Deep Learning Theory and Applications

Loss Landscapes

Yale

CPSC/AMTH 663

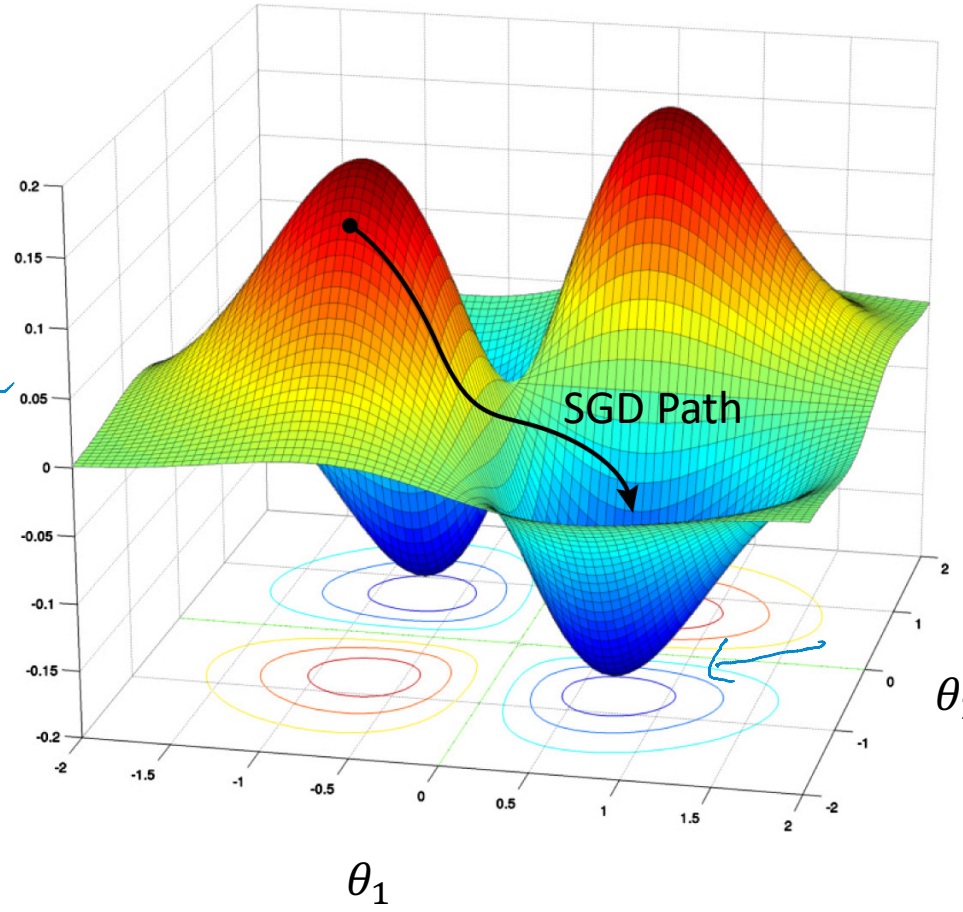
Loss landscape

(hills and valleys)

use SGD to train NN

Loss function

$L(\theta, X)$



a local minima

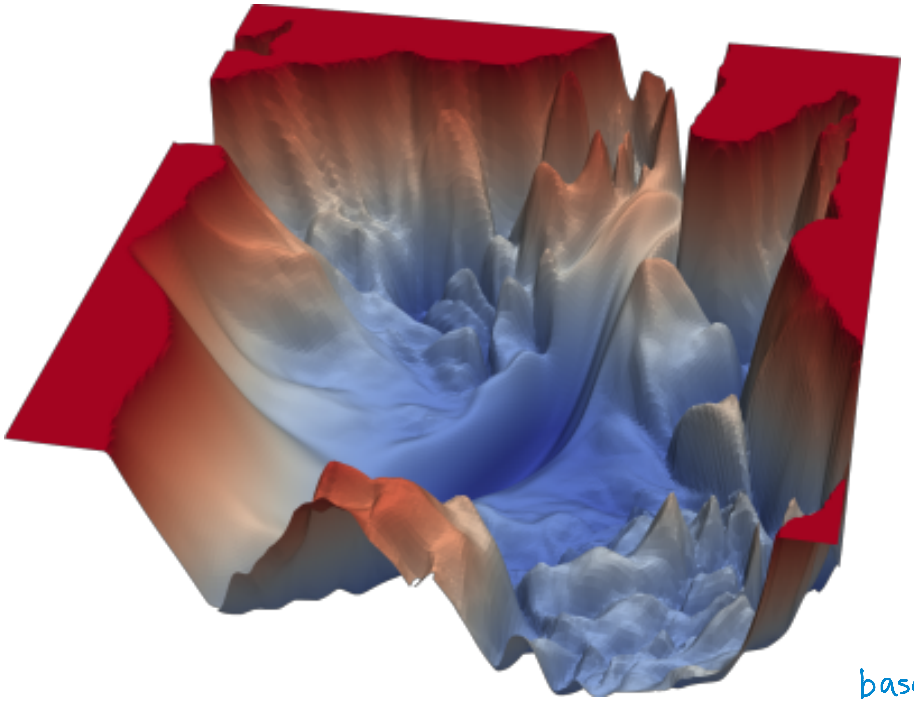
θ_2

θ_1

What can it tell us?

- If the loss landscape is very bumpy, then there are a lot of local minima

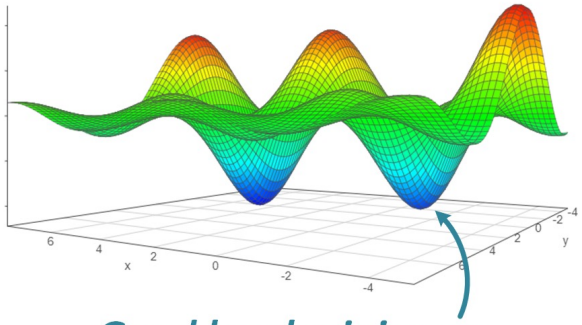
凹凸不平的



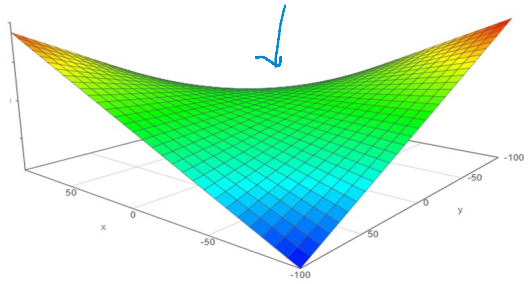
based on initialization

- SGD may not reach a good solution, and not very fast

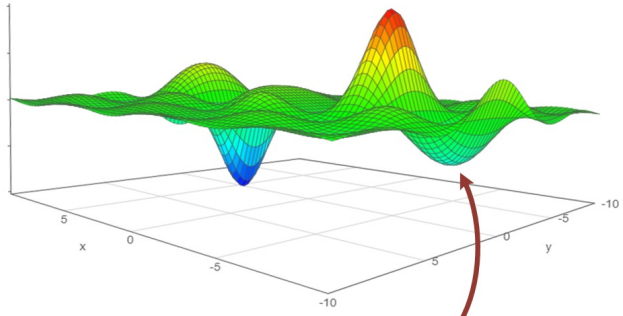
More information from landscapes



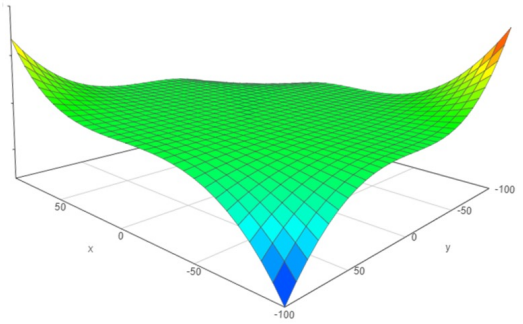
Good local minimum



Strict saddle



Poor local minimum



Non-strict saddle

Landscape analyses prove convergence to global minimum by disqualifying these

disregard these bad local minima

Dimensionality of Loss Landscape

- Hugely high dimensional space
- One ”dimension” for each weight and bias
- Difficult to visualize in its entirety
- But we can use tricks
  - Subsampling random directions
  - Subsampling the landscape around the minima
  - Random directions to get insight

Random direction visualization

- Train a neural network until it gets to a minima
- Add a small perturbation in a random direction
- See how the loss changes

- https://losslandscape.com/explorer

Filter-wise Normalization

- Start with a random Gaussian vector d, with the same dimension as number of parameters θ
- We normalize each direction in d denote $d_{i,j}$ to have the same norm as the corresponding weight in $\theta_{i,j}$ (this is the jth weight in the ith layer)

$$d_{i,j} \leftarrow \frac{d_{i,j}}{\|d_{i,j}\|} \|\theta_{i,j}\|$$

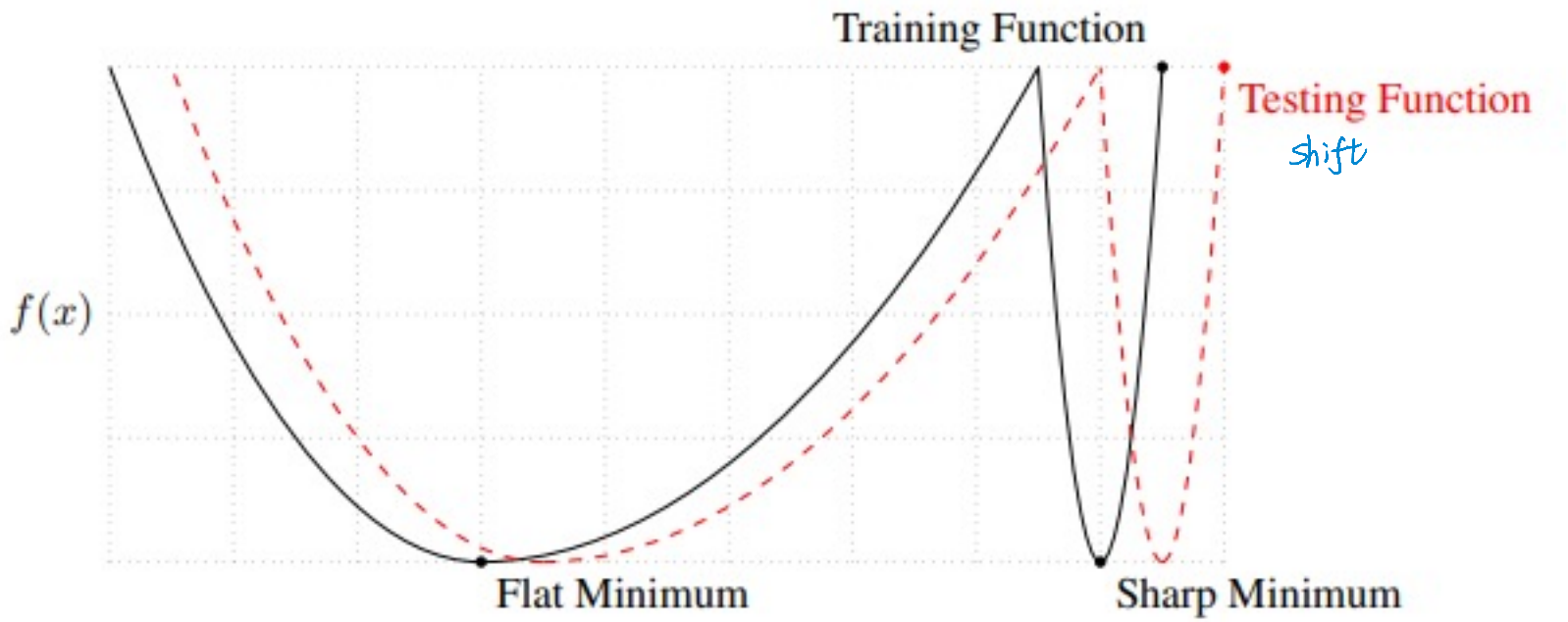
- Why would this help?

Scale Invariance in RELU Nets

- In a RELU network if weights in one layer are all multiplied by 10 and weights in the next layer are all divided by 10, network performance remains the same but loss landscape different
- Even more prominent during batch normalization regularization
  - Output of each layer is normalized before going to the next layer
- Without the filter-wise normalization A neural network with large weights may appear to have a smooth and slowly varying loss function
  - Perturbing the weights by one unit will have very little effect on network
- However, if the weights are much smaller than one, then that same unit perturbation may have a catastrophic effect, making the loss function appear quite sensitive to weight perturbations.

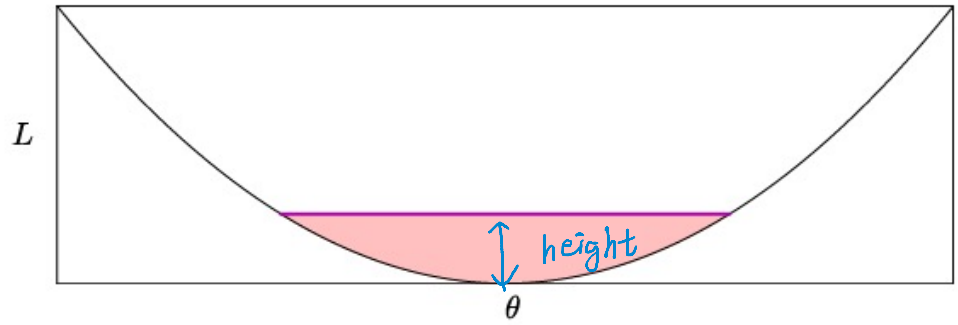
Flat Minima minima are thought to generalize better

reason resiliant to noise
test dist often differ from train dist but
flat minima don't change much



- Flatness definitions are not scale invariant

Volume



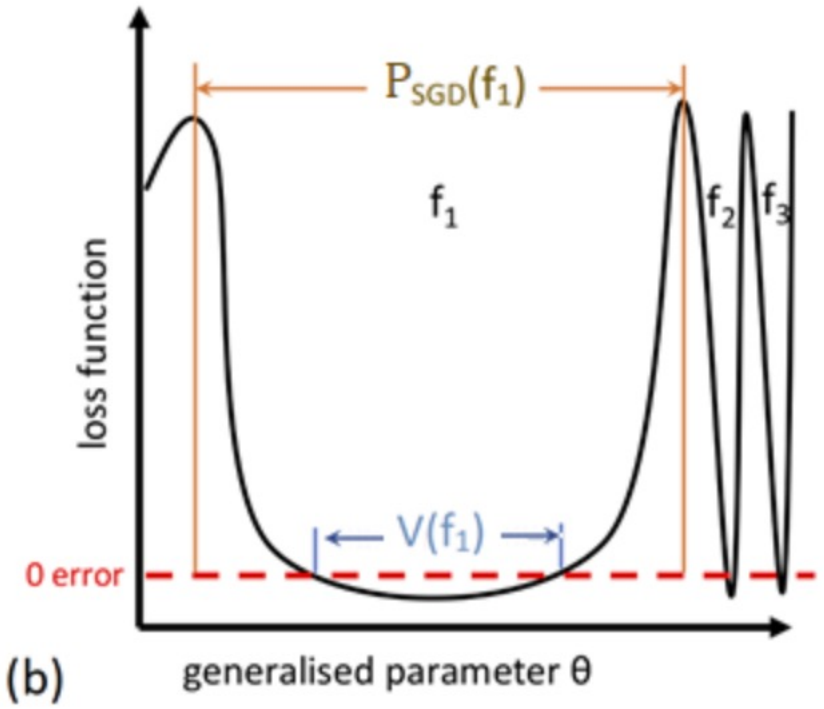
threshold for Loss
↓

- Height vs volume of red area is a measure of sharpness
- There have been many attempts at scale invariant definitions

Volume of minima argument

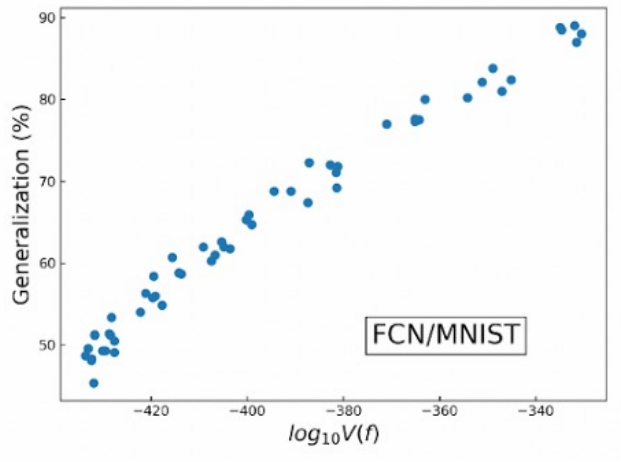
- Can think of the loss landscape as a function space
- Space of functions that are consistent with the minima and the probability of finding them with SGD can also be used

P_{SGD}

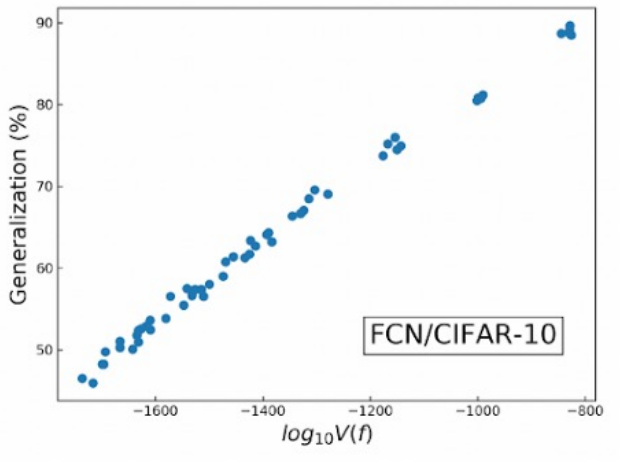


Empirical Correlation

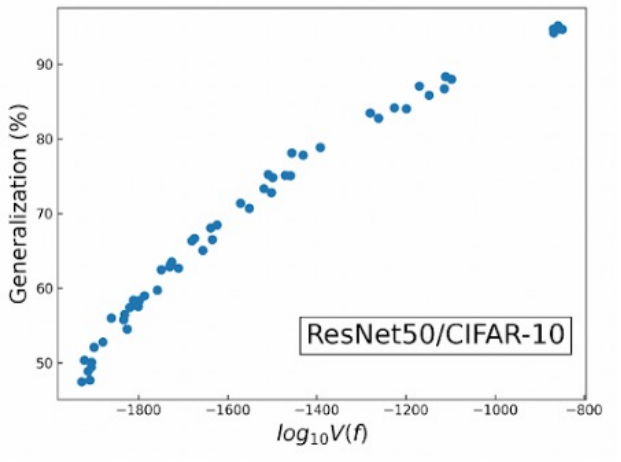
log volume and generalization: linear



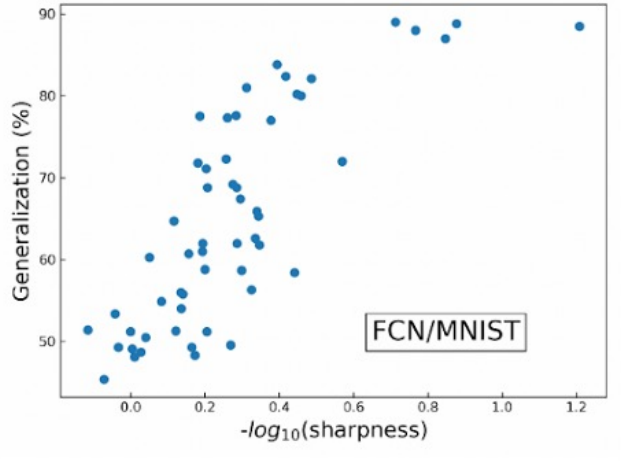
(a)



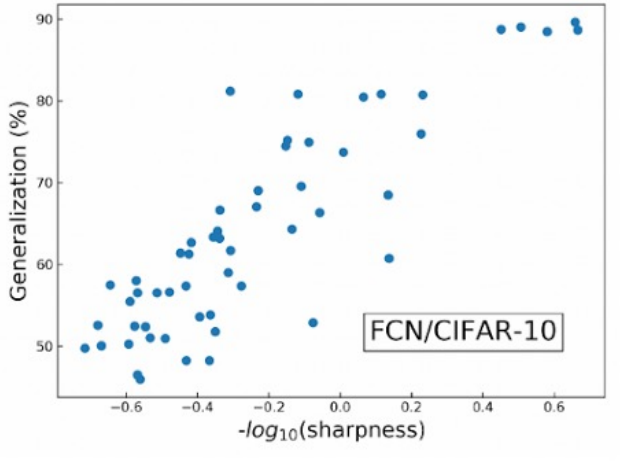
(b)



