The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks

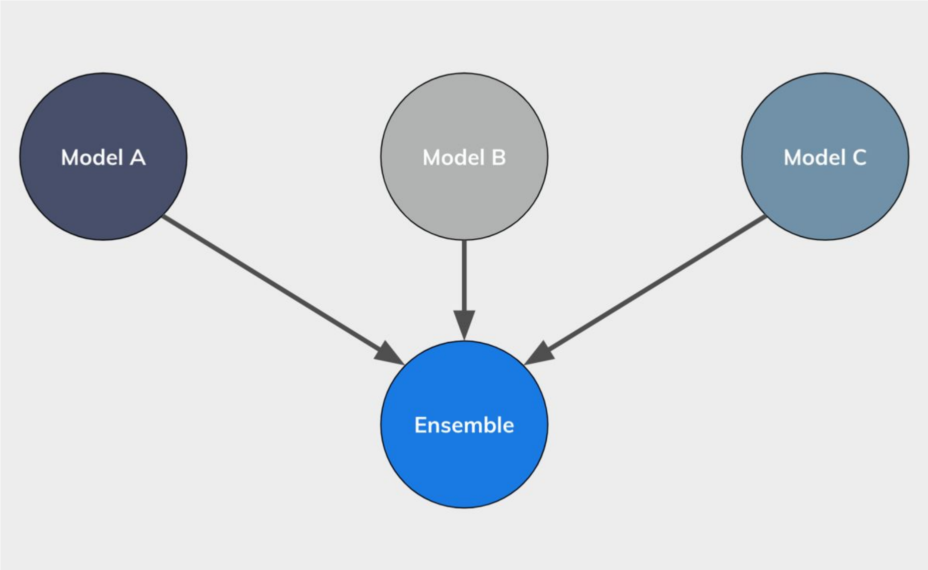
Rahim Entezari, Hanie Sedghi, Olga Saukh, Behnam Neyshabur

TU Graz
COMPLEXITY SCIENCE HUB VIENNA
Google Research

Motivation

Form an ensemble model

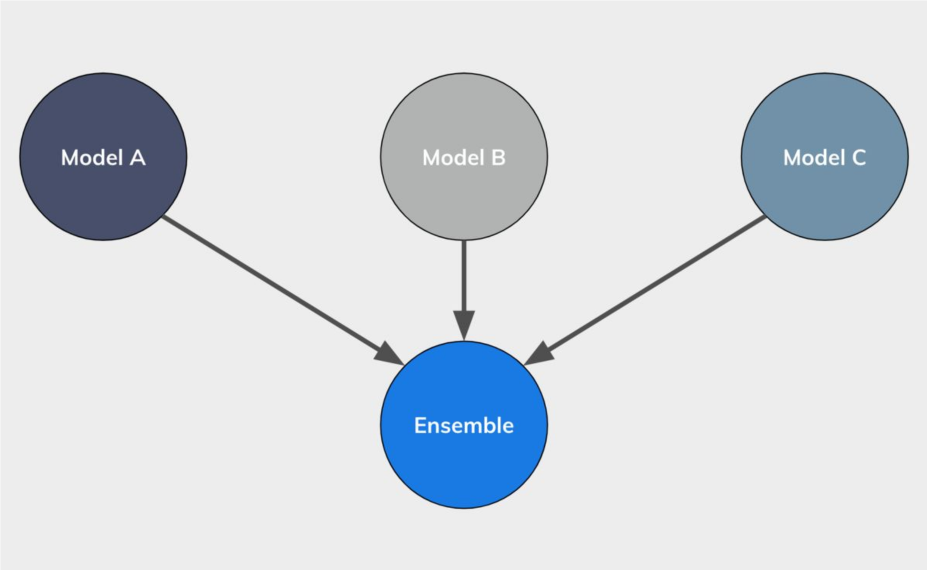
- In output space
- In weight space



Motivation

Form an ensemble model

- In output space
- **In weight space (Embedded ML)**

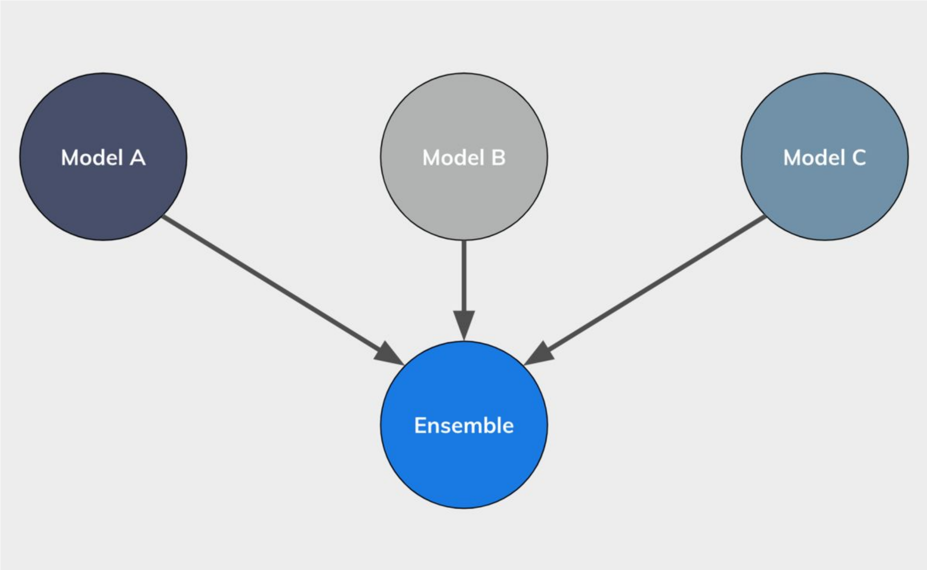


Motivation

Ensemble by weight averaging

requirements:

1. Functionally diverse solutions
2. Residing in one basin



Linear Mode Connectivity

Functionally different solutions:

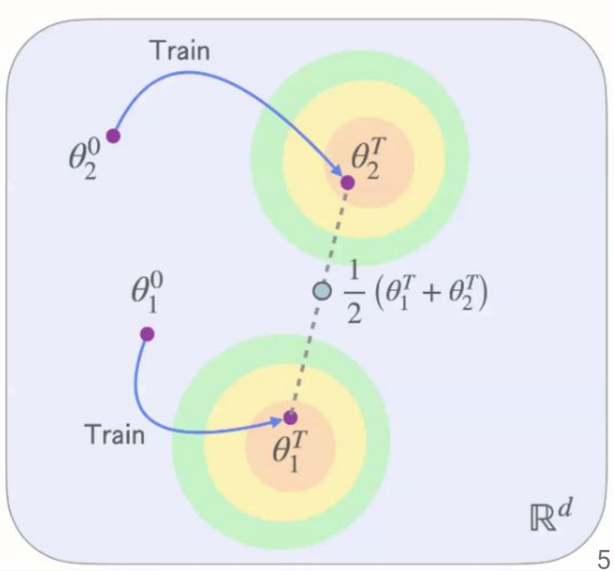


Image credit: Mitchell Wortsman

Linear Mode Connectivity

Functionally different solutions:

Weight space averaging fails

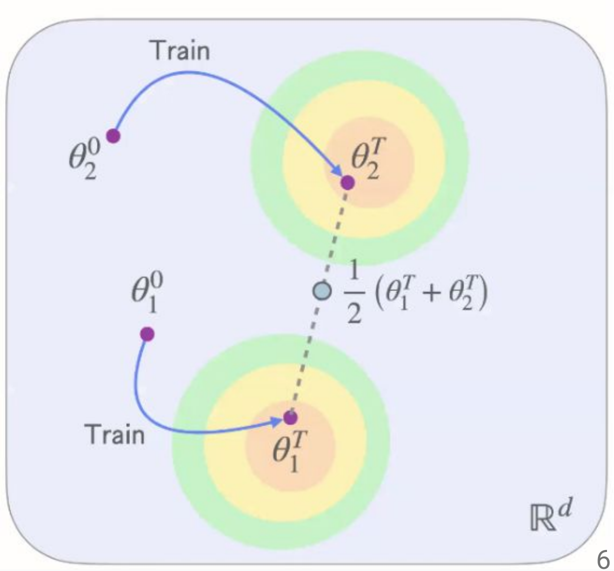


Image credit: Mitchell Wortsman

Linear Mode Connectivity

Same basin:

When part of training trajectory is shared, the solutions are **linearly mode connected** (Frankle et al., 2019) .

