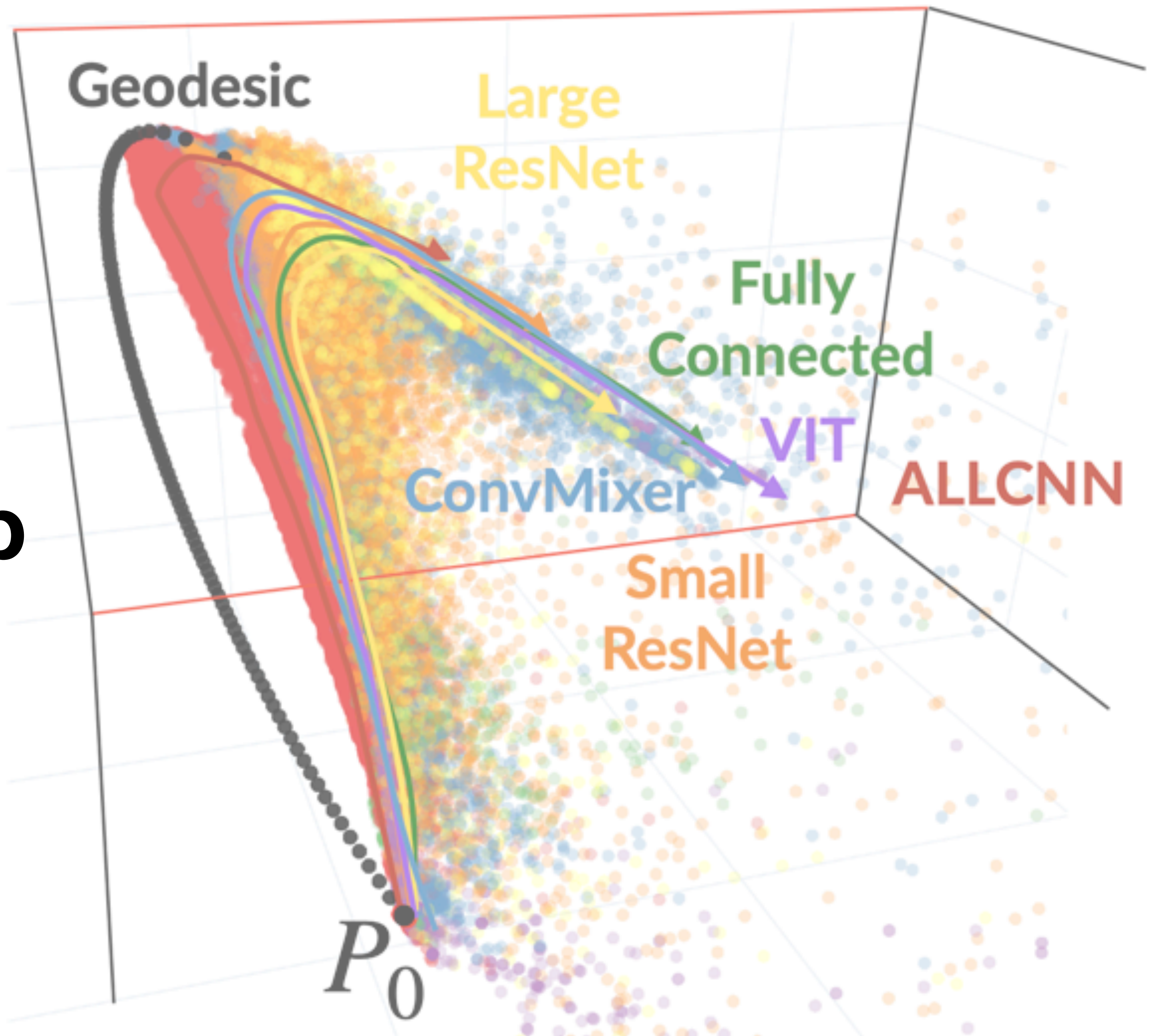
InPCA as a lense

The Training Process of Many Deep Networks Explores the Same Low-Dimensional Manifold

Itay Griniasty
griniasty@cornell.edu



- Mao, Griniasty, Yang, Teoh, Transtrum, Sethna & Chaudhari. *arXiv:2305.01604*.

DNNs the canonical synthetic emergent machine

Me: Hi, please create an image representing the complex functionalities large language models can perform.

ChatGPT4: Here is the image representing the complex functionalities of large language models. It depicts a futuristic and intricate machine symbolizing a large language model, surrounded by diverse groups of people engaging with it. The scene captures the advanced technology and intelligence of these models.

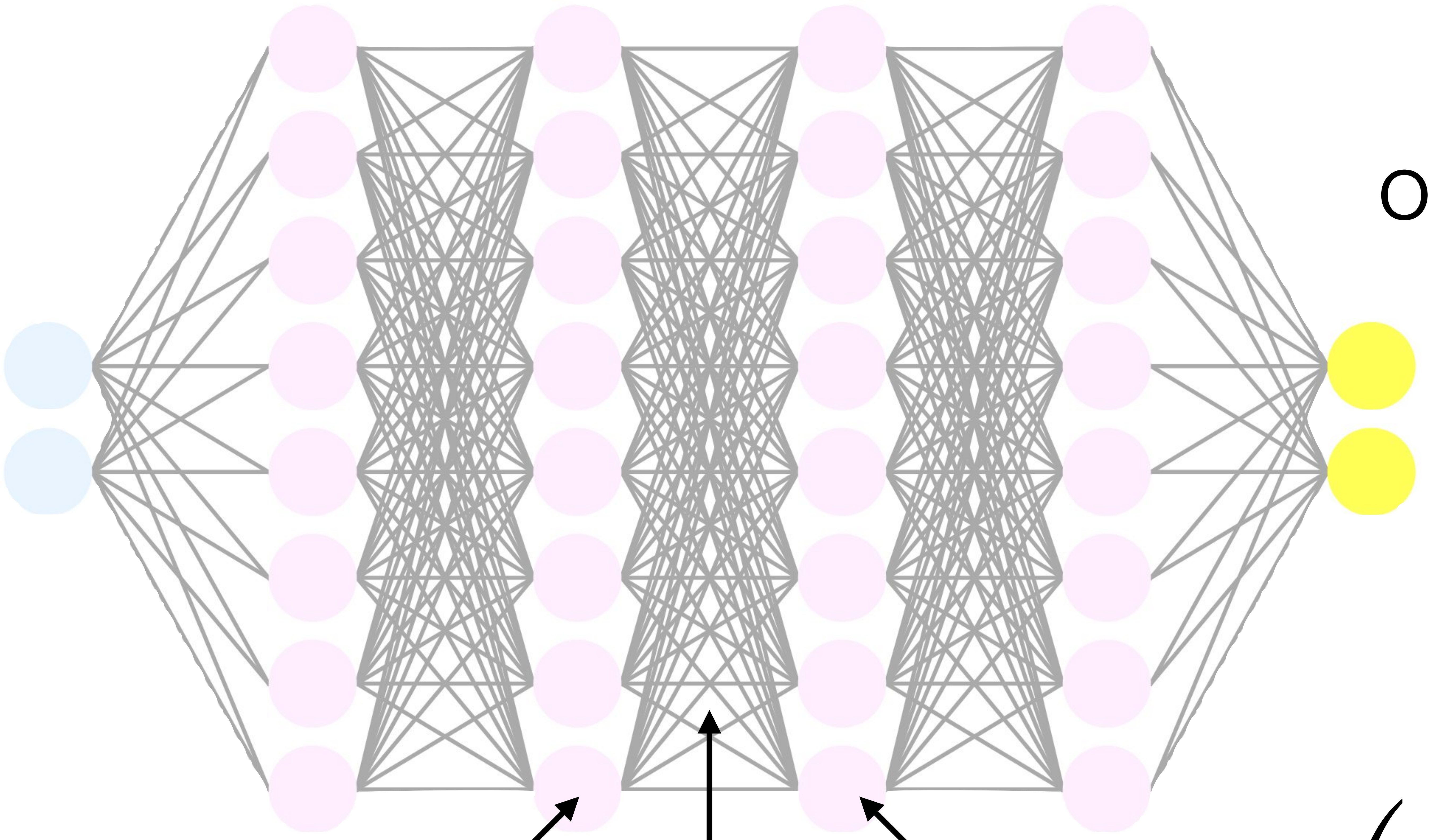
MANY parameters

MANY outputs

A Sketch of a Neural Network

Hidden Layers

Input



Output - Class

Dog
Car
Frog

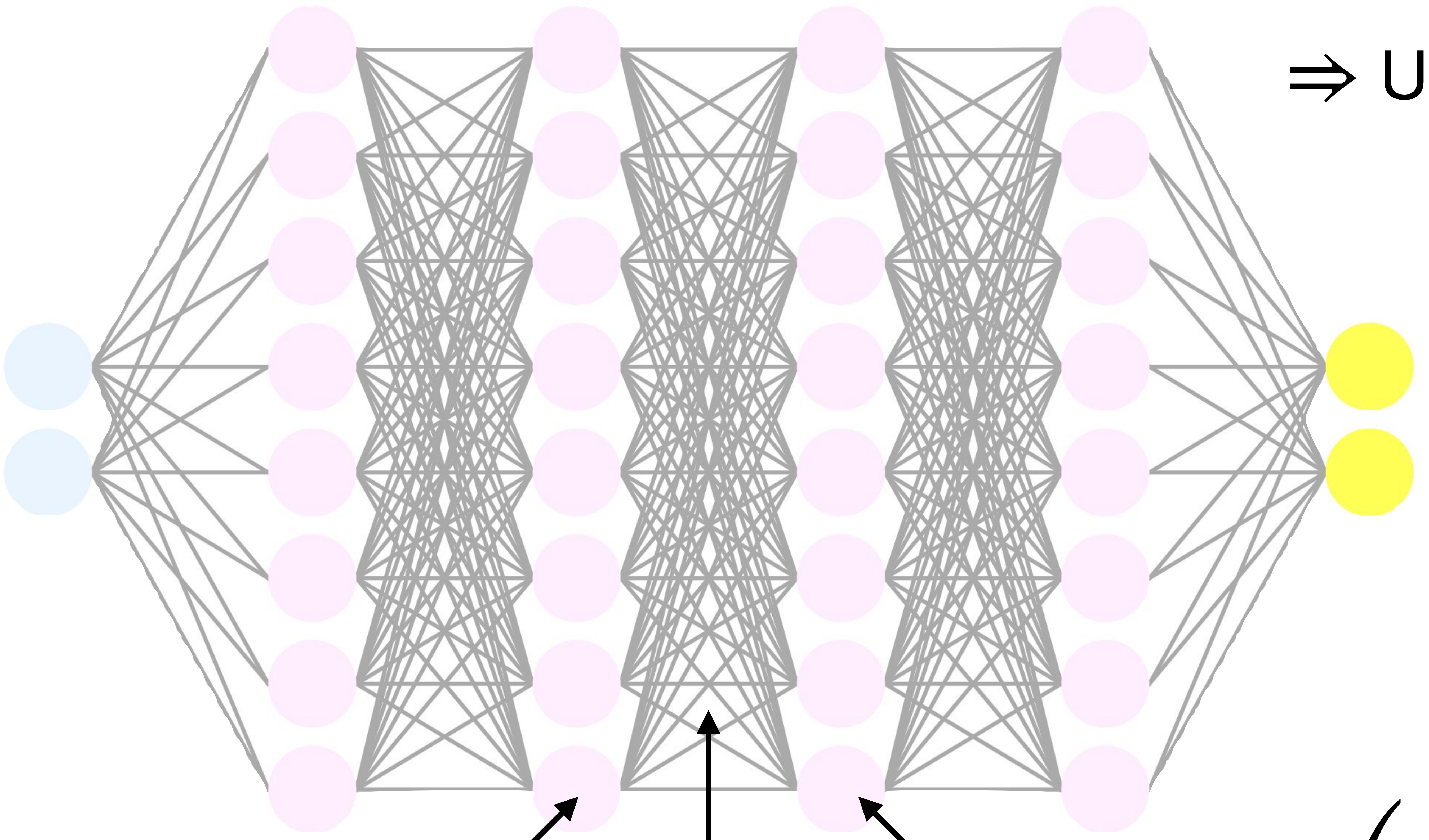
Node a_i Weights w_{ij} Node $a_j = f\left(\sum_i w_{ij} a_i\right)$

A Sketch of Training

Minimize **Loss** = Difference between Prediction and Truth

$\Rightarrow$ Update Weights

Input



Why Can We Train DNNs?

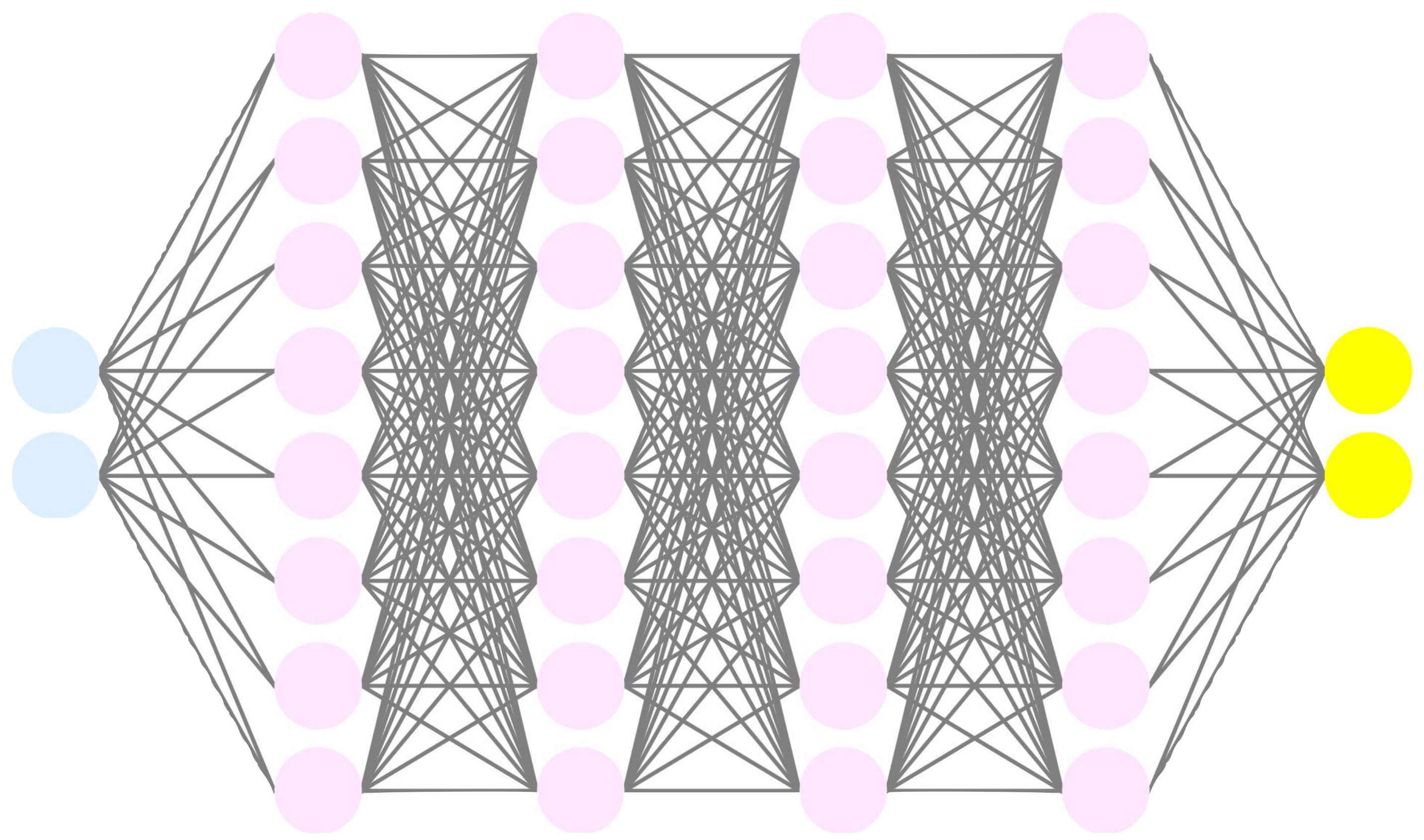
How can we train a 10^{11} dimensional machine in a year?

This is a high-dimensional $(10^6 - 10^{12}$ weights), large scale (millions of images) and non-convex optimization problem

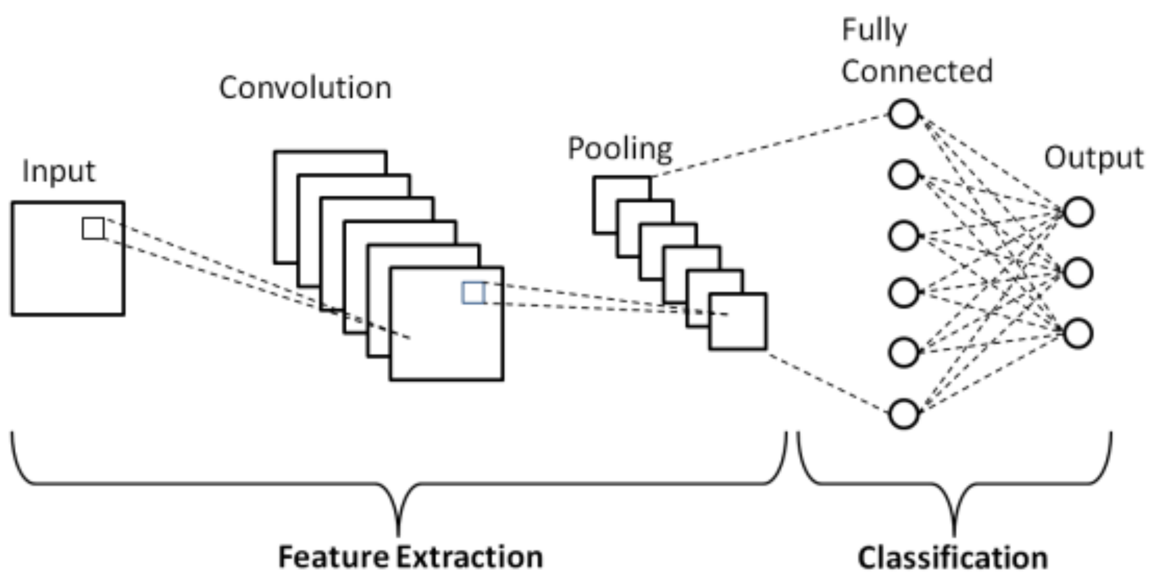
How Much Do The Architecture / Algorithm / Augmentation Matter?

Loss Landscape
Li et al. NeurIPS 2018

Many DNN Designs, What Is The Difference?