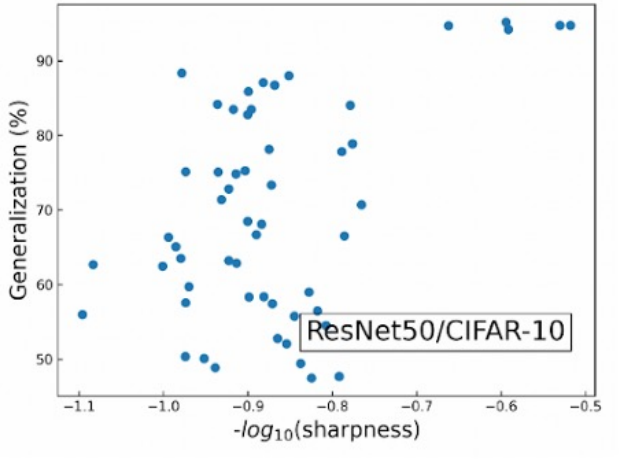
(c)



(d)



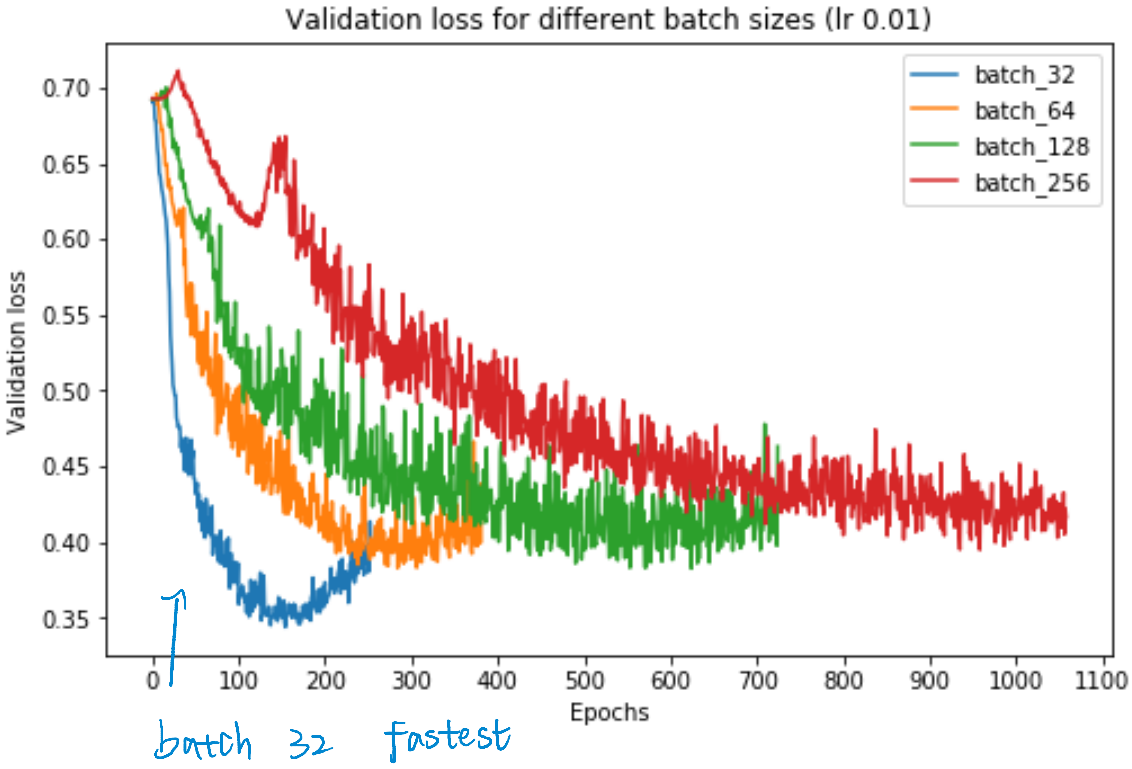
(e)



(f)

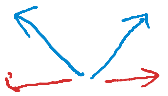
Large batch vs small batch training

- Training SGD with large batches has been associated with sharp minima, slow training
- Small batch training is thought to lead to more generalization and faster training

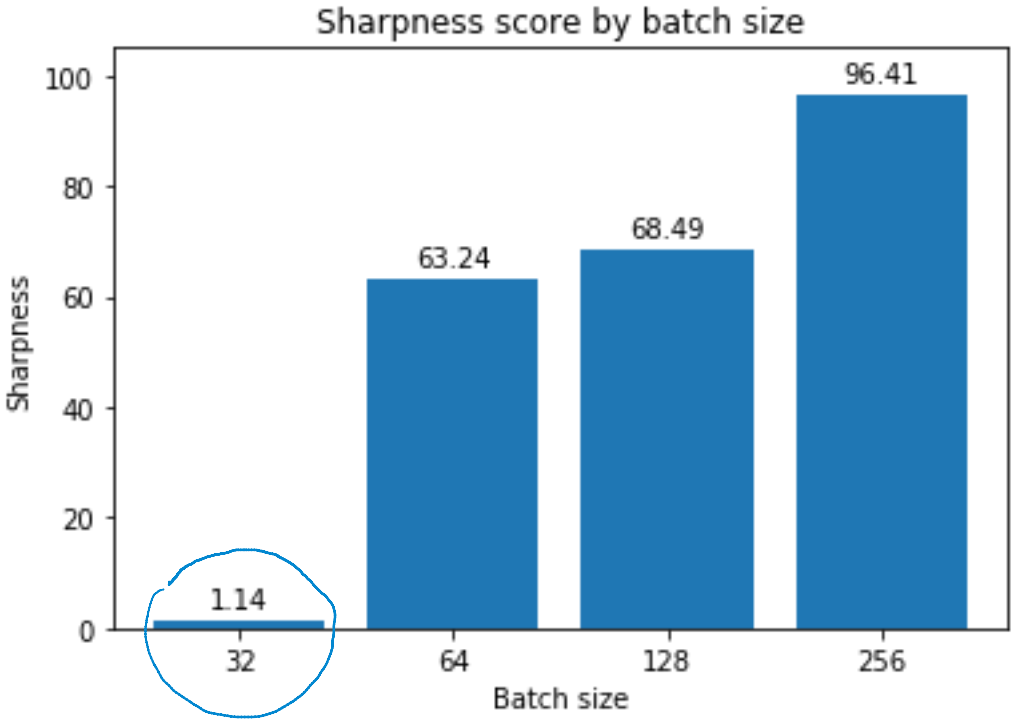


Reasons

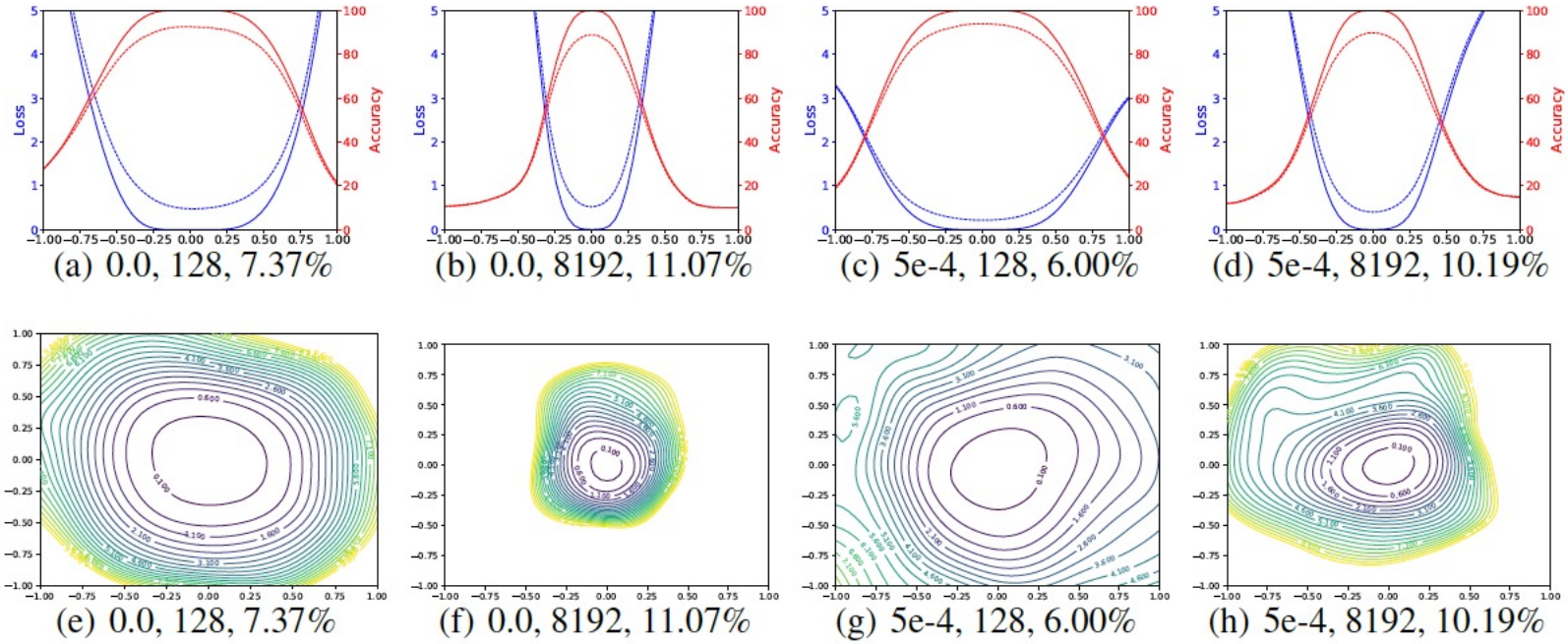
- Slow movement because of canceling directions



- Stochasticity of small batches gets out of local minima
- Tend to generalize to flatter minima

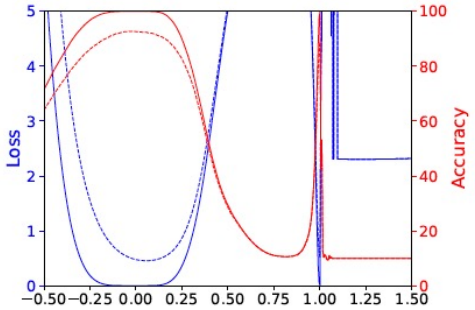


SB produces flatter Minima

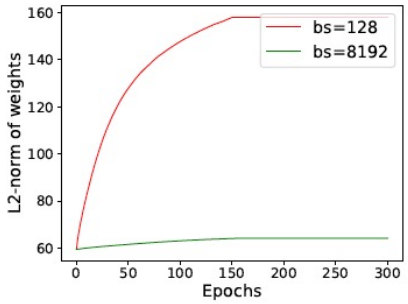


Reaffirms SB vs LB Hypothesis

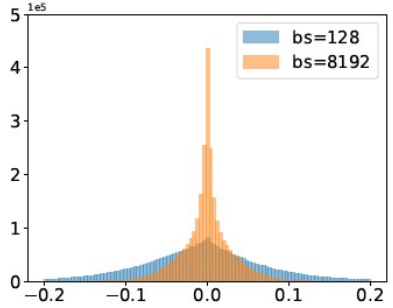
- Some works disputed the flat vs sharp loss landscape hypothesis, but visualization reaffirms it



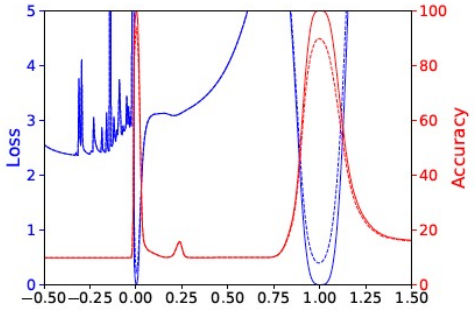
(a) 7.37% 11.07%



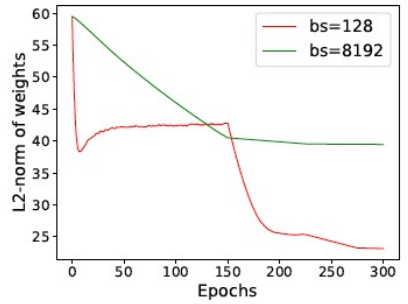
(b) $\|\theta\|_2$, WD=0



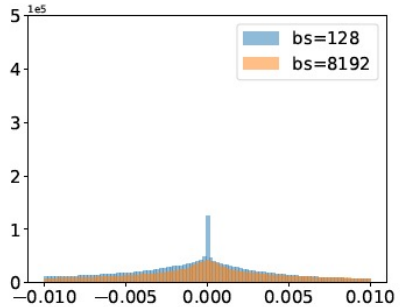
(c) WD=0



(d) 6.0% 10.19%

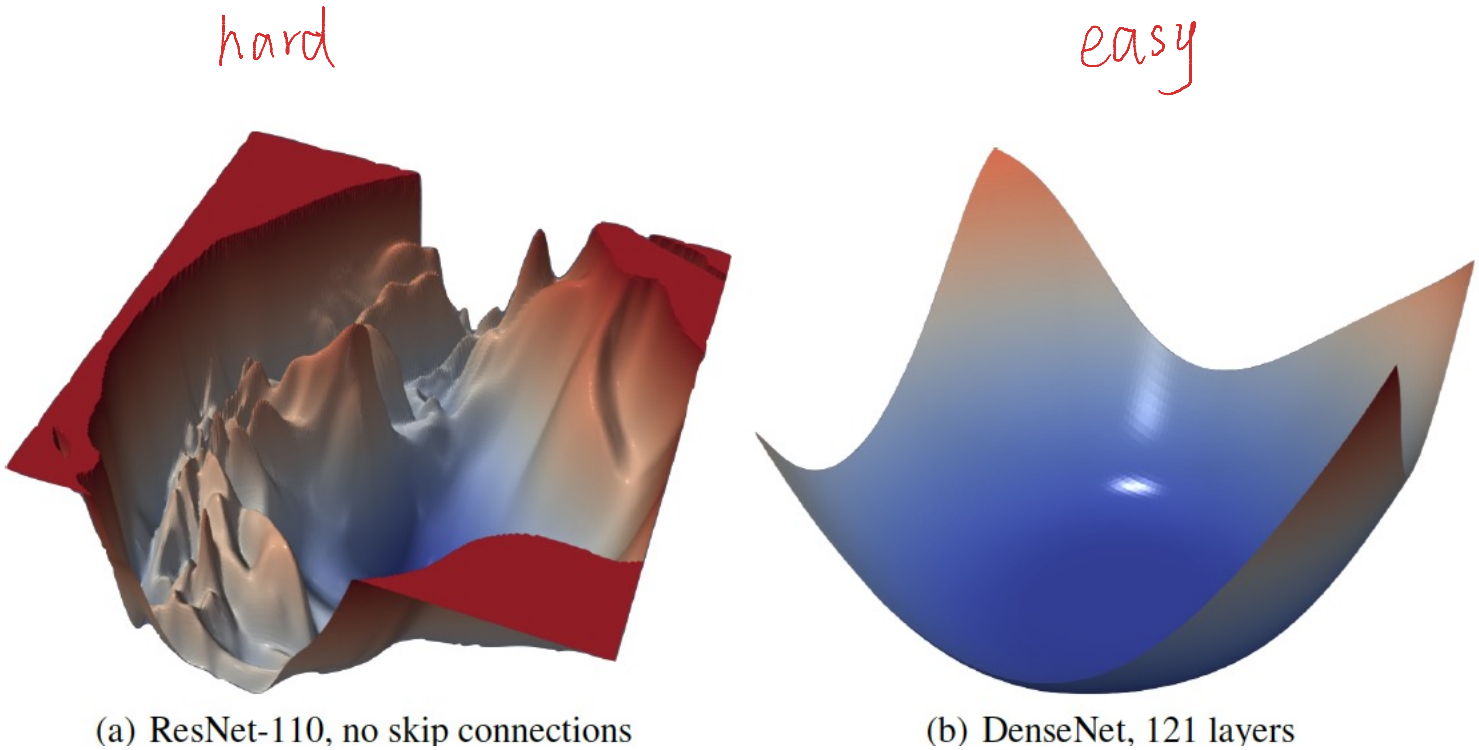


(e) $\|\theta\|_2$, WD=5e-4



(f) WD=5e-4

Provides Insights on Trainability



(a) ResNet-110, no skip connections

(b) DenseNet, 121 layers

Loss surfaces of ResNet and DenseNET 121 layers shows which is easier to train
https://losslandscape.com/

Wide vs Narrow Networks

NTK is wide NN

wider →

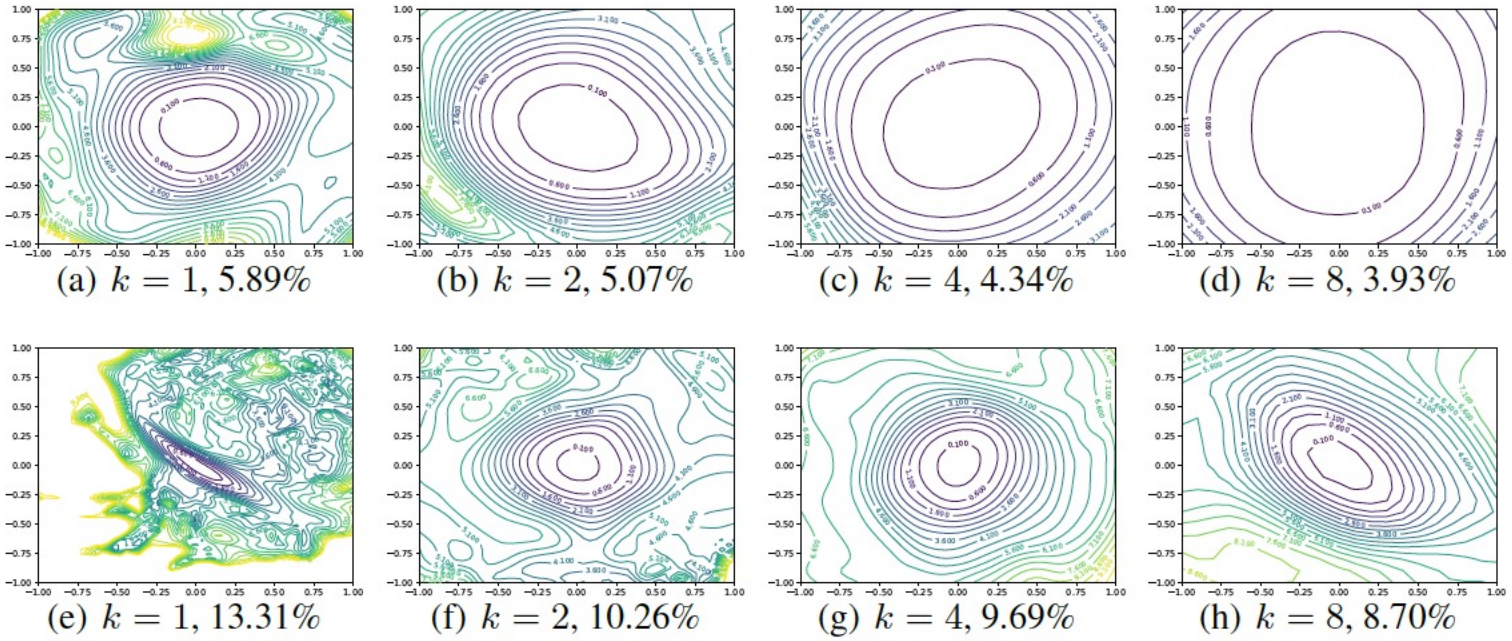


Figure 6: Wide-ResNet-56 on CIFAR-10 both with shortcut connections (top) and without (bottom). The label $k = 2$ means twice as many filters per layer. Test error is reported below each figure.