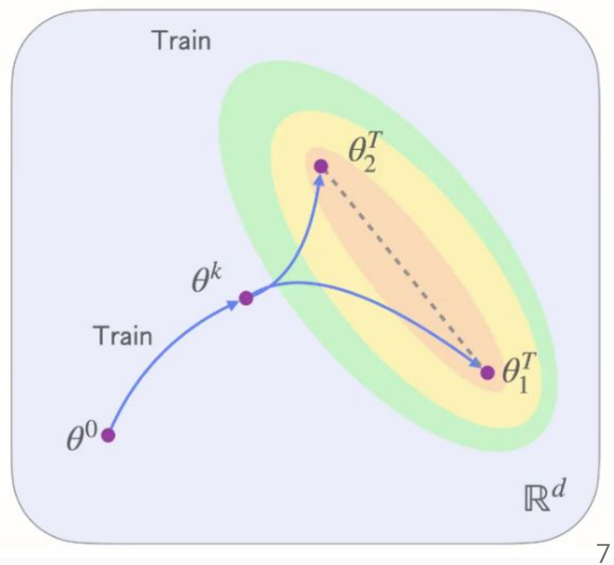


Image credit: Mitchell Wortsman

Stochastic Weight Averaging

Same basin:

e.g. SWA (Izmailov et al., 2018)

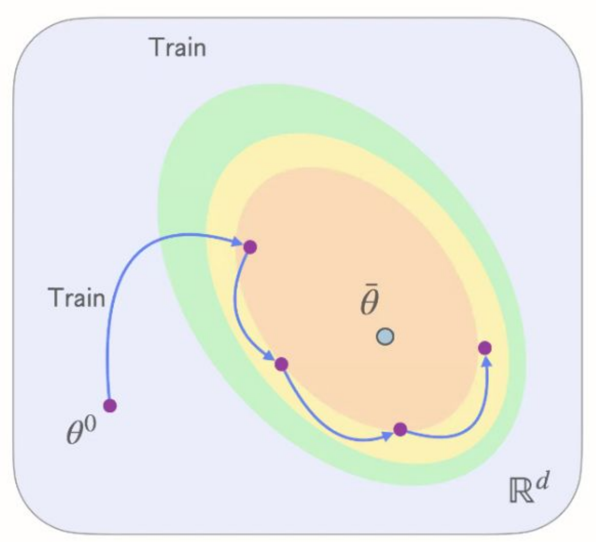


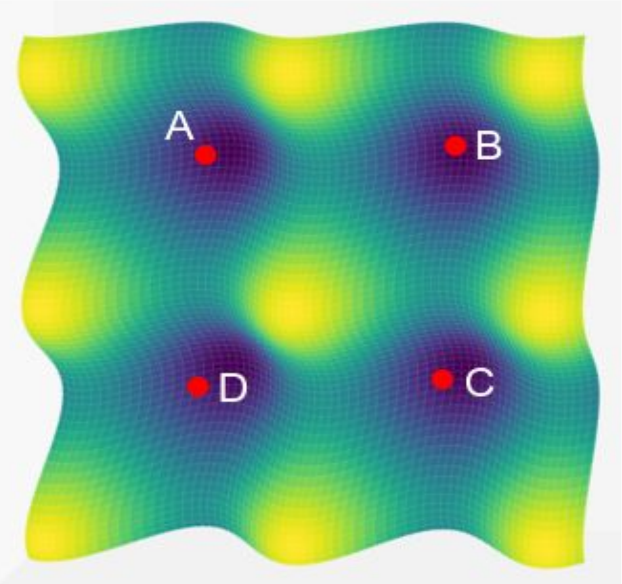
Image credit: Mitchell Wortsman

Question

Is there any way to make different solutions in one basin?

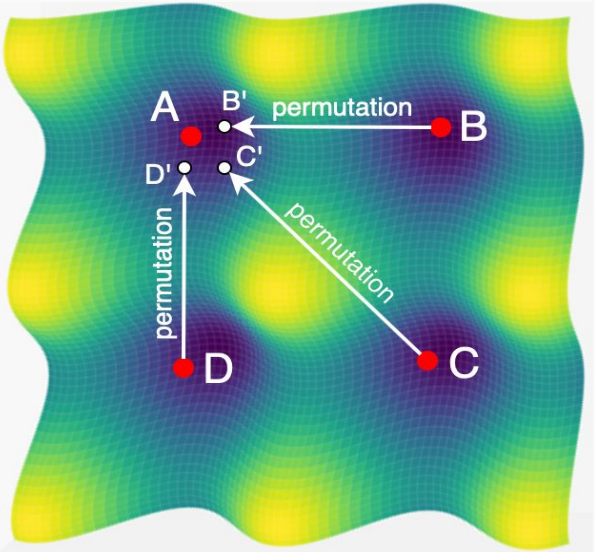
Conjecture

A, B, C, and D are minimas in different basins with barriers between pairs.



Conjecture

Taking permutations into account, there is likely no barrier in the linear interpolation between SGD solutions.



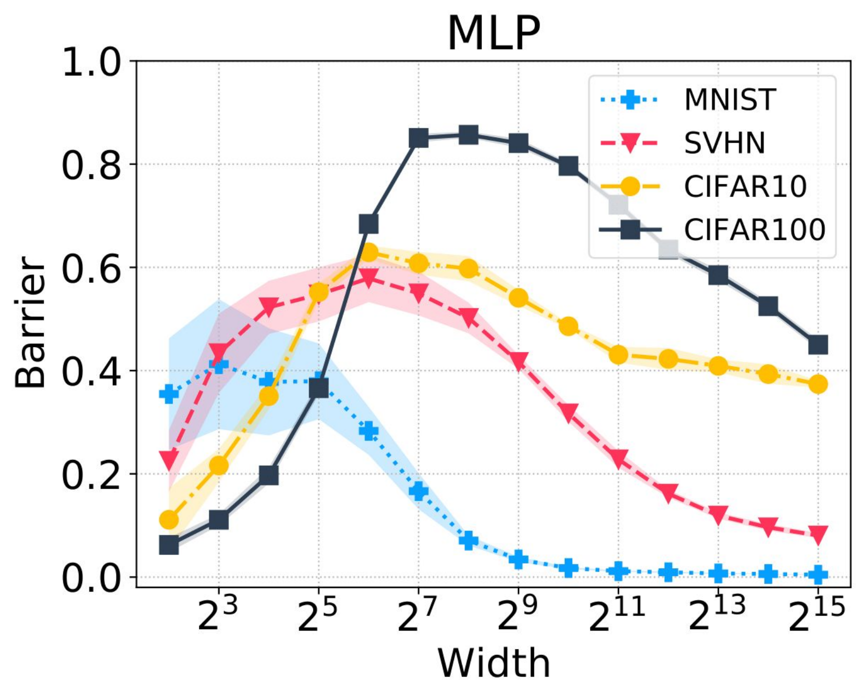
Part 1: Observations over loss landscape shape

Barrier

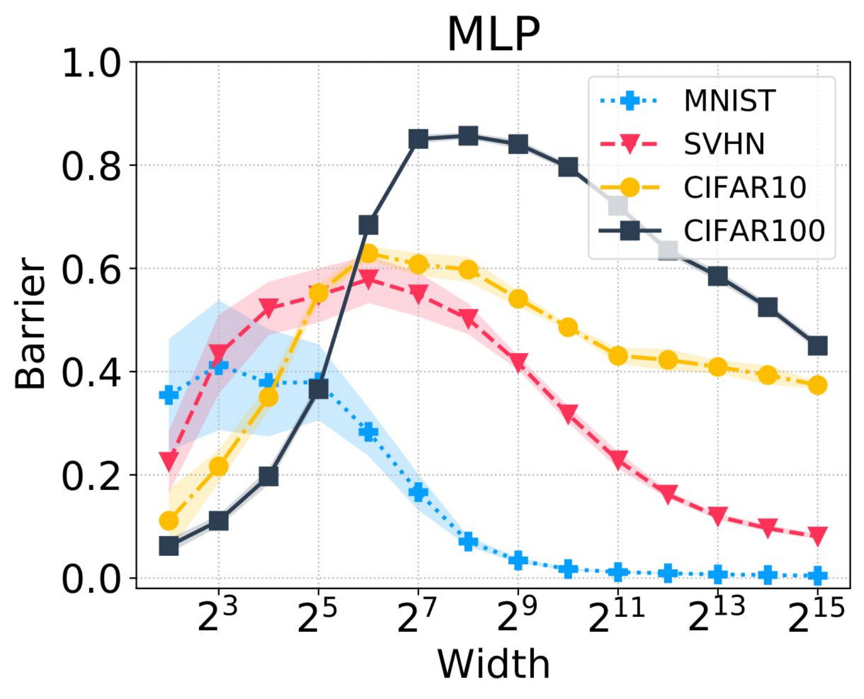
$$B(\theta_1, \theta_2) = \sup_{\alpha}[[\mathcal{L}(\alpha\theta_1 + (1-\alpha)\theta_2)] - [\alpha\mathcal{L}(\theta_1) + (1-\alpha)\mathcal{L}(\theta_2)]]$$