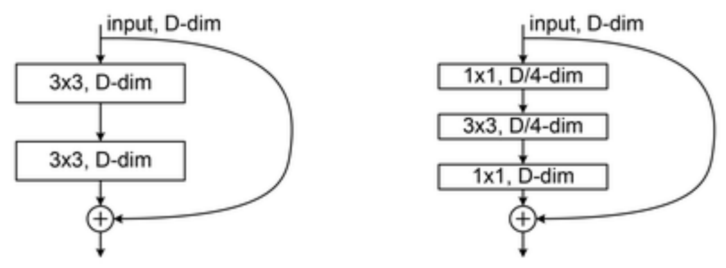


Fully Connected

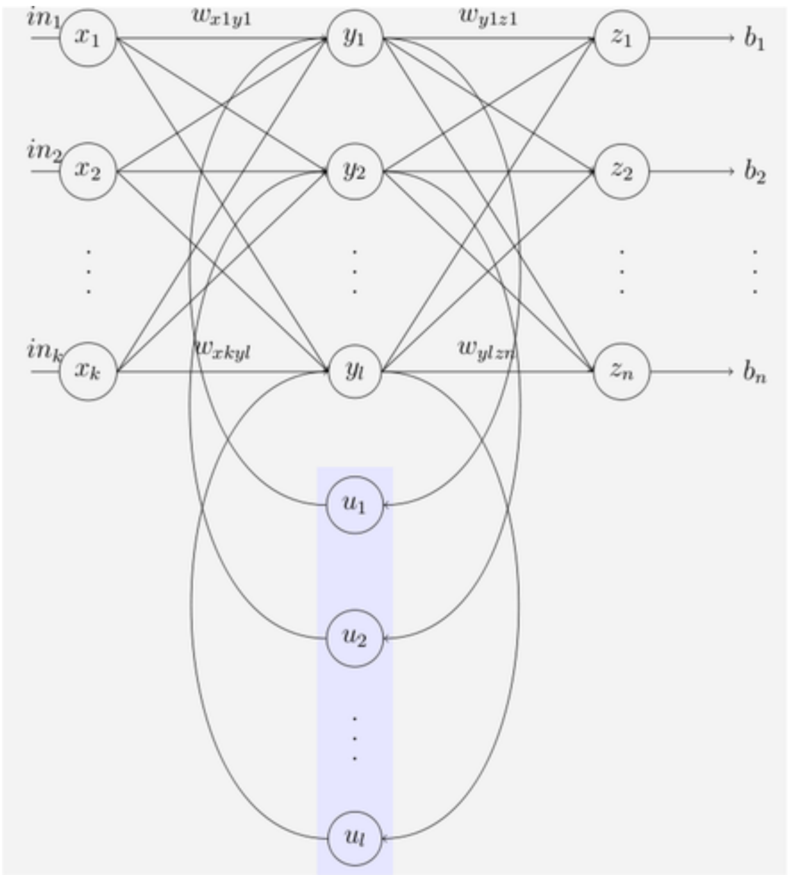


Convolutional Neural Network,

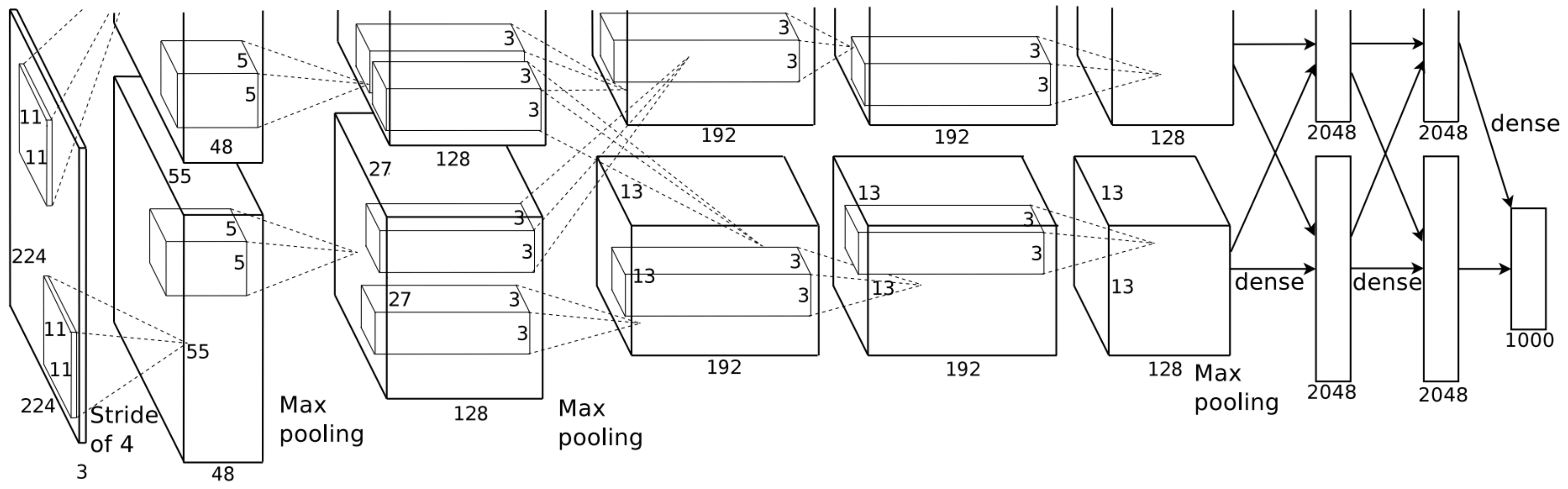
image: Phung and Rhee App. Sci. 2019



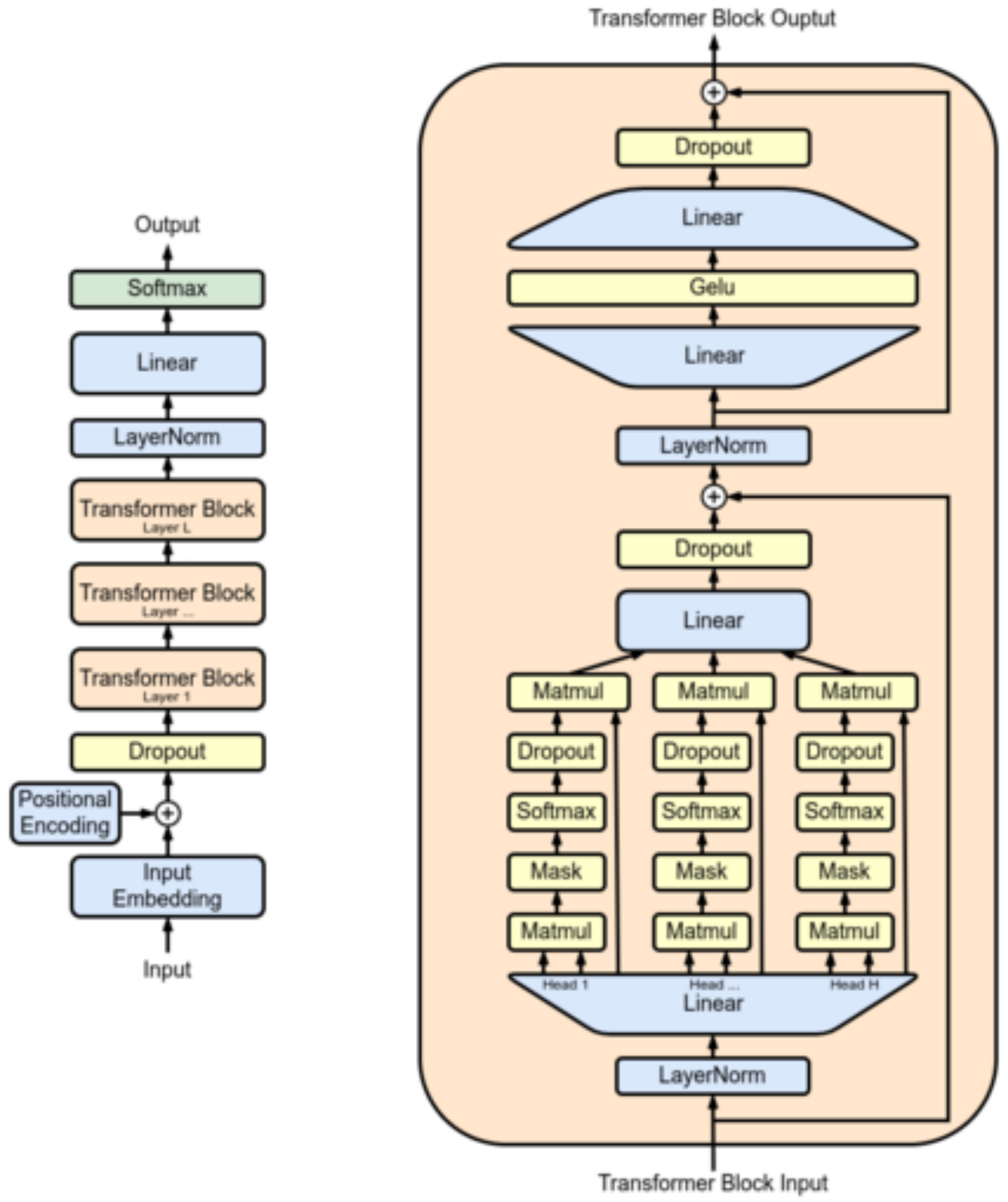
Residual Network



Recurrent Network

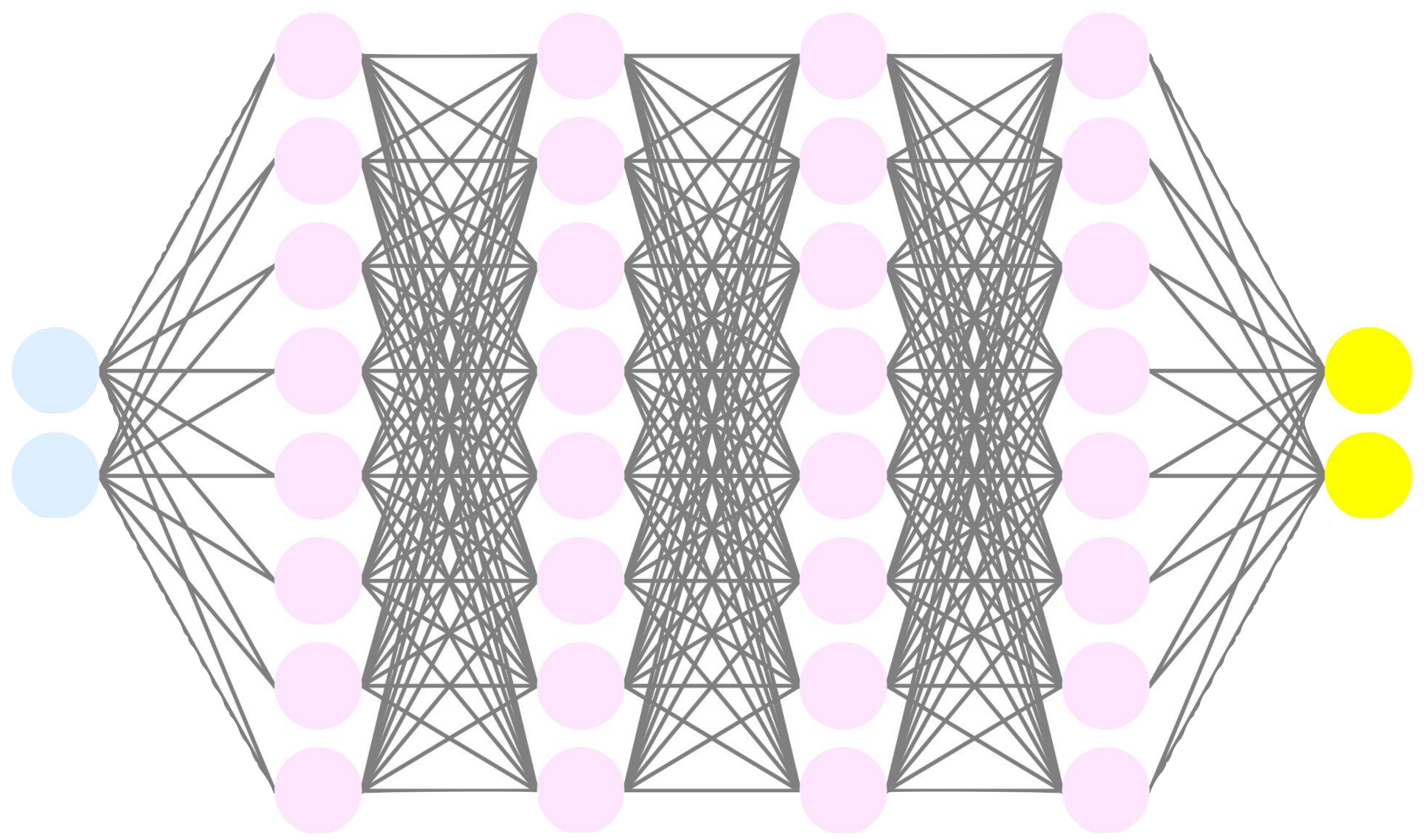


AlexNet

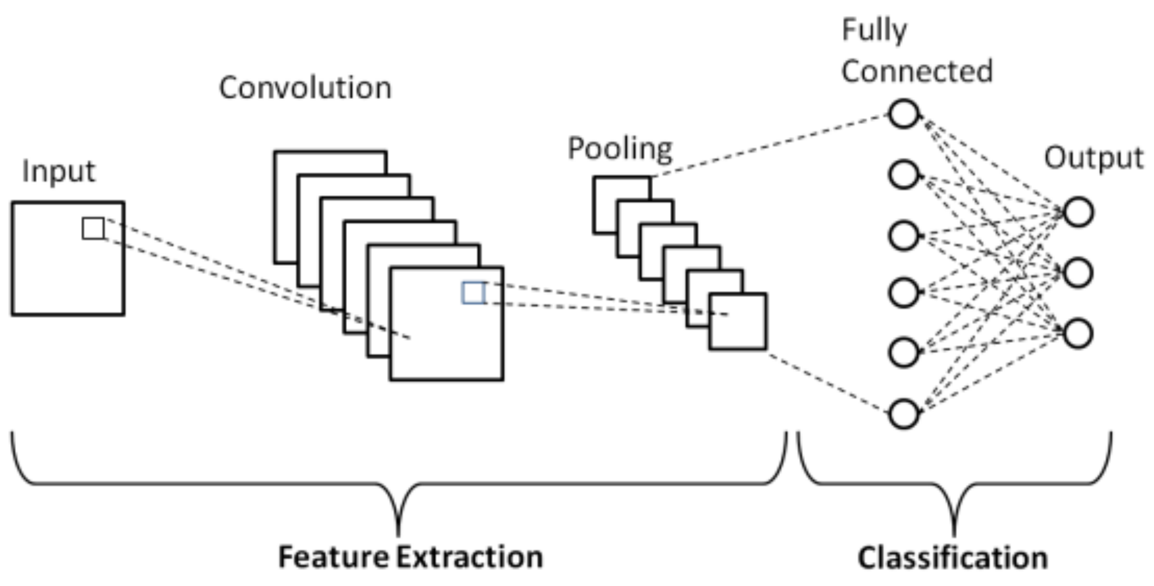


Transformer Network

Many DNN Designs, What Is The Difference?

