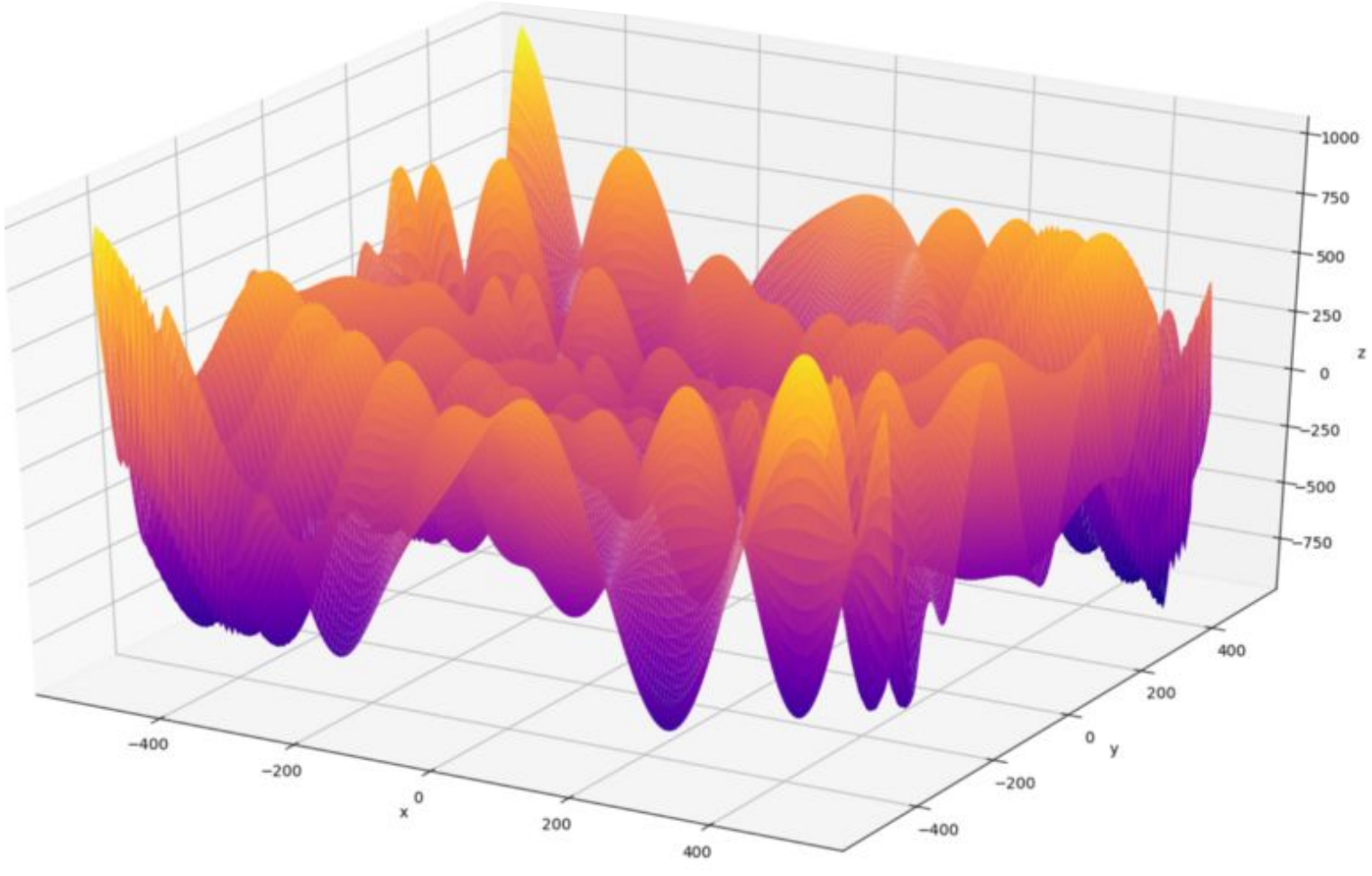
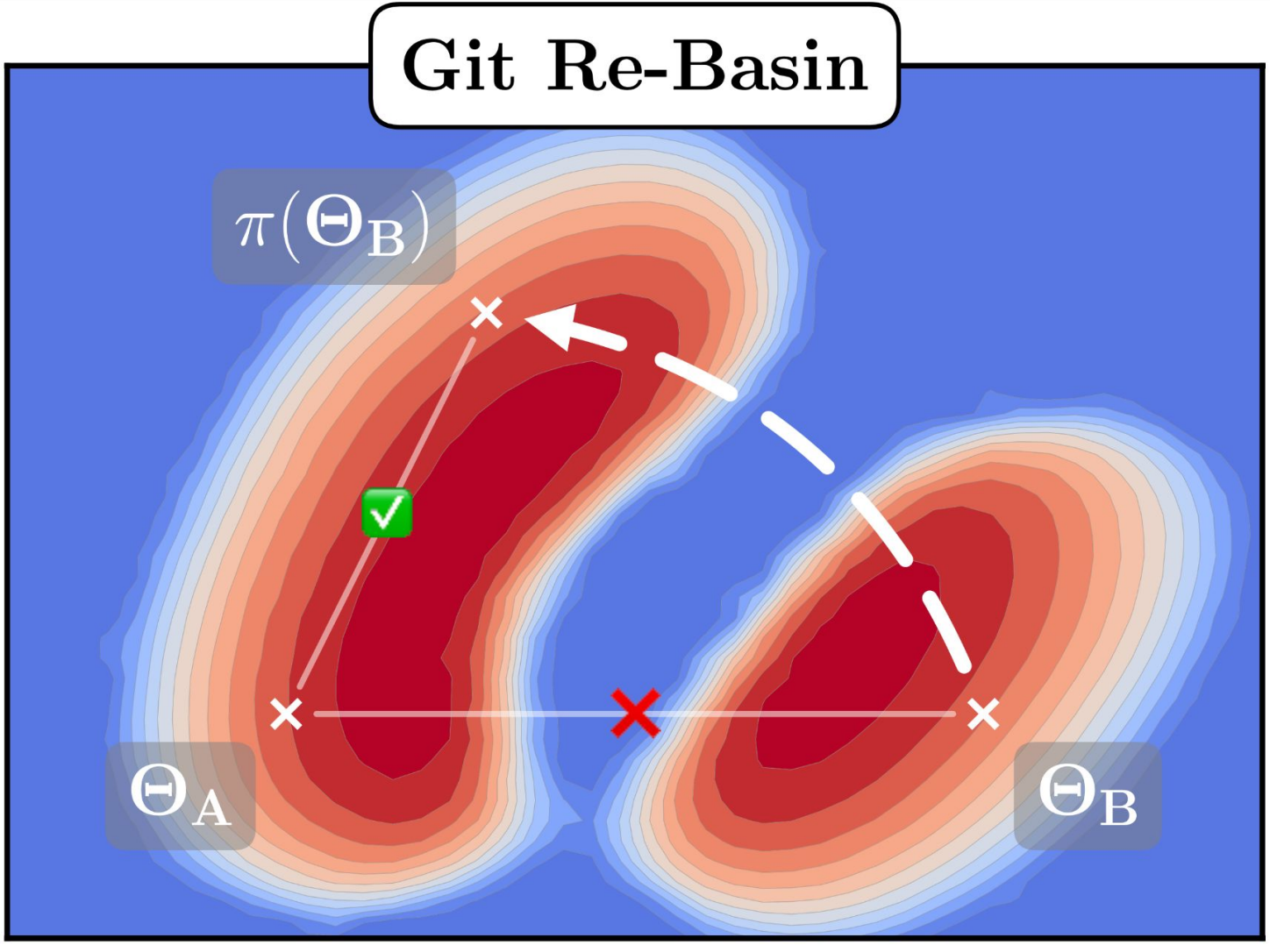
Git Re-Basin: Merging Models modulo Permutation Symmetries