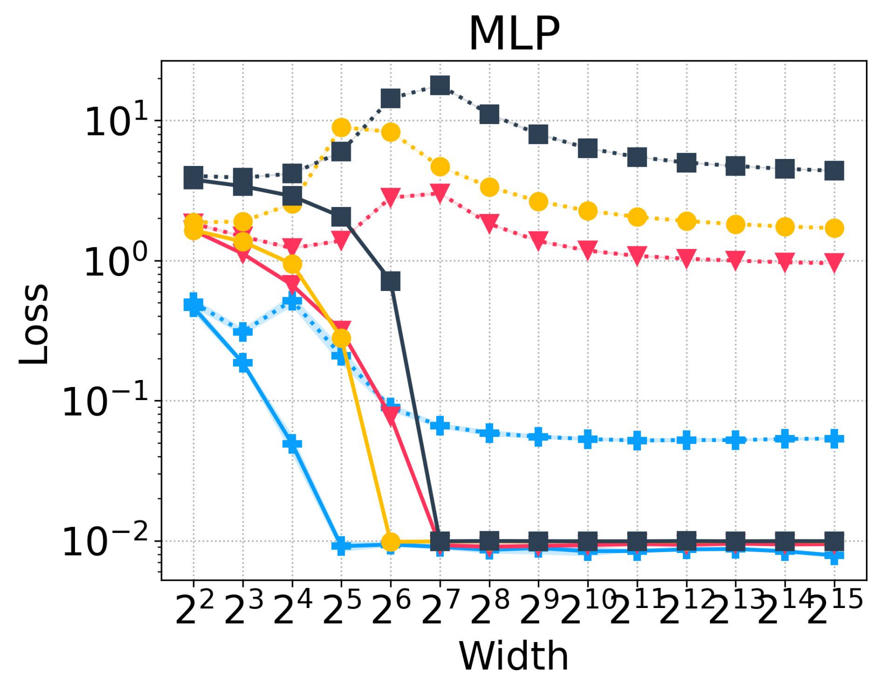
Effect of **Width** on barrier size

As the width increases, the barrier first increases and then decreases



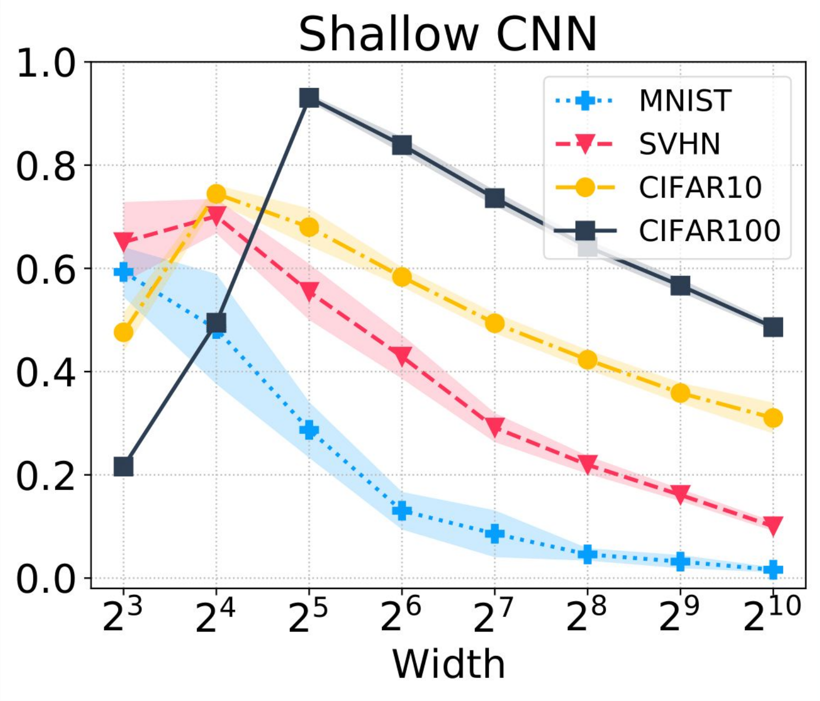
Deep Double Descent in Barrier





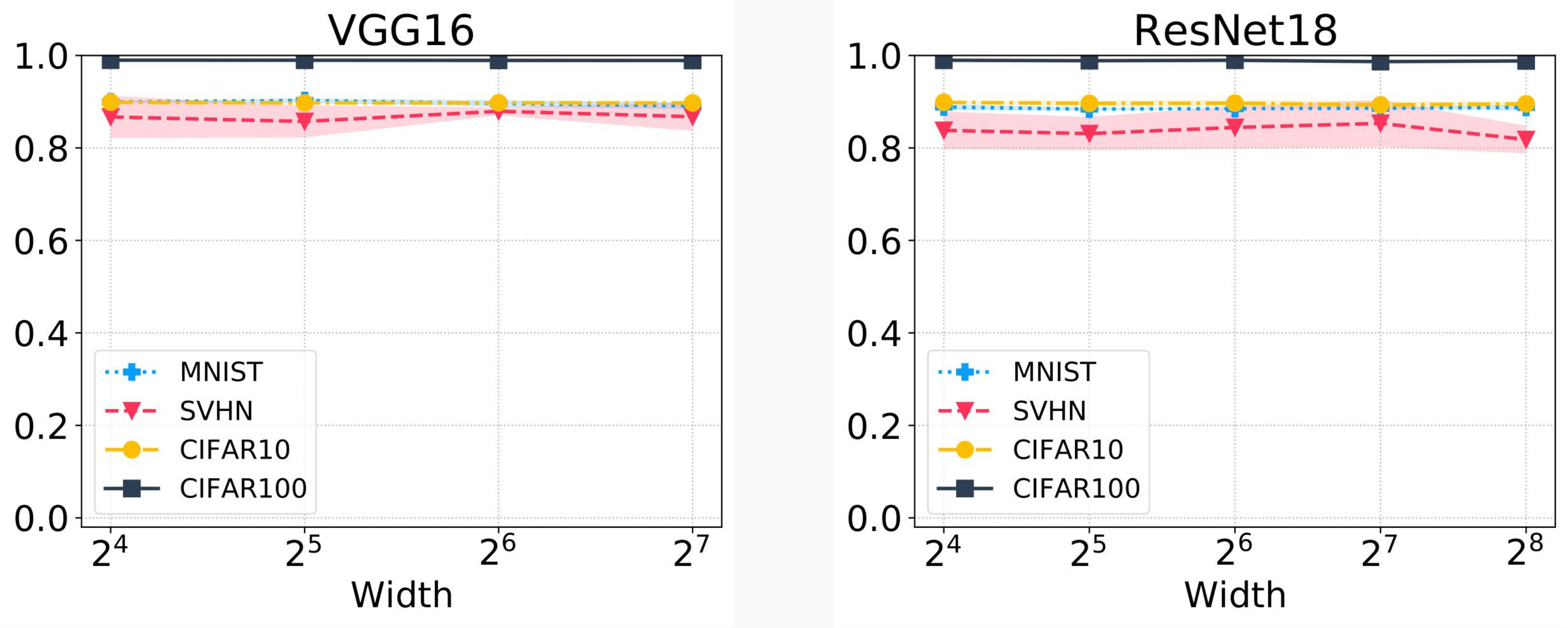
Effect of **Width** on barrier size

As the width increases, the barrier first increases and then decreases



Effect of **Width** on barrier size:

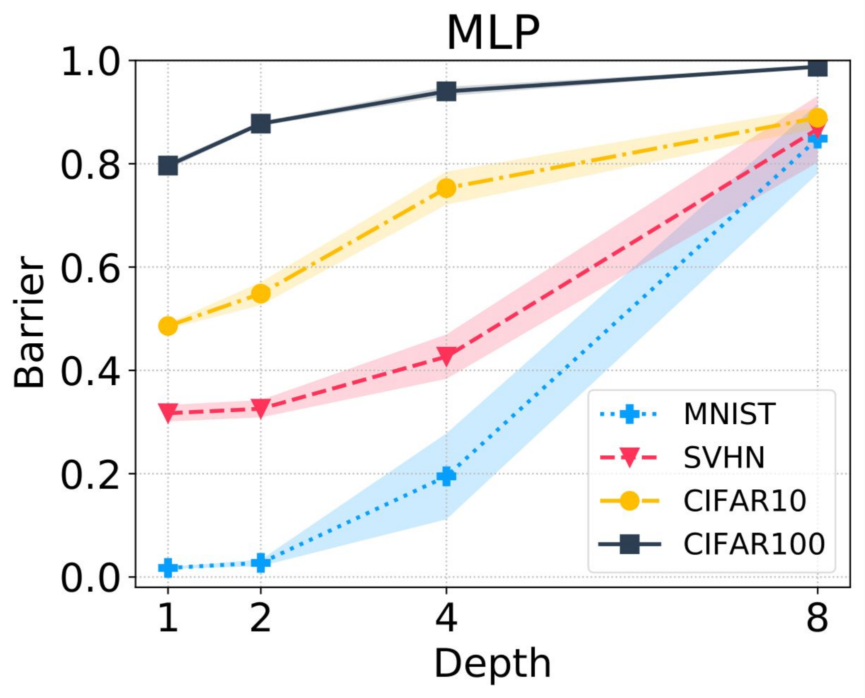
Barrier saturates at high level in deeper models



Effect of **Depth** on barrier size

Low barrier when number of layers are low

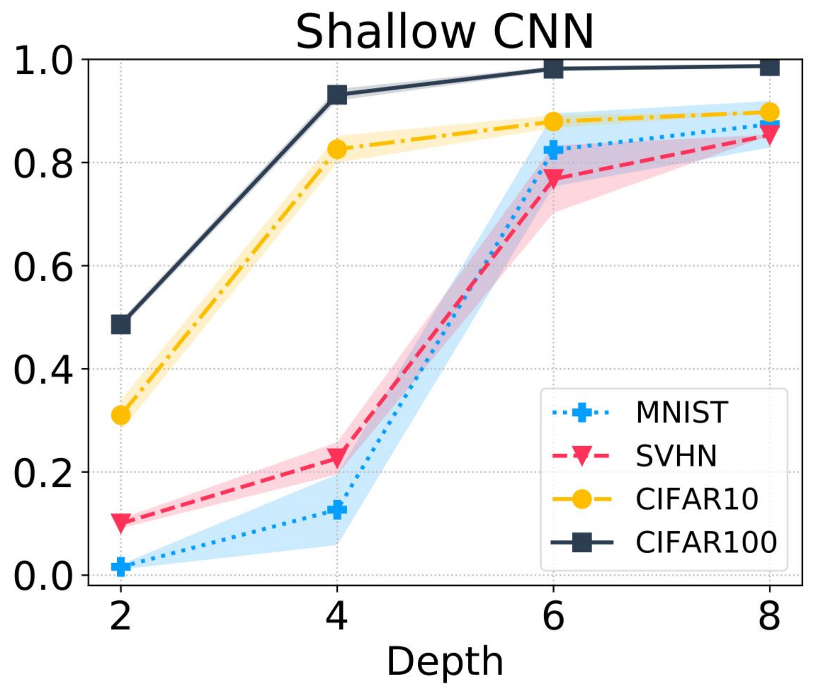
Fast and significant barrier increase as more layers are added