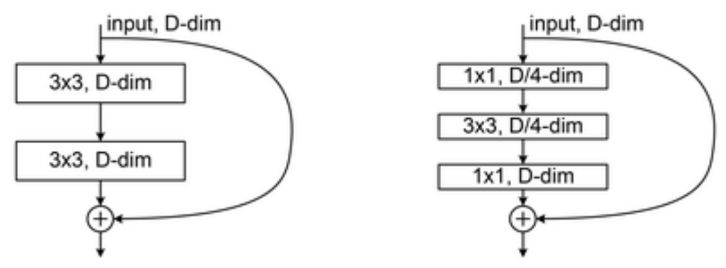


Fully Connected



Convolutional Neural Network,

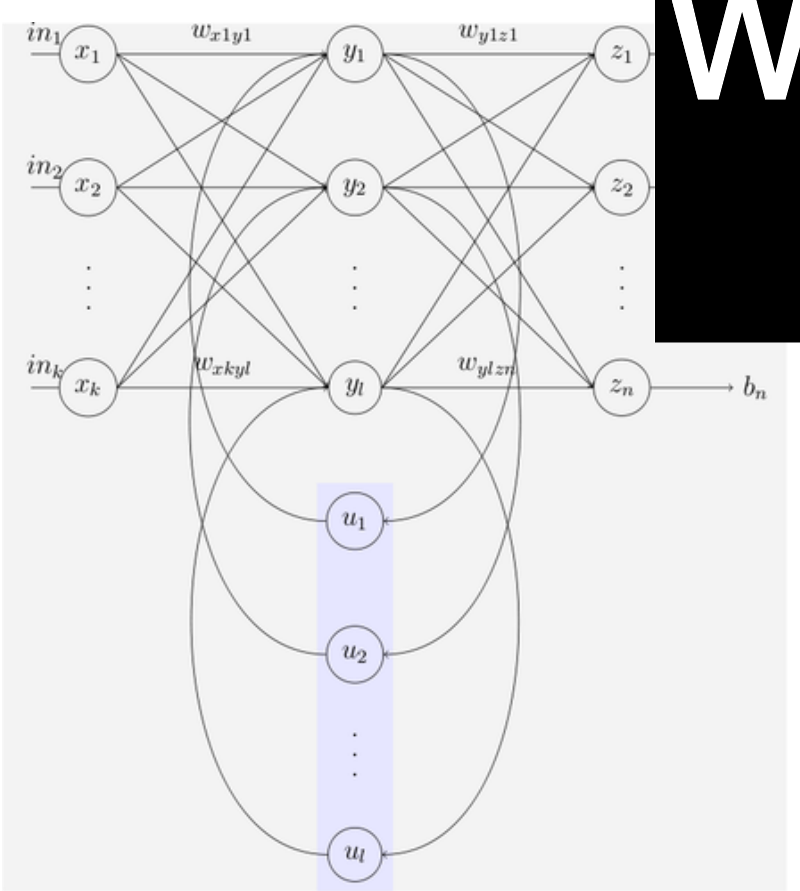
image: Phung and Rhee App. Sci. 2019



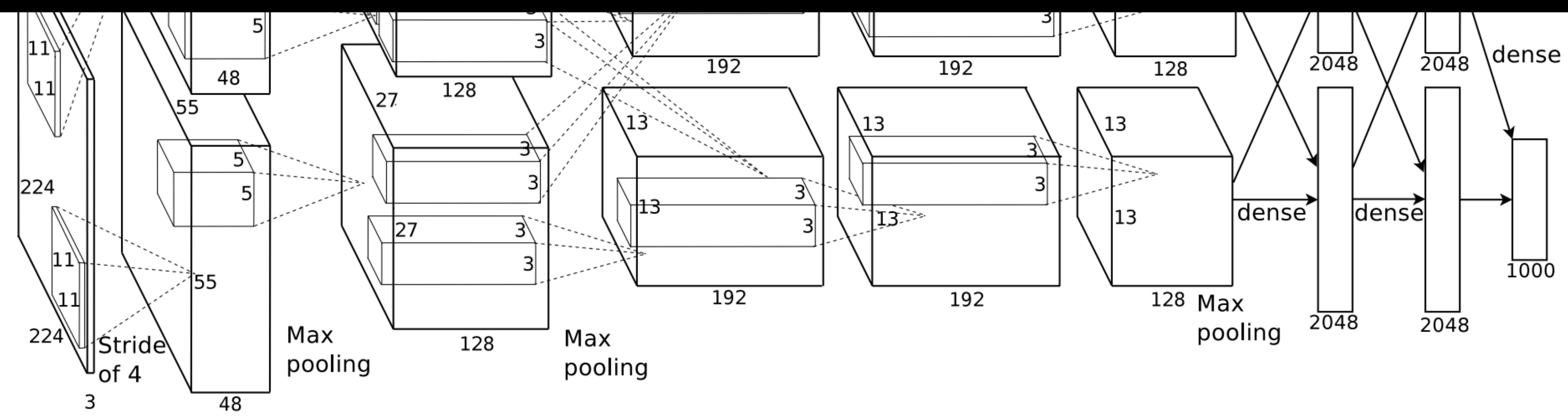
Residual Network

Weight Dynamics are Idiosyncratic

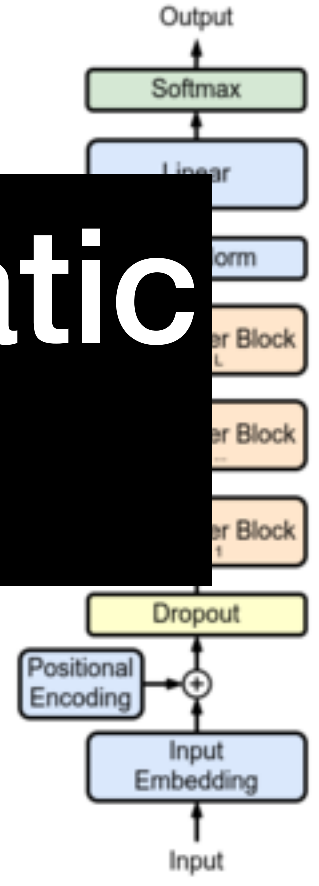
The Output is Common

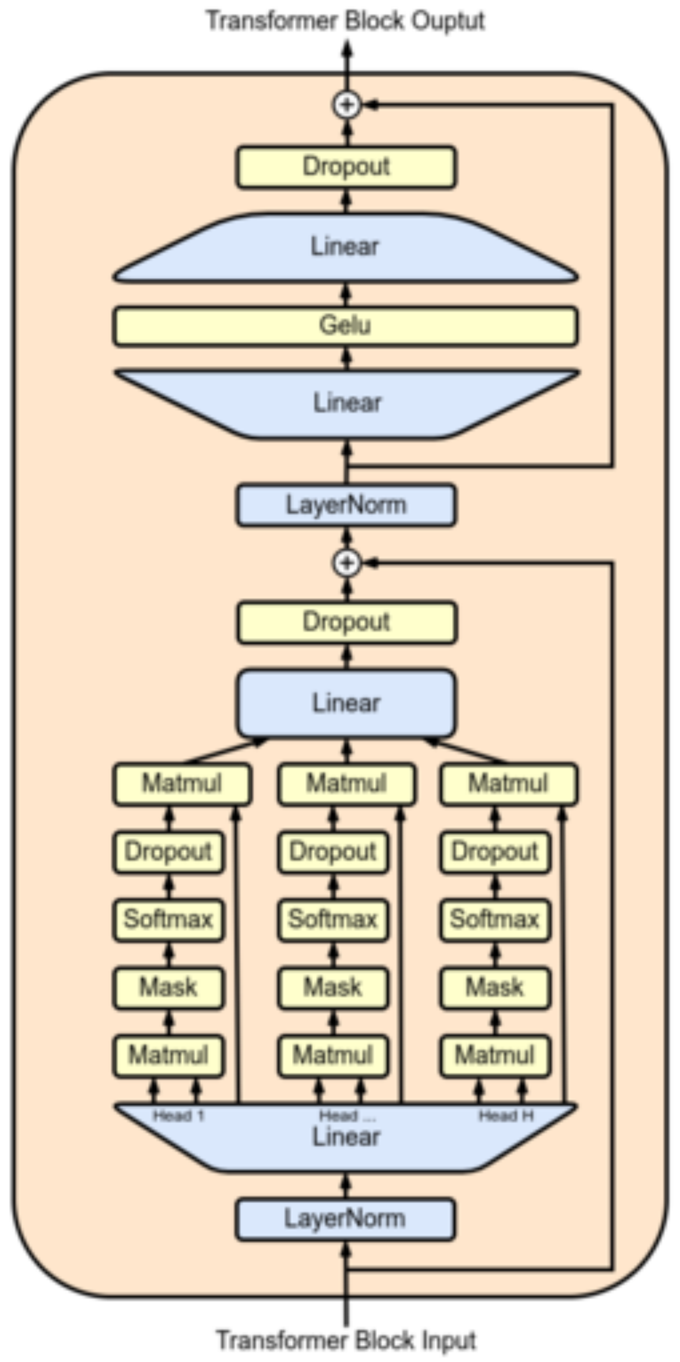


Recurrent Network



AlexNet



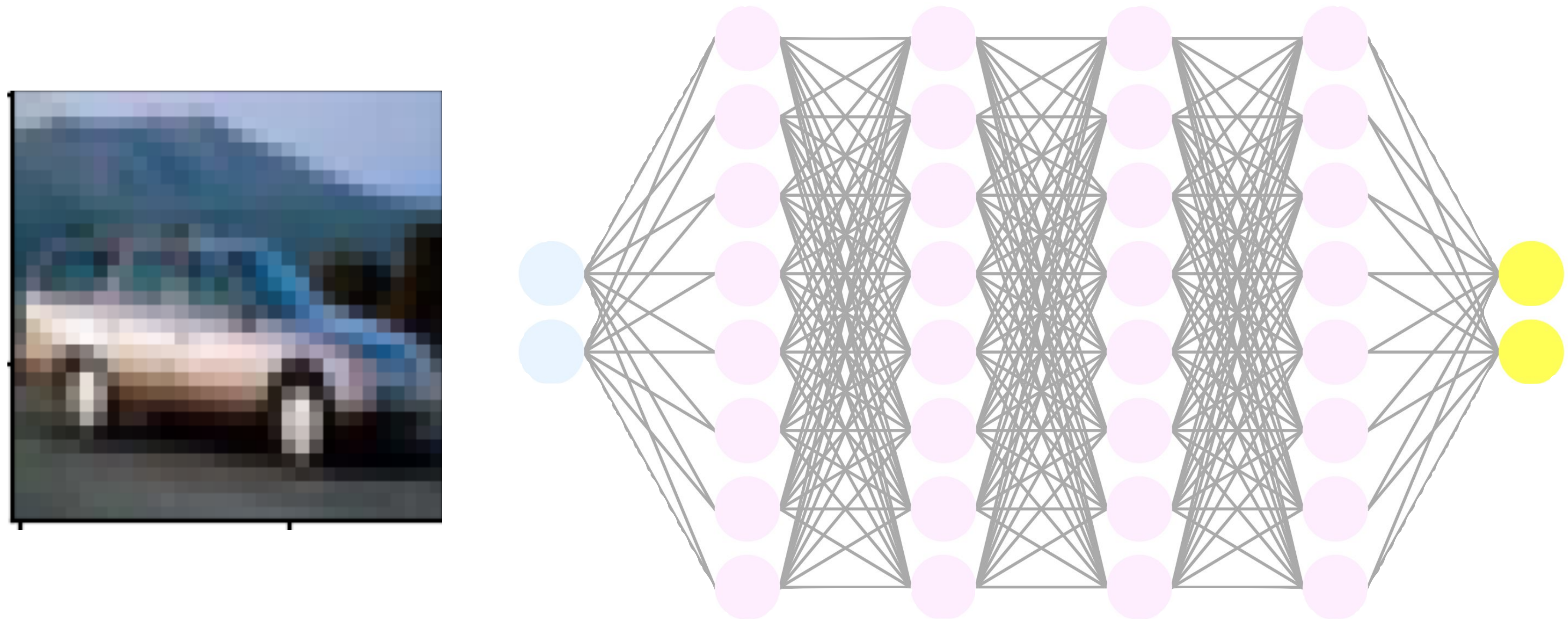


Transformer Network

How Do We Visualize the Output?

Network Represented over N Samples: $P_w(\vec{y}) = \prod_{n=1}^{N} p_w^n(y_n \,|\, x_n)$

Sample x Network with Weights w Class y



Dog

CIFAR-10: $5 \cdot 10^4$ Images, 10 classes $\sim 5 \cdot 10^5$ Dimensions

ImageNet: 10^6 Images, 1,000 Classes $\sim 10^9$ Dimensions

High Dimensional Output is Cursed

Even if $p_1 \cdot p_2 = 1 - \epsilon$

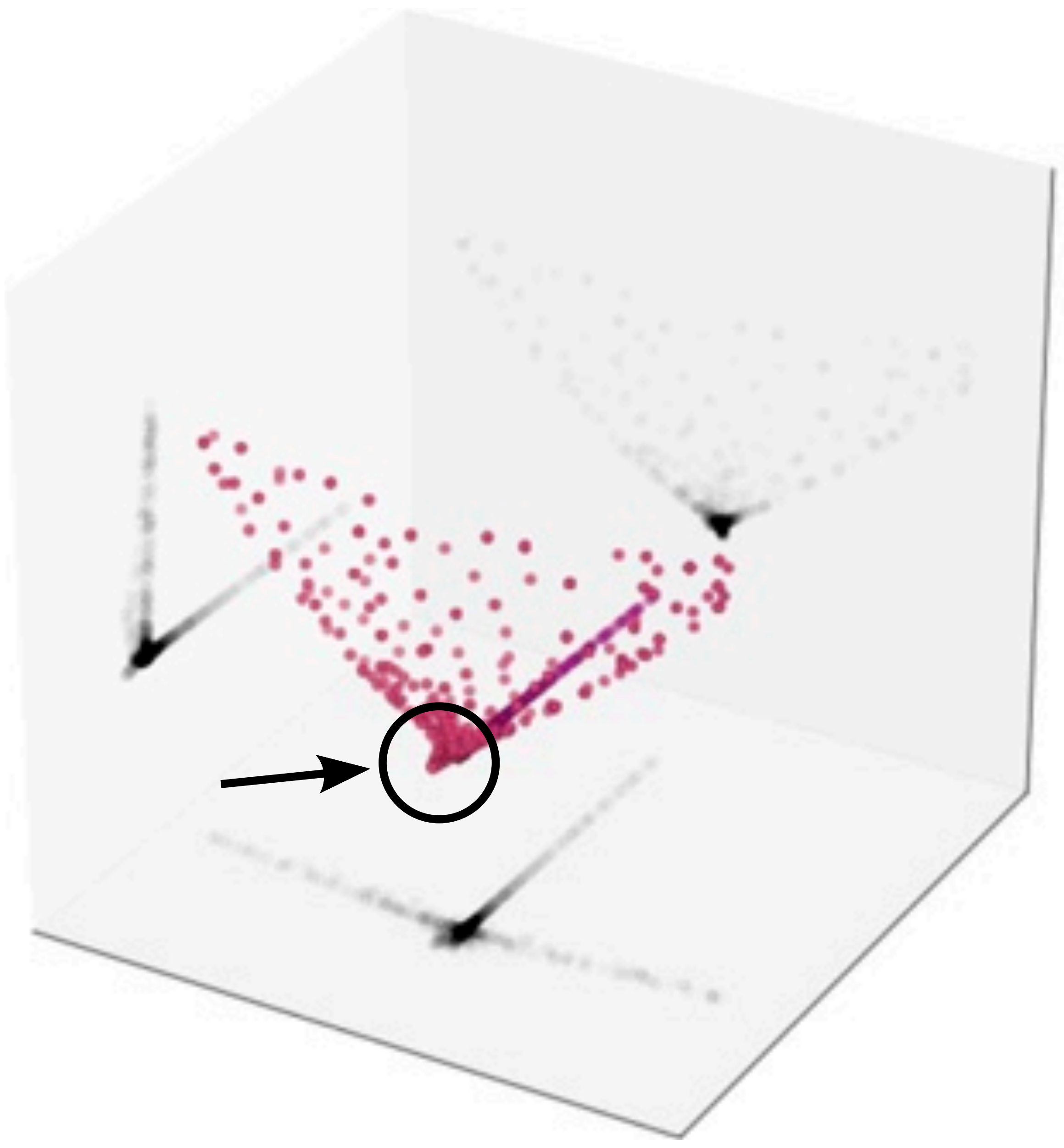
For many samples $P = \prod_{n=1}^{N} p^n$

all probabilities are orthogonal

$$P_1 \cdot P_2 = \prod_n p_1^n \cdot p_2^n \approx 0$$

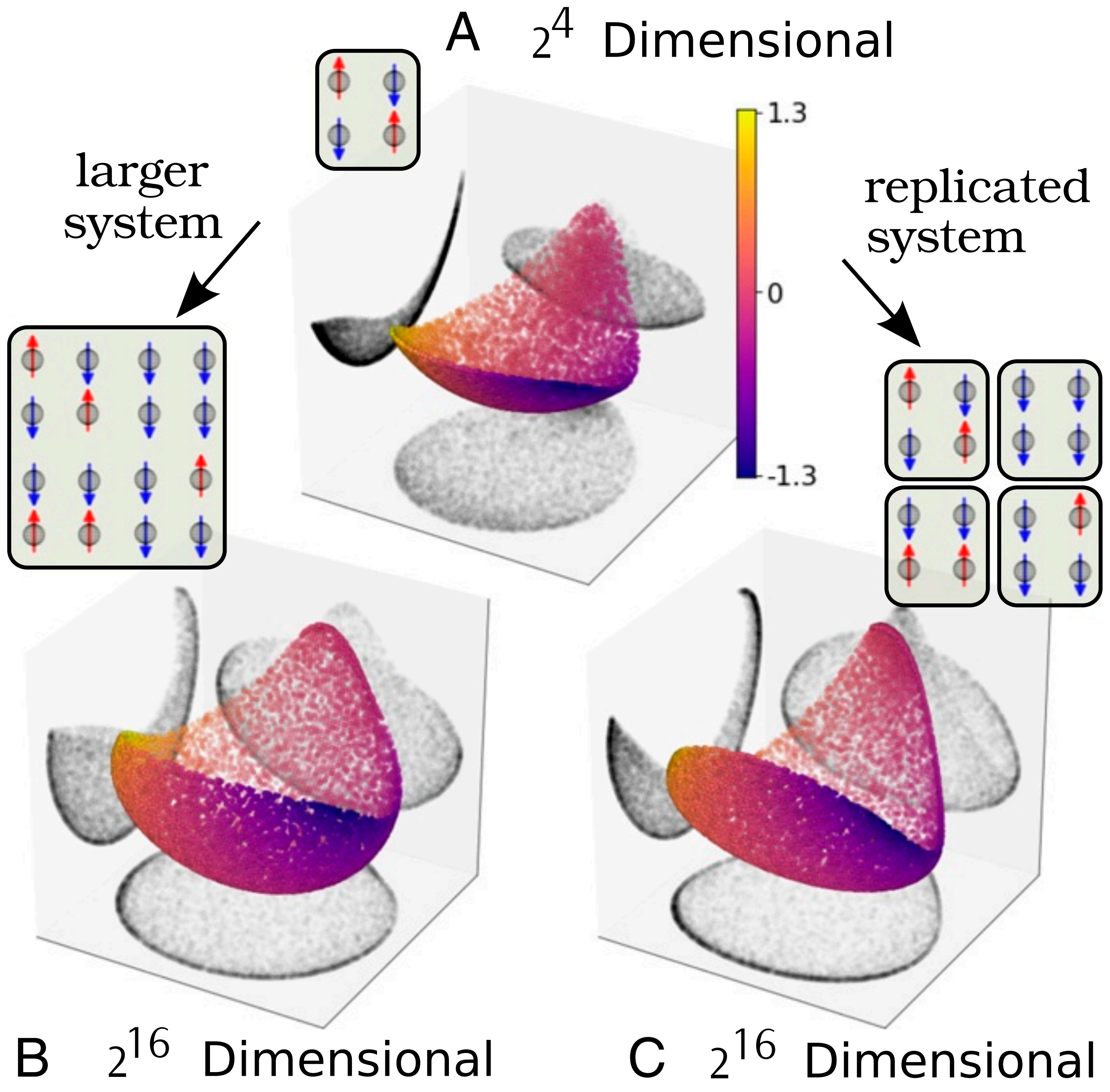
Natural metric can't discern close from far

$$d_H^2(P_1, P_2) = 0.5\|P_1 - P_2\|^2 = 1 - \prod_n p_1^n \cdot p_2^n \approx 1 - e^{-N\epsilon}$$



How Do We Visualize High Dimensional Data?