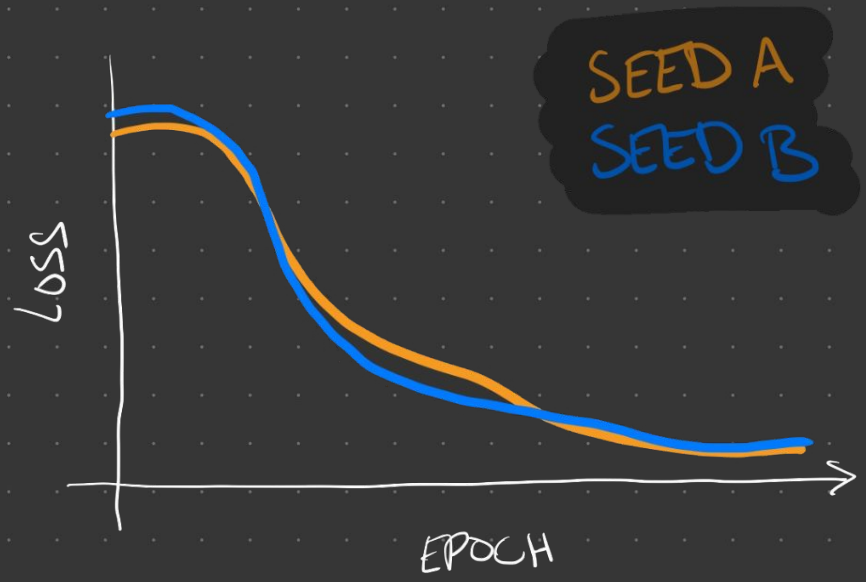
Samuel K. Ainsworth, Jonathan Hayase, Siddhartha Srinivasa