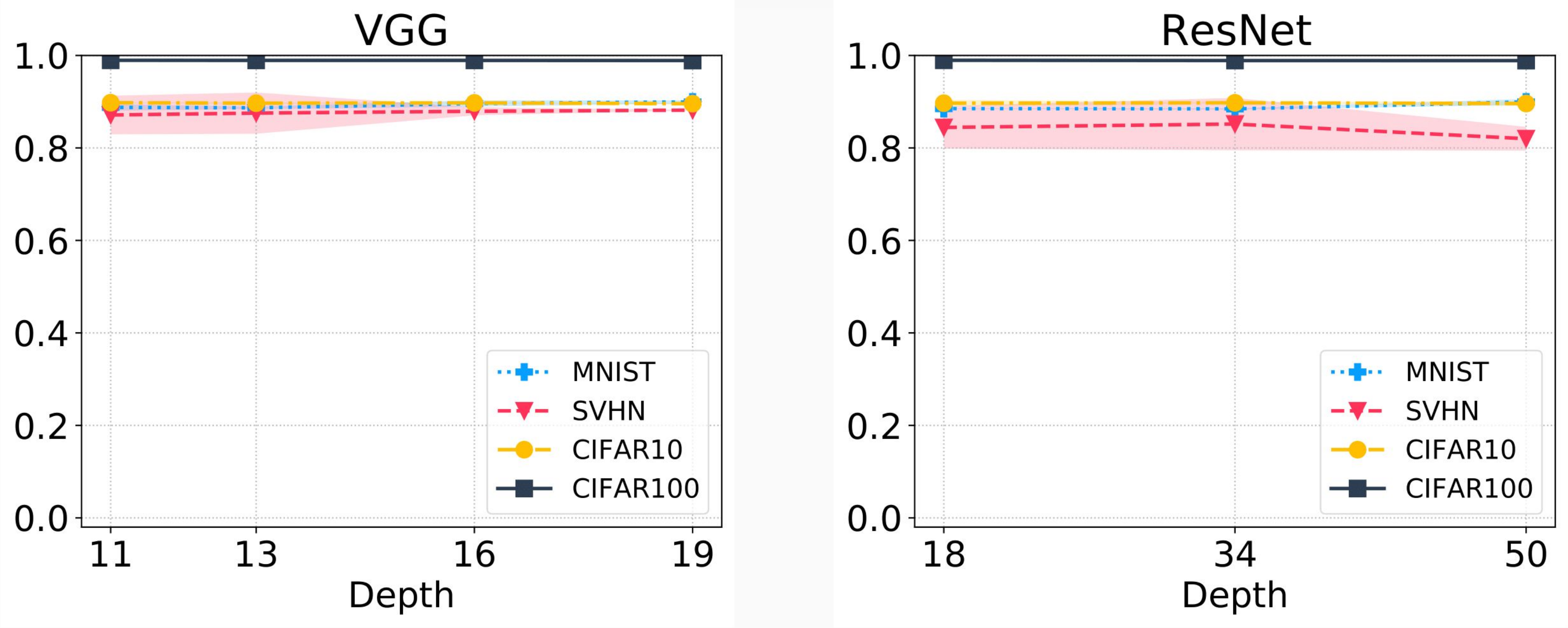
Effect of **Depth** on barrier size

Low barrier when number of layers are low

Fast and significant barrier increase as more layers are added



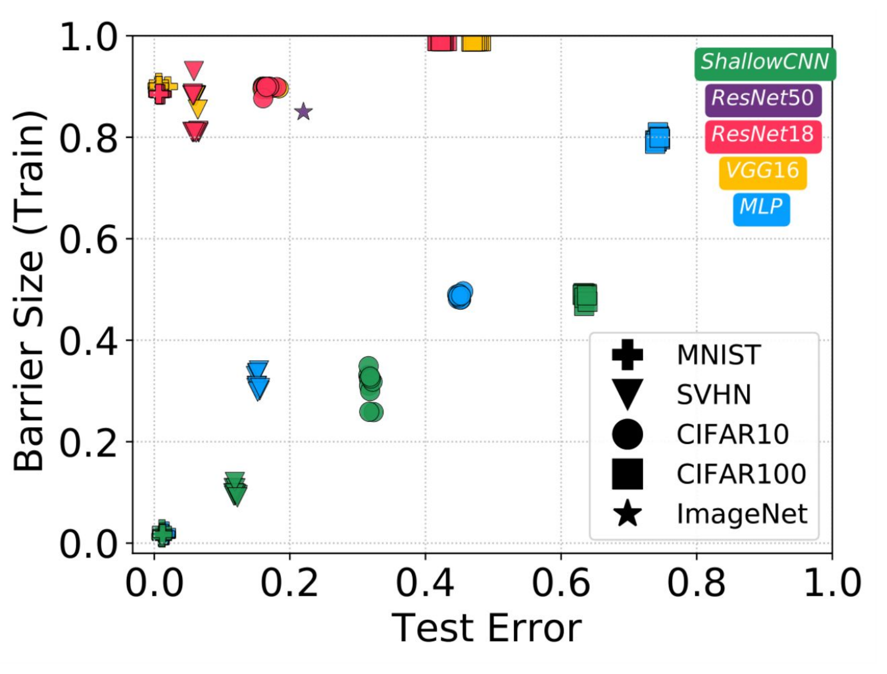
Effect of **Depth** on barrier size



Effect of **Task Complexity** on barrier size

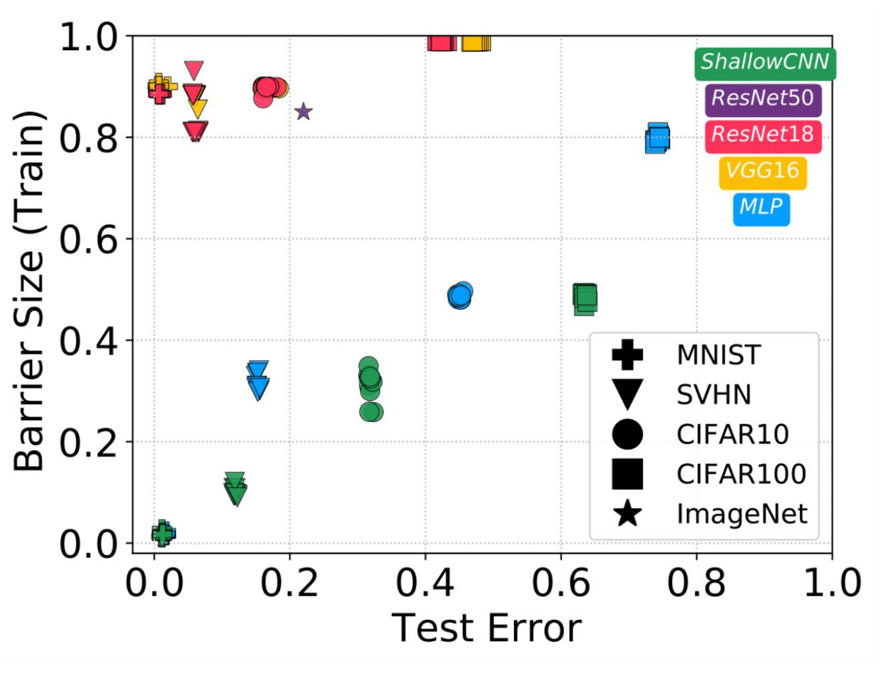
(architecture, task) has lower barrier if

the test error is lower



Effect of **Task Complexity** on barrier size

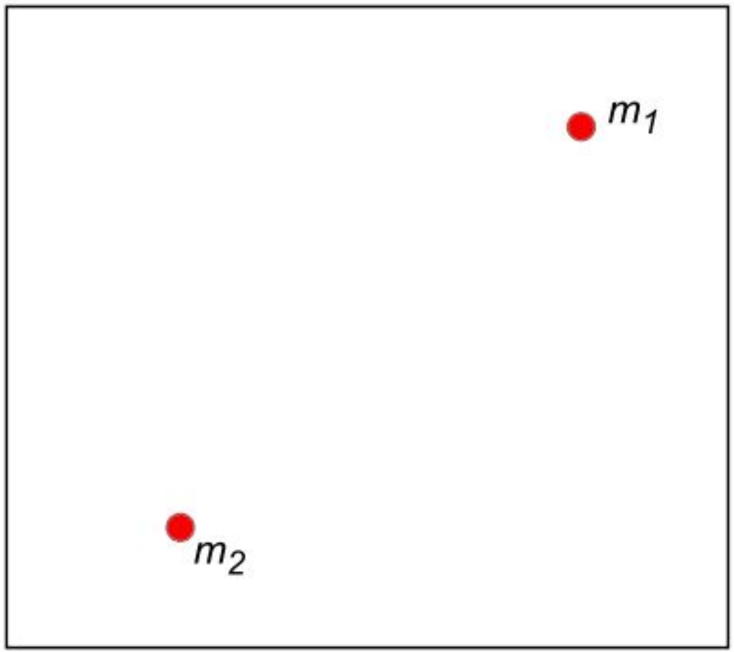
Effect of depth is stronger than (architecture, task) which leads to high barrier values for deep nets