Intensive Embeddings

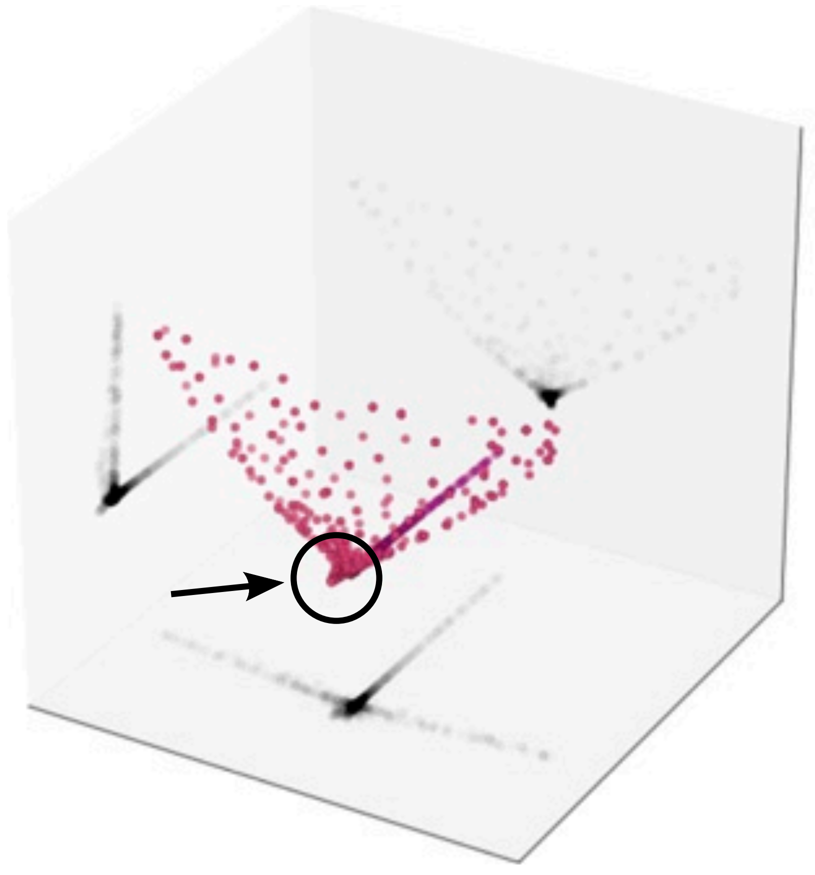


Quinn et al. PNAS 2019

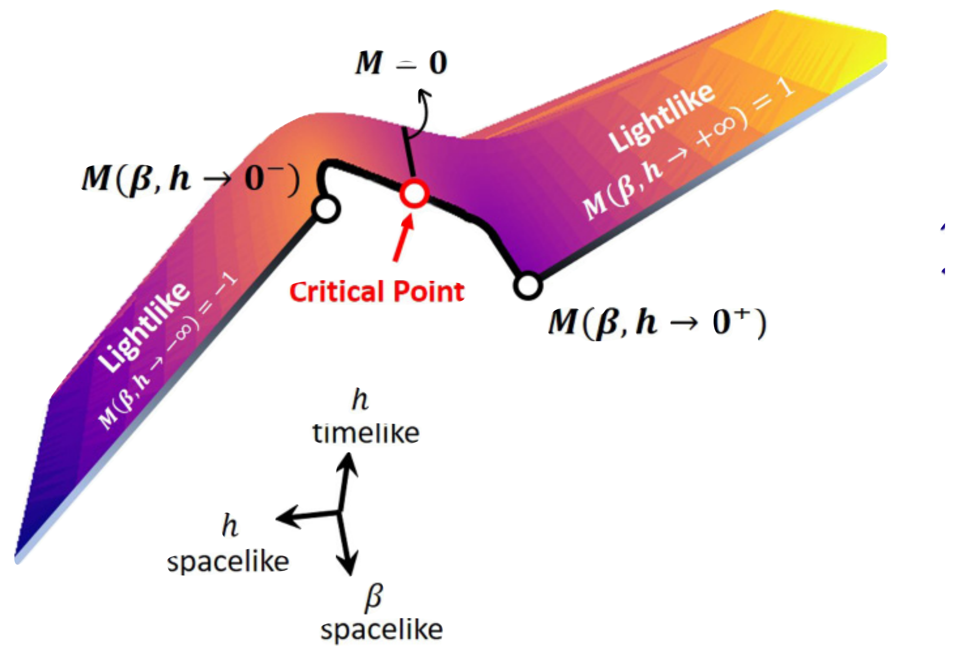
Zero Replica limit

$$\lim_{k\to 0} \frac{d_H(P_u^k, P_v^k)}{k} = -\log \sum_{\vec{y}} \prod_{n=1}^{N} \sqrt{p_u^n(y_n) p_v^n(y_n)}$$

$$= \ldots = \sum_{n}^{\mathbf{N}} d_B(p_u^n, p_v^n)$$



Natural Embedding

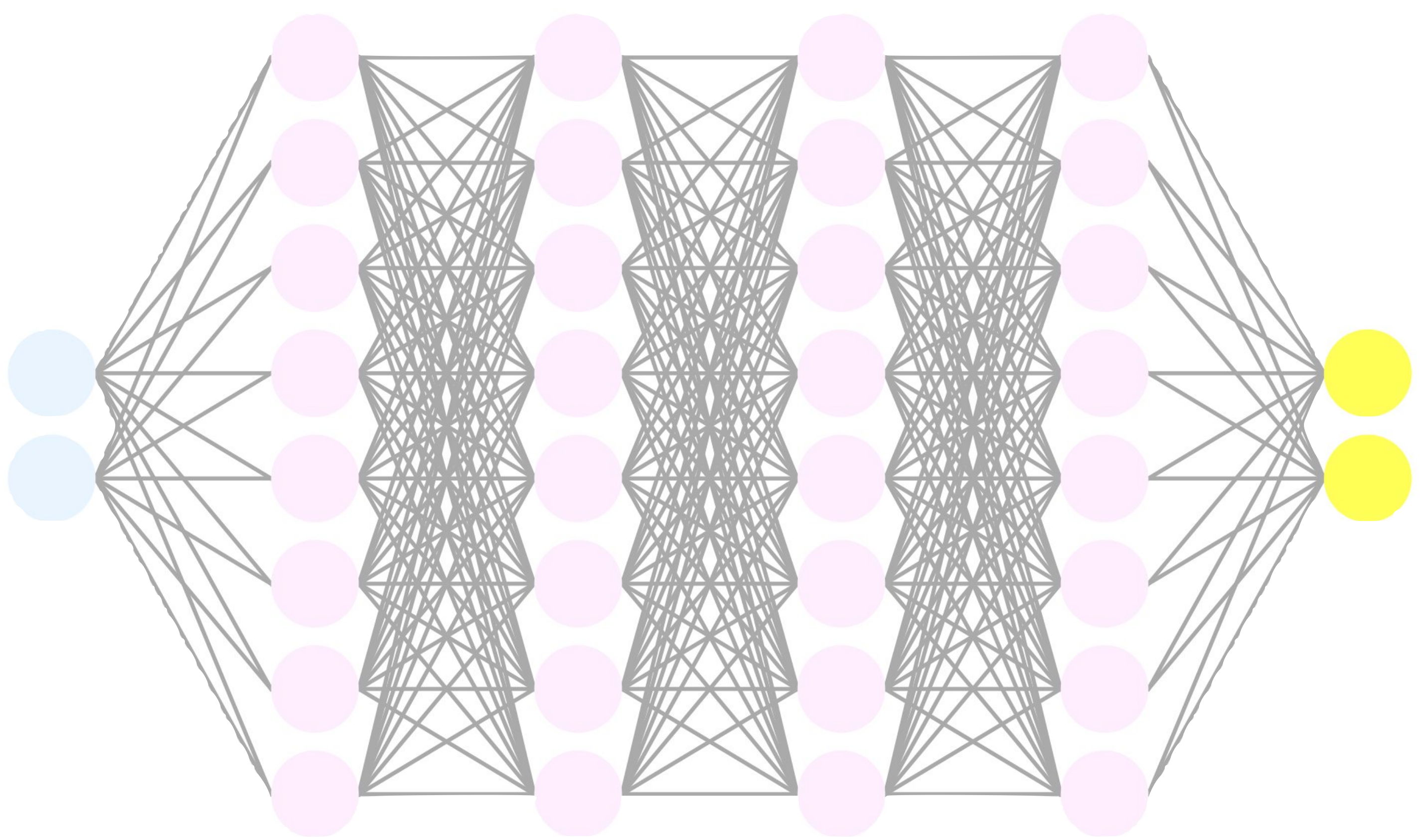


Intensive Embedding

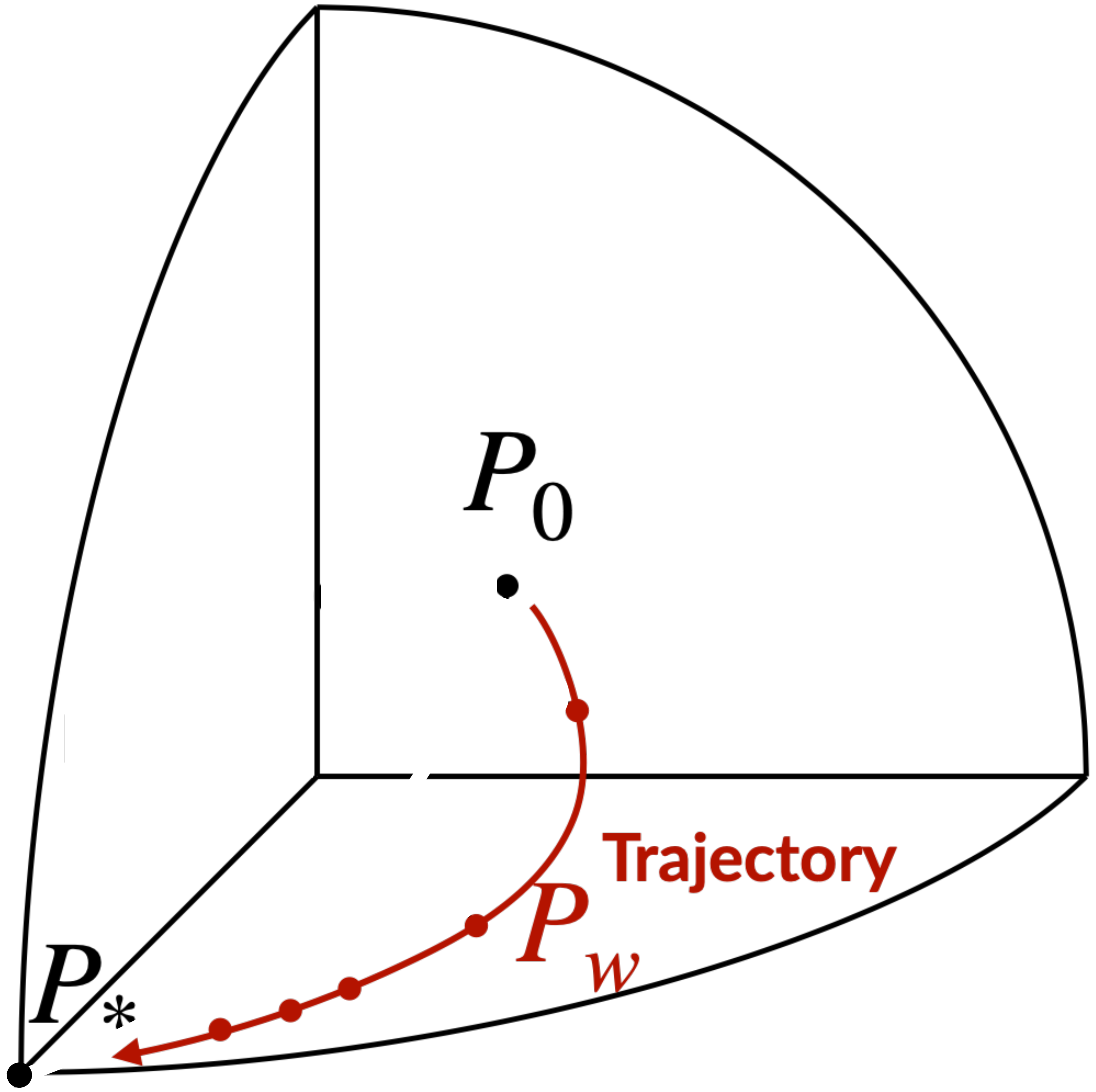
Computational Intensive Geometry

Network Represented over N Samples: $P_w(\vec{y}) = \prod_{n=1}^{N} p_w^n(y_n \,|\, x_n)$

Sample x Network with Weights w Class y



Dog



Ignorance: $p_0(y \,|\, x) = 1/C, \quad \forall y \in (1,\ldots,C)$

Truth: $P_* = \delta_{\vec{y}*}(\vec{y})$, $\vec{y}* =$ True label

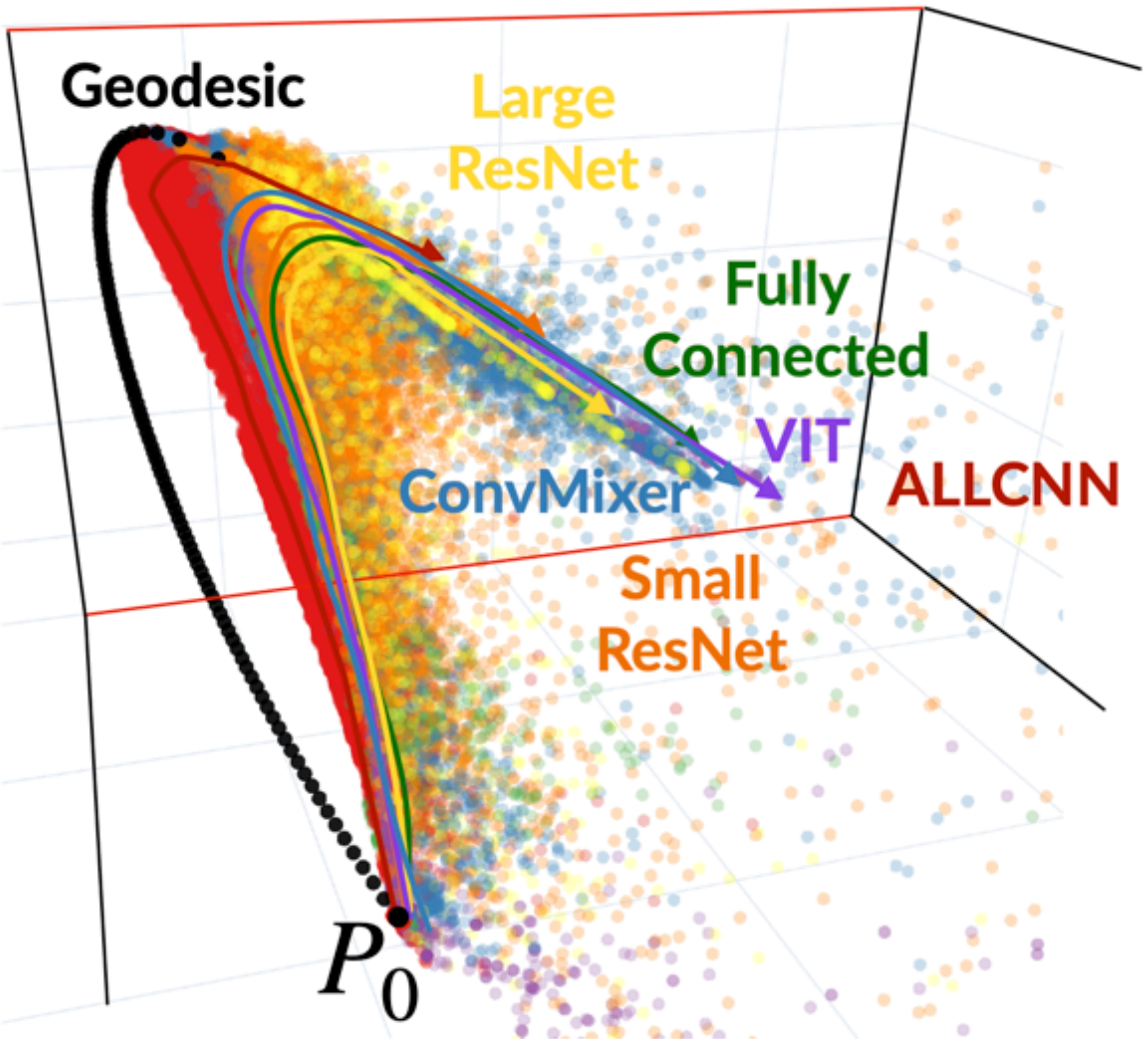
The Training Process explores a Low Dimensional Manifold

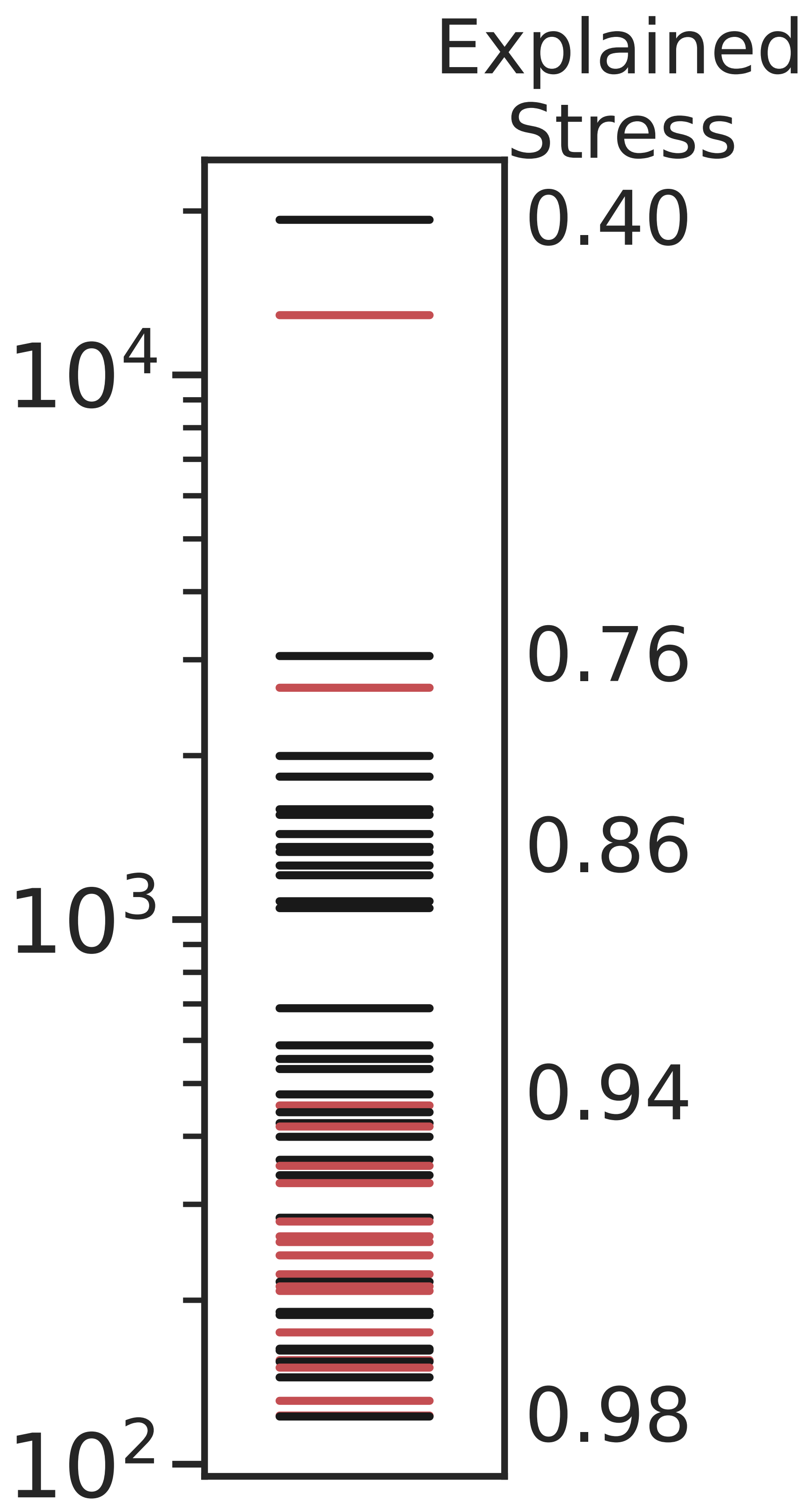
CIFAR-10 $\sim 10^6$ Dimensions

~2,000 Configurations

~150,000 Networks

- Architectures
- Optimization:
  - SGD, SGDN, ADAM
- Hyper Parameters
  - Learning Rate, Batch Size
- Regularization
- Data augmentation
- 10 Random seeds