1000
750
500
250
0
−250
−500
−750
z
−400
−200
0
200
400
x
400
200
0
−200
−400
y


Git Re-Basin
$\pi(\Theta_B)$
Θ_A
Θ_B

Unreasonable effectiveness of SGD

1. Why does SGD work in deep learning but fail elsewhere (policy learning, traj. opt., recommender systems)?
2. Where are all the local minima?
3. Why do independently trained models have the same training dynamics?



Idea: permutation symmetries to blame?

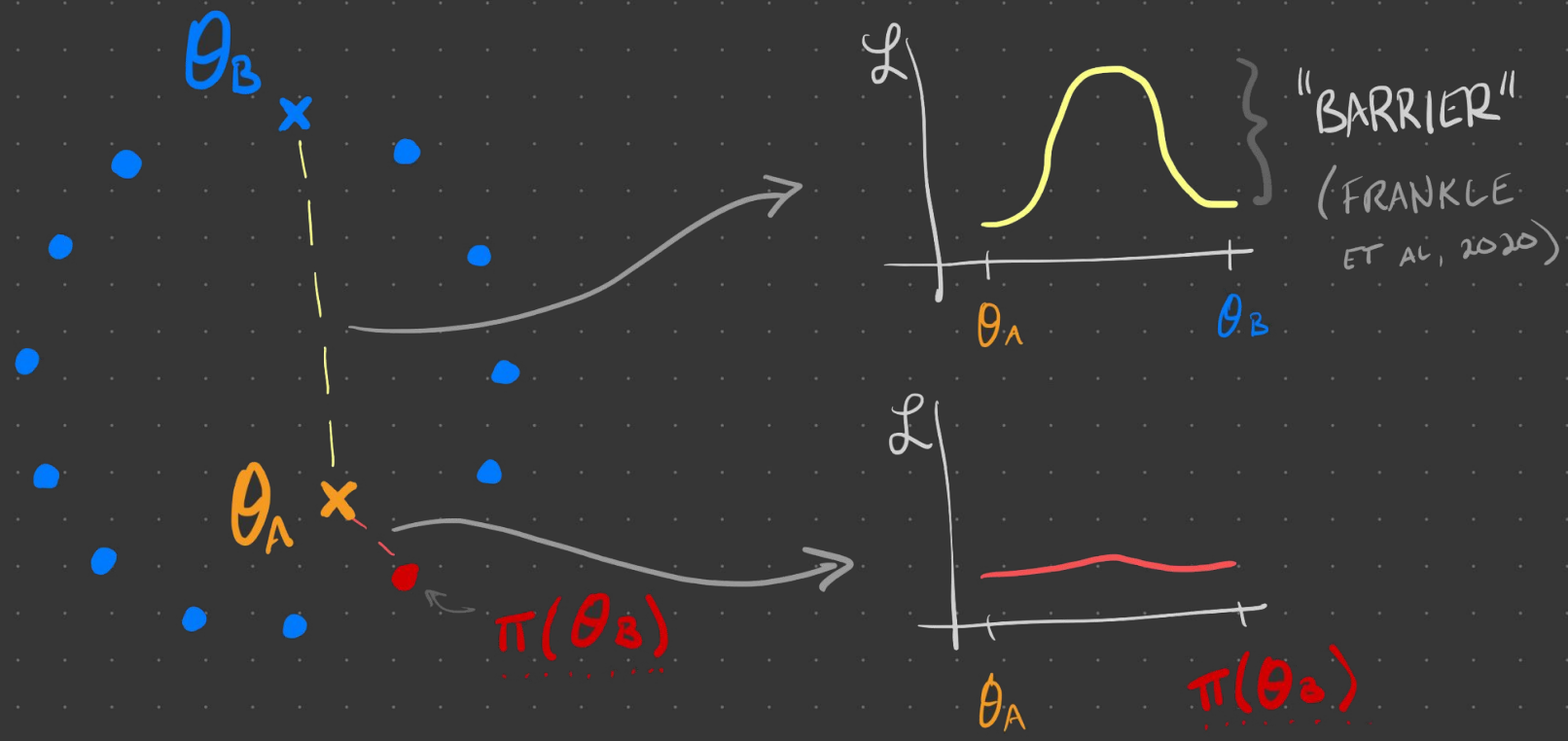
swap hidden units: functionally-equivalent models, different parameters.