Wide networks converge to wider basins

Exploring the Geometry and Topology of Neural Networks Loss Landscapes

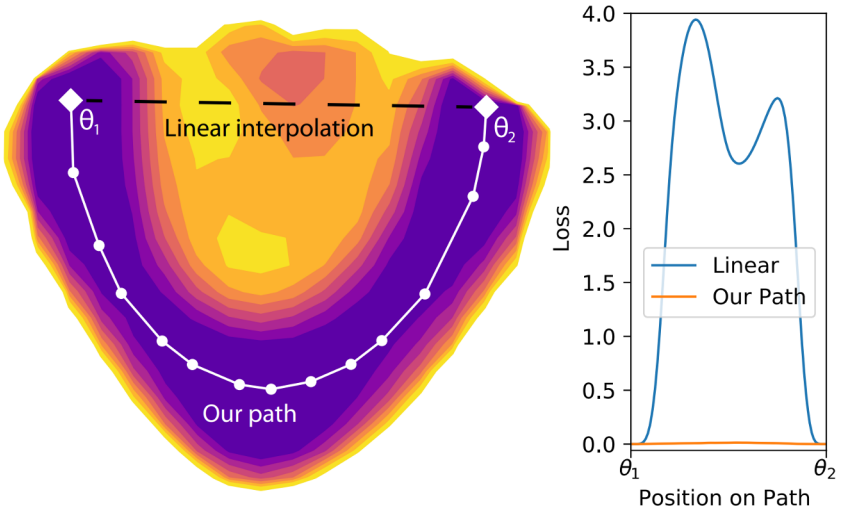
Stefan Horoi

Symposium on Intelligent Data Analysis (IDA) 2022

Existing methods cannot properly visualize complex high-dimensional loss landscape geometric characteristics

Current loss landscape visualization methods are limited by:

- Their linear nature
- The small number of dimensions sampled and visualized (1 or 2)



[Draxler et al., 2018]

Proposed method:

Dynamical “Jump and Retrain” loss landscape sampling

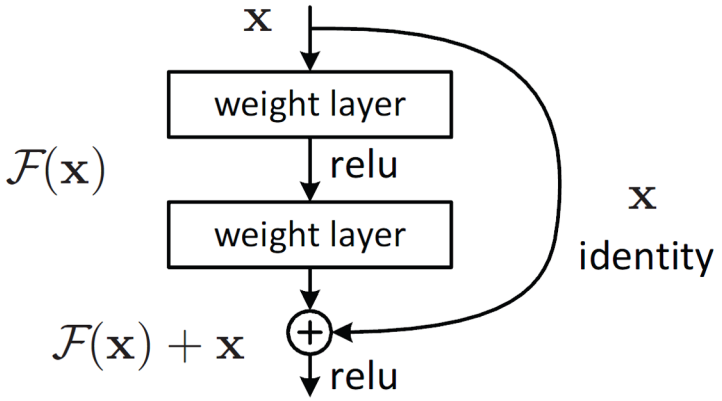
Visualizations based on PHATE dimensionality reduction

Quantification of topological activity using computational homology

Experimental set-up

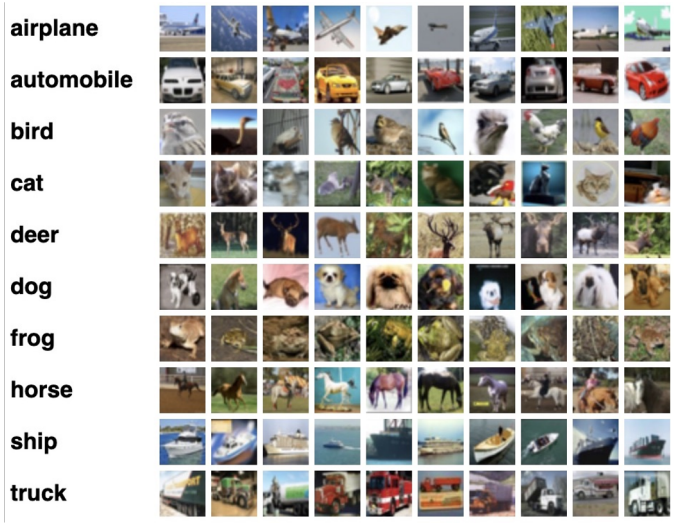
Wide ResNets

[Zagoruyko & Komodakis, 2016]



CIFAR10

[Krizhevsky, 2009]



108 networks: same initalization

{10, 16, 22}	{1,2}	{0, 0.0001, 0.001}	{32, 128, 512}	{no data augmentation,
Depth	Width	Weight decay	Batch size	random flips and crops}

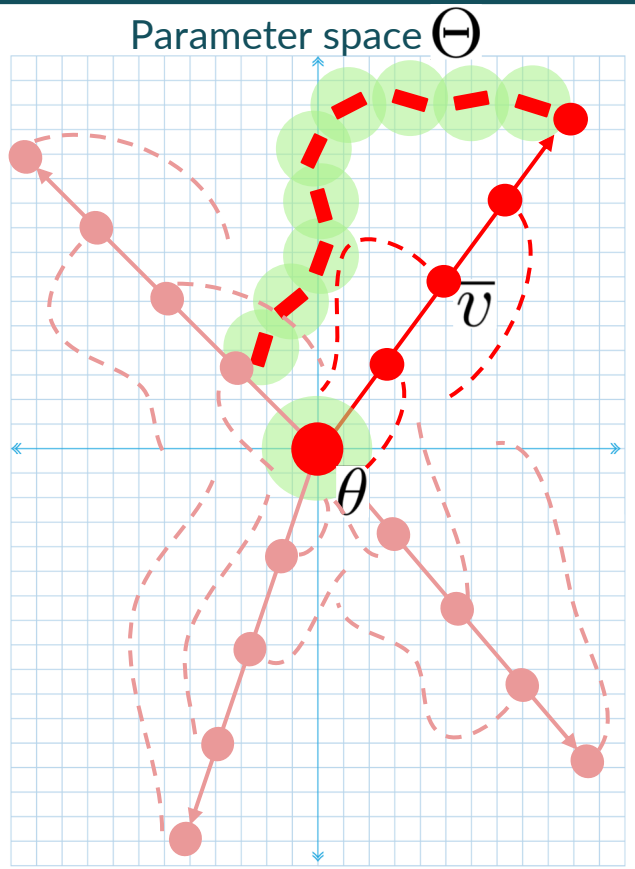
"Jump and retrain" sampling of relevant low-loss manifolds on the loss landscape that are reachable during training

Given a minimum: θ

1. Pick a random direction and filter-normalize it: $\overline{v}$
2. Pick a step_size in {0.25, 0.5, 0.75, 1}: k
3. Set the ANN parameters to be: $\theta + k\overline{v}$
4. Retrain the ANN and record data

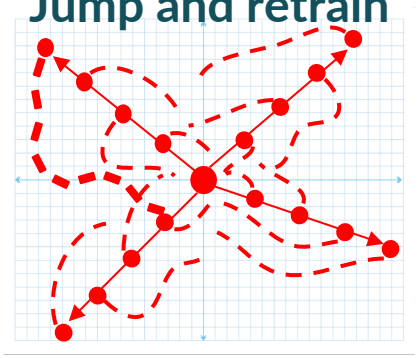
Repeat for 4 or 5 directions and all step sizes

Low loss regions of the loss landscape

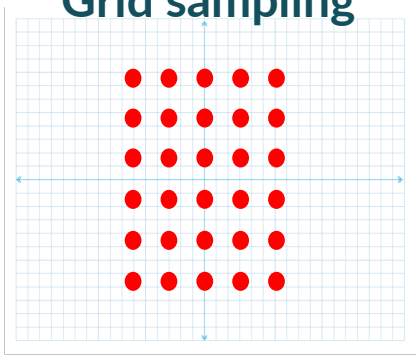


"Jump and retrain" sampling holds more information about generalization and training than past methods

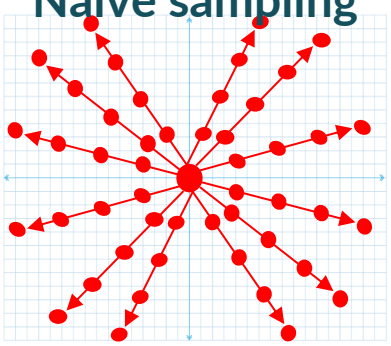
Jump and retrain



Grid sampling



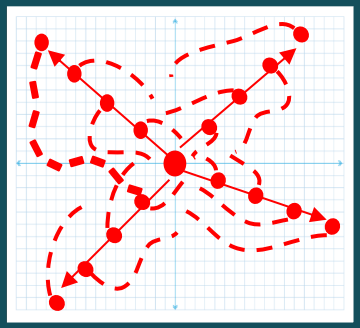
Naive sampling



Mean accuracy and standard error of 11 simple classifiers:

Features	5 class gen.	Weight decay	Data augmentation

Dynamical "Jump and Retrain" loss landscape sampling



Holds more information about generalization and training than past methods

Visualizations based on PHATE dimensionality reduction

Quantification of topological activity using computational homology

Dimensionality reduction using **PHATE** [Moon et al., Nature Biotechnology, 2019]

a) Given a data matrix

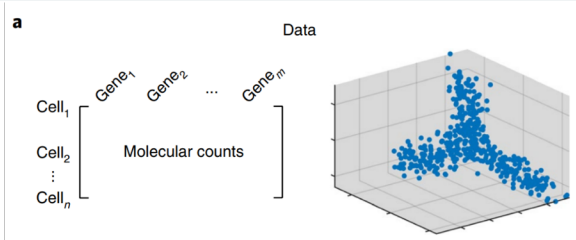
b) Compute pairwise distances

c) Transform distances to affinities to encode local information

d) Learn global relationships via diffusion

e) Encode the learned relationships using the potential distance

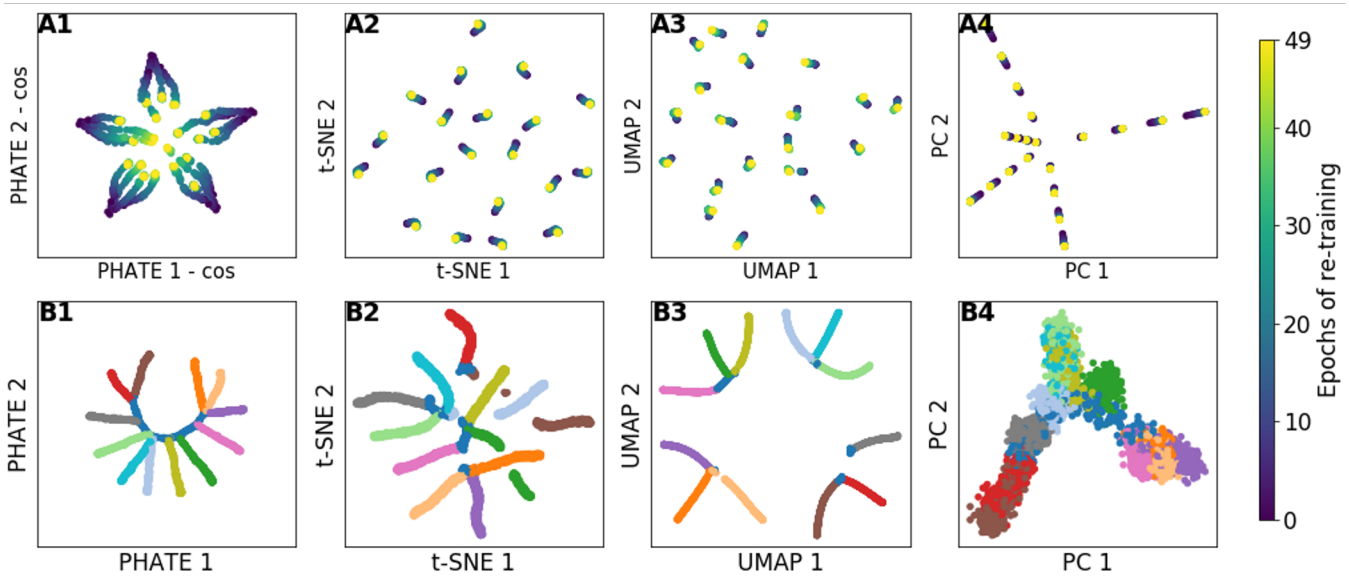
f) Embed the potential distance information into low dimensions via MDS



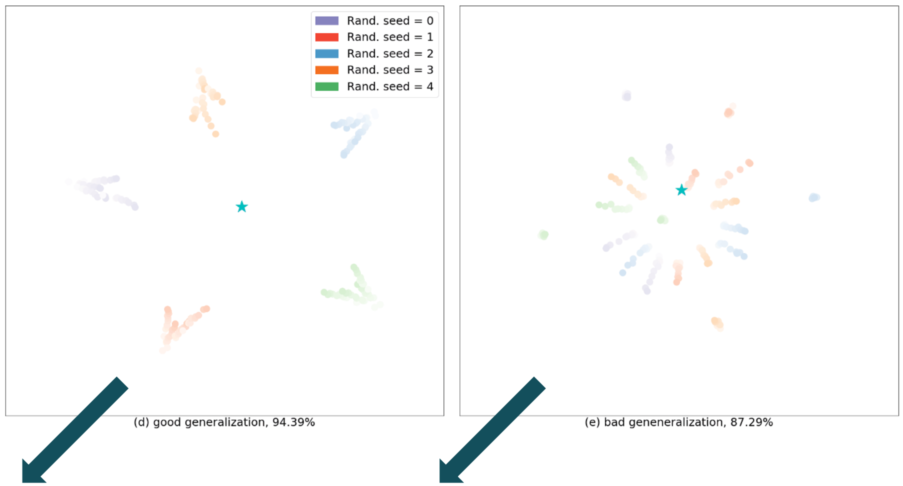
PHATE visualizations better preserve data variability and global structure

PHATE (with cosine distance) improves on past, linear, methods by:

- Capturing variance in sampled data from all relevant dimensions and embedding it in a low-D space
  - (as would've done other modern dimensionality reduction methods...)
- Preserving high-D trajectories and global structures of data



PHATE visualization results in stark differences between networks that memorize (overfit) vs. networks that generalize

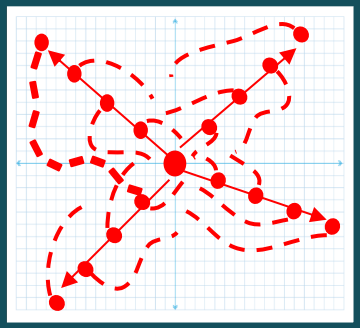


(d) good generalization, 94.39%

(e) bad geneneralization, 87.29%

to get bad minima : train on random labels first
then train on right labels

Dynamical “Jump and Retrain” loss landscape sampling



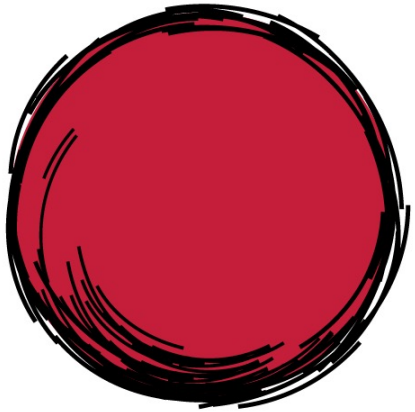
Holds more information about generalization and training than past methods

Visualizations based on PHATE dimensionality reduction

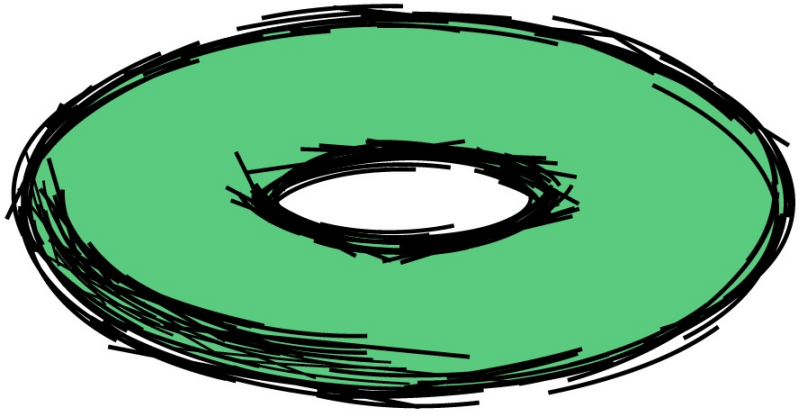
Highlights geometric differences in the low loss manifolds around different minima

Quantification of topological activity using computational homology

Characterization of D-dimensional Holes

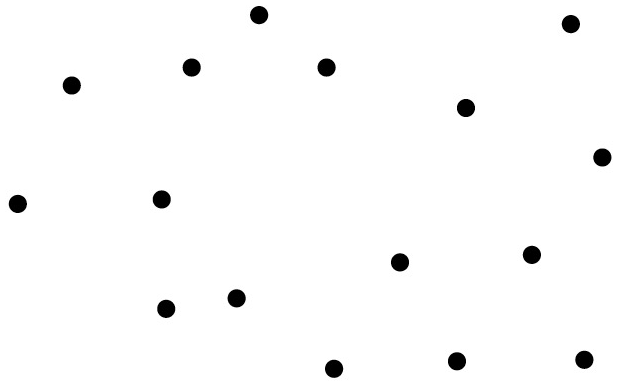


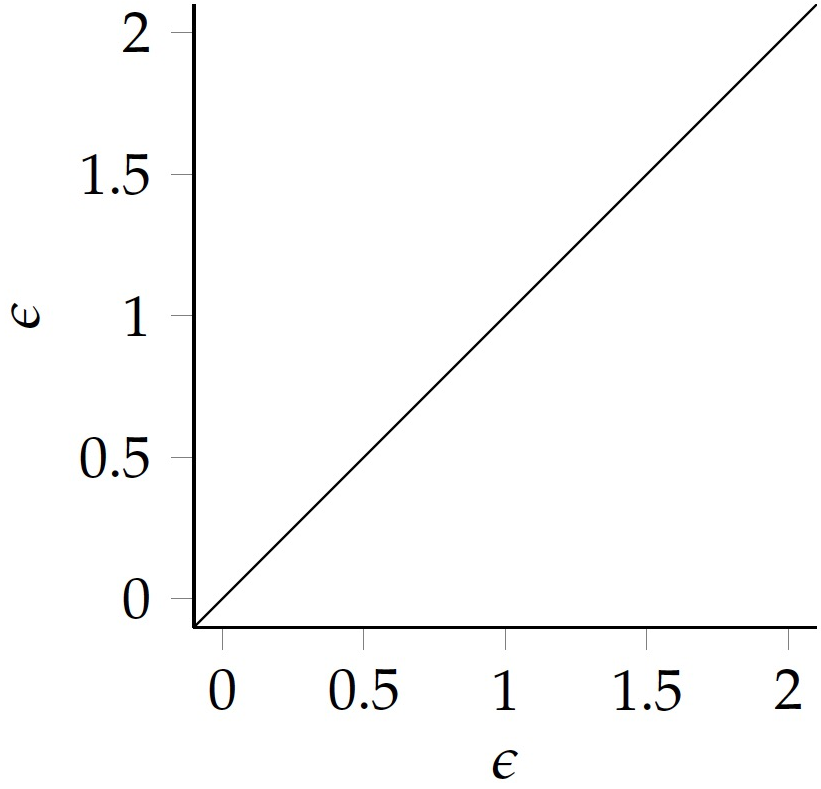
$\beta_0 = 1, \beta_1 = 0, \beta_2 = 1$



$\beta_0 = 1, \beta_1 = 2, \beta_2 = 1$

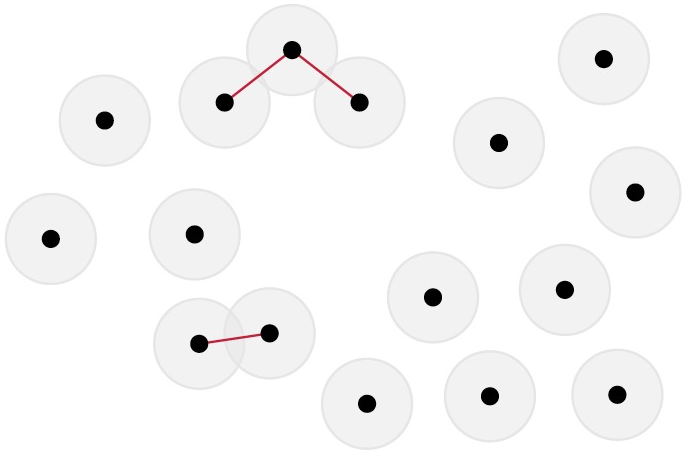
Computational Homology (Vietoris Rips Filtration)

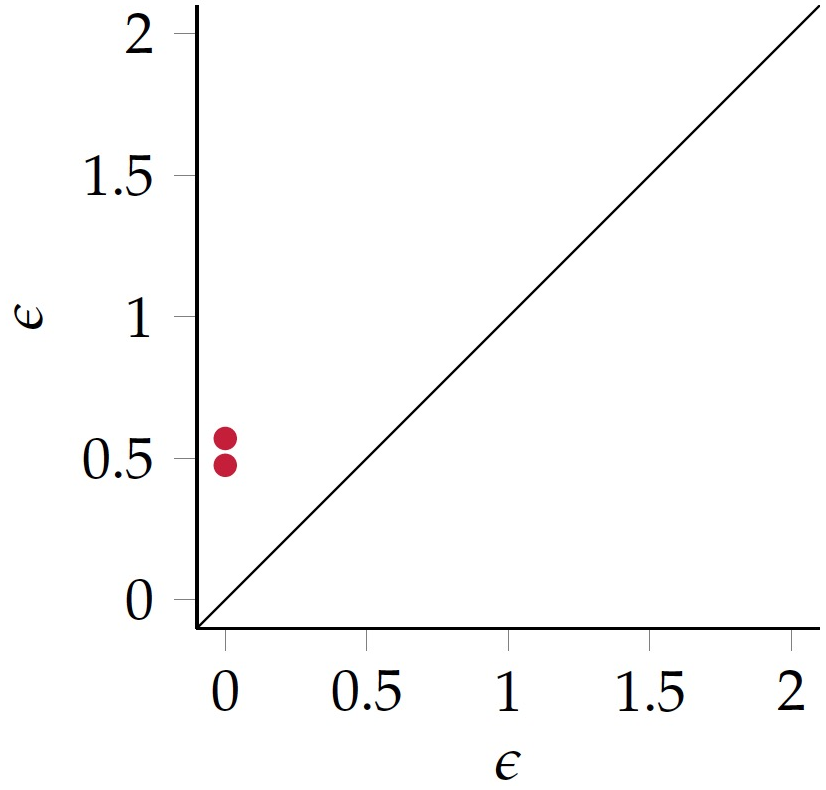




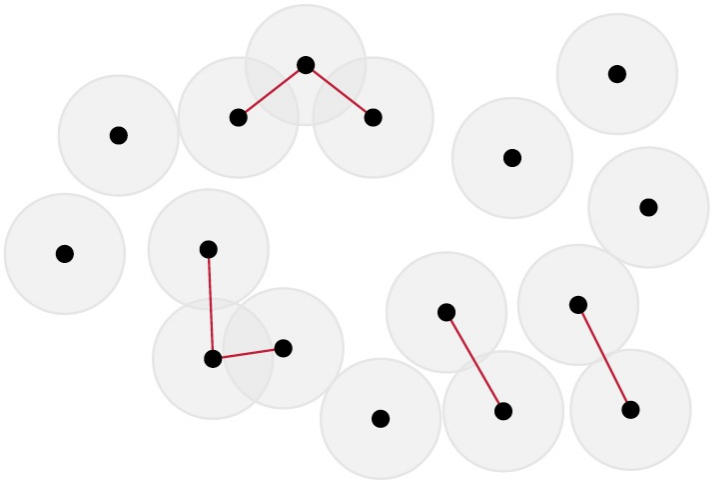
Computational Homology (Vietoris Rips Filtration)