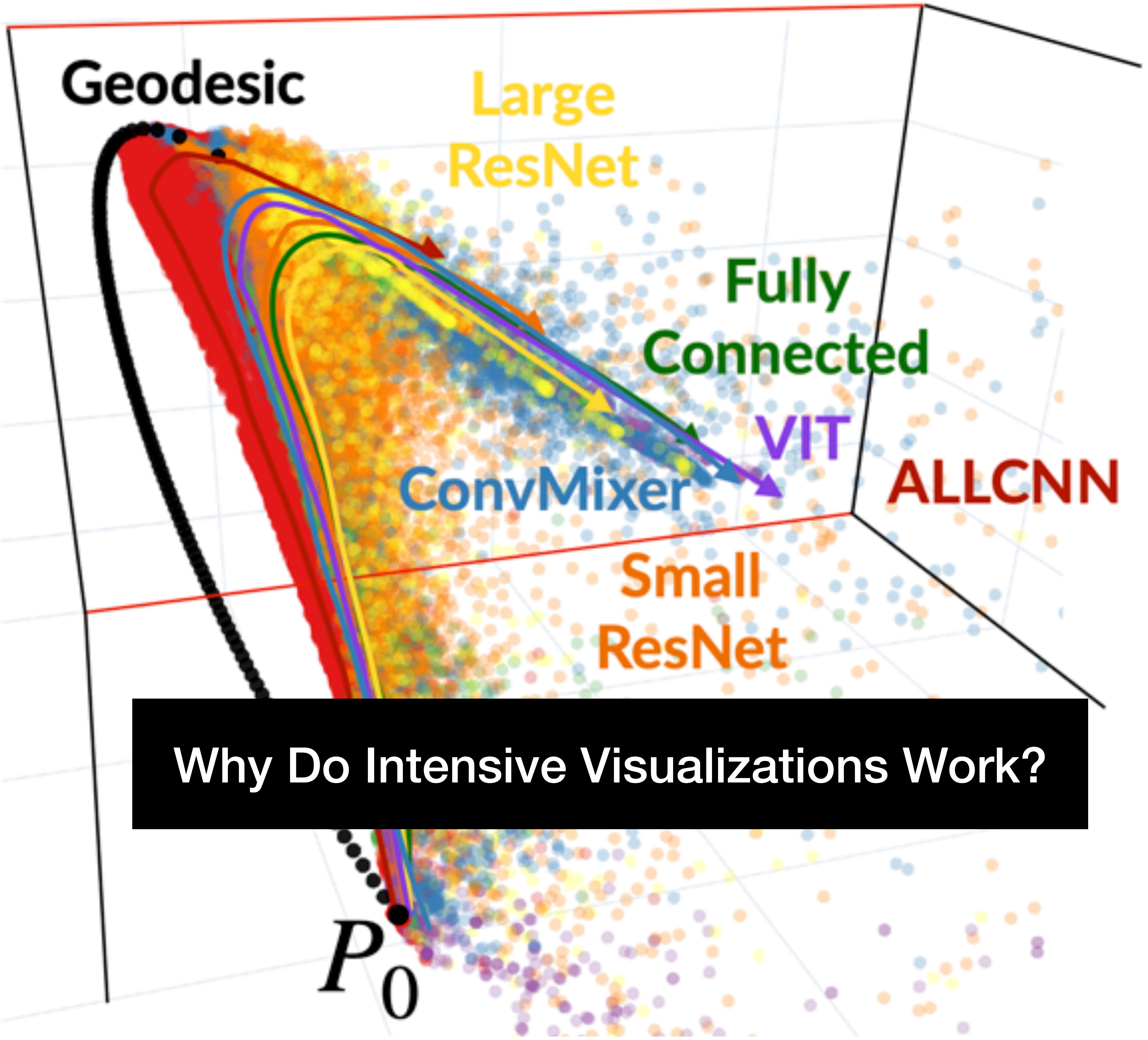
The Training Process explores a Low Dimensional Manifold

~2,000 Configurations

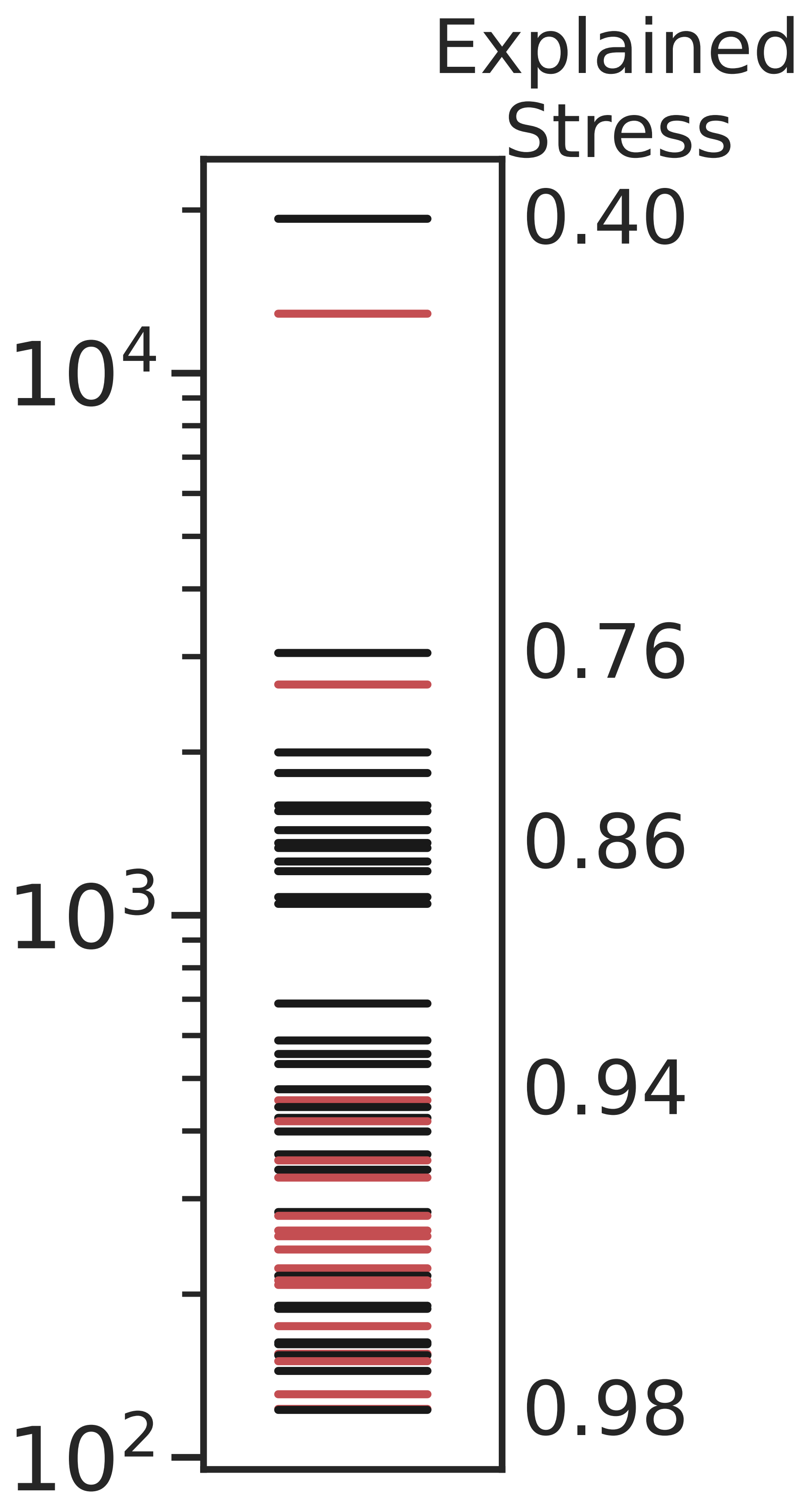
~150,000 Networks

- Architectures
- Optimization:
  - SGD, SGDN, ADAM
- Hyper Parameters
  - Learning Rate, Batch Size
- Regularization
- Data augmentation
- 10 Random seeds

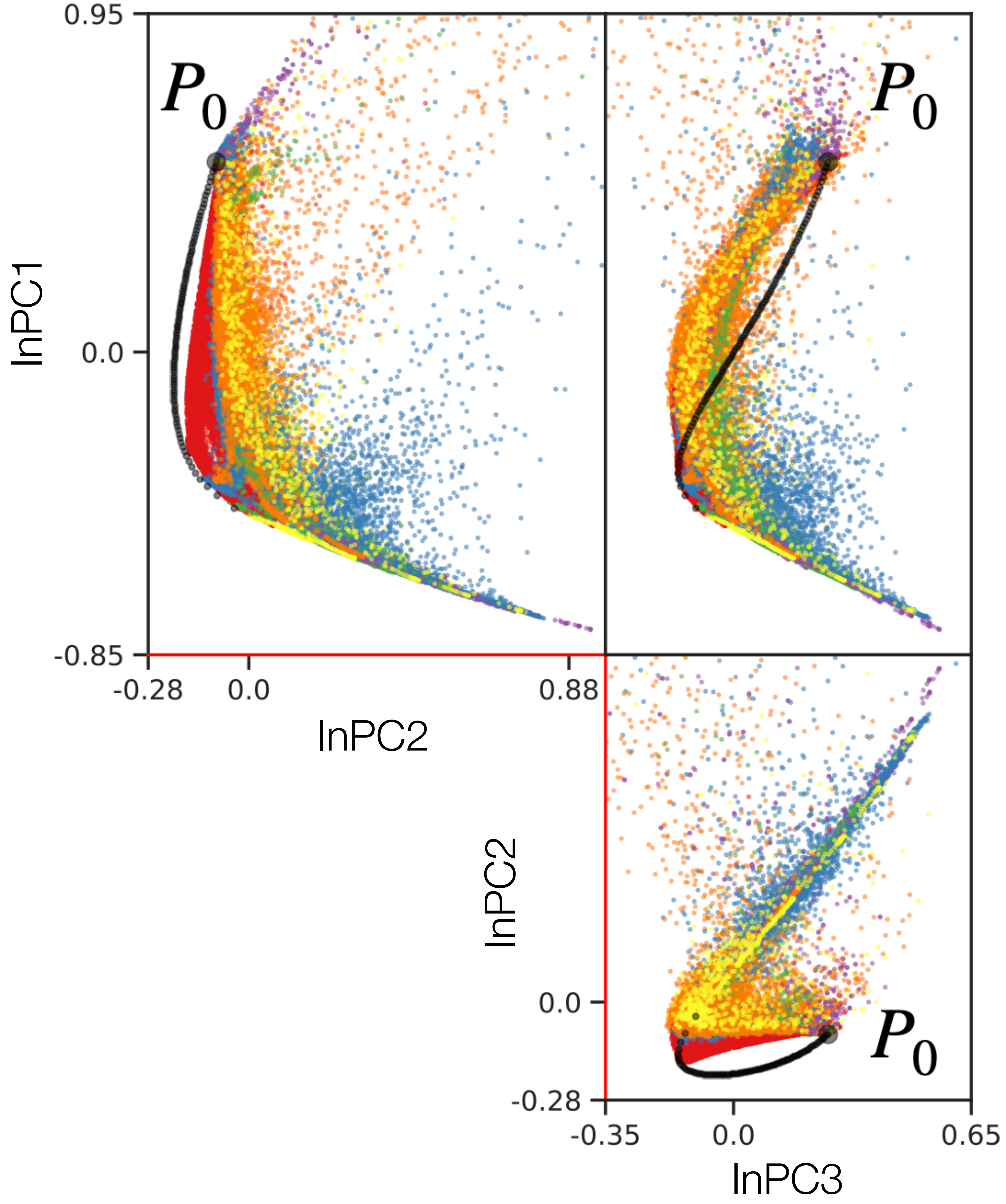
CIFAR-10 $\sim 10^6$ Dimensions

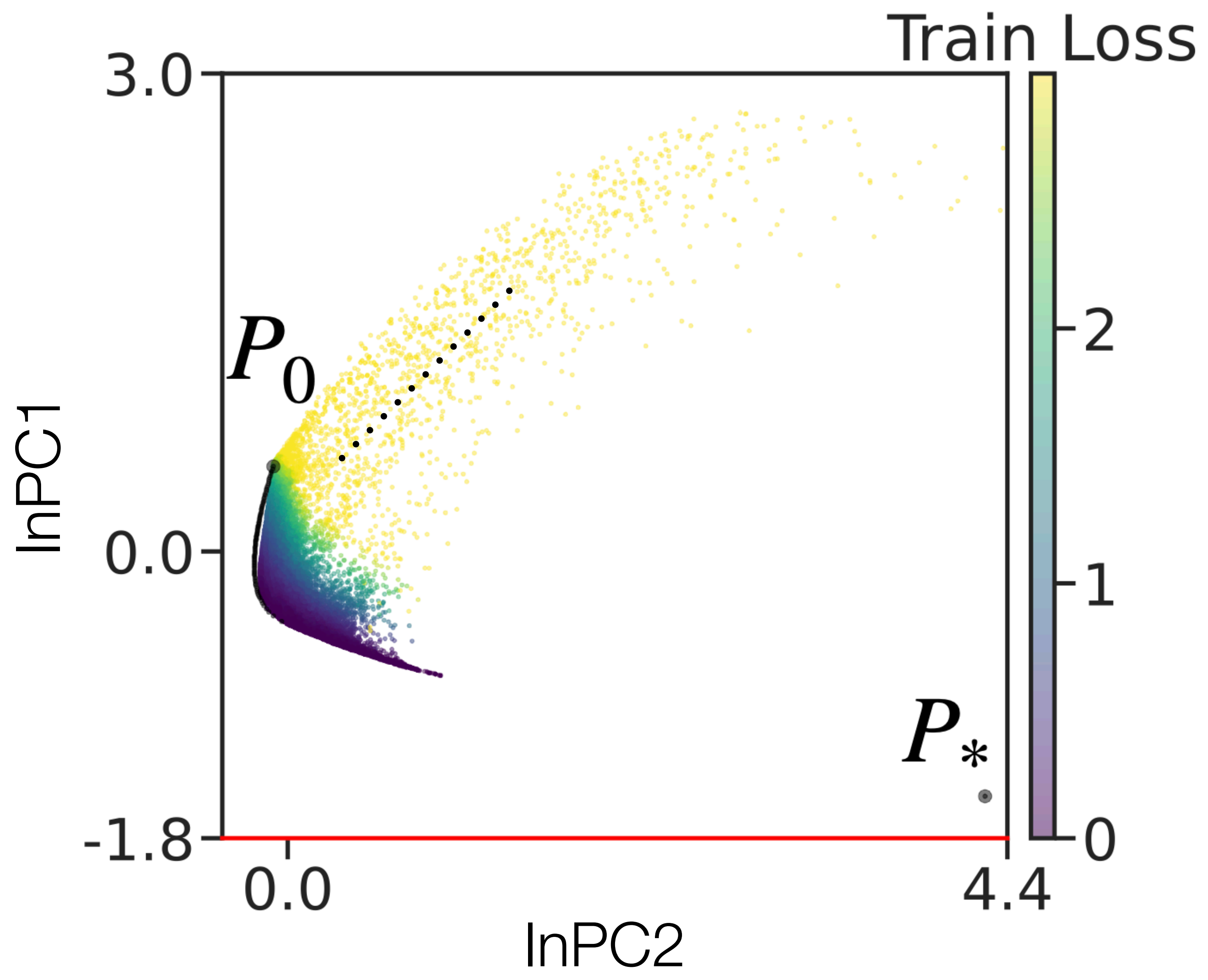


Why Do Intensive Visualizations Work?