ResNet50: 10^55109

atoms in universe: 10^82

See also *Entezari et al, 2021*!

Idea: permutation symmetries to blame?



How to find π?

methods

Alg. 1: Activation matching

IDEA: SIMILAR UNITS OUGHT TO HAVE CORRELATED ACTIVATIONS!

ALGORITHM: AT EACH LAYER, COLLECT ACTIVATIONS FROM BOTH

OLS CONSTRAINED TO S_d:

$$\underset{P \in S_d}{\operatorname{argmin}} \sum_{i=1}^{N} \| Z^{(A)}_{:,i} - P Z^{(B)}_{:,i} \|^2$$

$$= \underbrace{\underset{P \in S_d}{\operatorname{argmax}} \left\langle P, Z^{(A)} Z^{(B)T} \right\rangle_F}$$

"LINEAR ASSIGNMENT PROBLEM"

$\Rightarrow$ SOLVE W/ HUNGARIAN ALGO, ETC.

Alg. 2: Weight matching

IDEA: SIMILAR UNITS OUGHT TO HAVE SIMILAR WEIGHTS.

$$\underset{\pi}{\arg\min} \underbrace{\| \mathrm{vec}(\theta_A) - \mathrm{vec}(\pi(\theta_B)) \|^2}_{L2} = \underset{\pi}{\arg\max} \underbrace{\mathrm{vec}(\theta_A) \cdot \mathrm{vec}(\pi(\theta_B))}_{\text{INNER PROD.}}$$

$$= \underset{\pi = \{P_i\}}{\arg\max} \left\langle W_1^{(A)}, P_1 W_1^{(B)} \right\rangle_F + \left\langle W_2^{(A)}, P_2 W_2^{(B)} P_1^T \right\rangle_F + \cdots + \left\langle W_L^{(A)}, W_L^{(B)} P_{L-1}^T \right\rangle_F$$

LEMMA: THIS IS NP-HARD!

ALGORITHM:

- RANDOMLY PICK P_i,
- SOLVE JUST THAT ONE (REDUCES TO LINEAR ASSIGNMENT PROBLEM),
- REPEAT UNTIL CONVERGENCE.

Alg. 3: Straight-through estimator

IDEA: LEARN $\pi(\theta_B)$ BASED ON THE DATA DISTRIBUTION.

$$\min_{\tilde{\theta}_B} \mathcal{L}\left(\frac{1}{2}(\theta_A + \text{proj}(\tilde{\theta}_B))\right) \qquad \text{proj}(\theta) \triangleq \underset{\pi(\theta_B)}{\arg\max}\ \text{VEC}(\theta) \cdot \text{VEC}(\pi(\theta_B))$$

PROBLEM: proj IS NOT DIFFERENTIABLE...

$\Rightarrow$ USE STRAIGHT-THRU ESTIMATOR!

$\Rightarrow$ SOLVE proj W/ WEIGHT MATCHING ALGO,

FWD PASS: USE $\text{proj}(\tilde{\theta}_B)$

BWD PASS: USE $\tilde{\theta}_B$

Bonus: Merging more than 2 models?

Algorithm 3: MERGEMANY

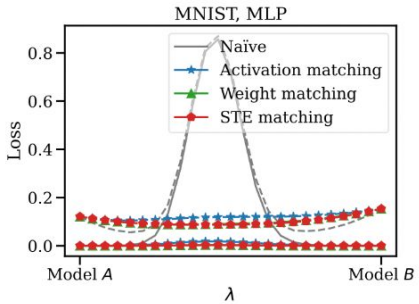
Given: Model weights $\Theta_1, \ldots, \Theta_N$

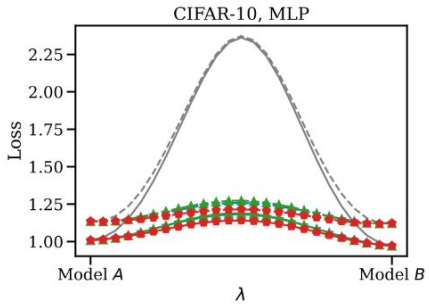
Result: A merged set of parameters $\tilde{\Theta}$.