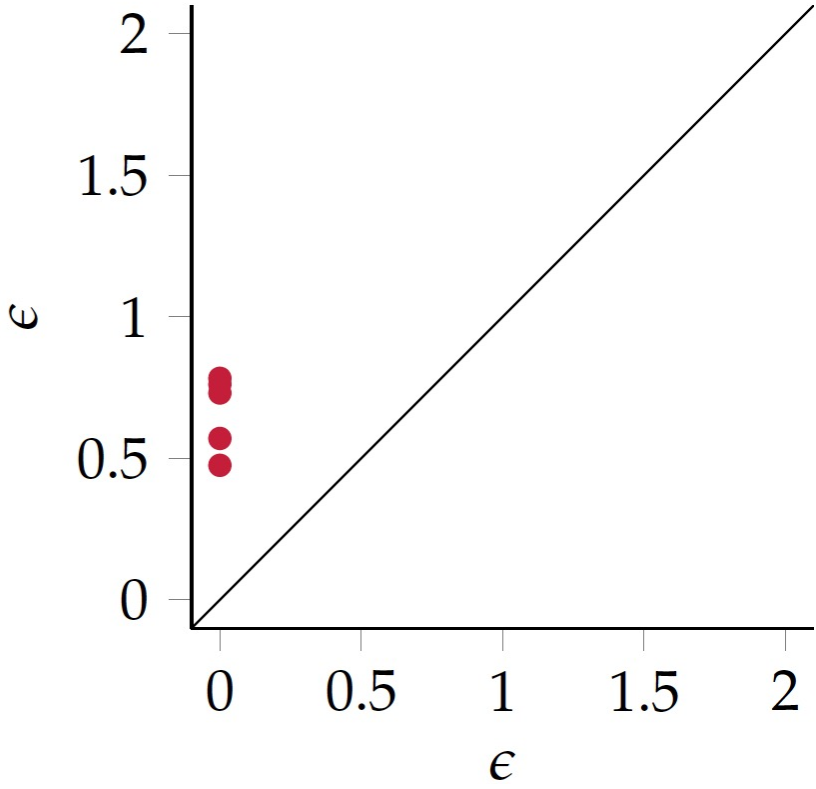
过滤



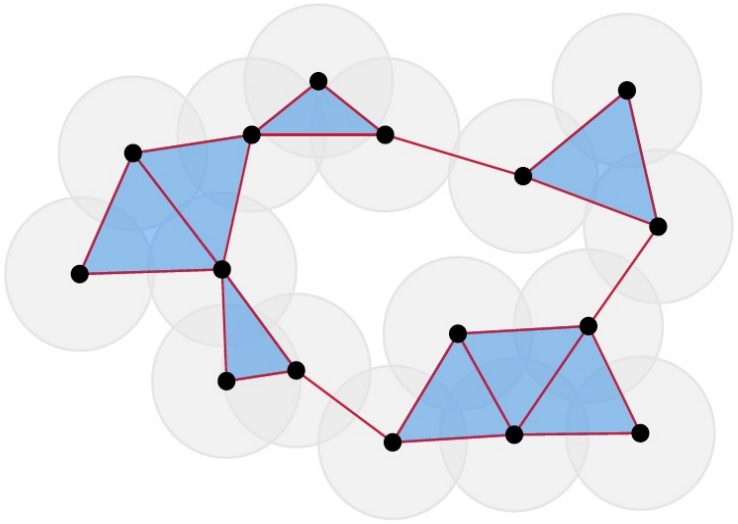


Computational Homology (Vietoris Rips Filtration)

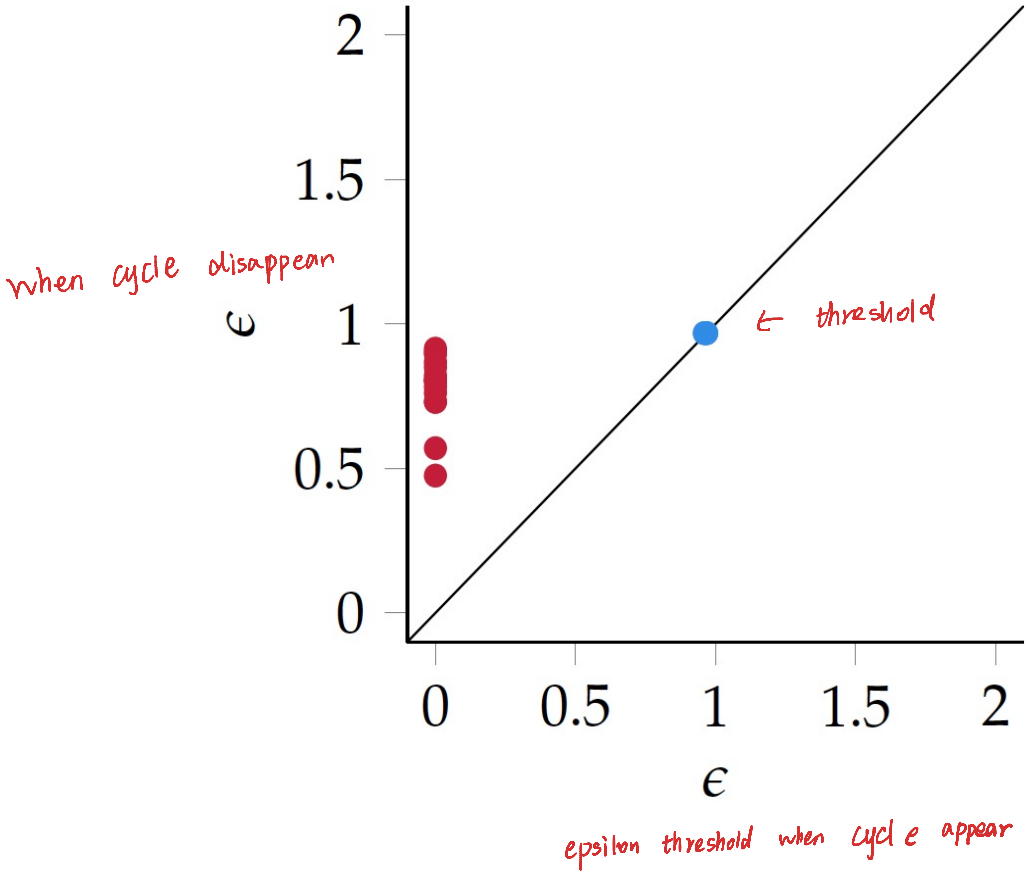




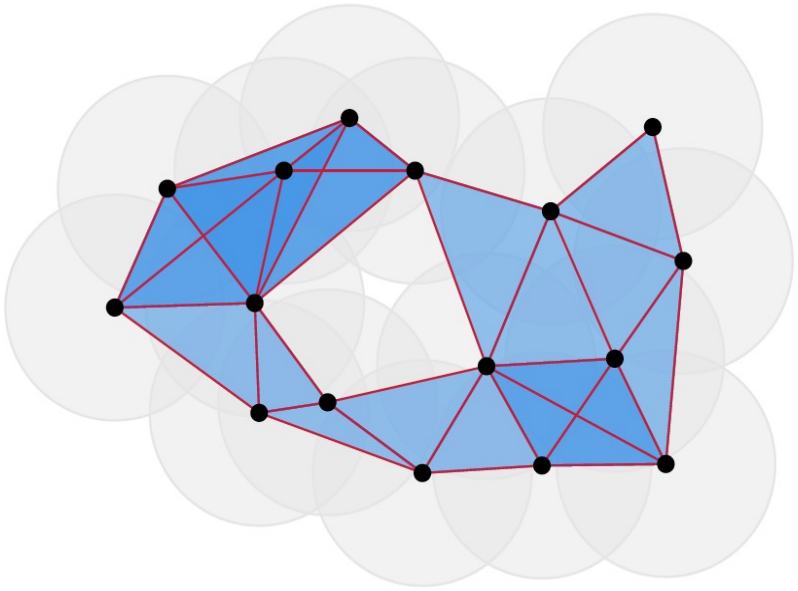
Computational Homology

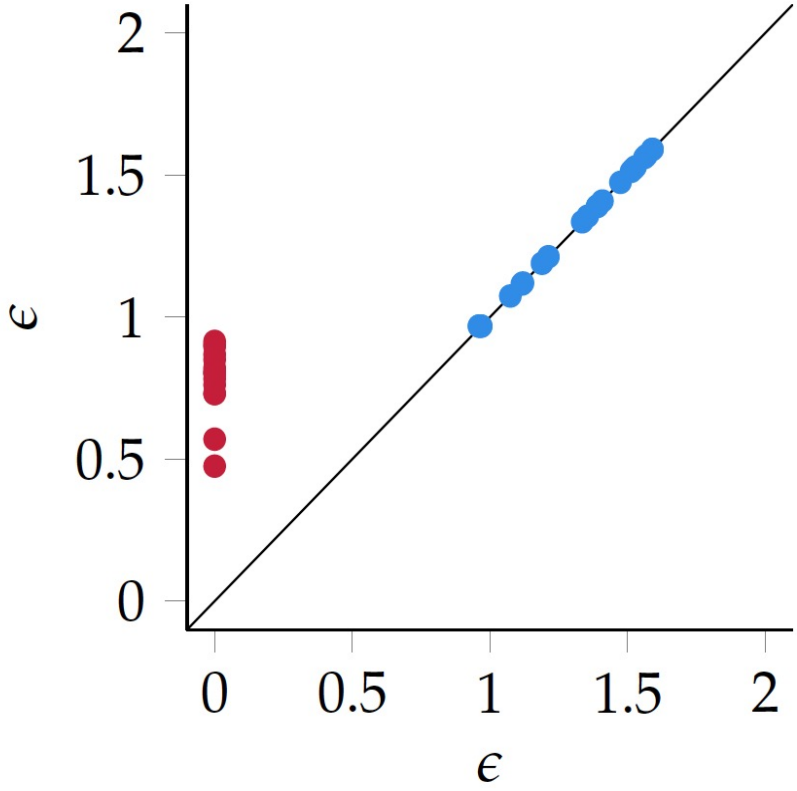


persistence diagram

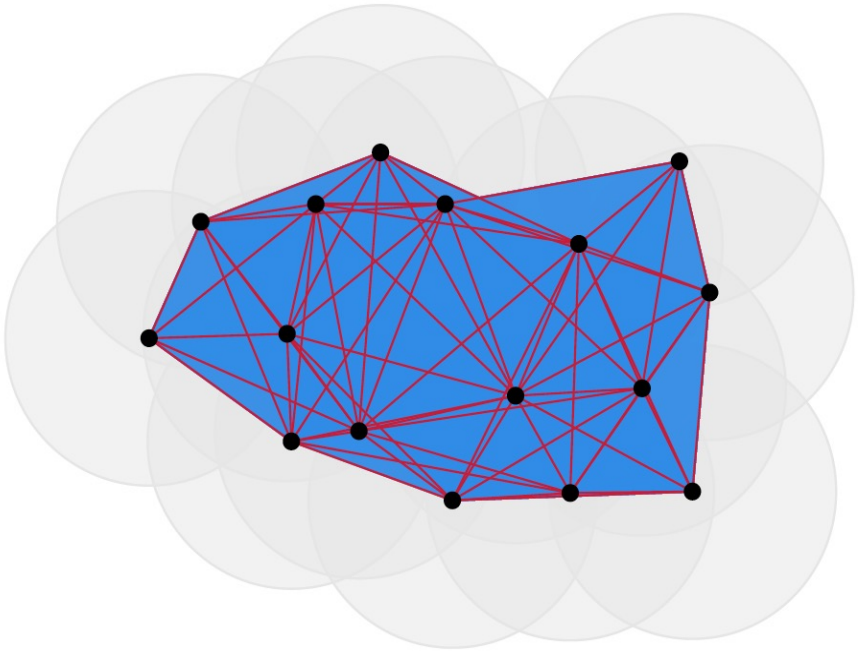


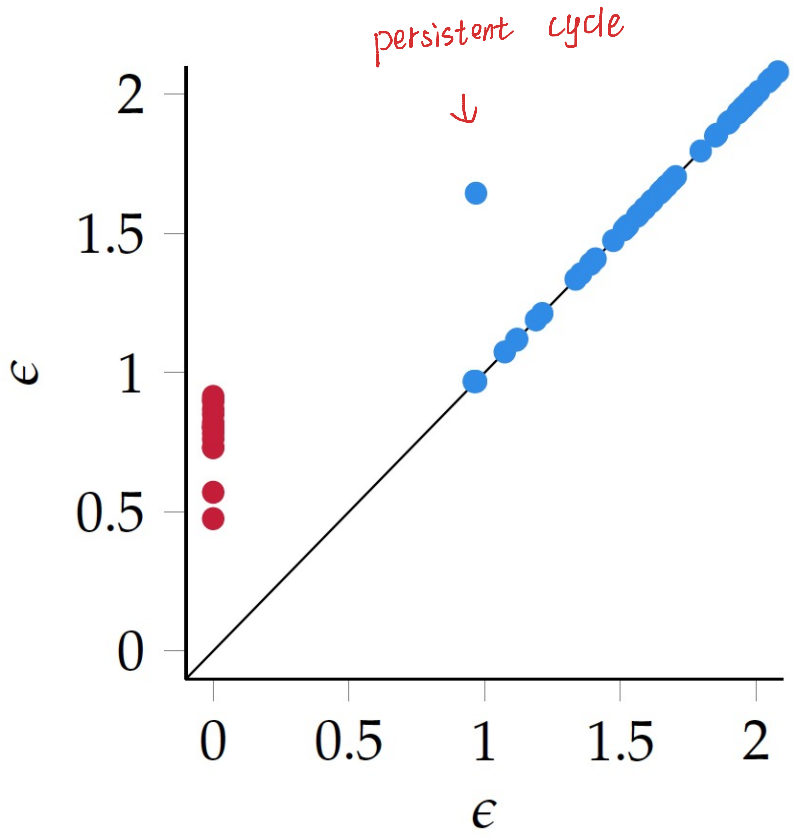
Computational Homology





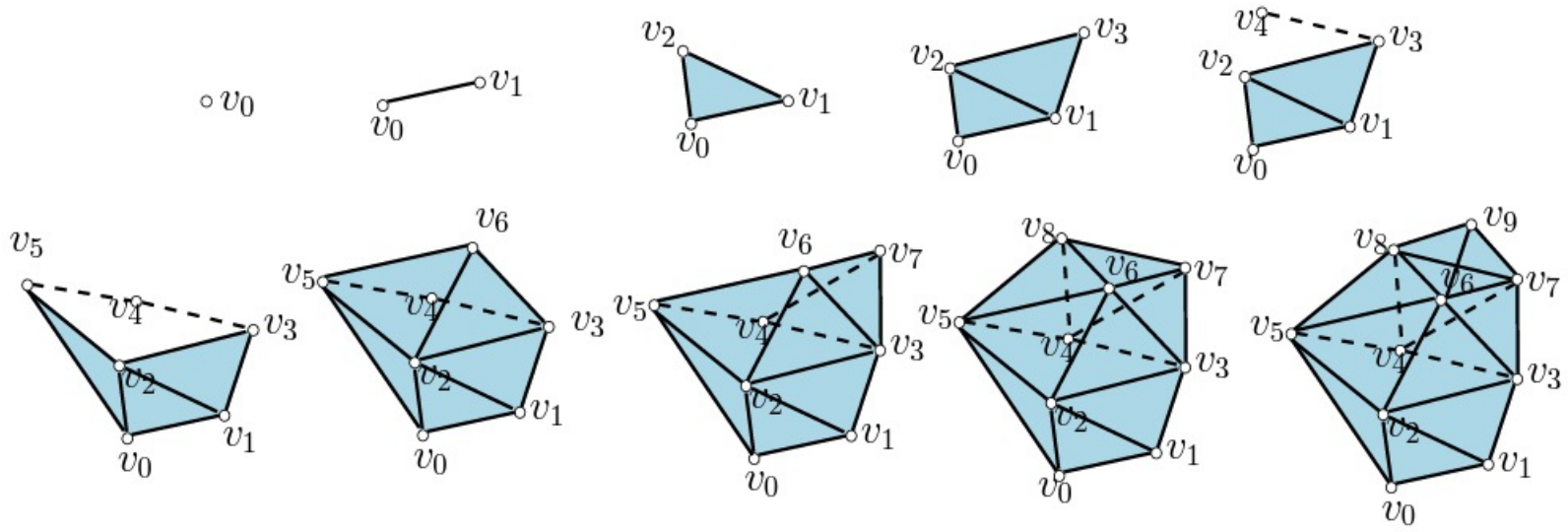
Computational Homology





Vertex Value Filtration 过滤

now each vertex has a value eg. loss

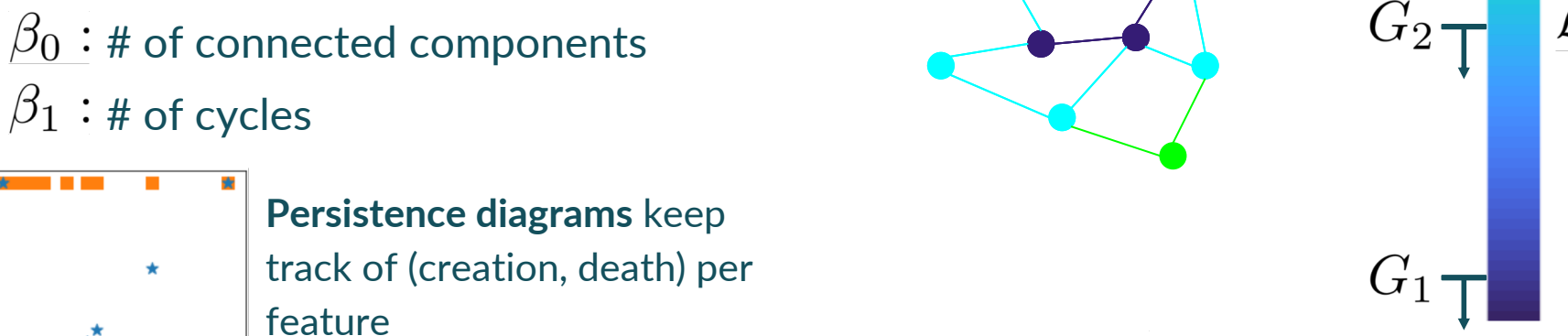


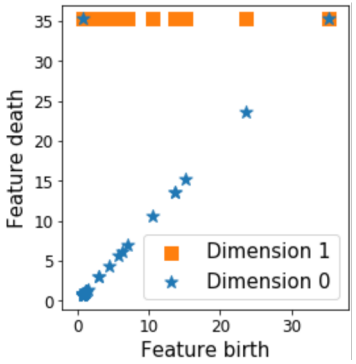
Quantifying the topological features of the J&R sampled trajectories using computational topology

1. Construct 20-NN graph using PHATE distances
2. Assign to each node the associated loss value
3. Create a graph filtration by increasing the loss
4. Compute topological features for each G_i

β_0 : # of connected components

β_1 : # of cycles

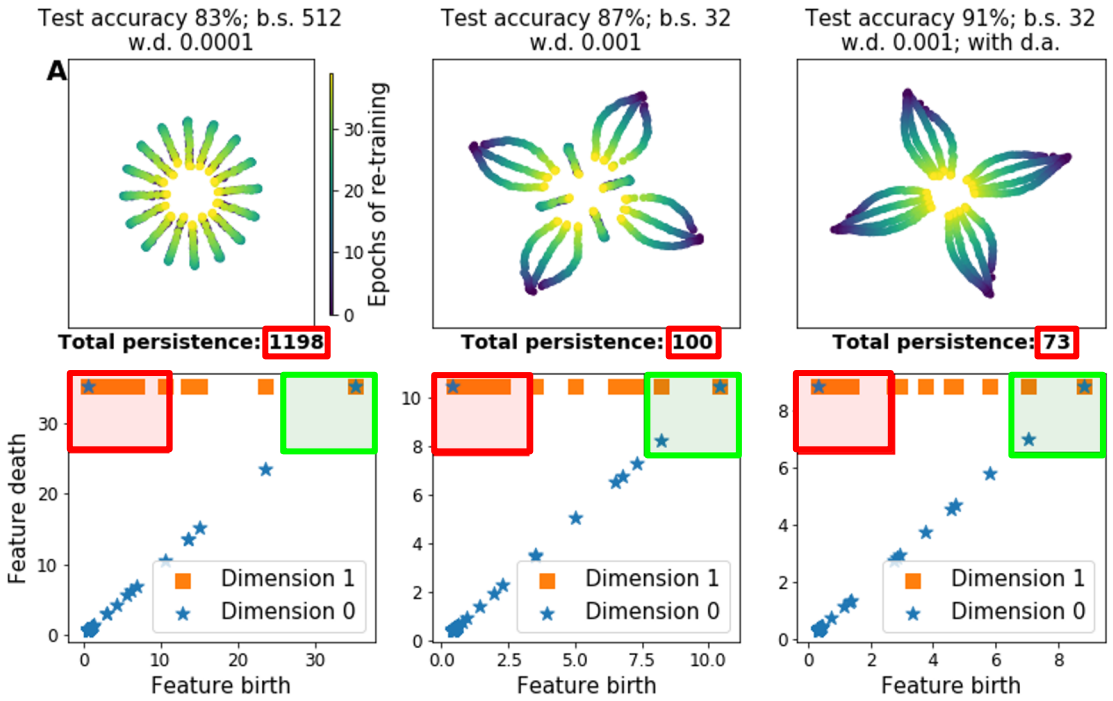




Persistence diagrams keep track of (creation, death) per feature

Total persistence: $\mathrm{pers}(\mathcal{D}) := \sum_{(c,d)\in\mathcal{D}} \underbrace{|d-c|}_{\text{persistence}}^2$

Minima that generalize well are surrounded by low loss manifolds with less topological activity (low total persistence)



Future directions

- Dimensionality reduction methods that better take into account the time dependency of the sampled data
- Principled approaches for the empirical analysis of loss landscapes

A lot more work is required!

