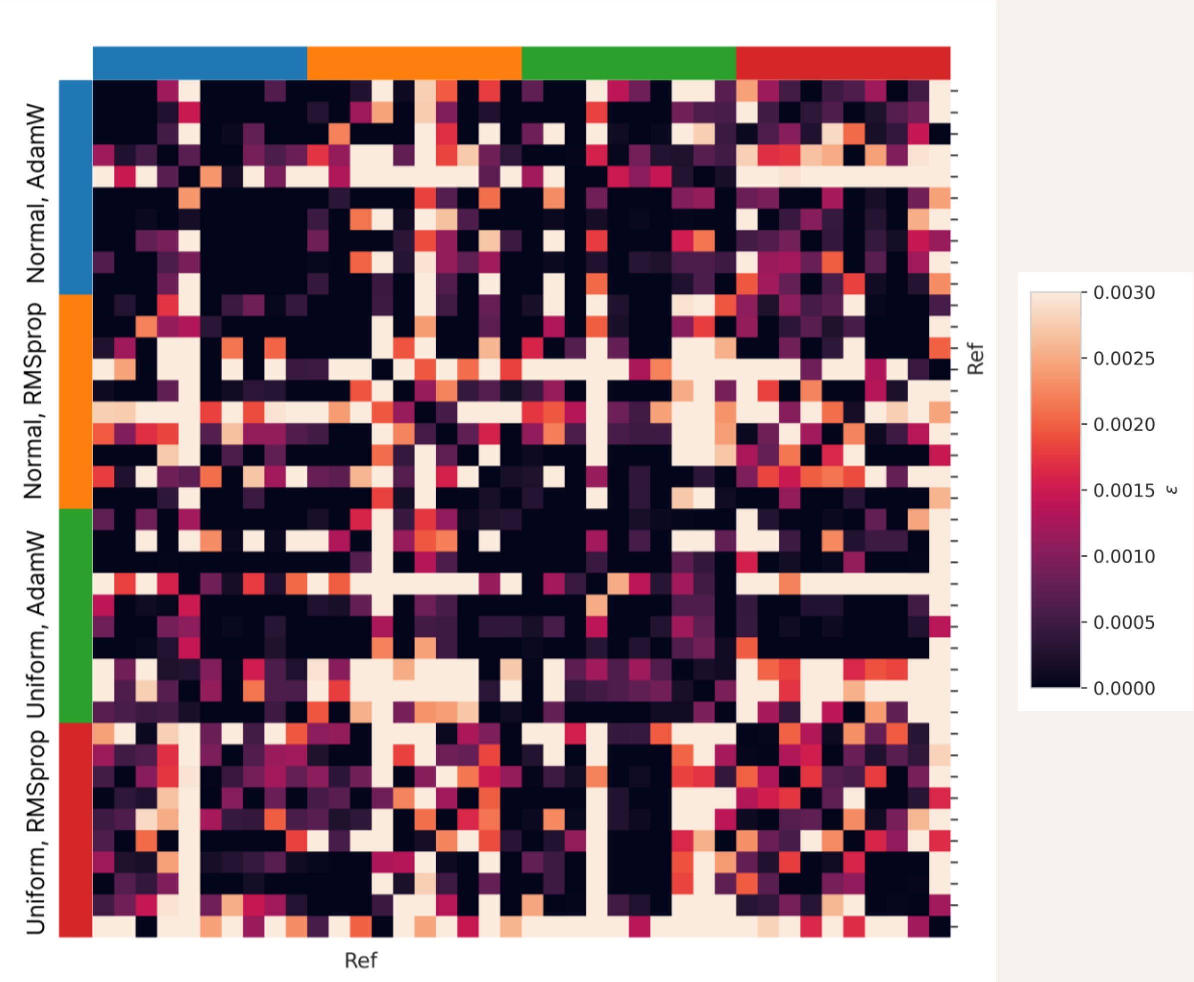
Using ε-loss as a measure of similarity, we do hierarchical clustering of networks. Reference appears to affect the result

Note: ε-loss is not a true metric as triangle inequality does not hold

How robust are results to initialization & optimizer?