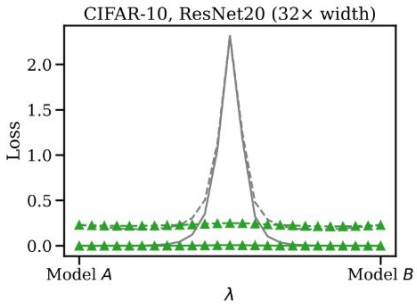
repeat
  for $i \in$ RANDOMPERMUTATION$(1, \ldots, N)$ **do**
    $\Theta' \leftarrow \frac{1}{N-1} \sum_{j \in \{1,\ldots,N\} \setminus \{i\}} \Theta_j$
    $\pi \leftarrow$ PERMUTATIONCOORDINATEDESCENT(Θ', Θ_i)
    $\Theta_i \leftarrow \pi(\Theta_i)$
  end
until *convergence*
return $\frac{1}{N} \sum_{j=1}^{N} \Theta_j$

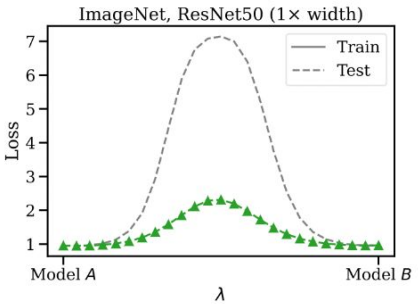
experiments!