Further reading

- Draxler et al. 2019 Essentially No Barriers in Neural Network Energy Landscape
- Li et al. 2018 Visualizing the Loss Landscape of Neural Nets
- Dinh et al 2019 Sharp Minima Can Generalize For Deep Nets
- Keskar, Nocedal, Mudigere, Smelyanskiy, and Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. https://arxiv.org/pdf/1609.04836.pdf
- Li, Xu, Taylor, Studer, and Goldstein. Visualizing the Loss Landscape of Neural Nets. https://papers.nips.cc/paper/7875-visualizing-the-loss-landscape-of-neural-nets.pdf.
- Horoi et al. Exploring the Geometry and Topology of Loss Landscapes https://arxiv.org/abs/2102.00485

Essentially No Barriers in Neural Network Energy Landscape

Felix Draxler[12], Kambis Veschgini[2], Manfred Salmhofer[2], Fred A. Hamprecht[1]

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

1 IWR Interdisciplinary Center for Scientific Computing

Heidelberg Collaboratory for Image Processing

2 ITP

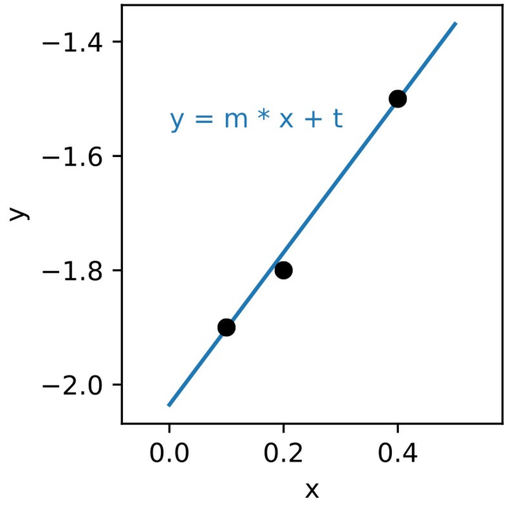
Neural Network Energy Landscape

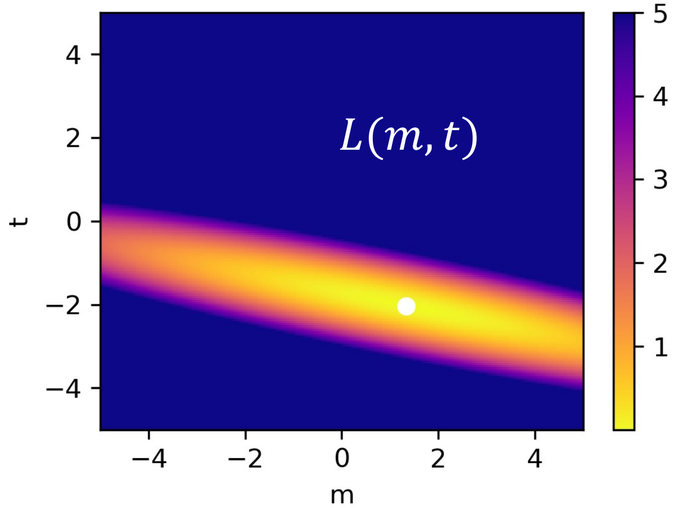
Loss

$$L : \Theta \rightarrow \mathbb{R}^+$$

Architecture (CNN, …)
Data (CIFAR, …)

Network parameters





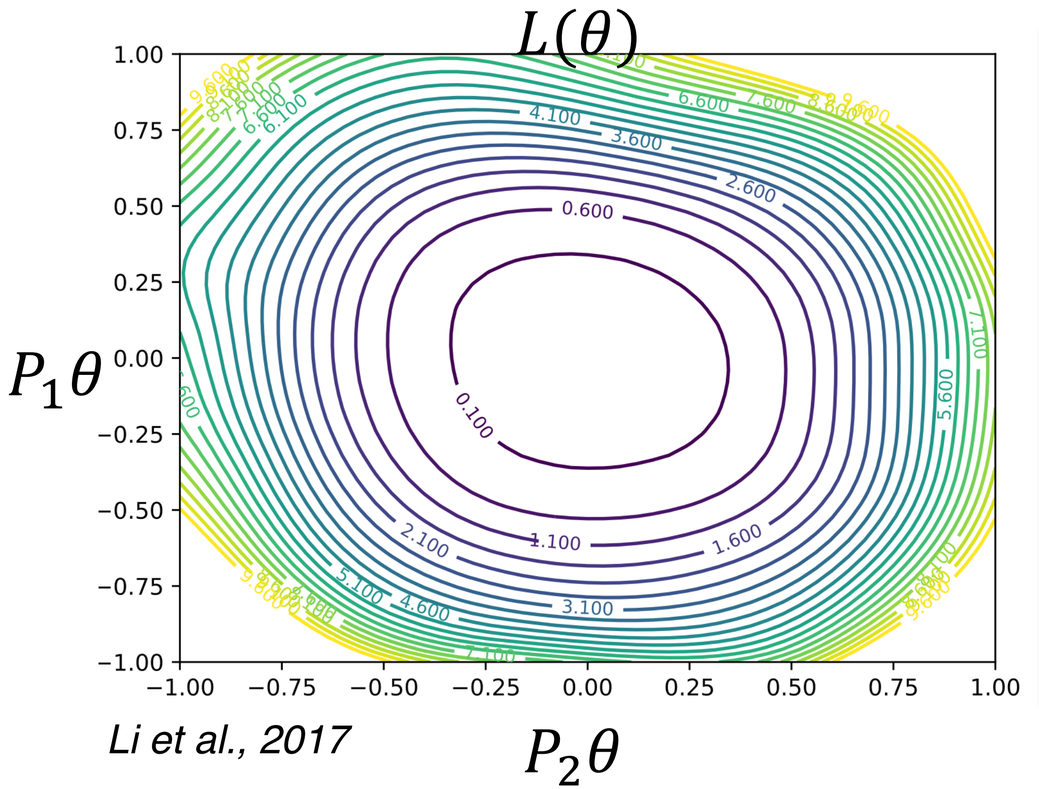
Neural Network Energy Landscape

Loss $L : \Theta \rightarrow \mathbb{R}^+$

non-convex
high-dimensional

no bad local minima
good generalisation

Minima

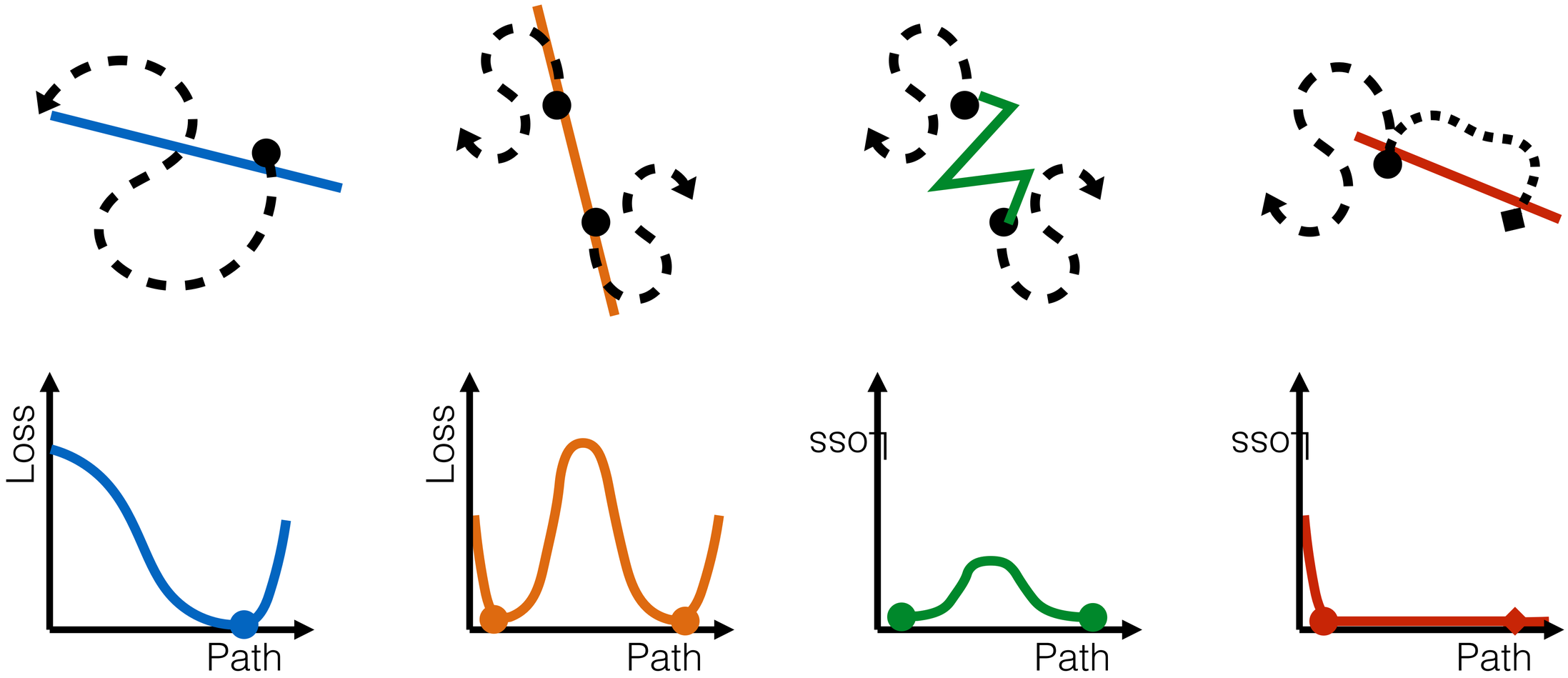


Li et al., 2017

- Minima at bottom of **valleys**
- **Wide** minima generalise **better** (Keskar et al., ICRL 2017)
- Training choices **shape** the valley (Li et al., 2017)

Method: **Sample loss** around minima.

Paths through Landscape



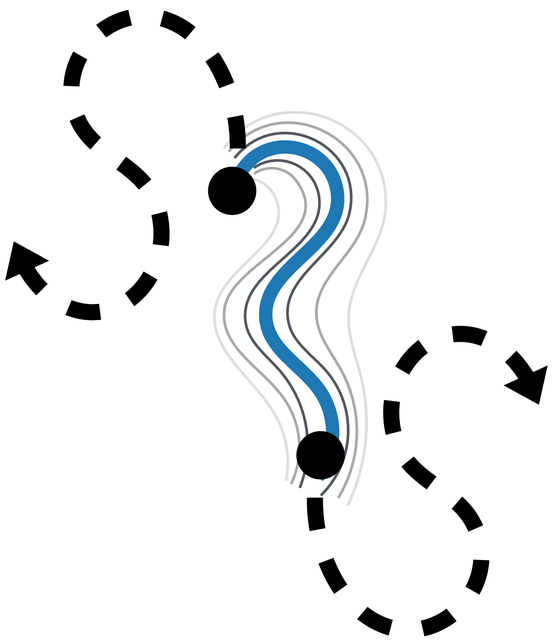
Goodfellow et al., ICRL 2015

Keskar et al., ICRL 2017

Freeman & Bruna, ICRL 2017

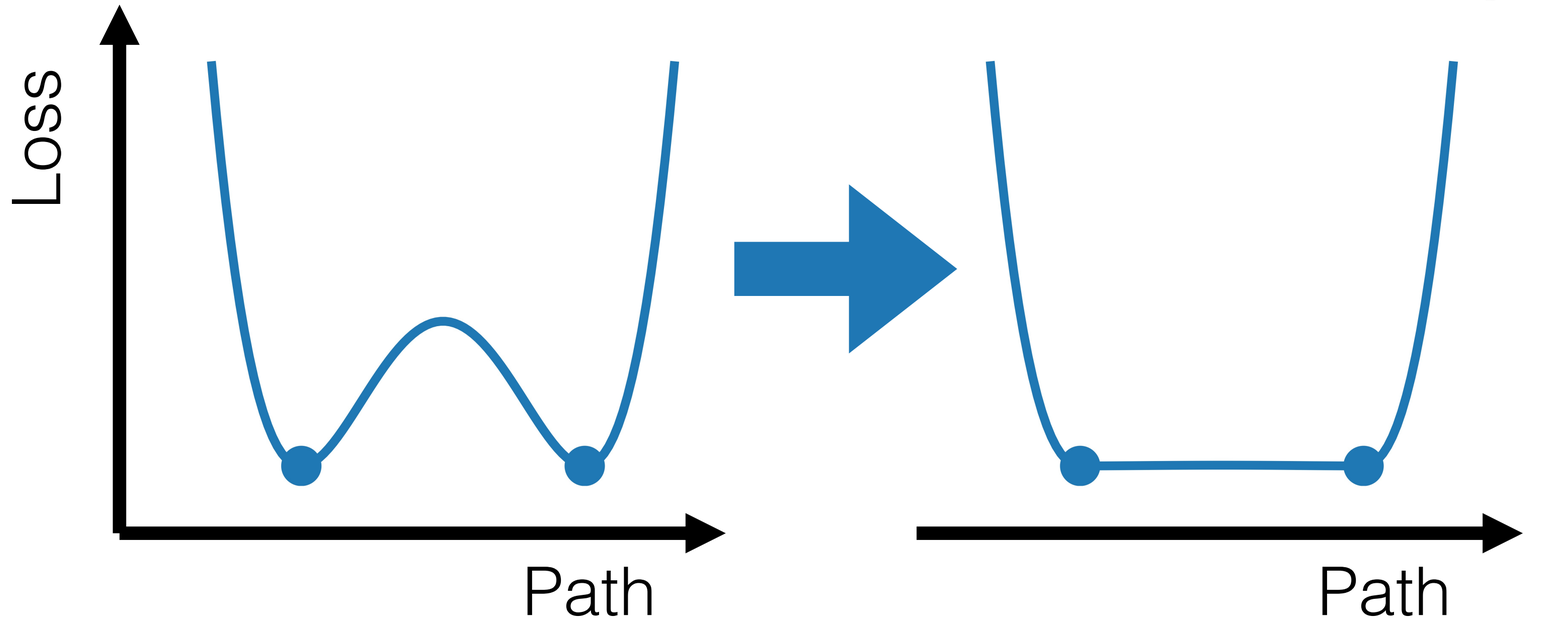
Sagun et al., ICRL 2018

Our Contribution



~~**Isolated Minima**~~

Connected Minima
Mode Connectivity



➡️ **No Bad Minima**

Minima are not equal

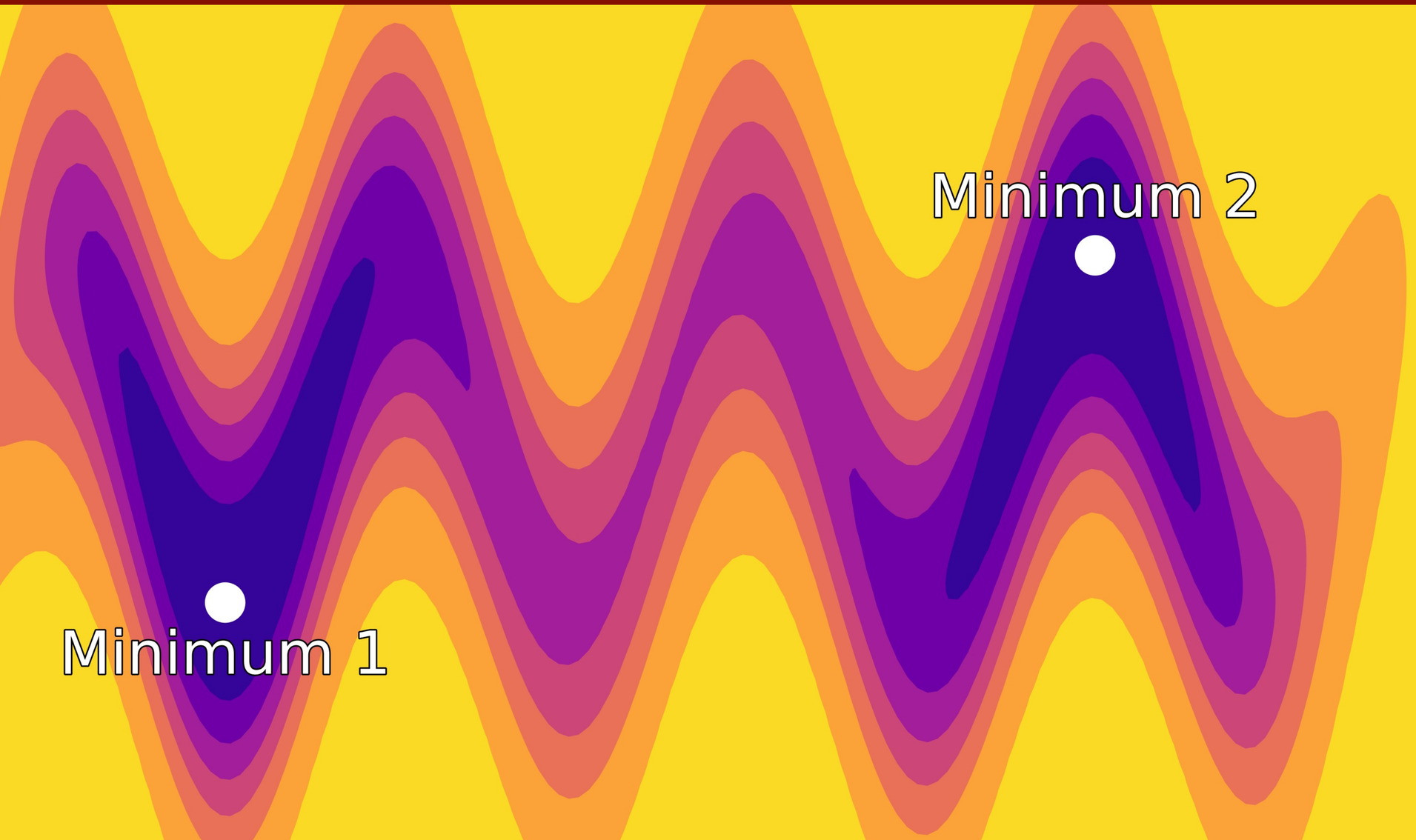
Overlap of misclassified instances **< 50%**