Intensive Embedding are Minkowski $\mathbb{R}^{q,p-q}$



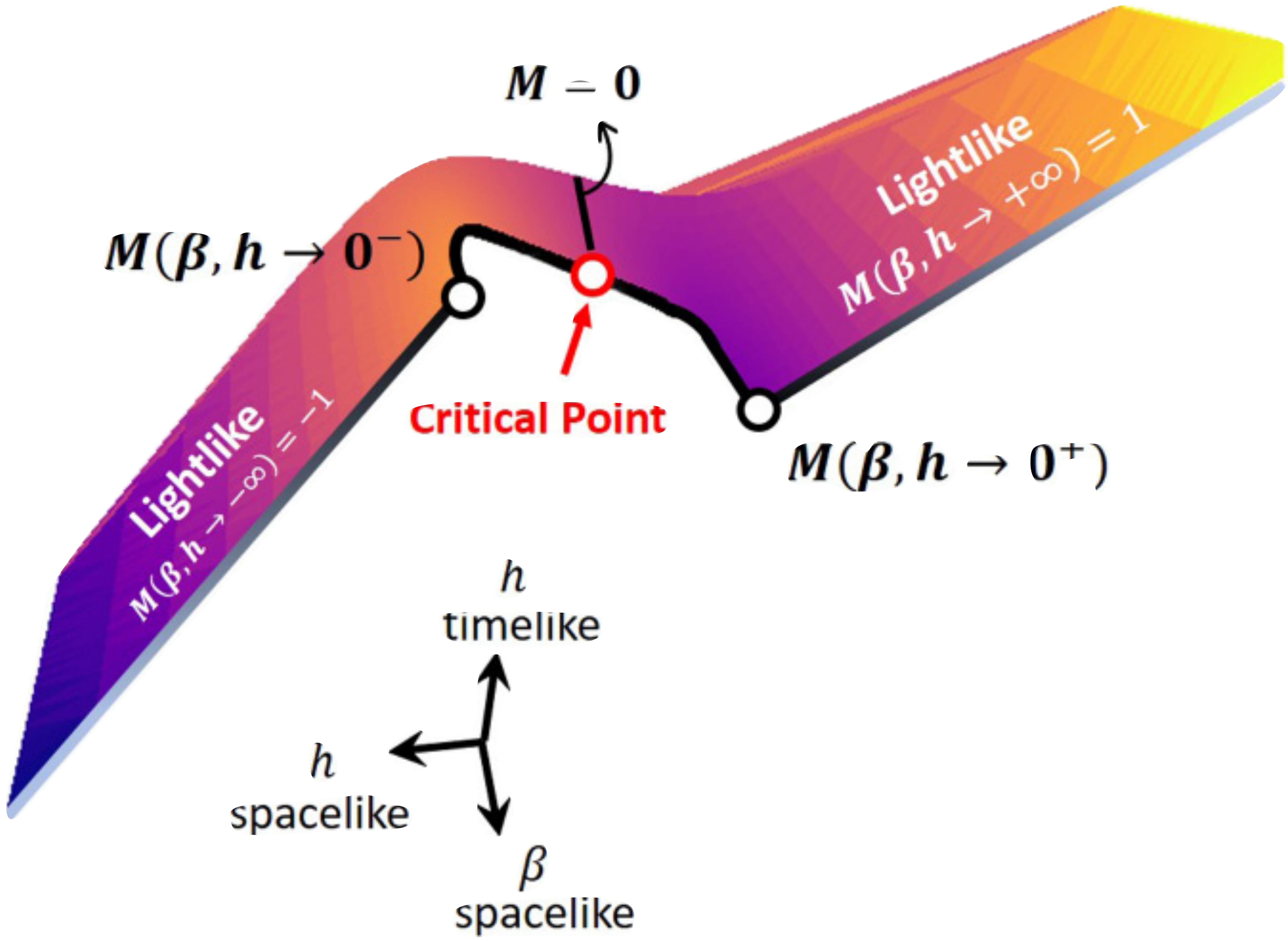


$$ds^2 = dx^2 - dt^2$$

Light Cones $dx = \pm\, dt$

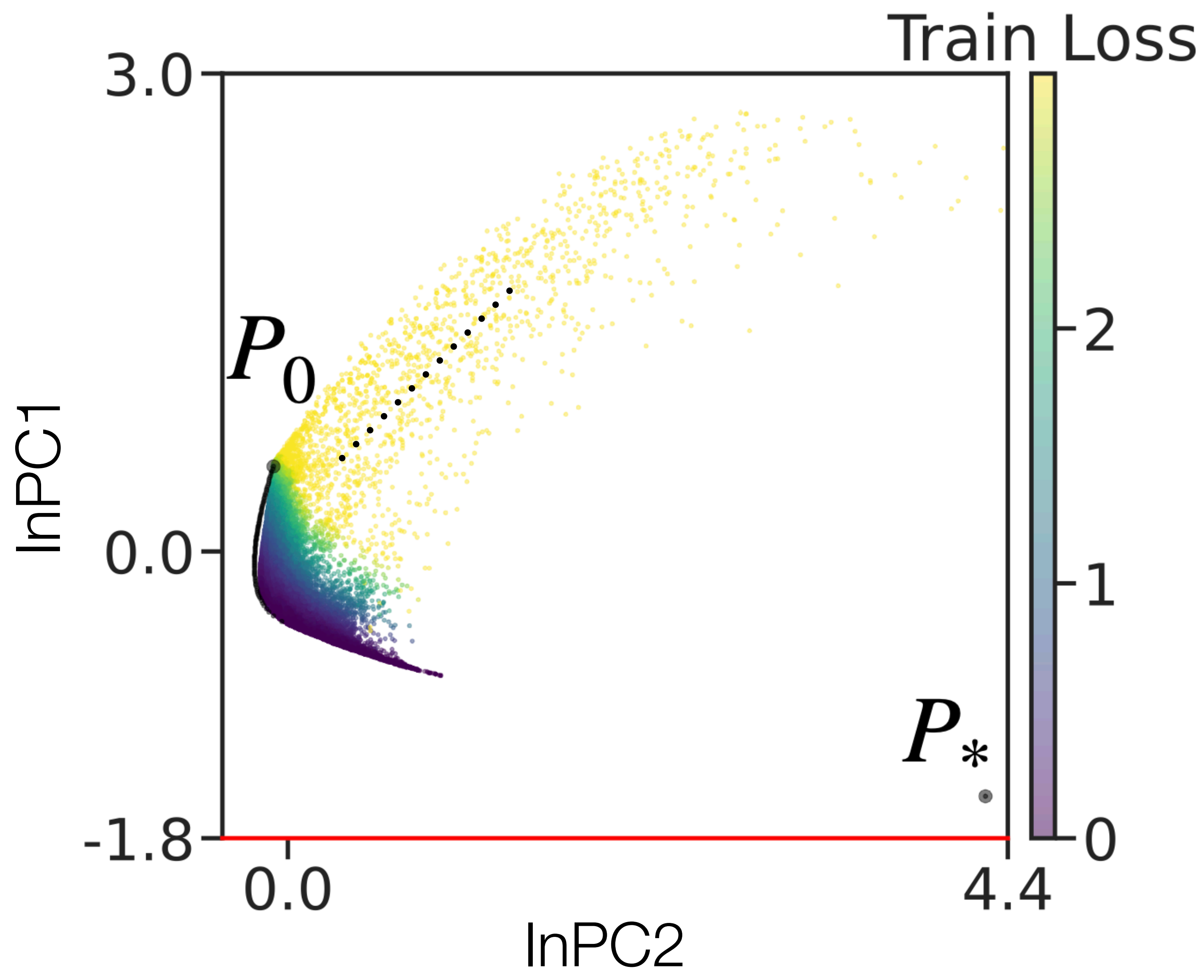
Connects Different Models With Equal Predictions

Intensive Embedding are Minkowski $\mathbb{R}^{q,p-q}$



Model Manifold of the 2D Ising model

Teoh et al. PRR 2020

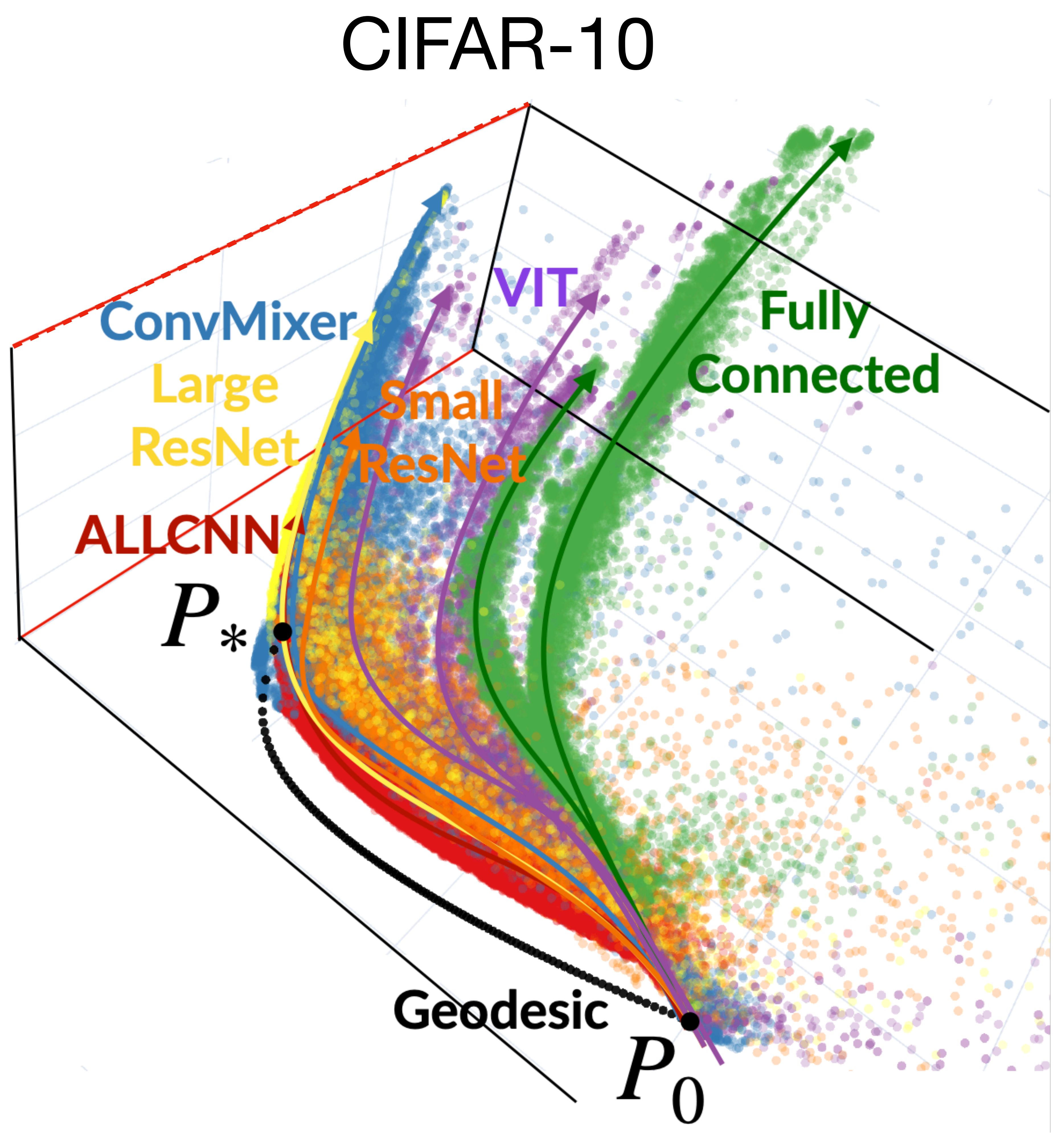


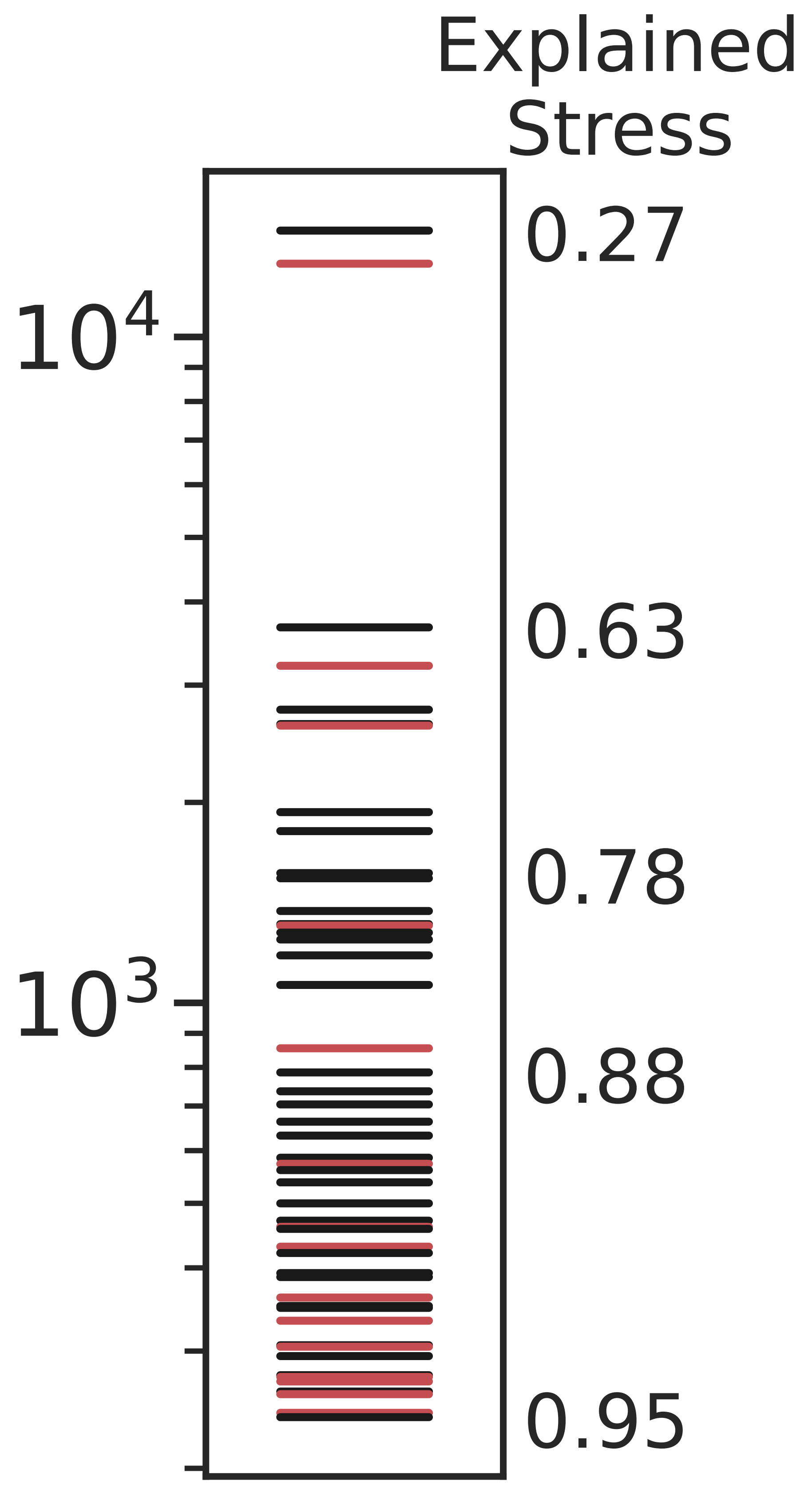
$$ds^2 = dx^2 - dt^2$$

Light Cones $dx = \pm\, dt$

Connects Different Models With Equal Predictions

Test Embedding is also Low Dimensional



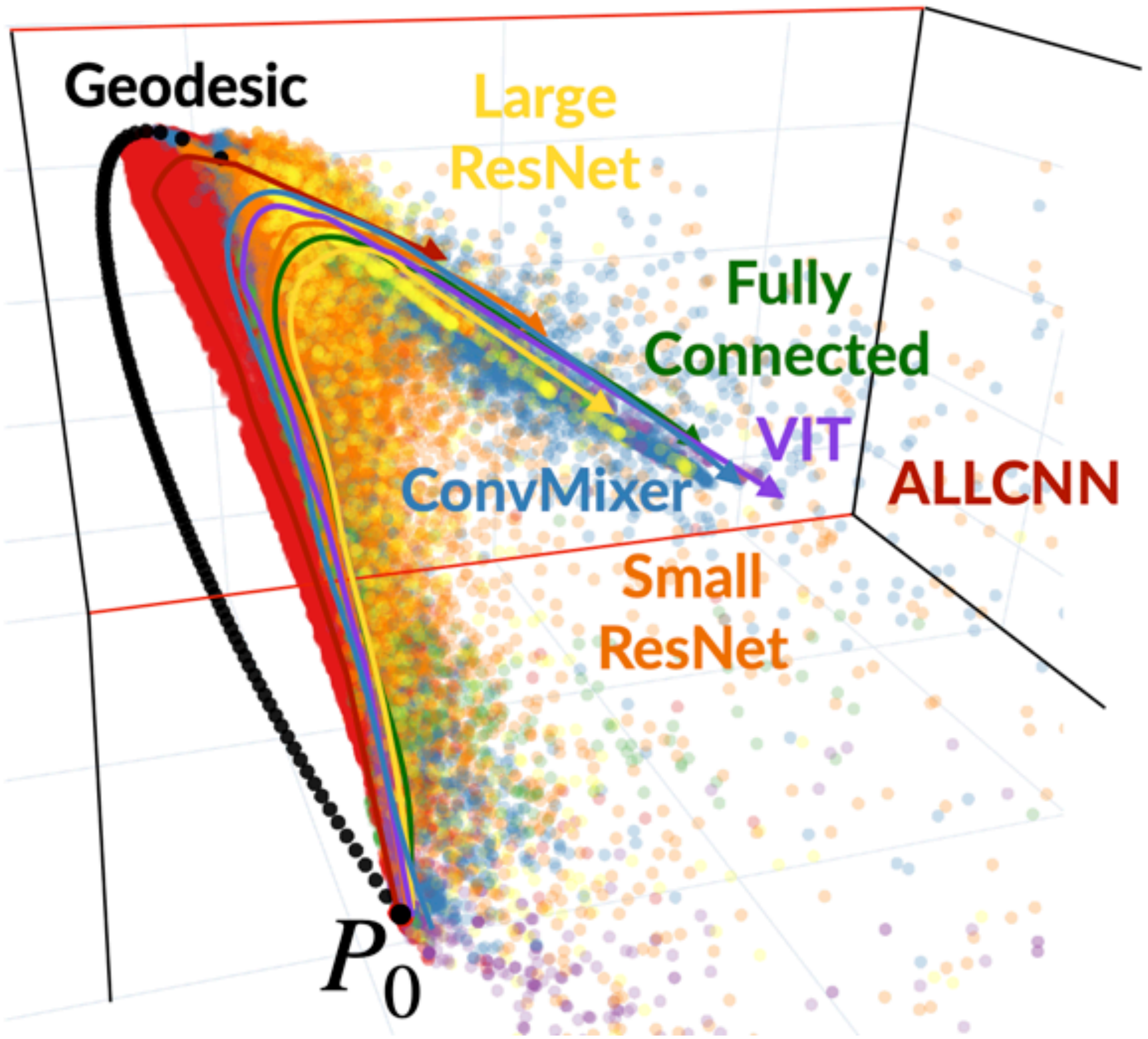


What Can We Learn About Different Configurations?