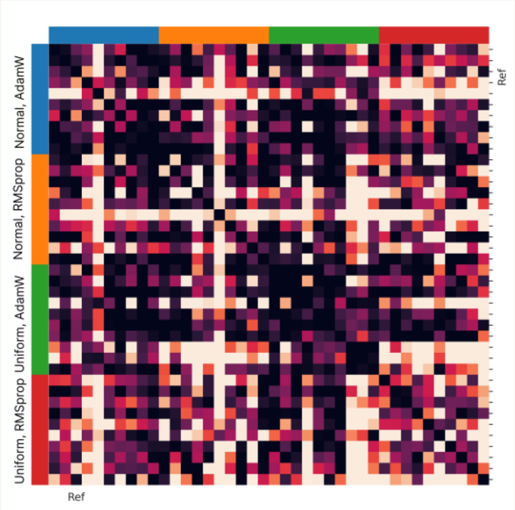
Config.	Train loss	Test loss	Train acc.	Test acc.
Norm., AdamW	0.0 ± 0.0	0.003 ± 0.0	1.0 ± 0.0	0.998 ± 0.0
Norm., RMSprop	0.0 ± 0.0	0.003 ± 0.001	1.0 ± 0.0	0.999 ± 0.001
Unif., AdamW	0.001 ± 0.0	0.003 ± 0.0	1.0 ± 0.0	0.998 ± 0.001
Unif., RMSprop	0.0 ± 0.0	0.003 ± 0.0	1.0 ± 0.0	0.999 ± 0.001

RMSprop networks on Moons appear to be less connected

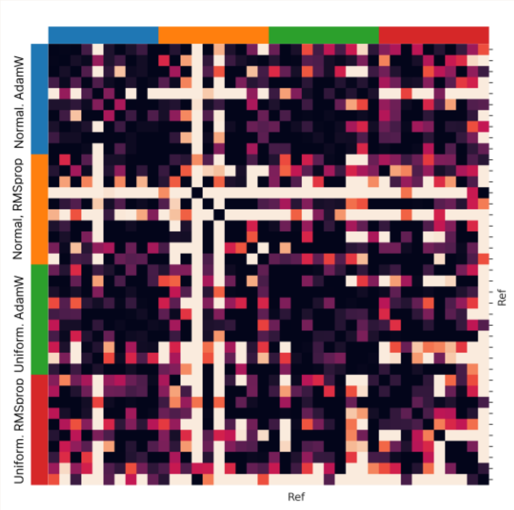


How robust are results to reference choice?

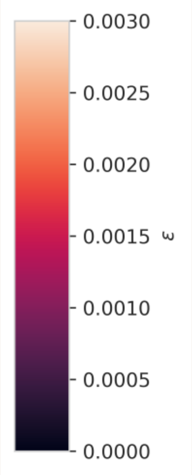
Reference choice impacts the results, but qualitatively similar