Part 2:
evidences to support conjecture

Conjecture: recall

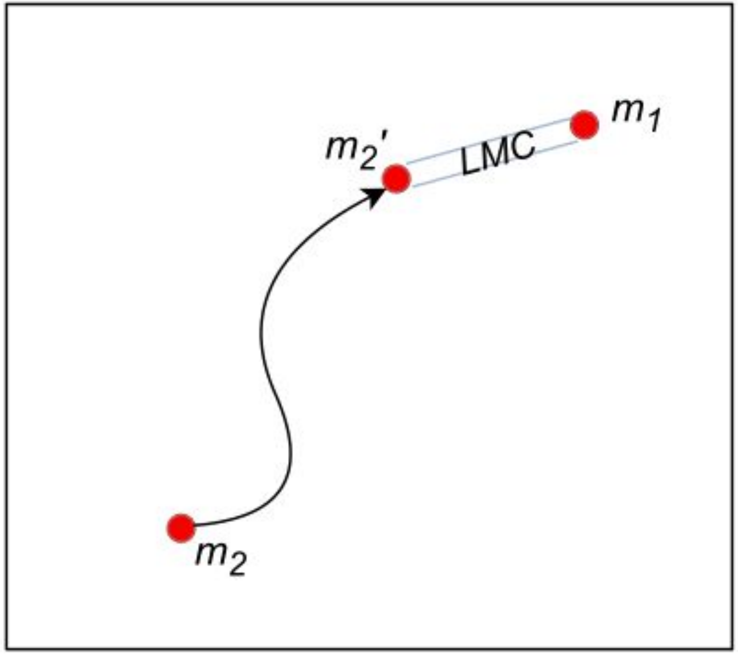
m_1, m_2 are trained and converged



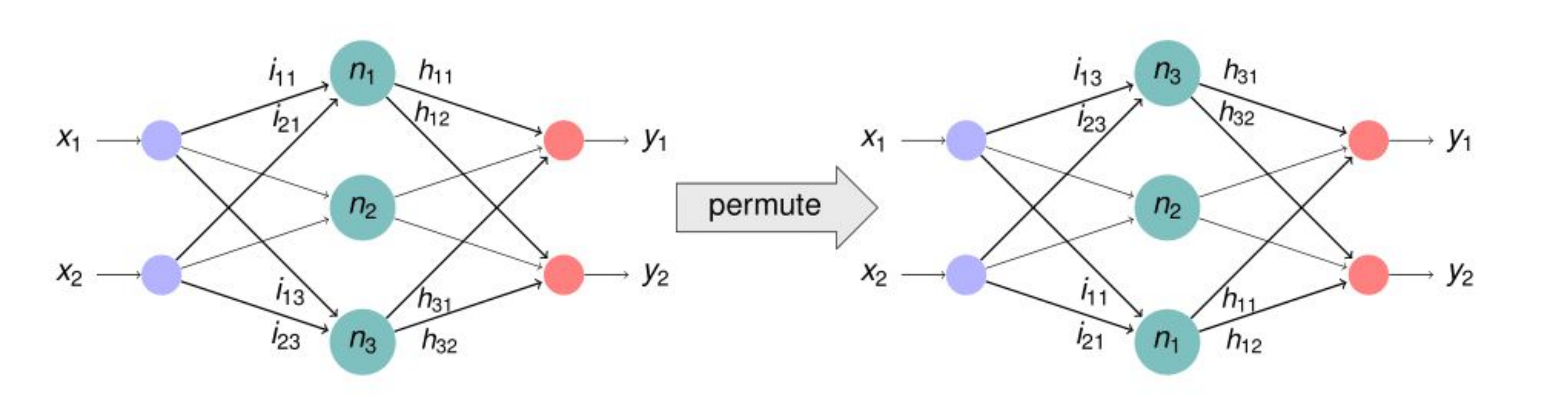
Conjecture: recall

m_1, m_2 are trained and converged

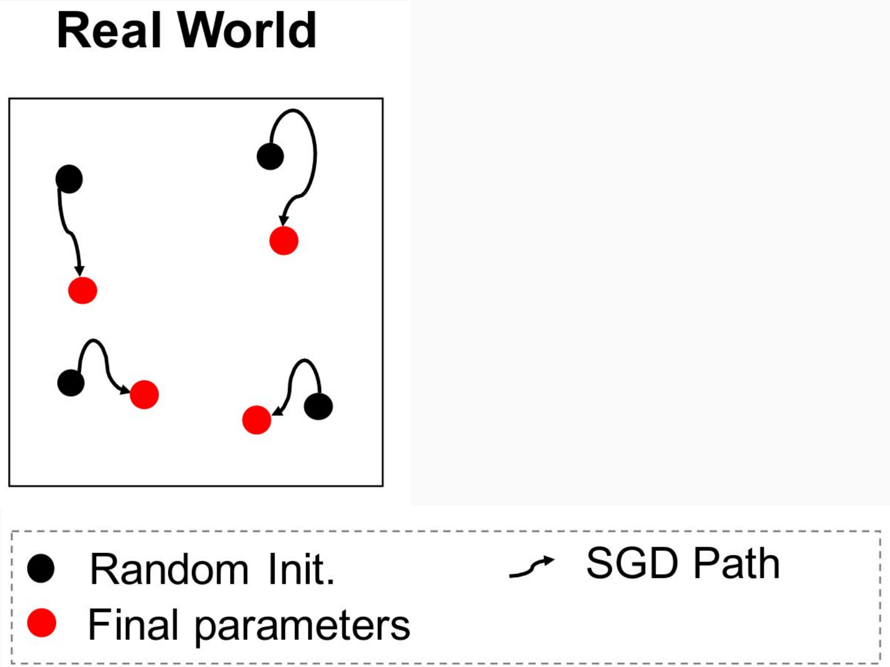
There exists a permutation applied to m_2, making m_1 and m_2' Linearly Mode Connected.



Permutation

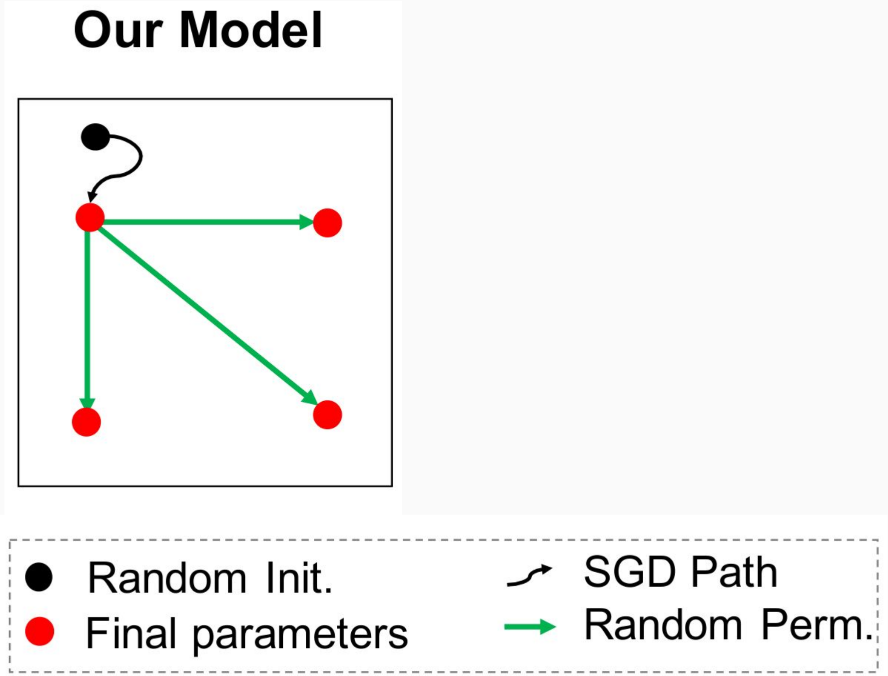


Real-world vs. Our model



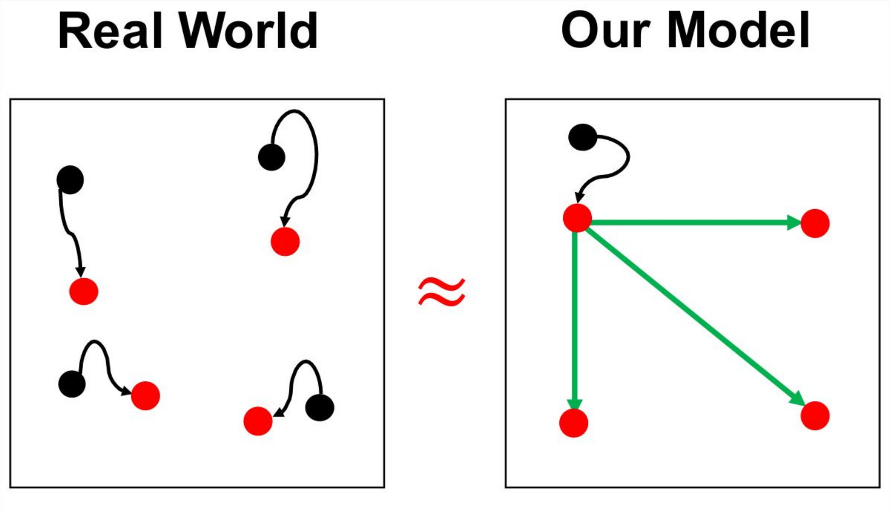
We train networks by running SGD with different random seeds and different initialization.

Real-world vs. Our model



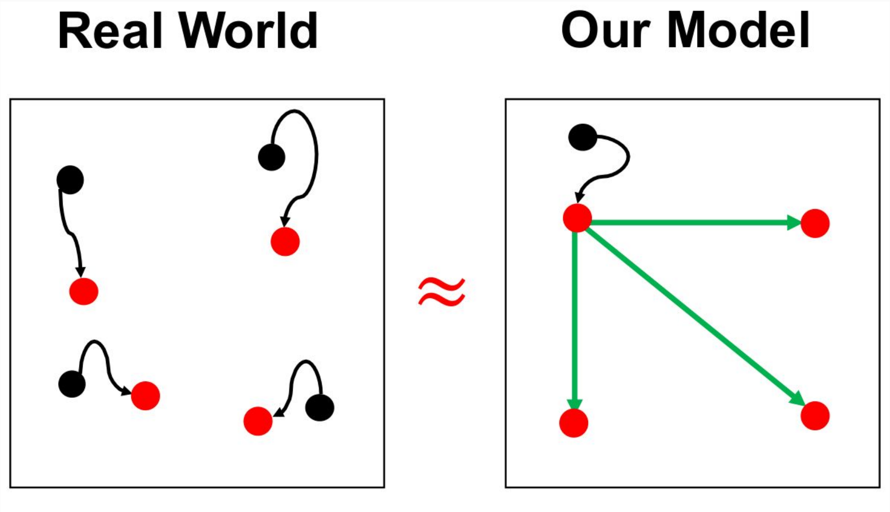
Different final networks are obtained by applying random permutations to the same SGD solution.