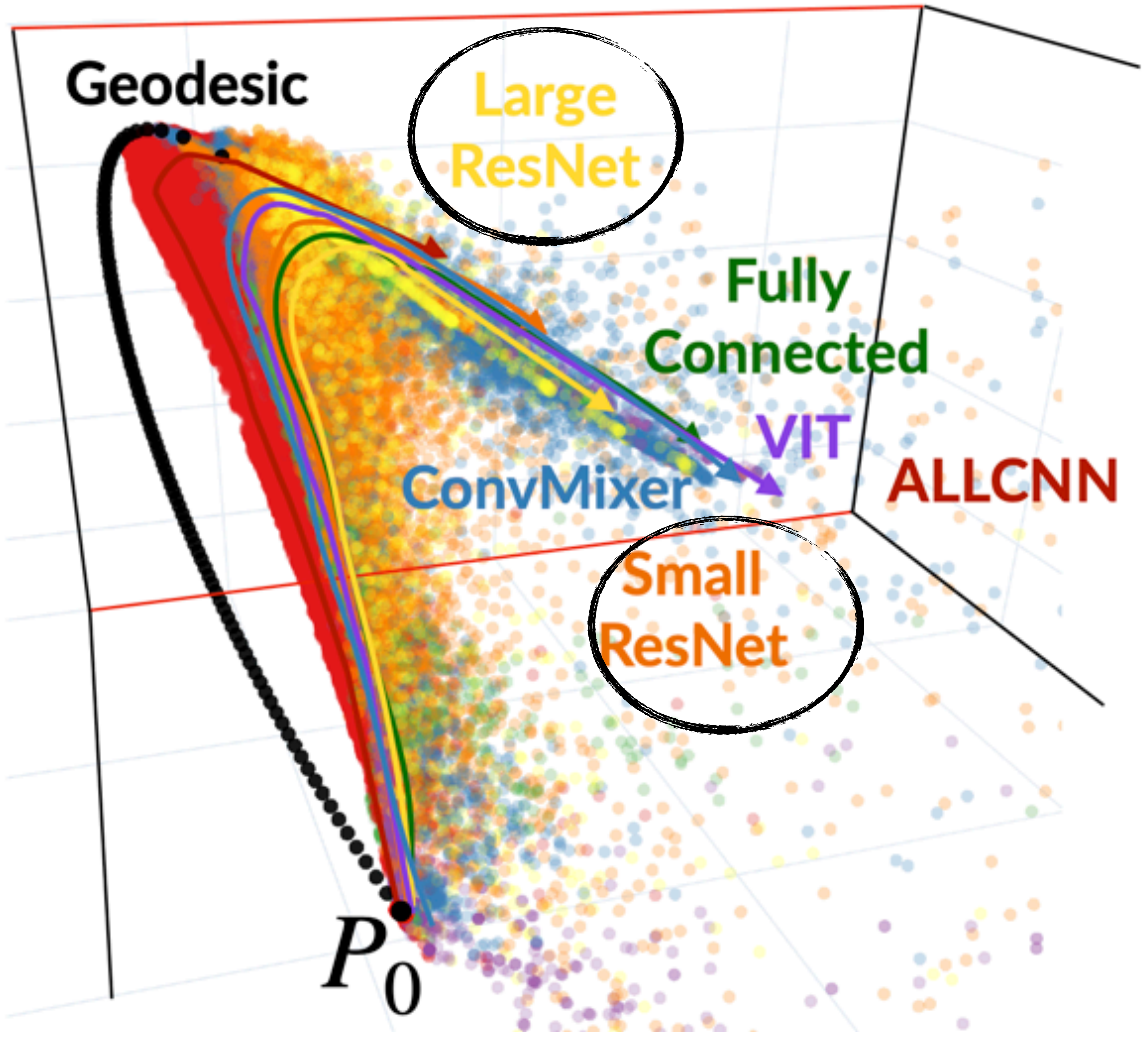
~2,000 Configurations

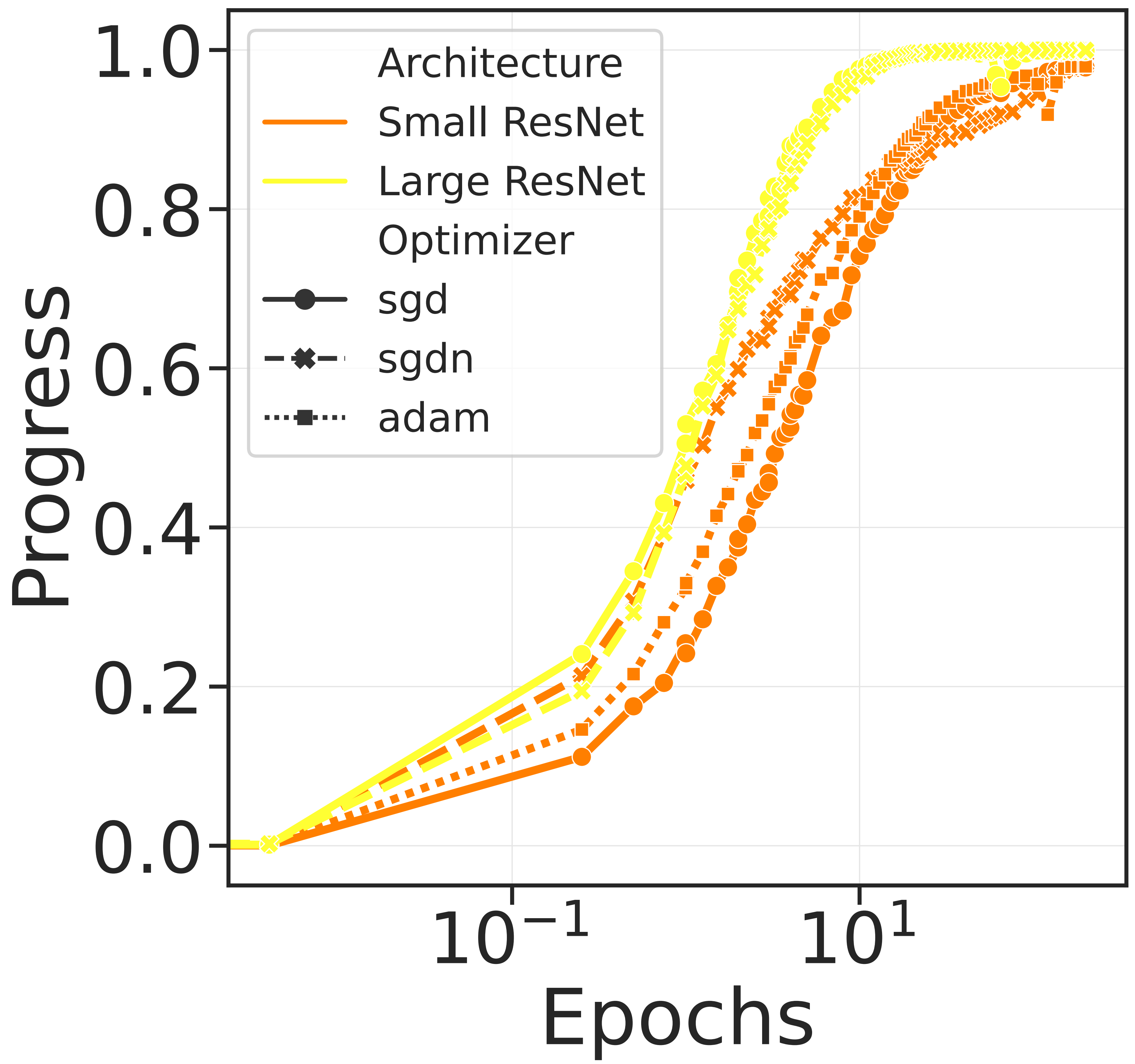
~150,000 Networks

- Architectures
- Optimization:
  - SGD, SGDN, ADAM
- Hyper Parameters
  - Learning Rate, Batch Size
- Regularization
- Data augmentation
- 10 Random seeds



A Larger Network Trains Along the Same Manifold as a Smaller Network With a Similar Architecture (But is Faster)

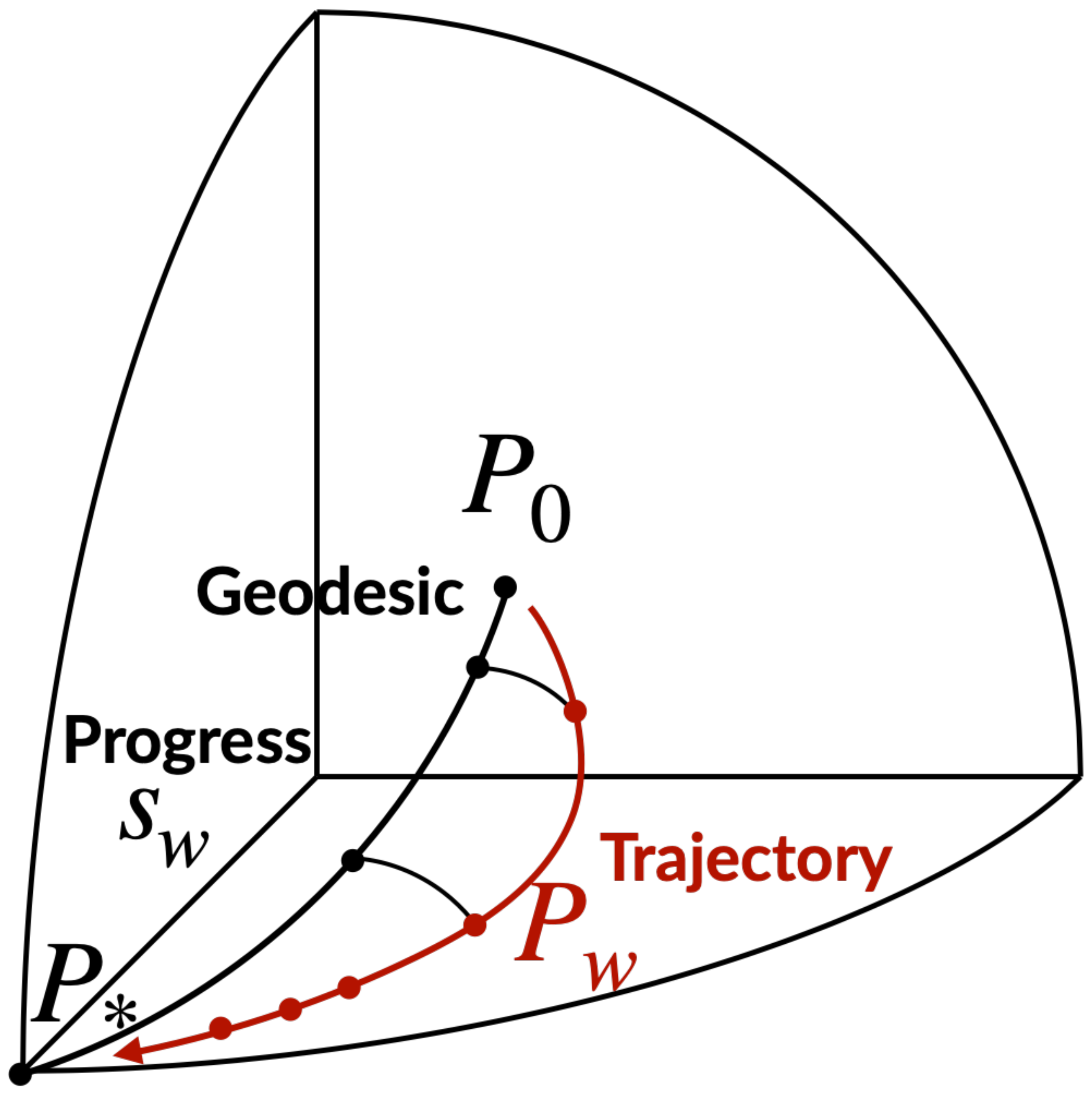


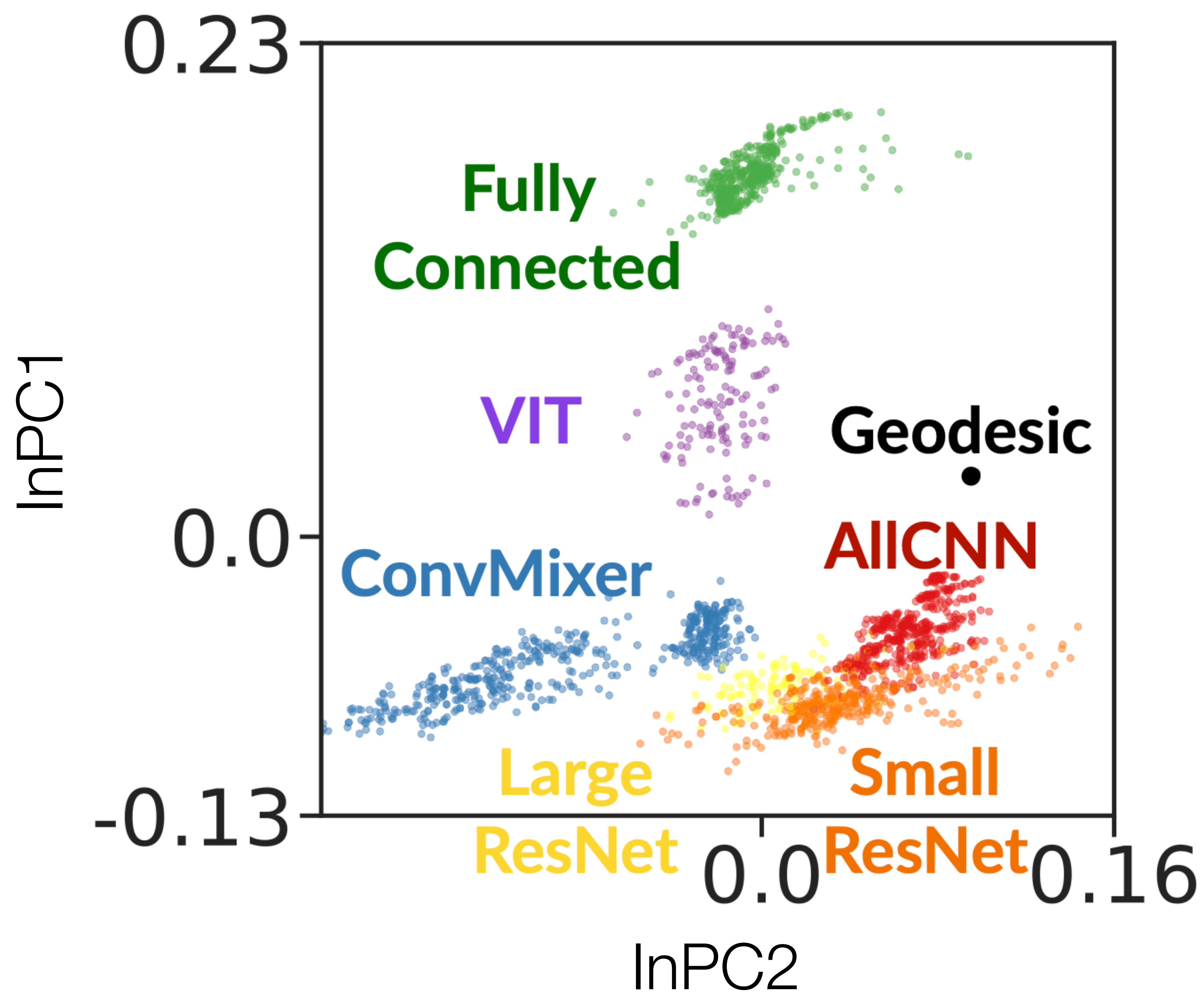


Distance between trajectories

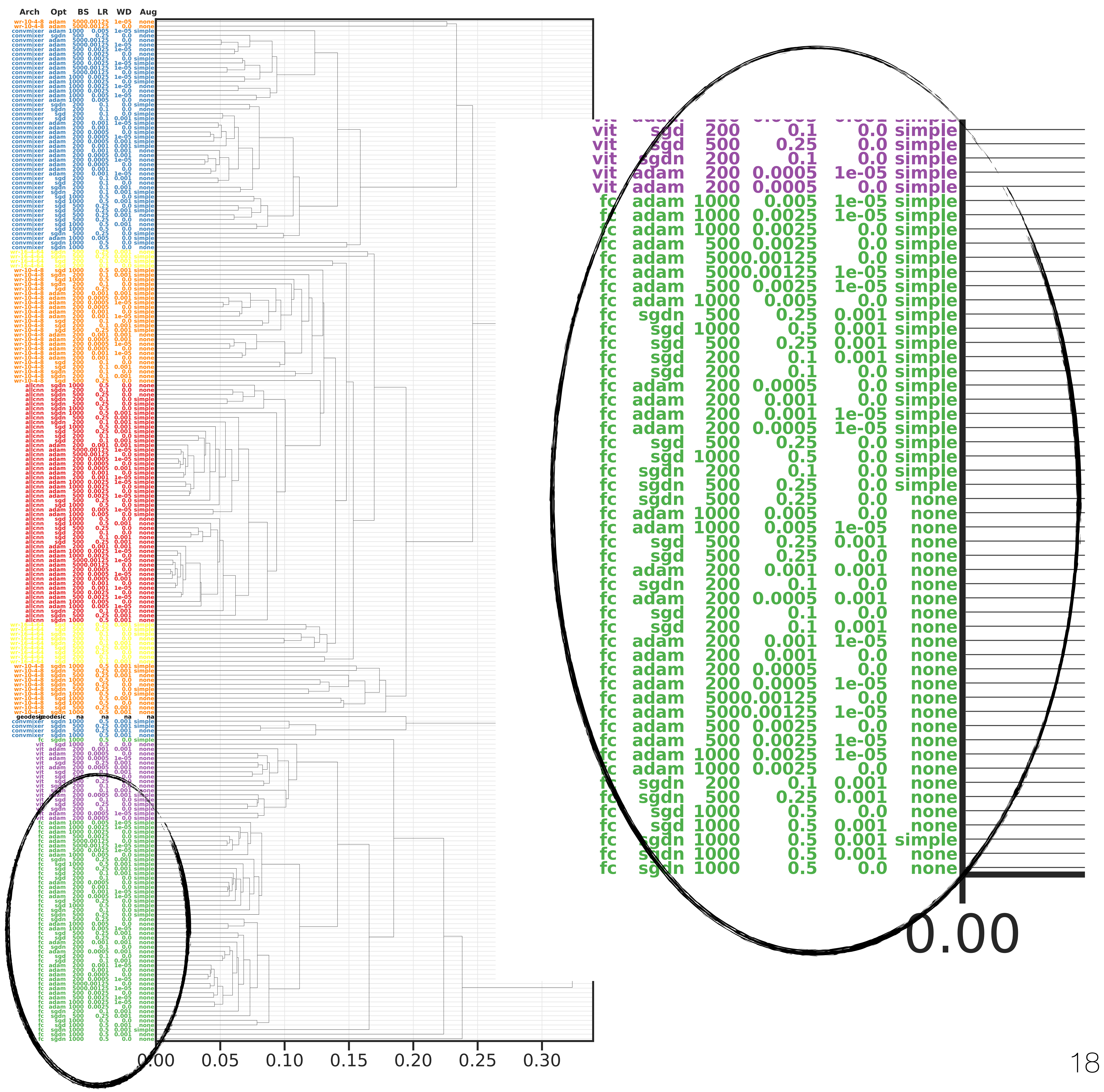
Define progress ≈ “geodesic arclength parametrization” to remove speed

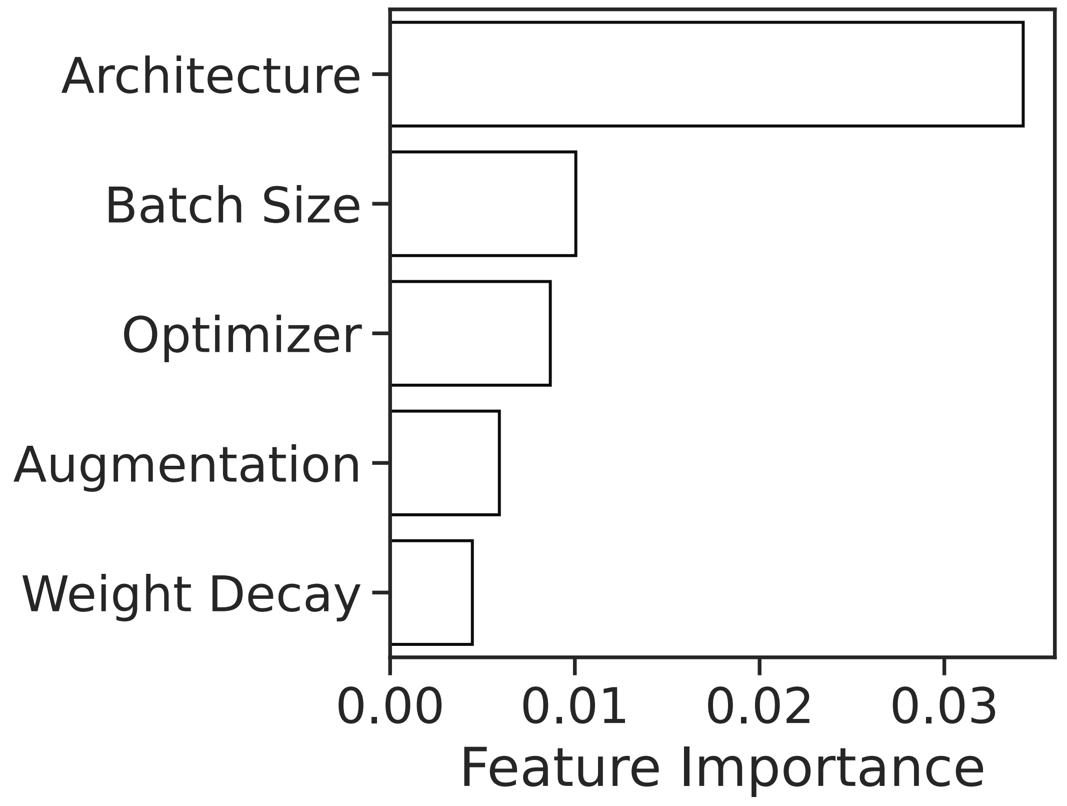
$$d(\tau_u, \tau_v) = \int d_B(P_{u(s)}, P_{v(s)})ds$$