CIFAR10 dataset

Misclassified
by Minimum 1

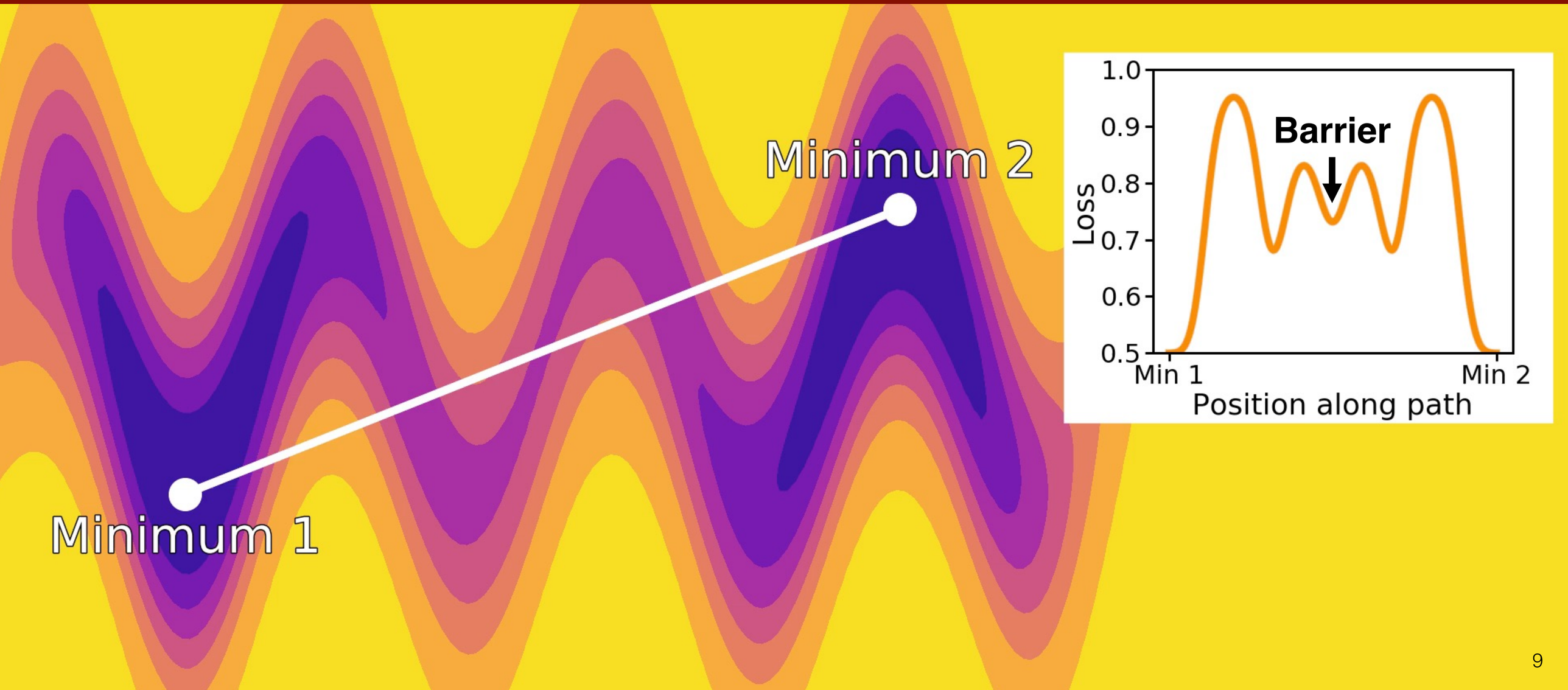
Misclassified
by Minimum 2

Method



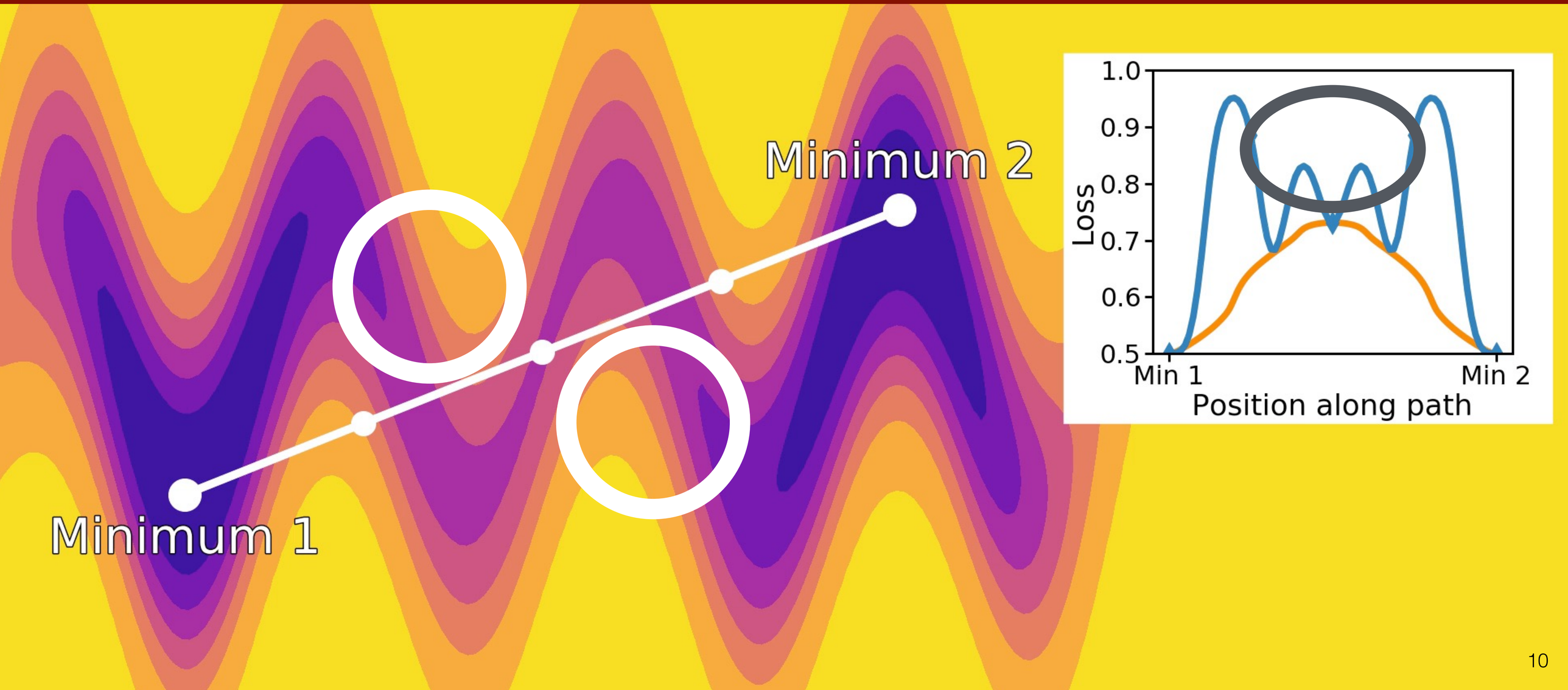
Task:
Find path
with the lowest
highest point.

Method


Minimum 2
Minimum 1

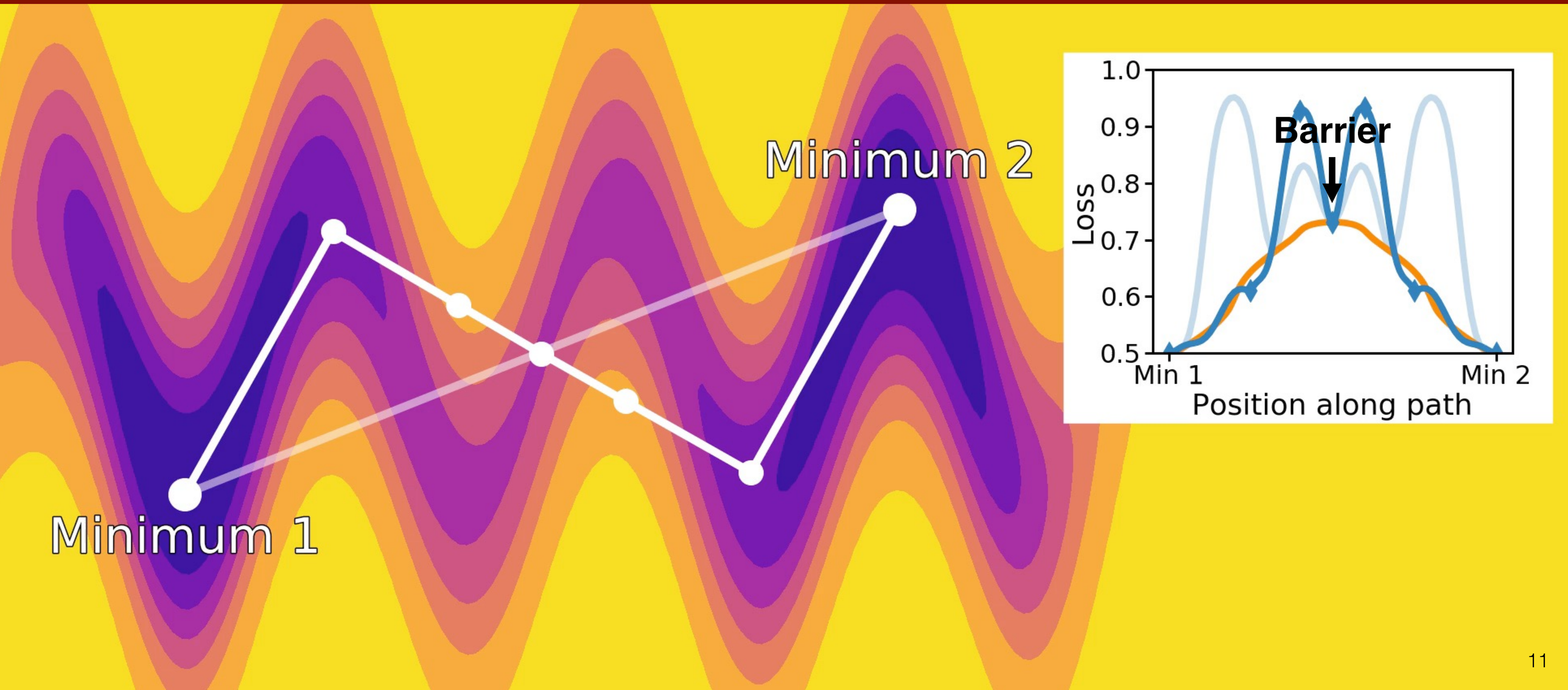
1.0
0.9
0.8
0.7
0.6
0.5
Loss
Barrier
Min 1
Min 2
Position along path

Method


Minimum 2
Minimum 1

1.0
0.9
0.8
0.7
0.6
0.5
Loss
Min 1
Min 2
Position along path

Method


Minimum 2
Minimum 1

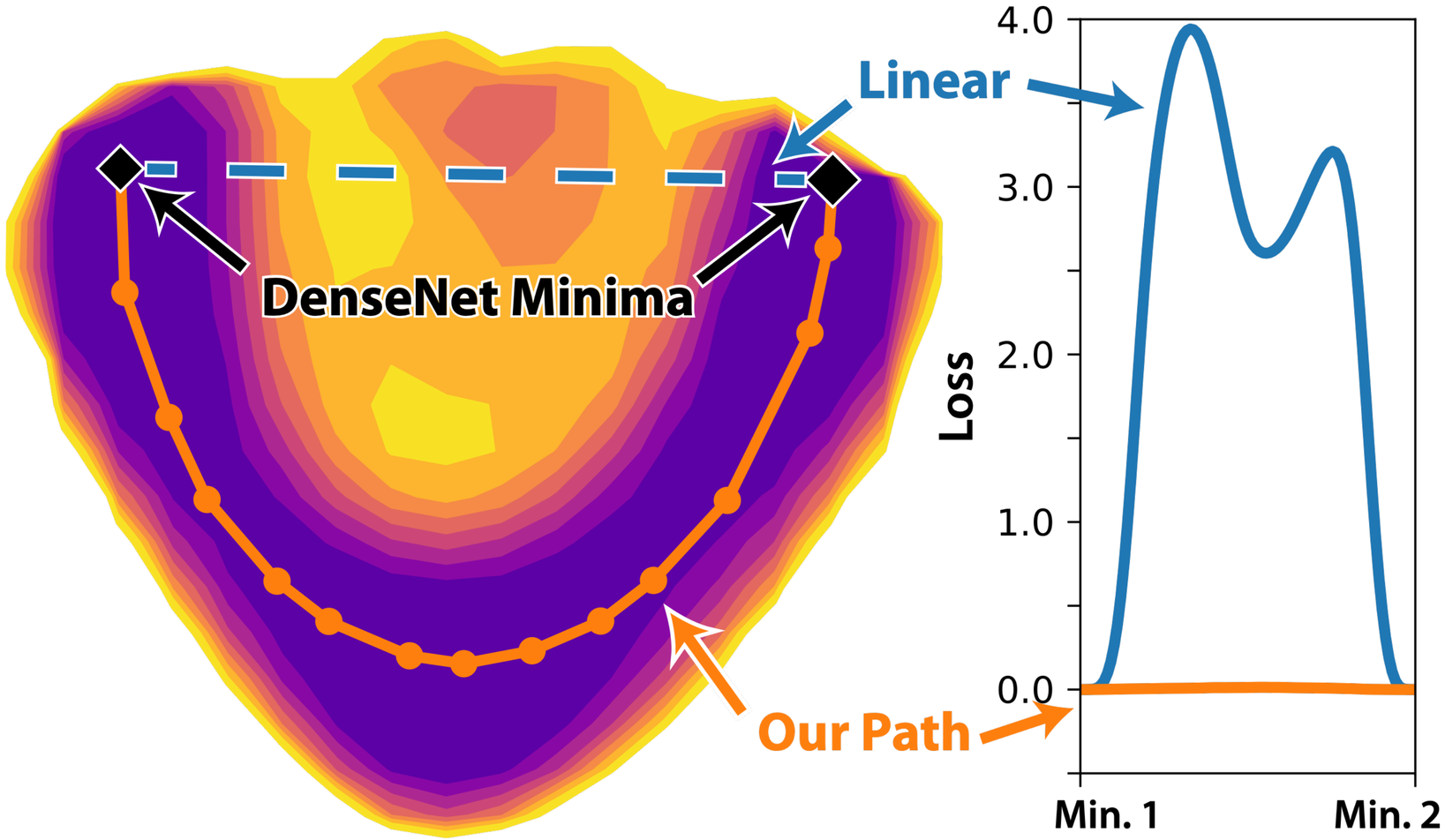
1.0
0.9
0.8
0.7
0.6
0.5
Loss
Barrier
Min 1
Min 2
Position along path

Method

1. Initialise linear path between independent minima
2. Iterate:
    1. Move pivots by loss gradient
    2. Insert pivots where needed
3. Read off barrier loss

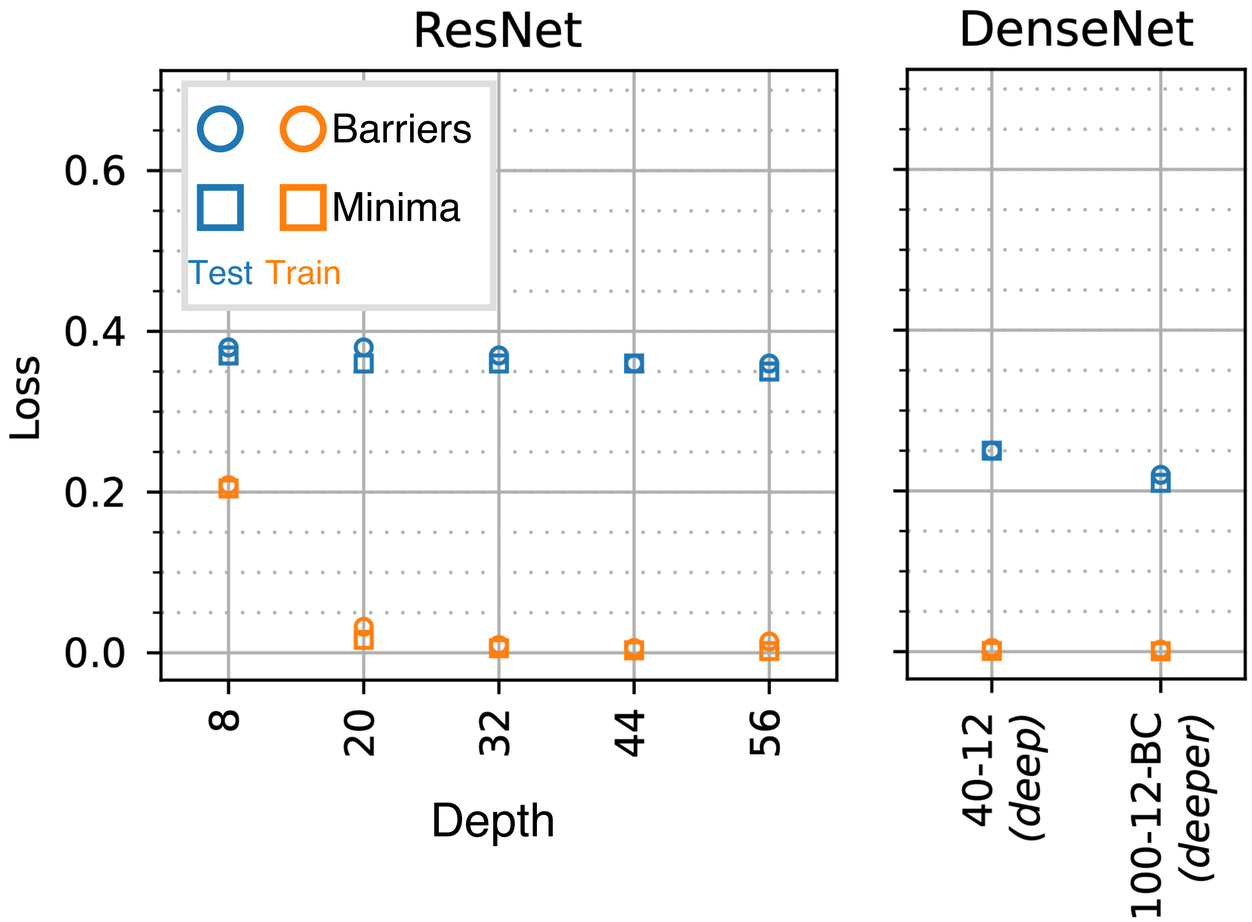
NEB: Nudged Elastic Band (Jónsson et al., 1998)
AutoNEB: Automated Nudged Elastic Band (Kolsbjerg et al., 2016)