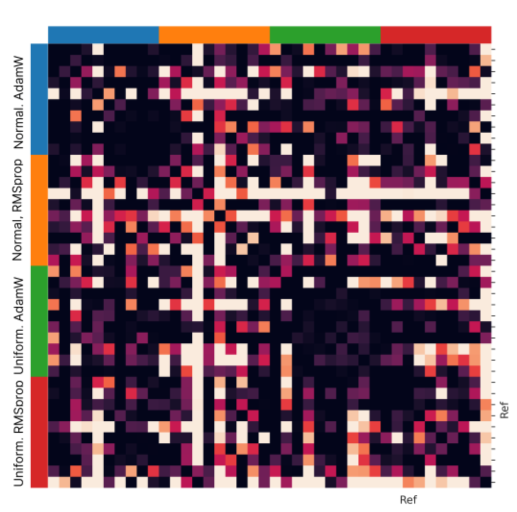


Normal, AdamW

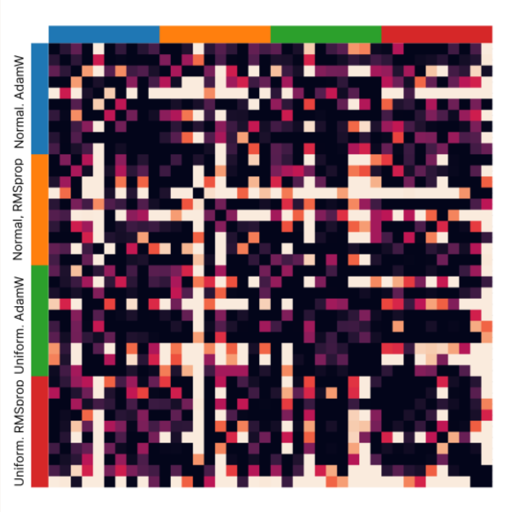


Uniform, AdamW

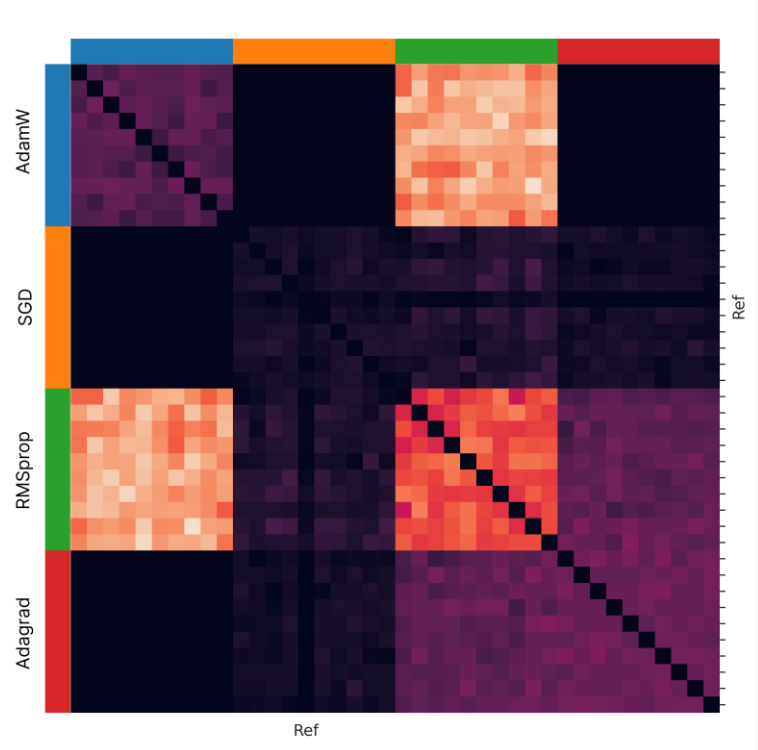




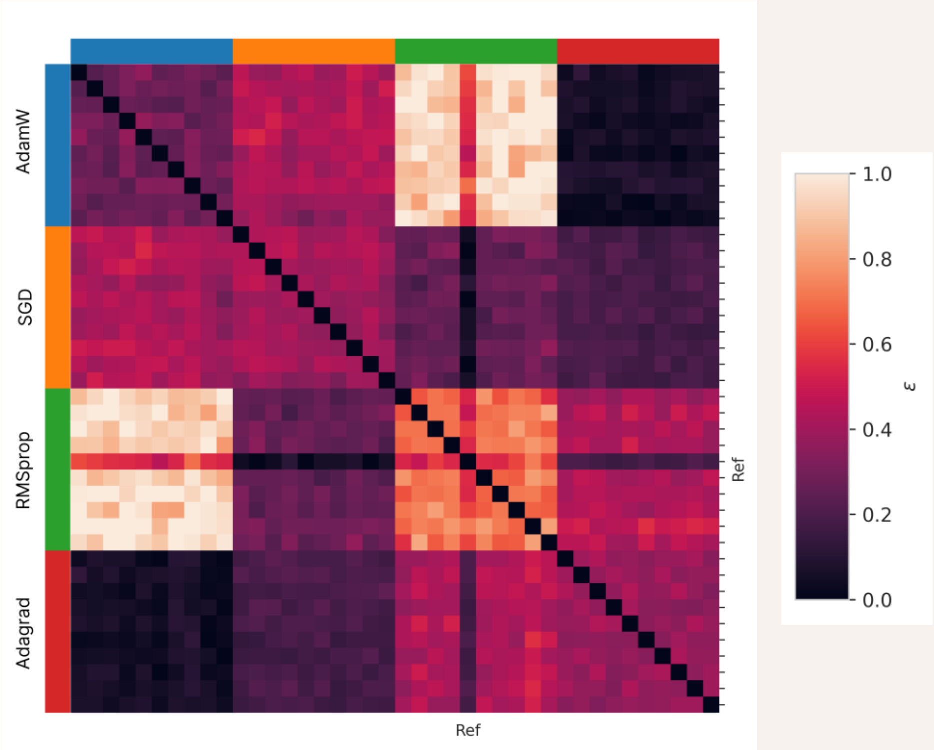
Uniform, RMSprop



Pairwise



Ref: SGD

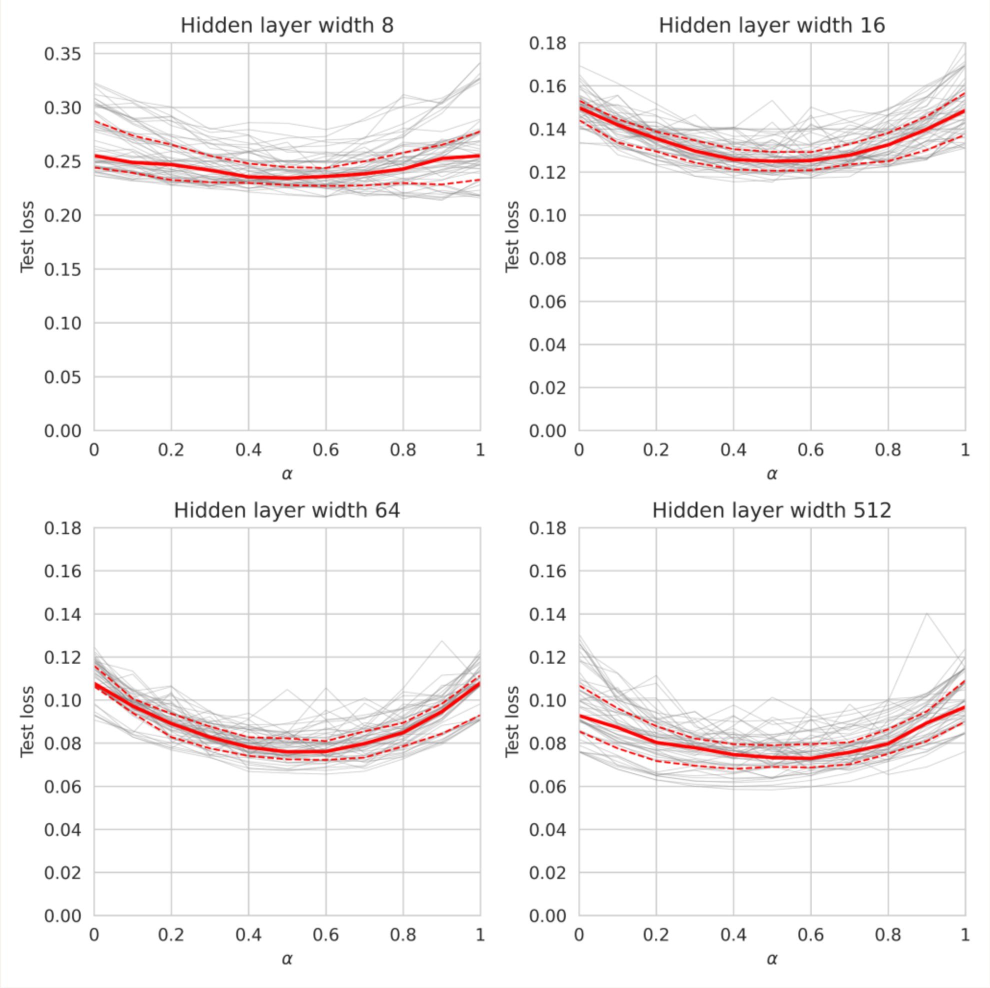


Ref: RMSprop

How robust are results to data complexity?

4-layer fully connected networks trained on CIFAR 10

Analyzing SWA



Starting with trained MNIST network, we run for 20 more epochs, and collect weights at end of each epoch as samples.

The loss along pairs of these samples are shown for different hidden layer widths.

SWA works as the samples are implicitly permutation aligned

Closing Remarks