Real-world vs. Our model



- Our model satisfies the conjecture

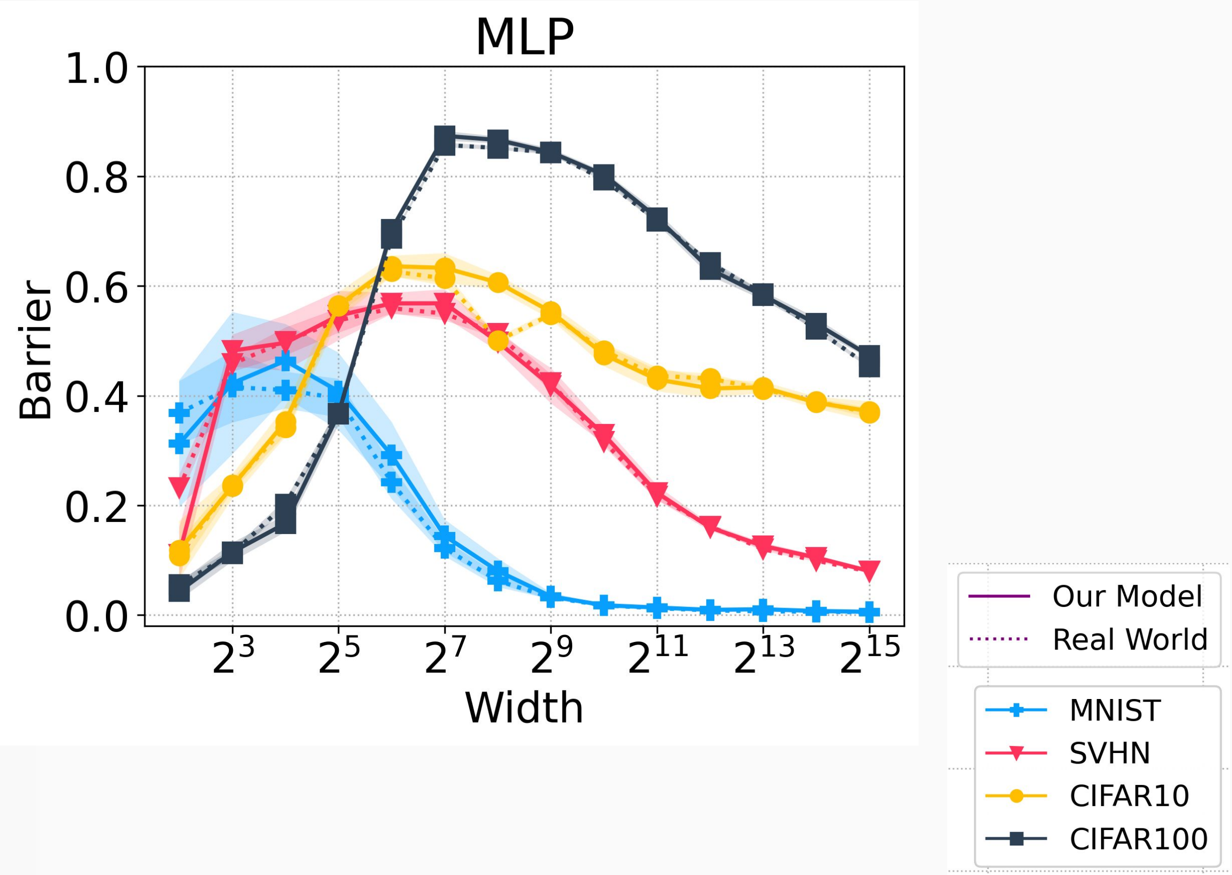
Real-world vs. Our model



- Our model satisfies the conjecture
- We show that Real world ~ Our model

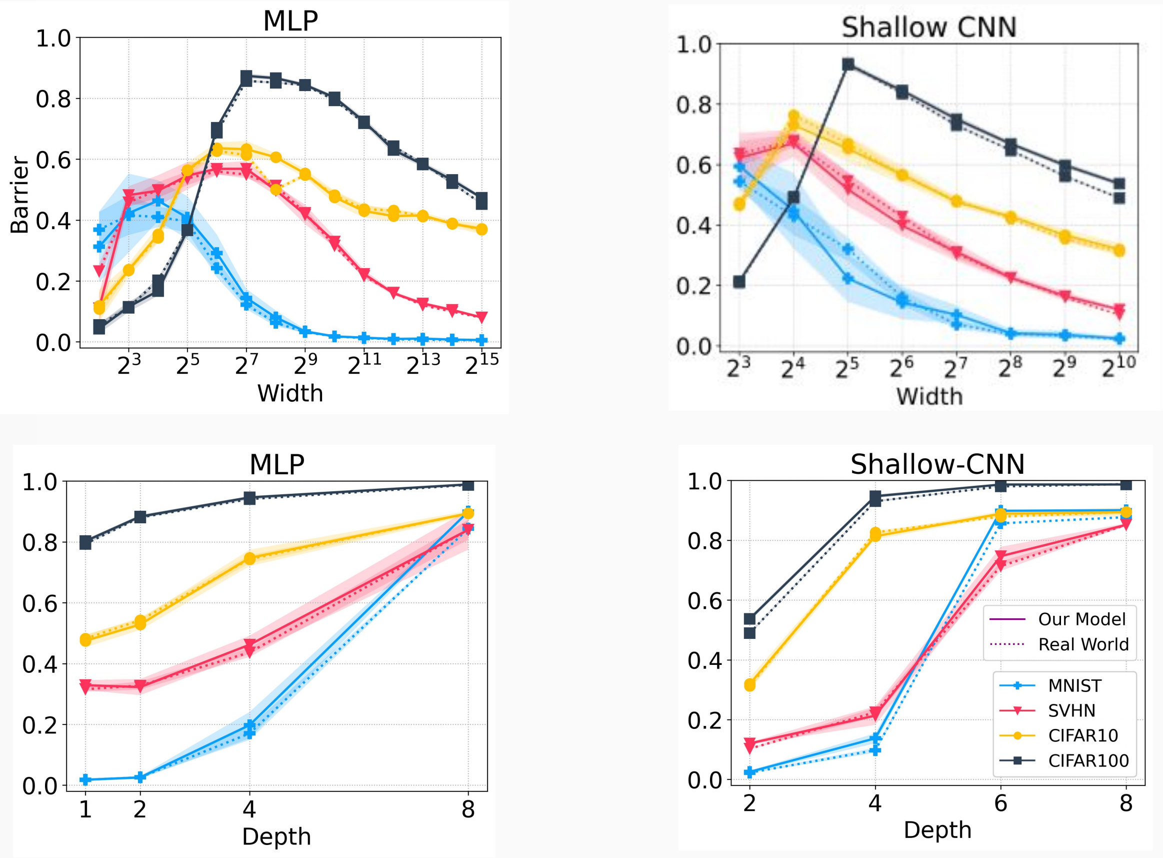
Real-world vs. Our model

Similar loss barrier between real world and our model **BEFORE** permutation search



Real-world vs. Our model

Similar loss barrier between real world and our model **BEFORE** permutation search, across all datasets, architectures, width, and depth