ResNet and DenseNet

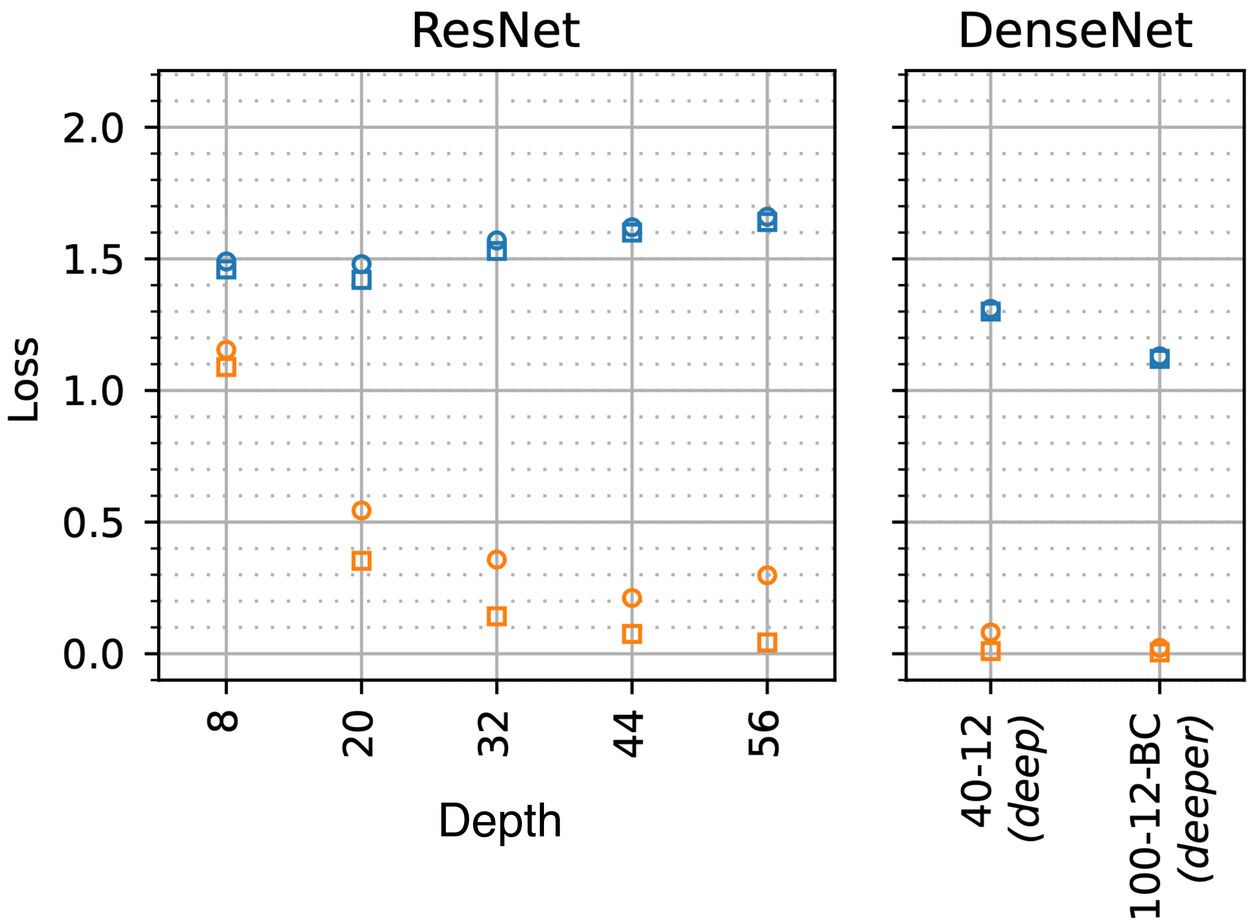


ResNet and DenseNet

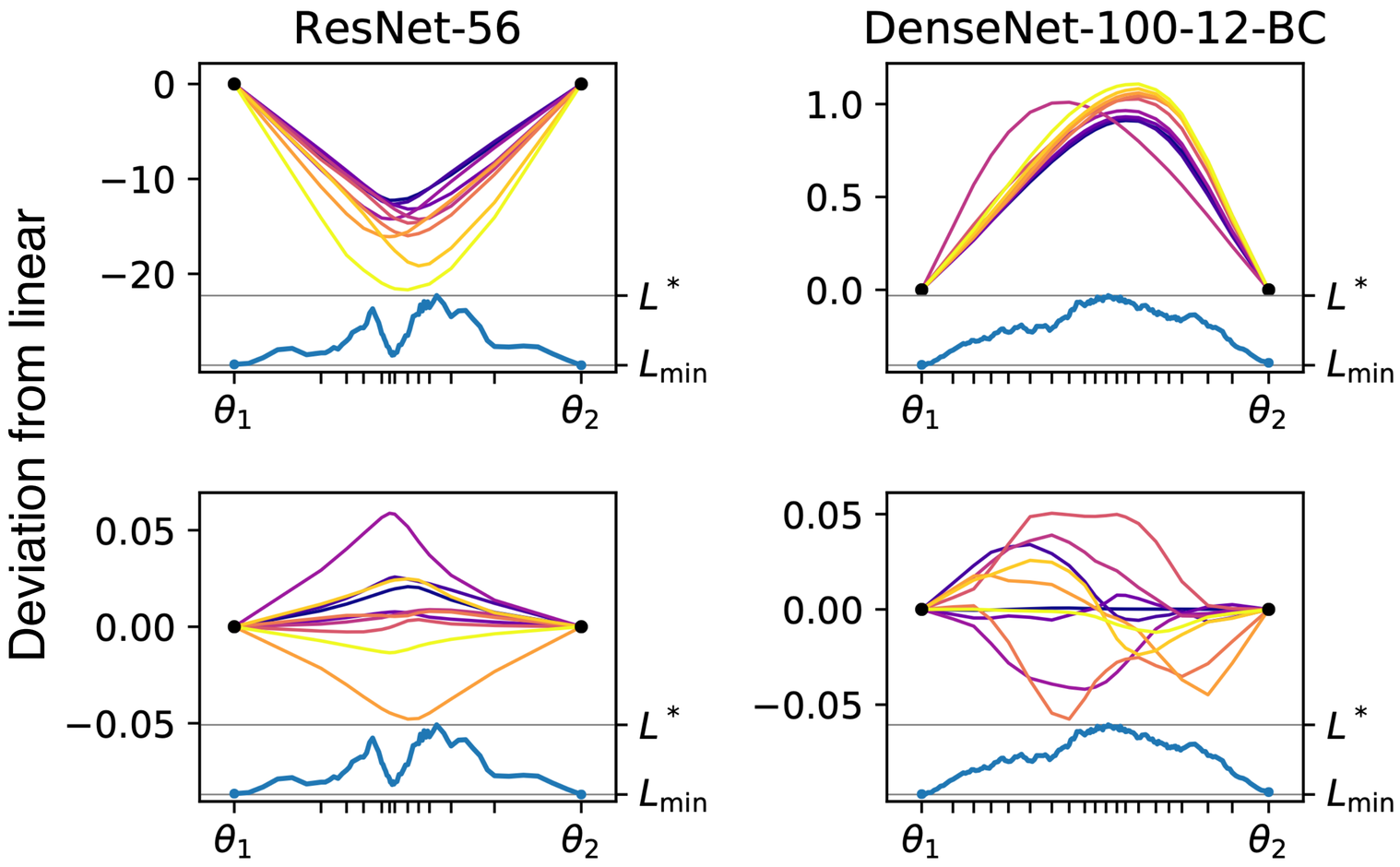
CIFAR10: No barriers



CIFAR100: Very low barriers



Smooth Paths

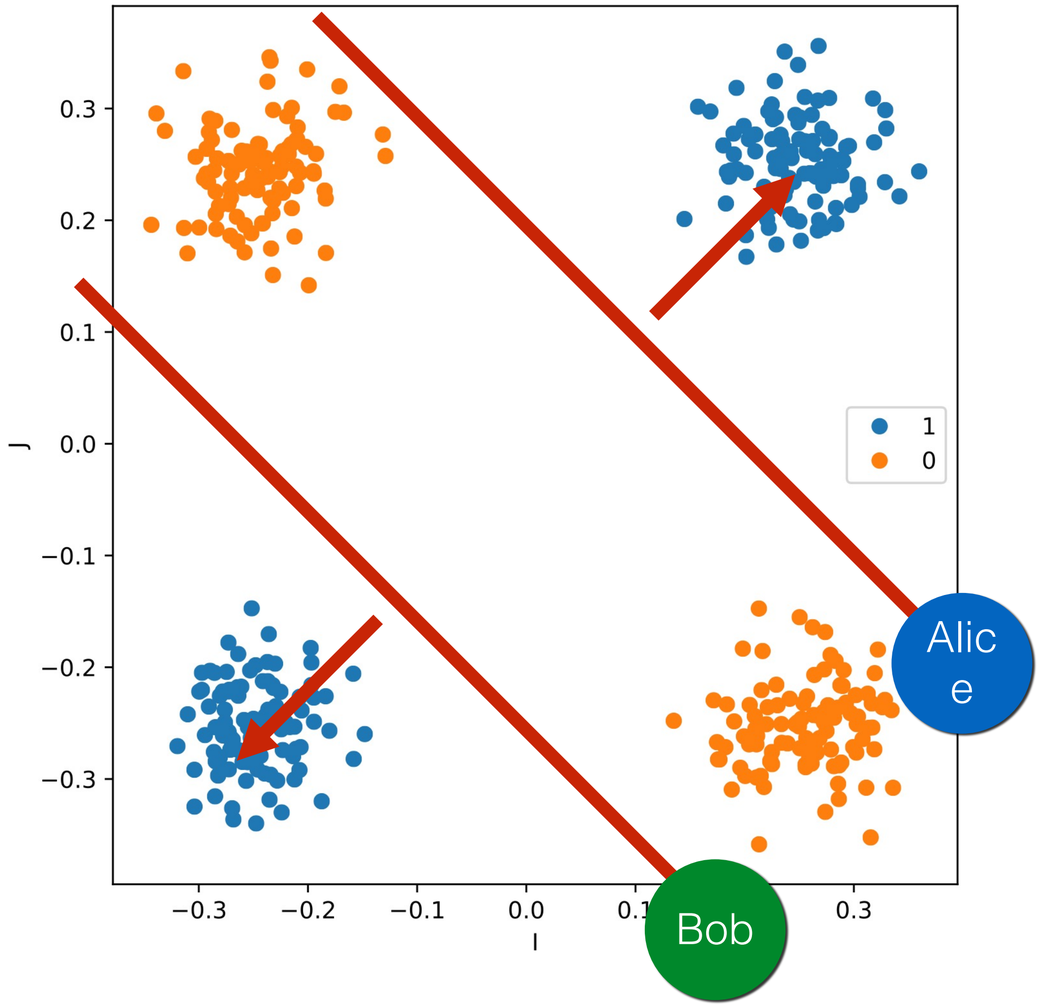


Peaked distance: Most deviating coordinates

Smooth trajectories: Randomly chosen coordinates

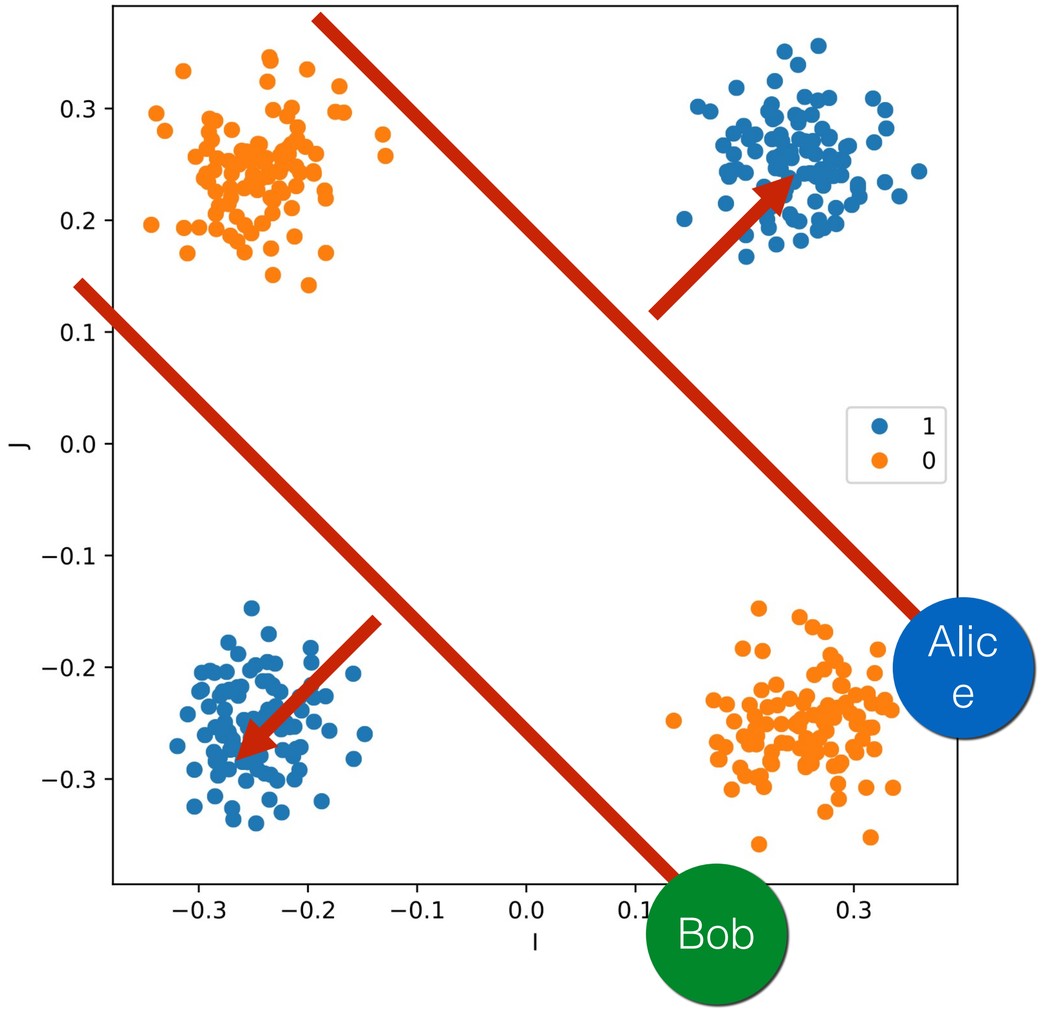
Redundancy

XOR textbook sample



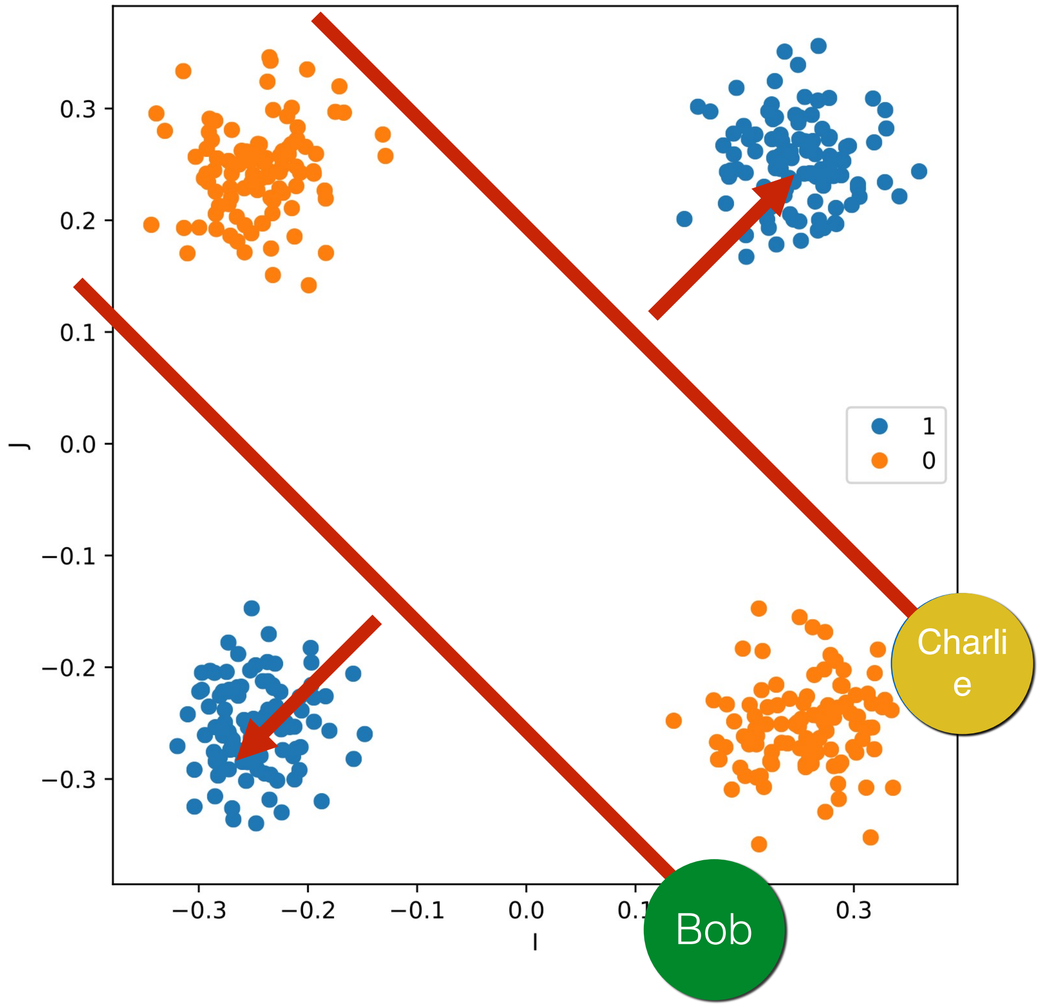
Redundancy

XOR textbook sample



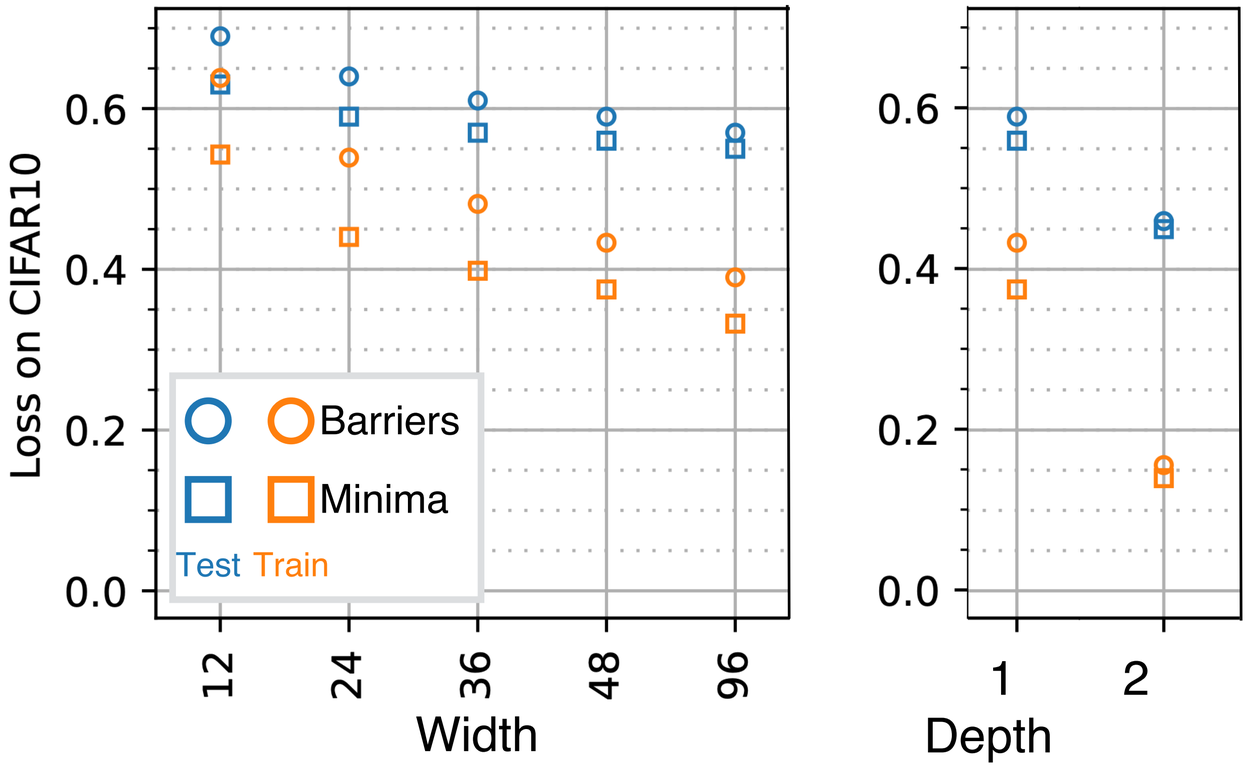
Redundancy

XOR textbook sample

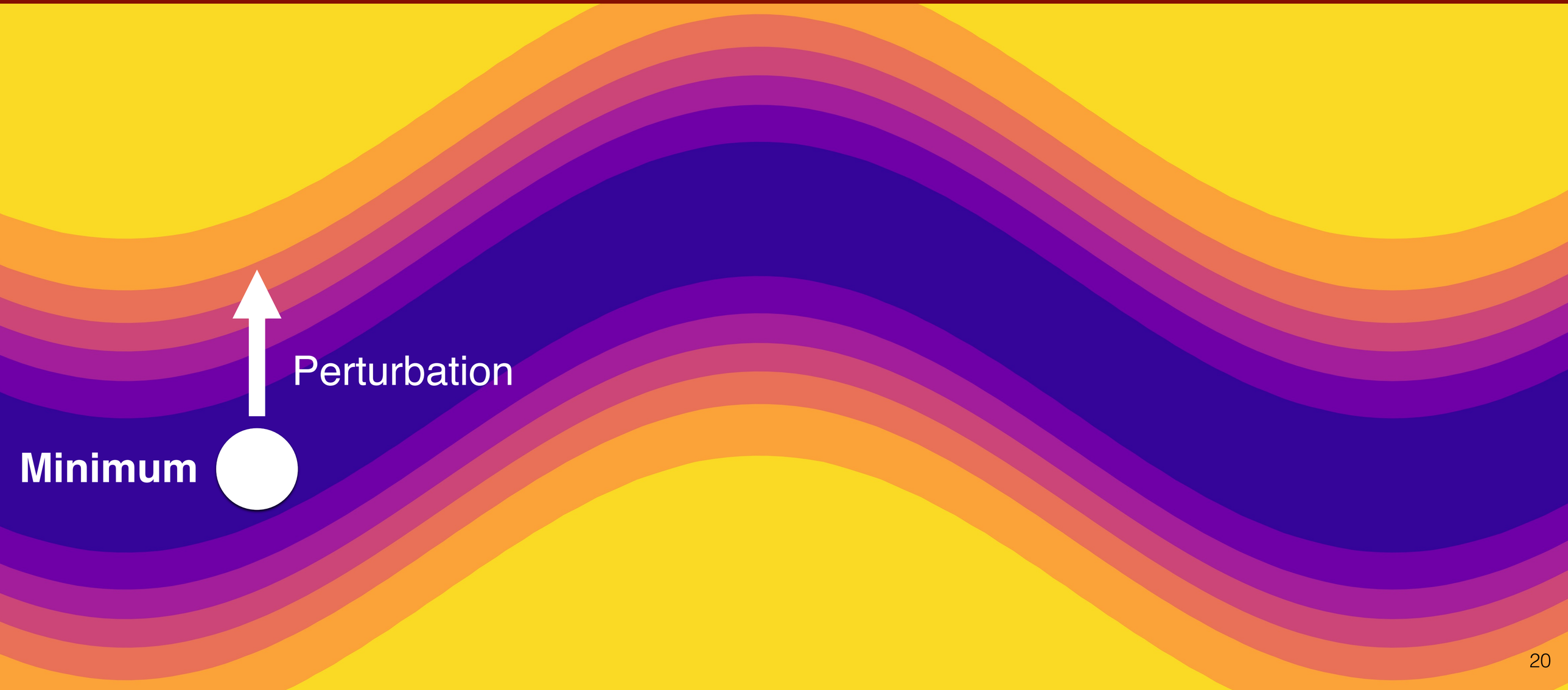


Width and Depth

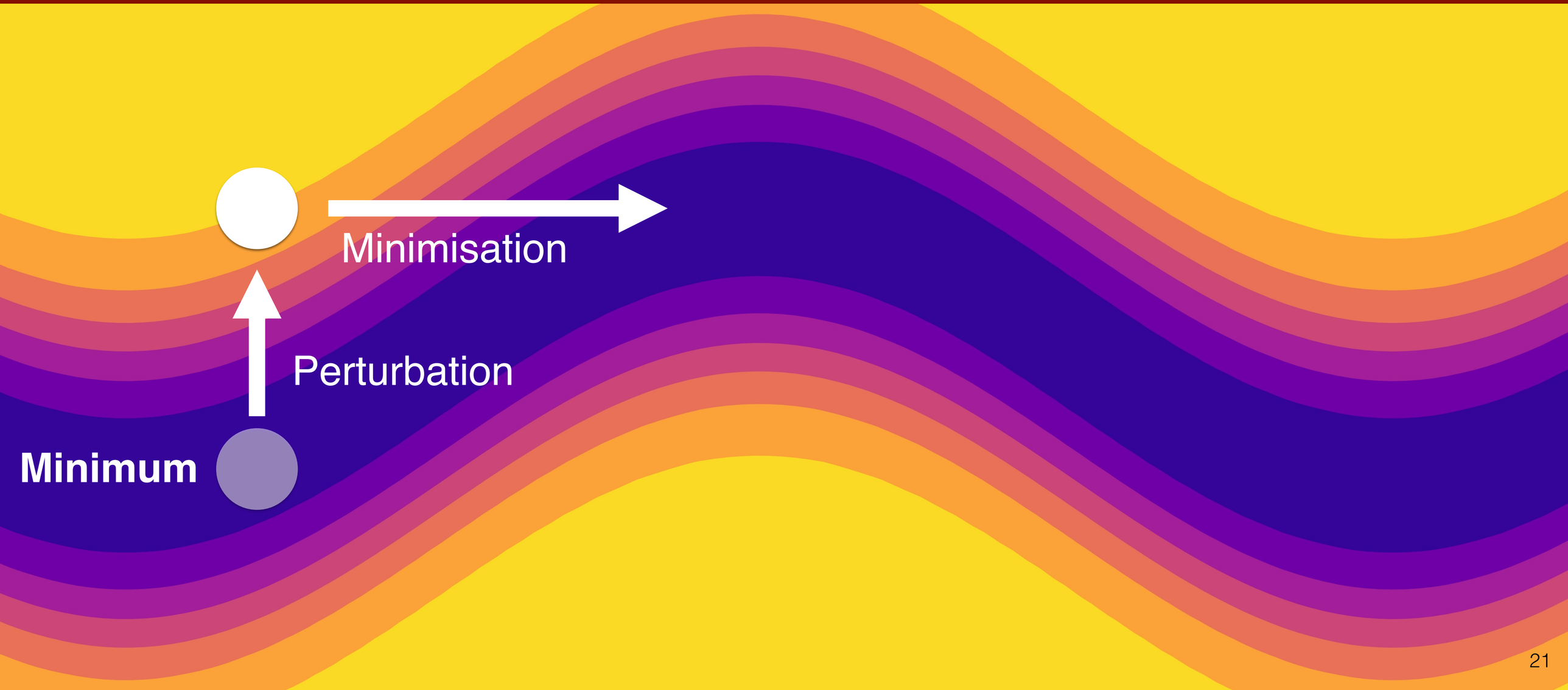
- Parametric architecture: **Width** and **Depth**
- **Wider** and **deeper**: Lower barriers



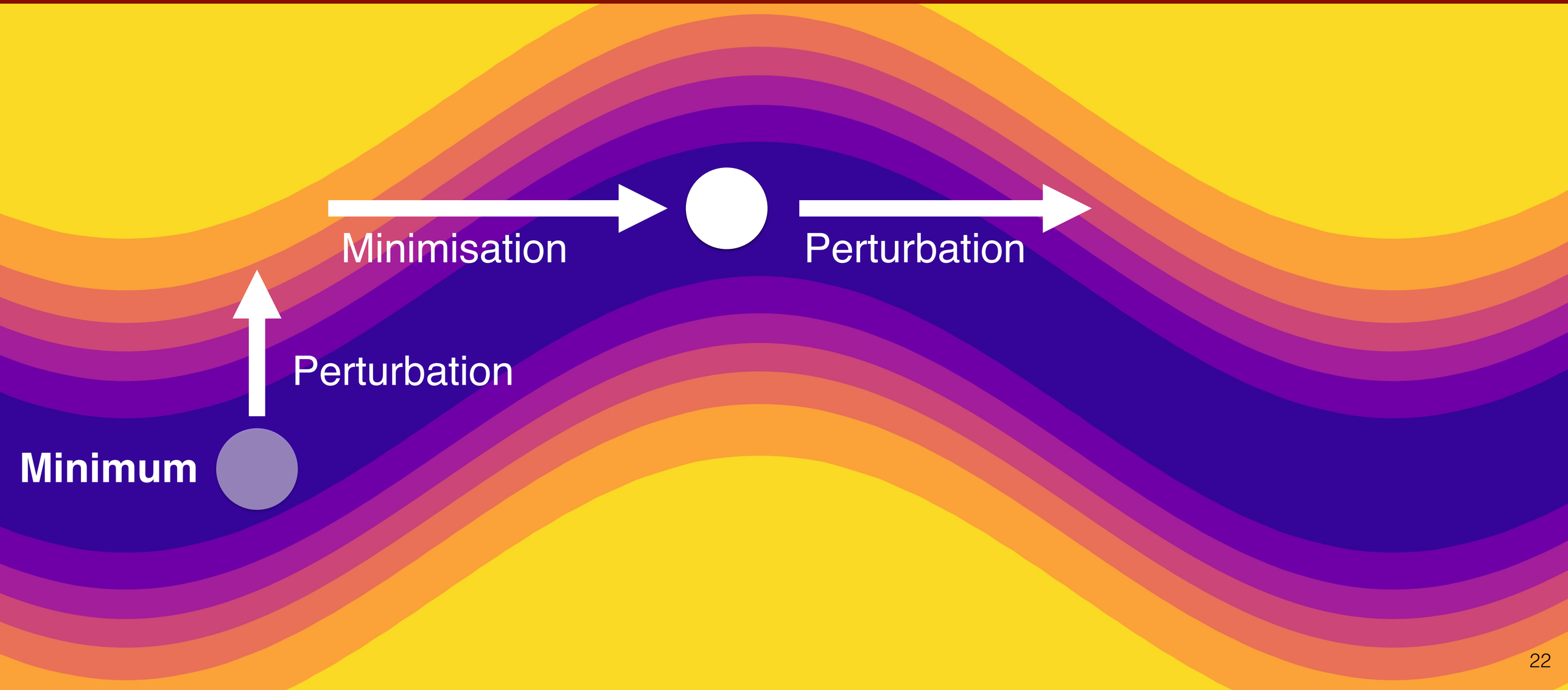
Resilience



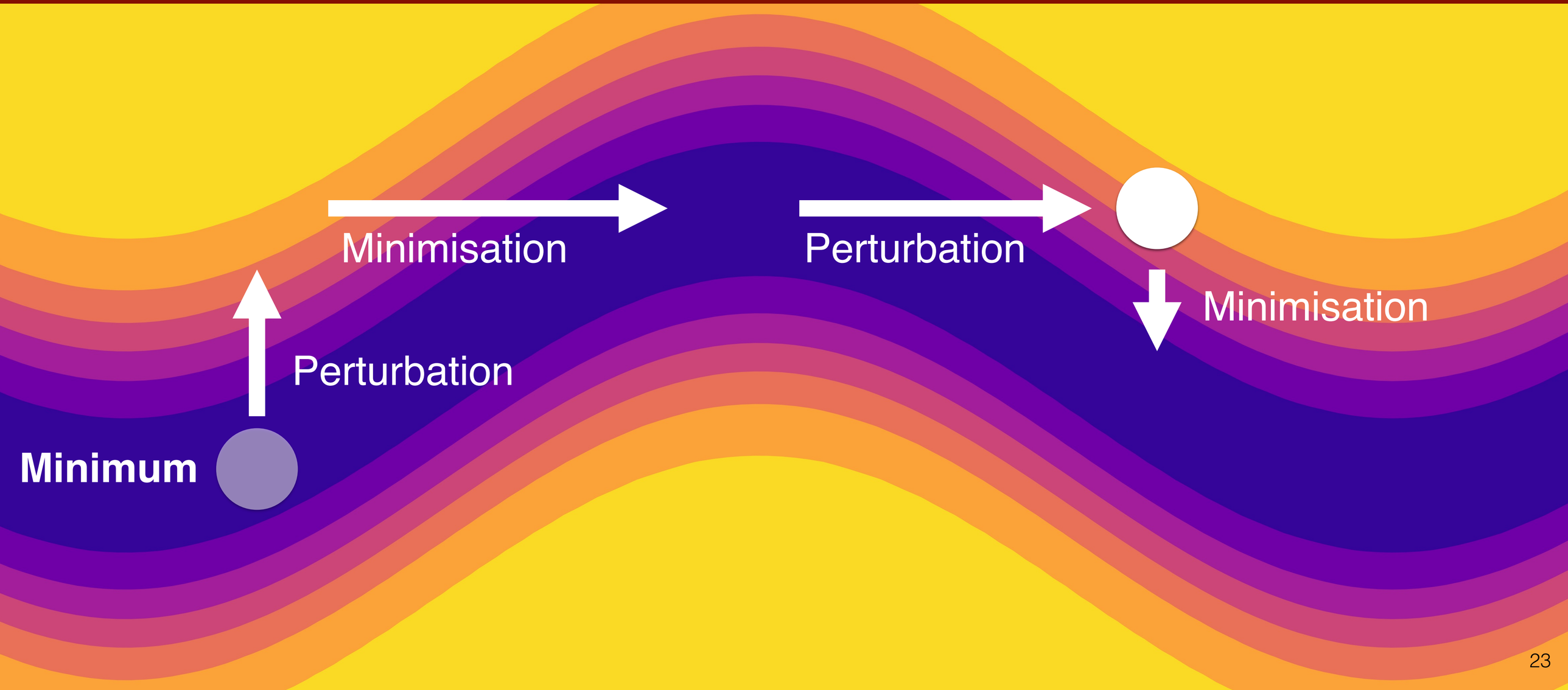