LMC before/after matching

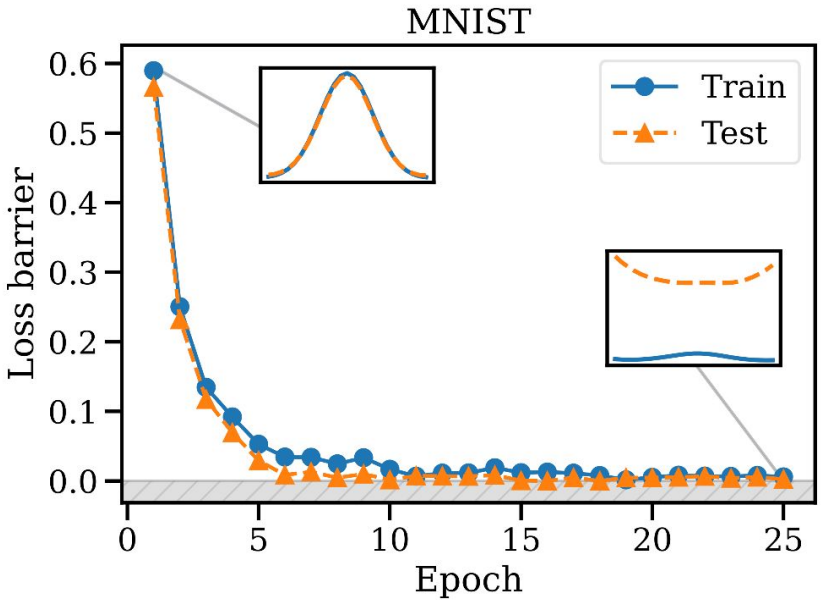


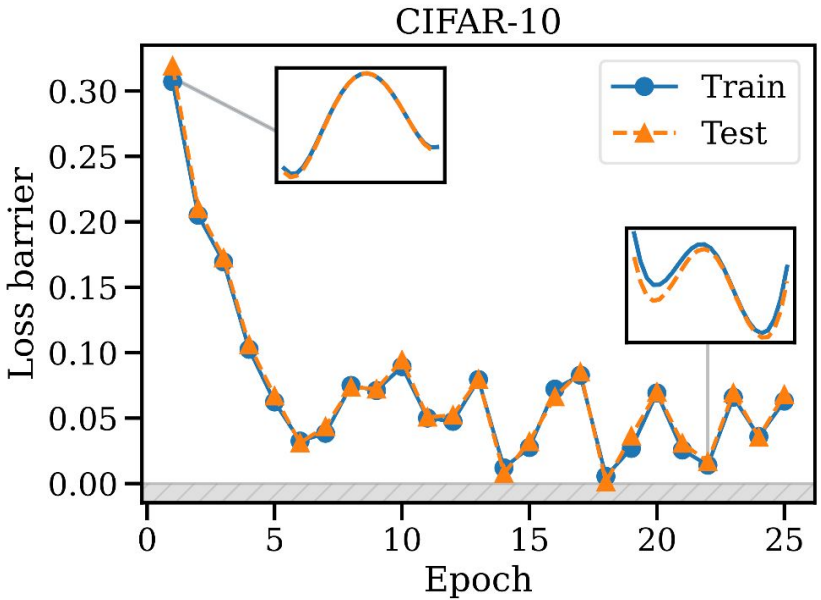






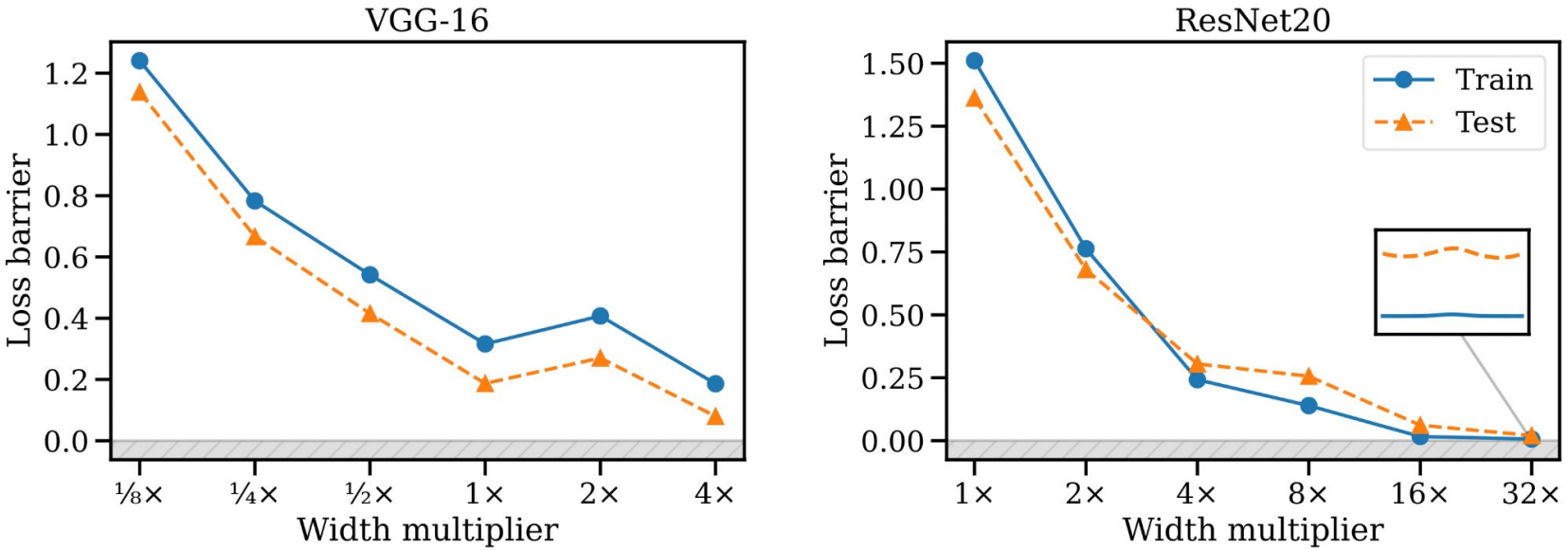
LMC is an emergent property of training



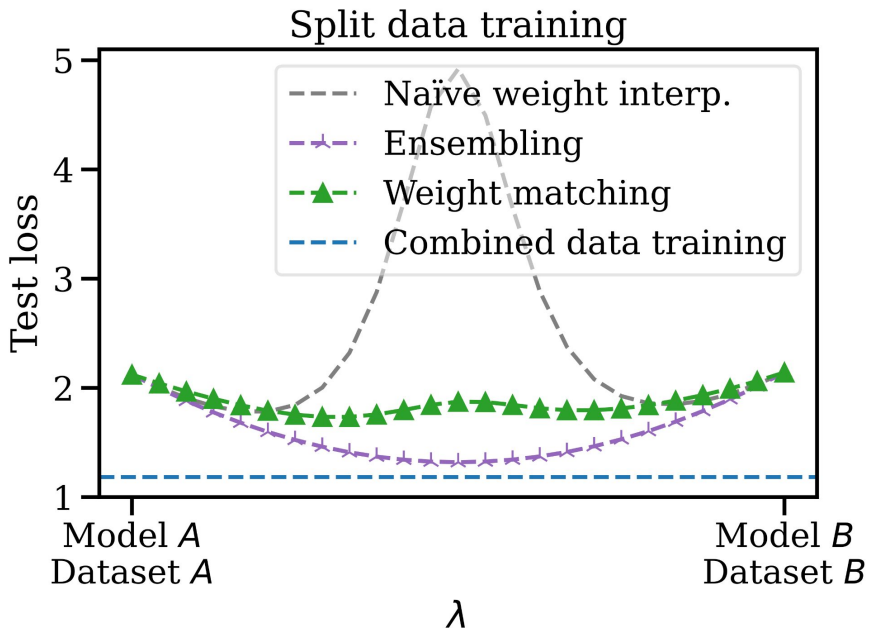


See also Benzing et al, 2022!