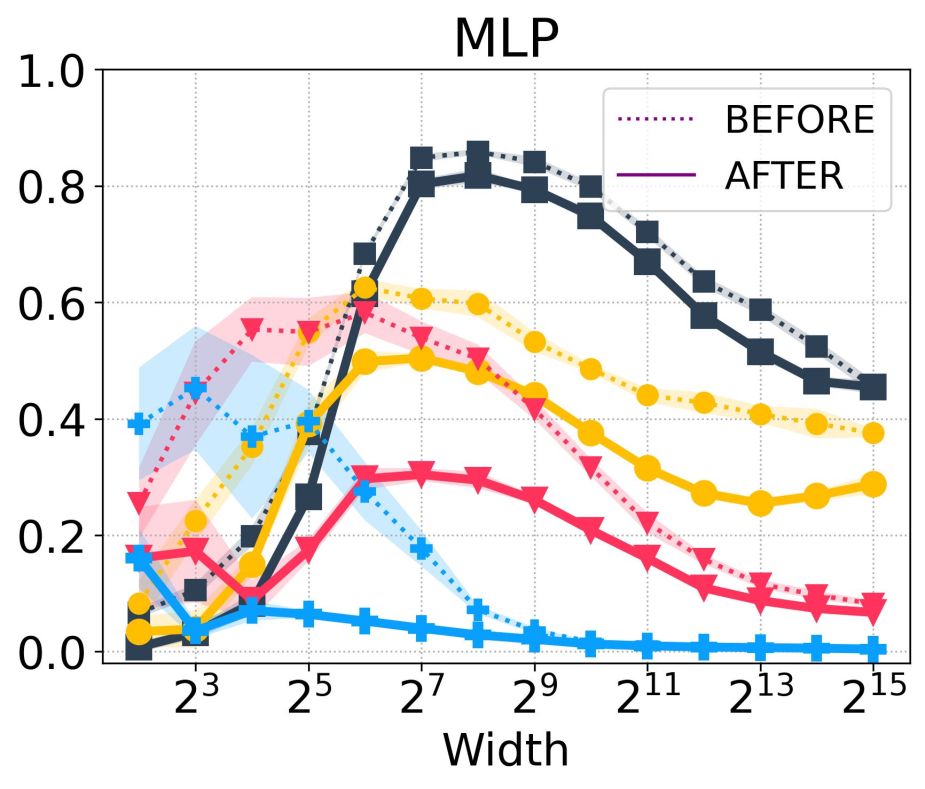


Permutation Search: Simulated Annealing

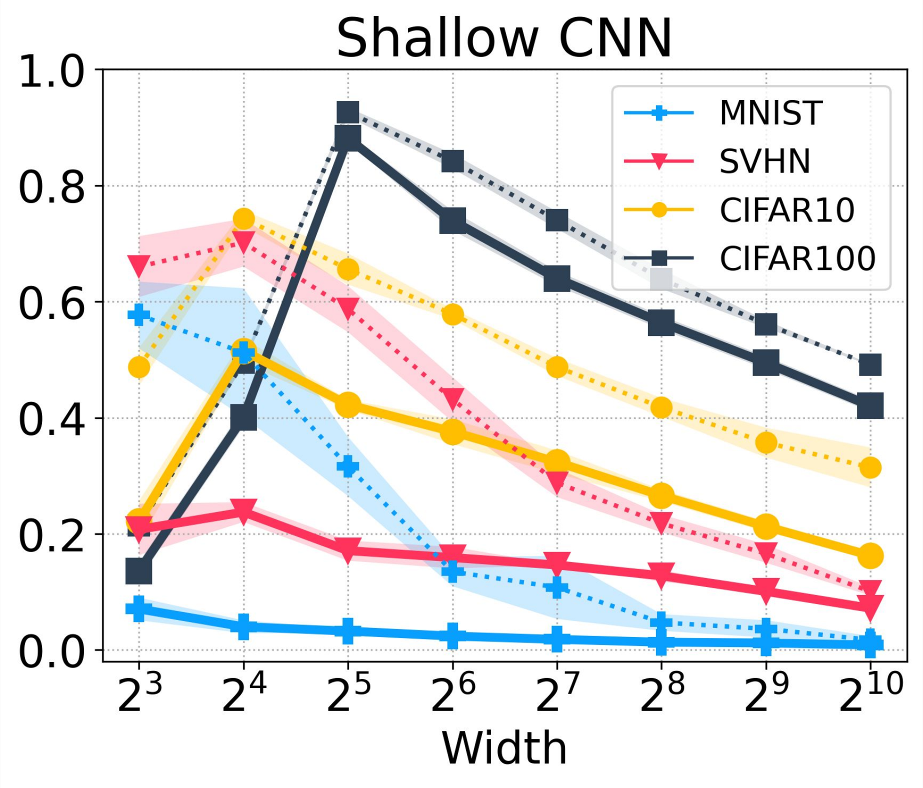
Algorithm 1 Simulated Annealing (SA) for Permutation Search

```
procedure SA({θ_i}, i = 1..n, n ≥ 2)                ▷ Goal: minimize the barrier between n solutions
    π_i = π_0, ∀i = 1..n
    for k = 0; k < k_max; k++ do
        T ← temperature((k+1)/k_max)
        Pick random candidate permutations {π̂_i}, ∀i = 1..n
        if Ψ(P(θ_i, π̂_i)) < Ψ(P(θ_i, π_i)) then    ▷ Ψ: barrier objective function
            π_i ← π̂_i
    return {π_i}
```

Simulated Annealing: Performance

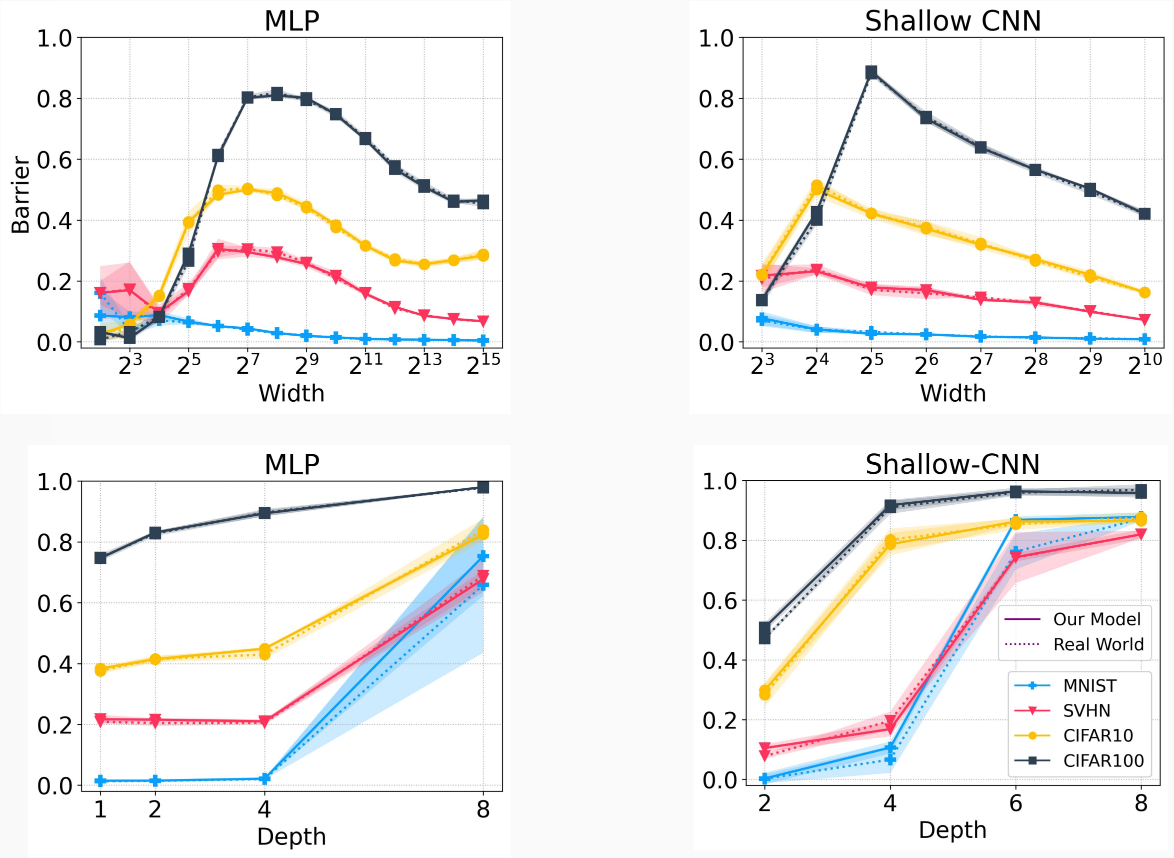


Simulated Annealing: Performance

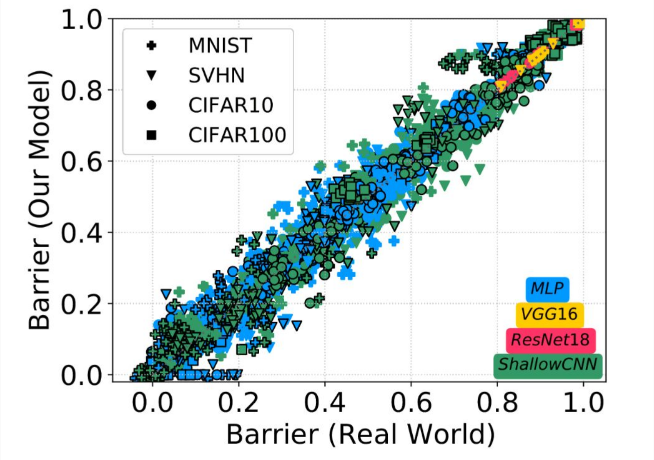


Real-world vs. Our model

Similar loss barrier between real world and our model **AFTER** permutation search



Real-world vs. Our model



- We show that Real world ~ Our model

Takeaways

- One way to form ensembles is to weight average solutions
- Conjecture: we can make different SGD solutions in one basin using permutations
- Our theoretical results + extensive experiments fall short of refuting our bold conjecture.

Thanks

Code: https://github.com/rahimentezari/PermutationInvariance

entezari@tugraz.at