Resilience



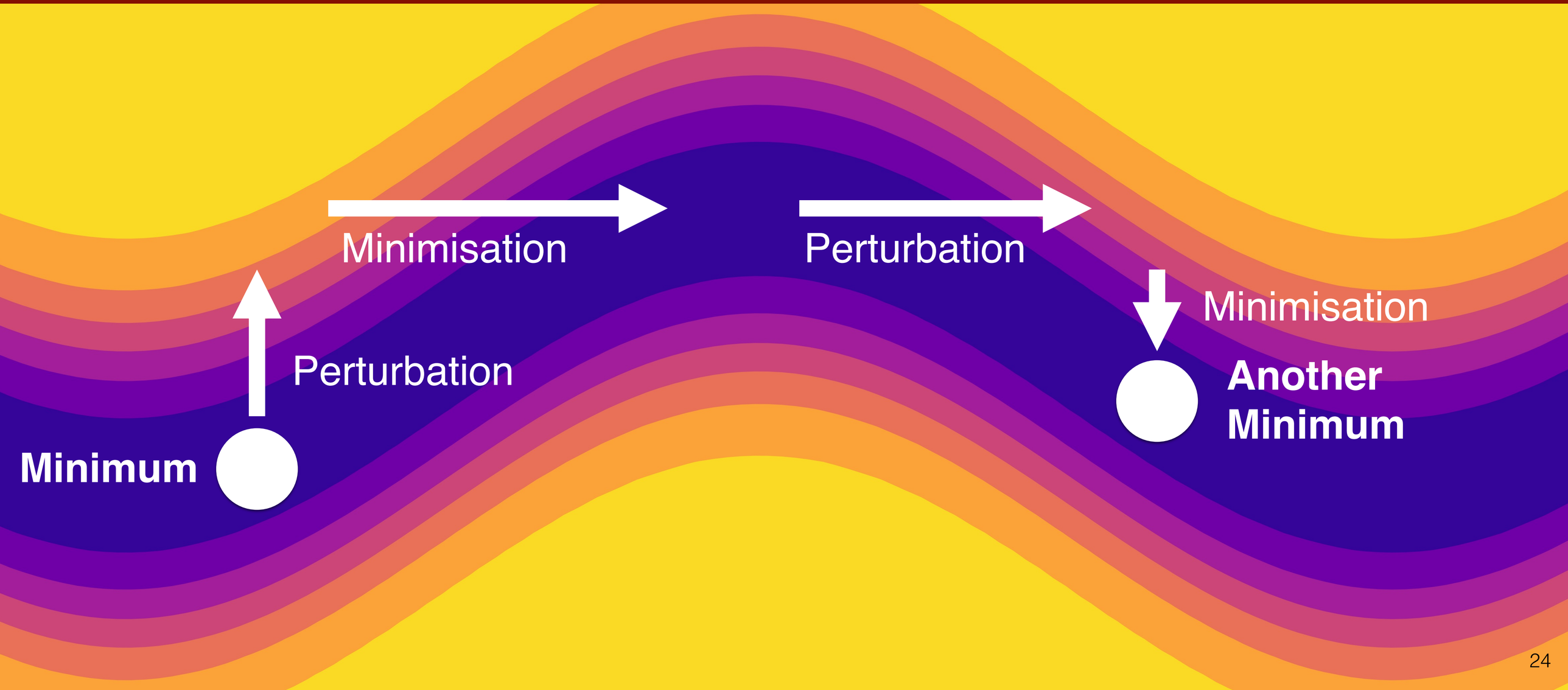
Resilience



Resilience


Minimisation
Perturbation
Minimisation
Perturbation
Minimum

Resilience


Minimisation
Perturbation
Perturbation
Minimisation
Another
Minimum
Minimum

Essentially No Barriers in Neural Network Energy Landcape

Contributions:

- **AutoNEB**: New method for landscape analysis
- Loss surface of deep NNs: **Very low barriers**
- Minima form **one single connected component** of low loss

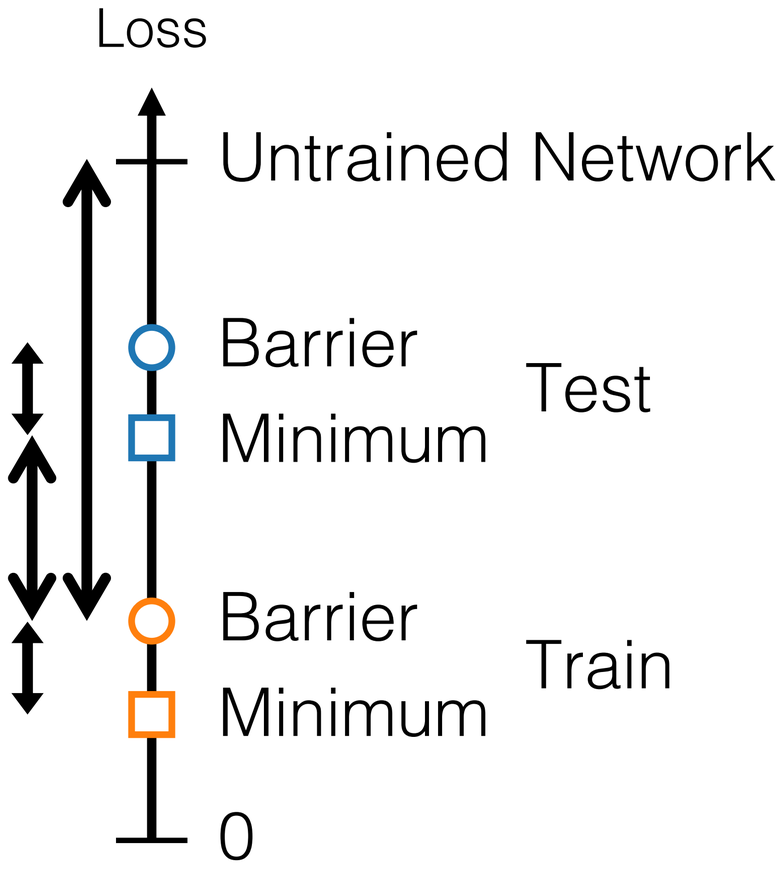
Outlook:

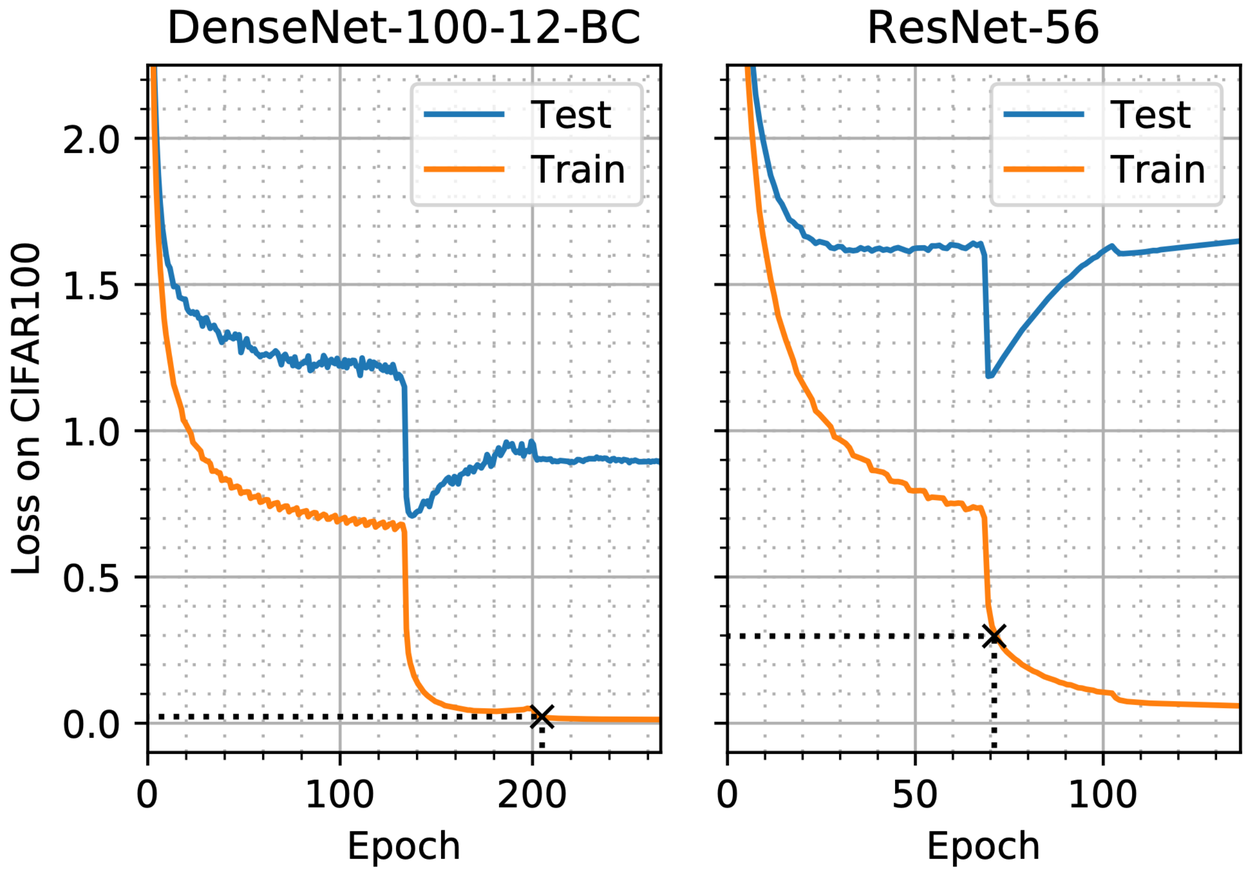
→ **Characterise loss landscape** further (e.g. Wales et al., ’98)

→ Connected minima inspired **Fast Geometric Ensembling** (Garipov et al., ’18)

Code available at:
github.com/fdraxler/PyTorch-AutoNEB

Connectivity Measures





Spurious High Barriers