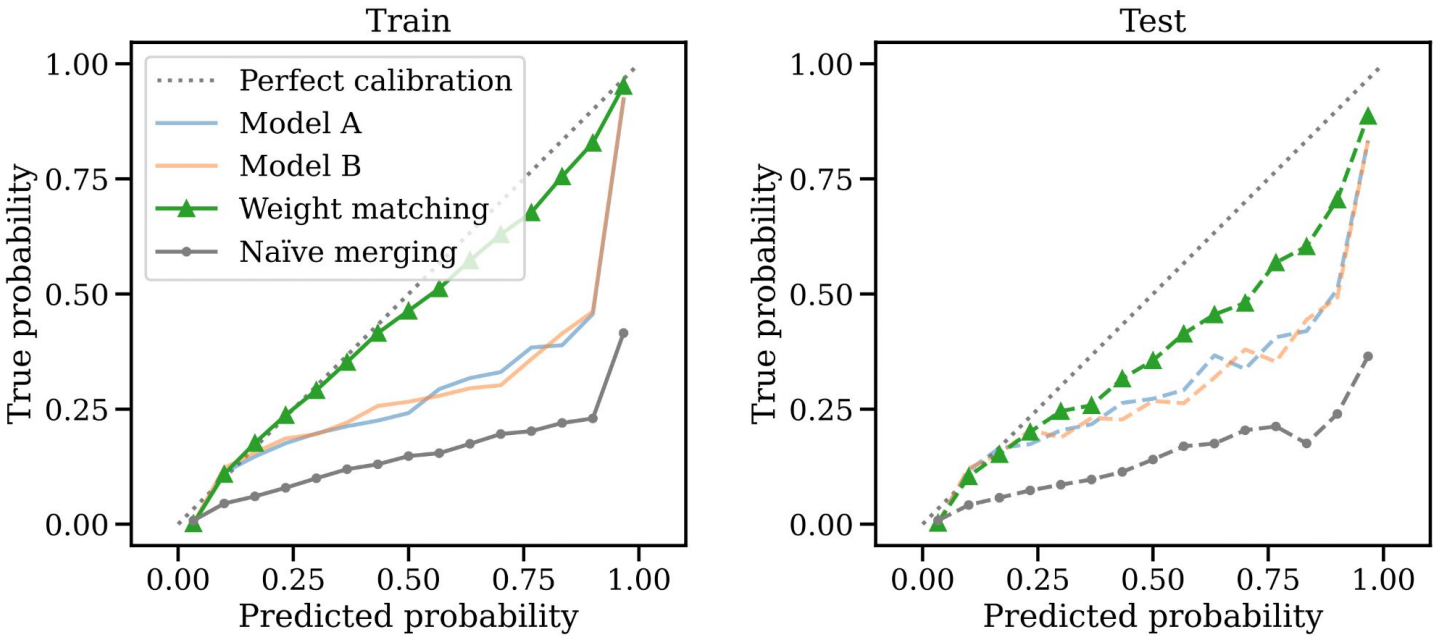
Wider models are better models



Model patching/disjoint datasets



Model patching/disjoint datasets, calibration

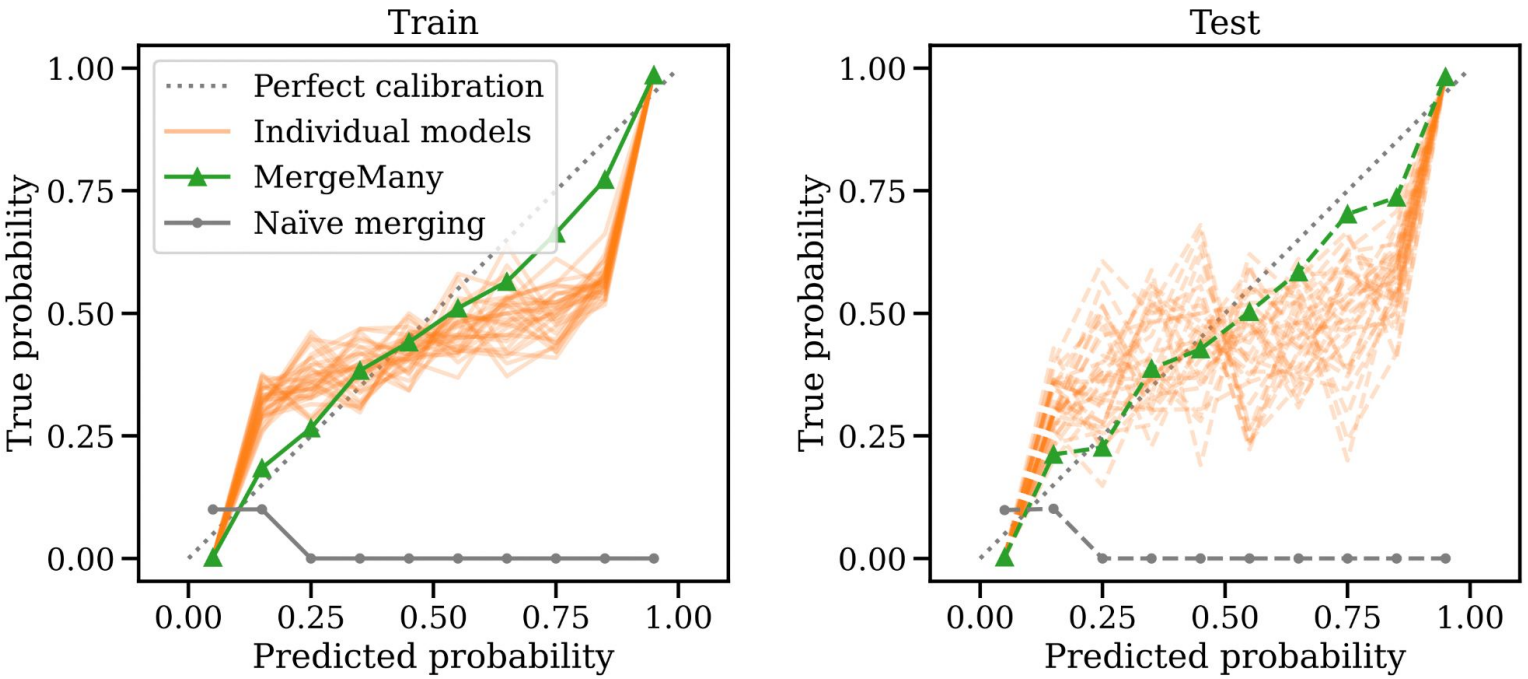


Bonus: MergeMany results

	Train Loss	Train Acc.	Test Loss	Test Acc.
Seed 1	0.0000	1.0000	0.1153	0.9856
Seed 2	0.0000	1.0000	0.1531	0.9854
Seed 3	0.0000	1.0000	0.1229	0.9855
Seed 4	0.0000	1.0000	0.1108	0.9865
Seed 5	0.0000	1.0000	0.1443	0.9871
MERGEMANY	0.0141	0.9952	**0.0727**	0.9831

43% decrease in test loss!

Bonus: MergeMany results (con't)



conclusion

1.

Loss landscapes seem to contain only a single basin mod. permutation symmetries in many settings. And wider is better.

2.

Independently trained models can be merged in weight space by teleporting into the same basin.

3.

Merged models are better calibrated and tend to outperform single models on test loss. Methodology for improving model performance?

Open questions and future work

- Federated learning? Distributed training?
- Cross validation for deep learning?
- What about thin models? They seem to exhibit similar behavior in training but don't work as well as wide ones...
  - Are thin models just wide models in superposition? Connection to optimal transport?
  - Hypothesis: Activations between different models can be linearly related. In the infinite width limit it just so happens that a sufficient permutation relationship exists.
- When does Git Re-Basin fail and why? Why is SGD implicitly biased towards solutions that admit LMC?
- Security implications for model merging? How safe is your data?

Thanks to my collaborators!

Samuel Ainsworth
(UW, Cruise AI Research)

Jonathan Hayase
(UW)

Sidd Srinivasa
(UW)

thank you!

questions?