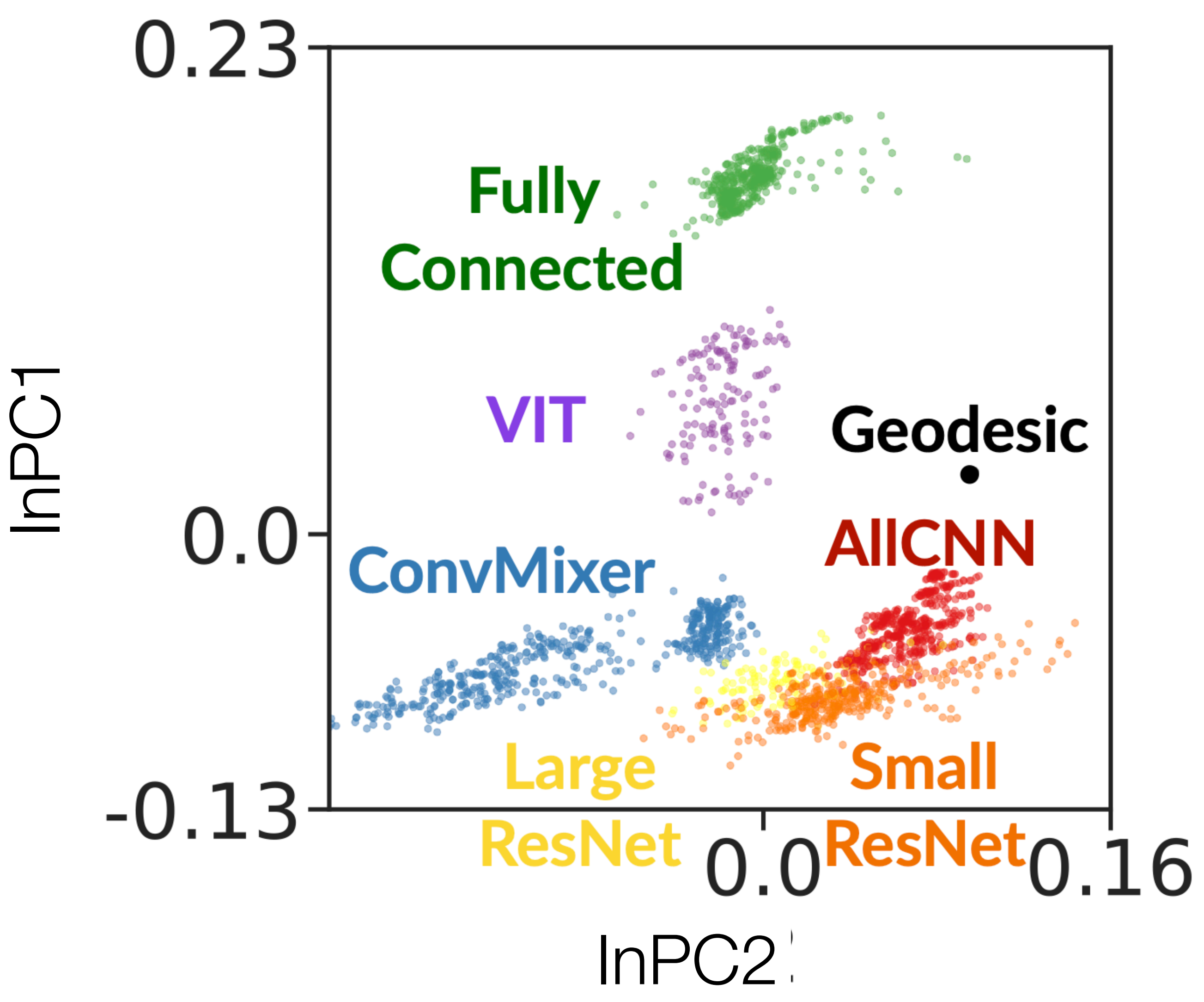




Architecture- Not training or regularization - primarily distinguished trajectories







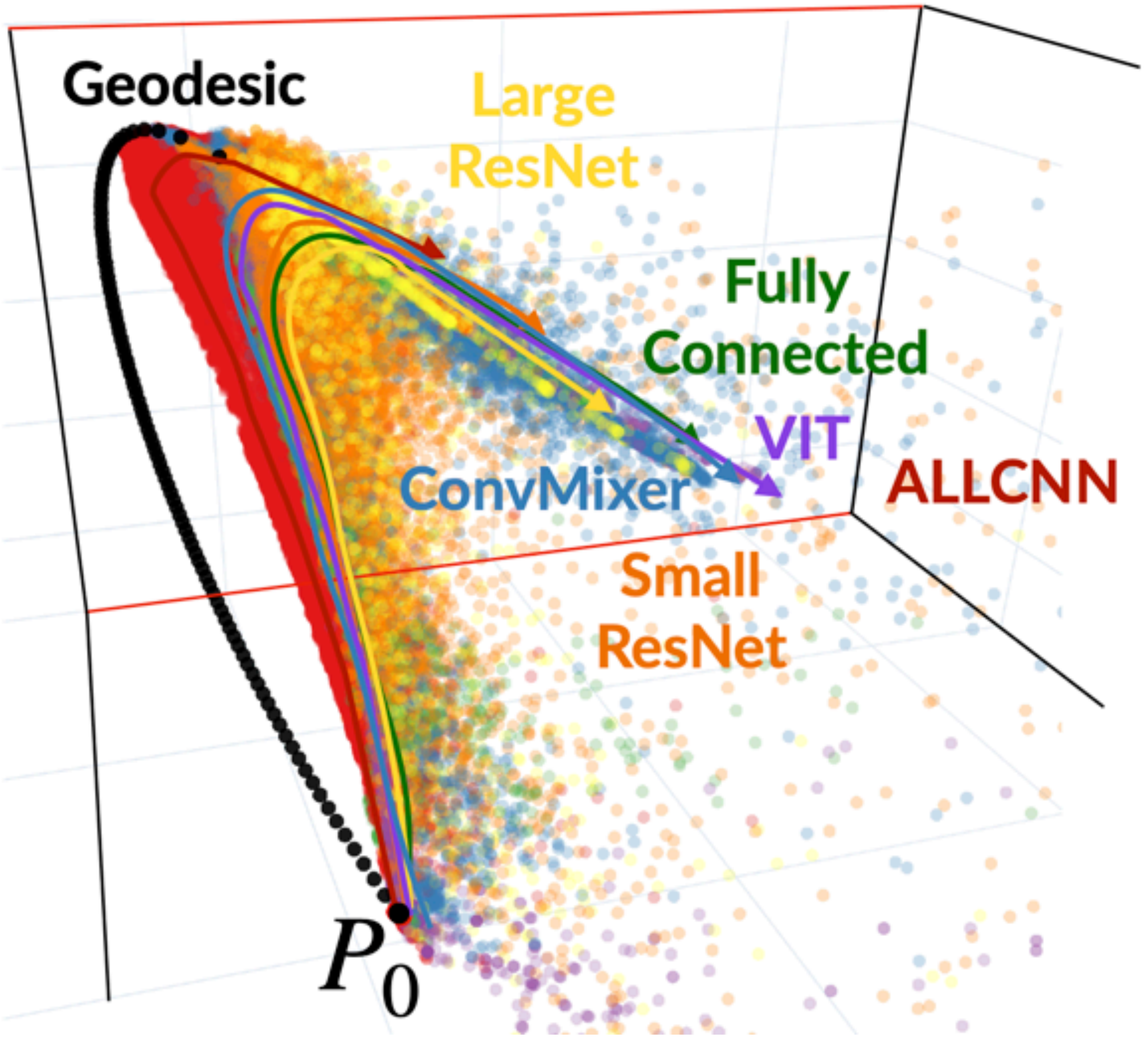
Why is the training low dimensional?

Suspects

1. Data is Structured - Easy and hard Images are common across networks.

2. Weights Initialize at ignorance P_0

3. Data is Low Dimensional



Why Are The Training Manifolds Low Dimensional?

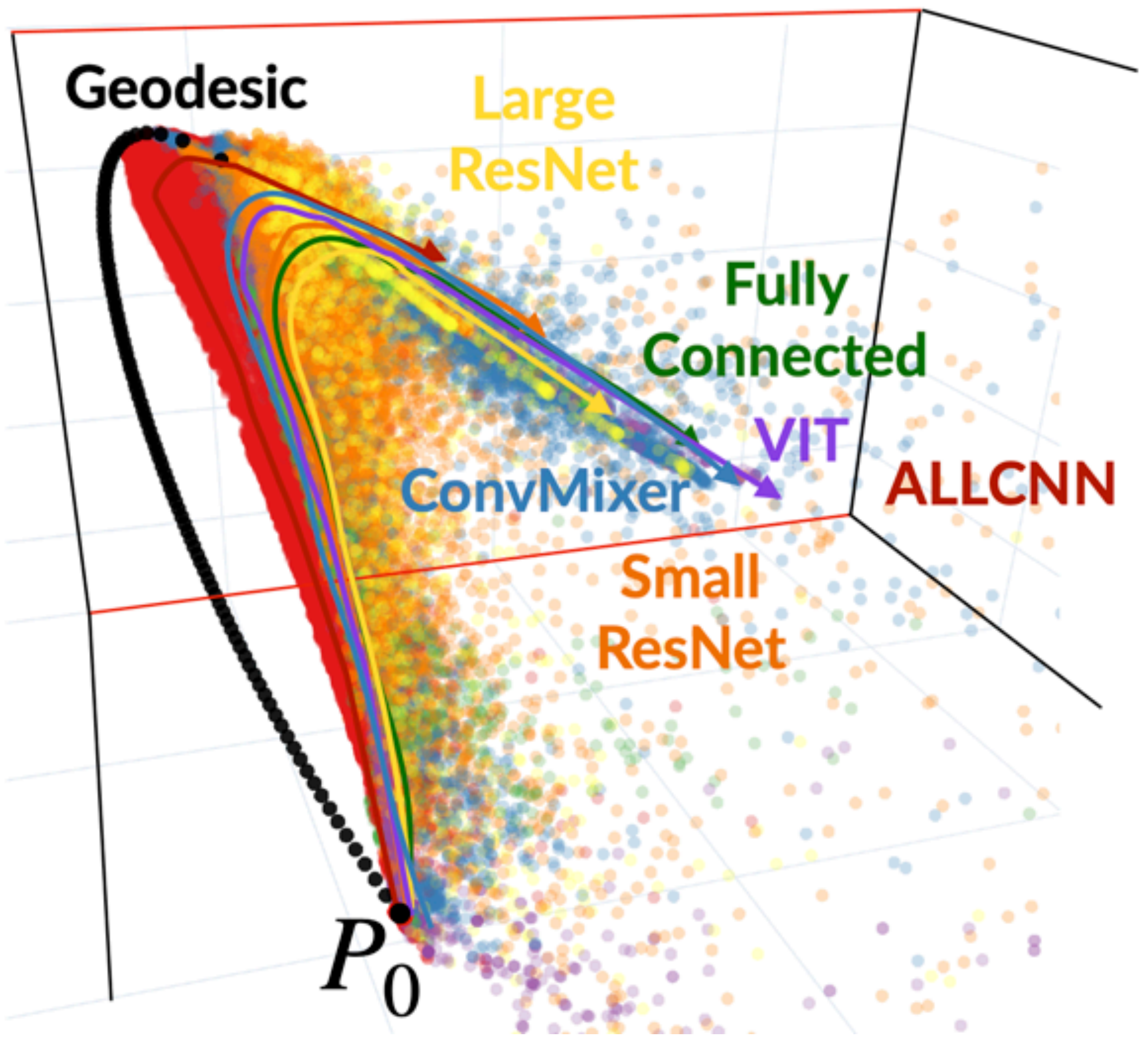
Suspects

1. Data is Structured - Easy and hard Images are common across networks.
2. Weights Initialize at ignorance P_0
3. Data is Low Dimensional

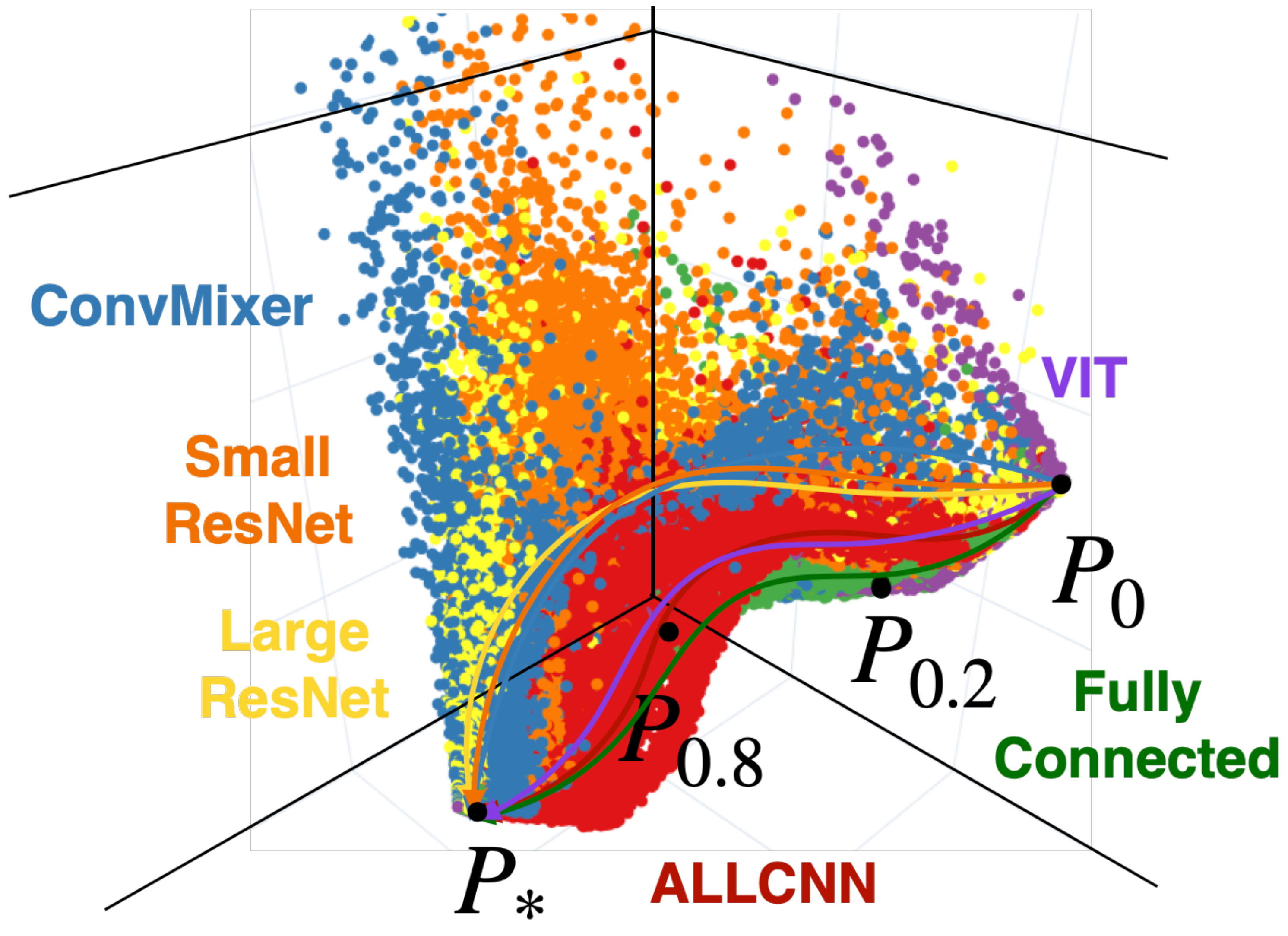
Experiments

1. Embed Data using Initial and Final Tangents
2. Initialization in multiple corners
3. Train on synthetic data with varying effective dimensionality & initialization

Experiment 1: Embed Data using Initial and Final Tangents

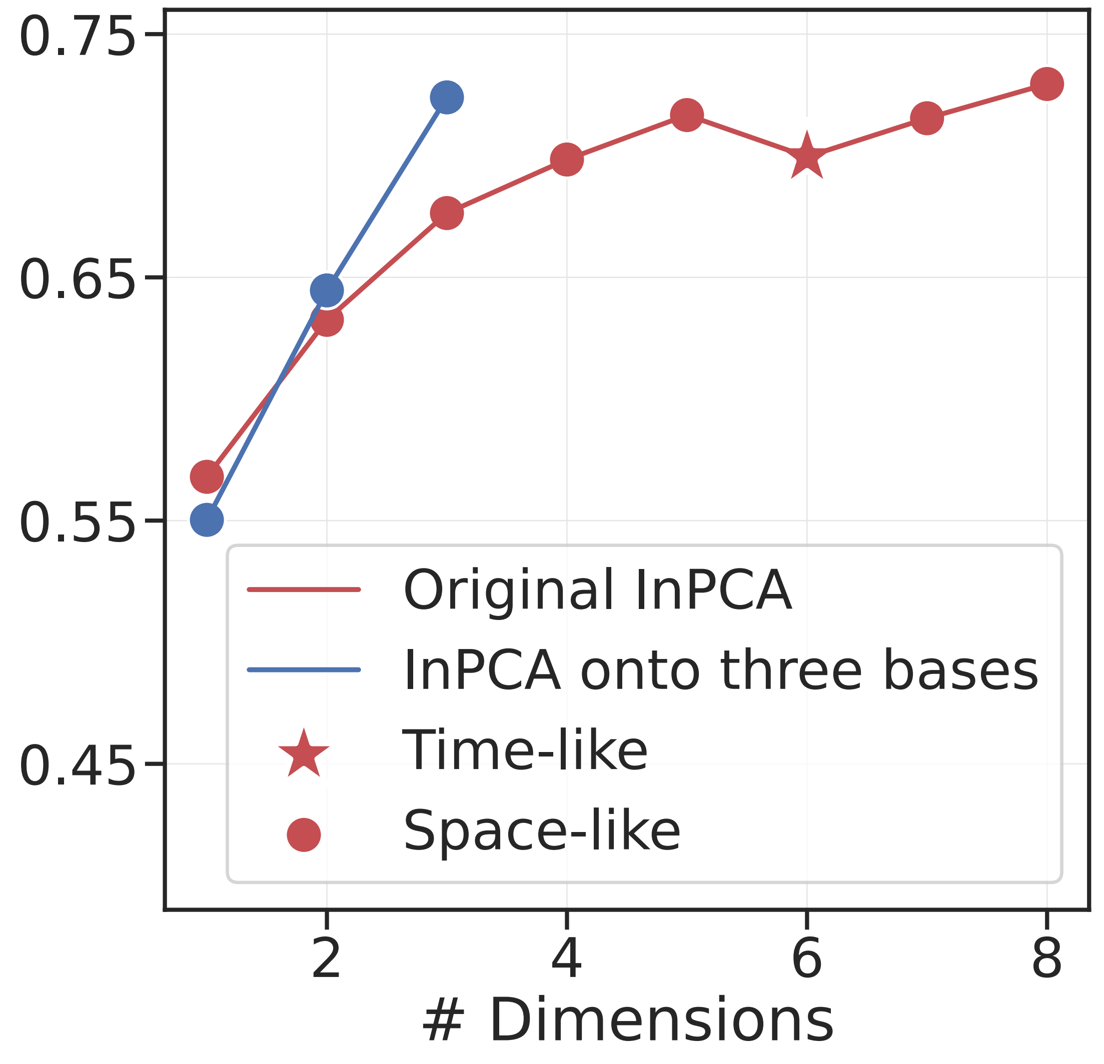


Original InPCA



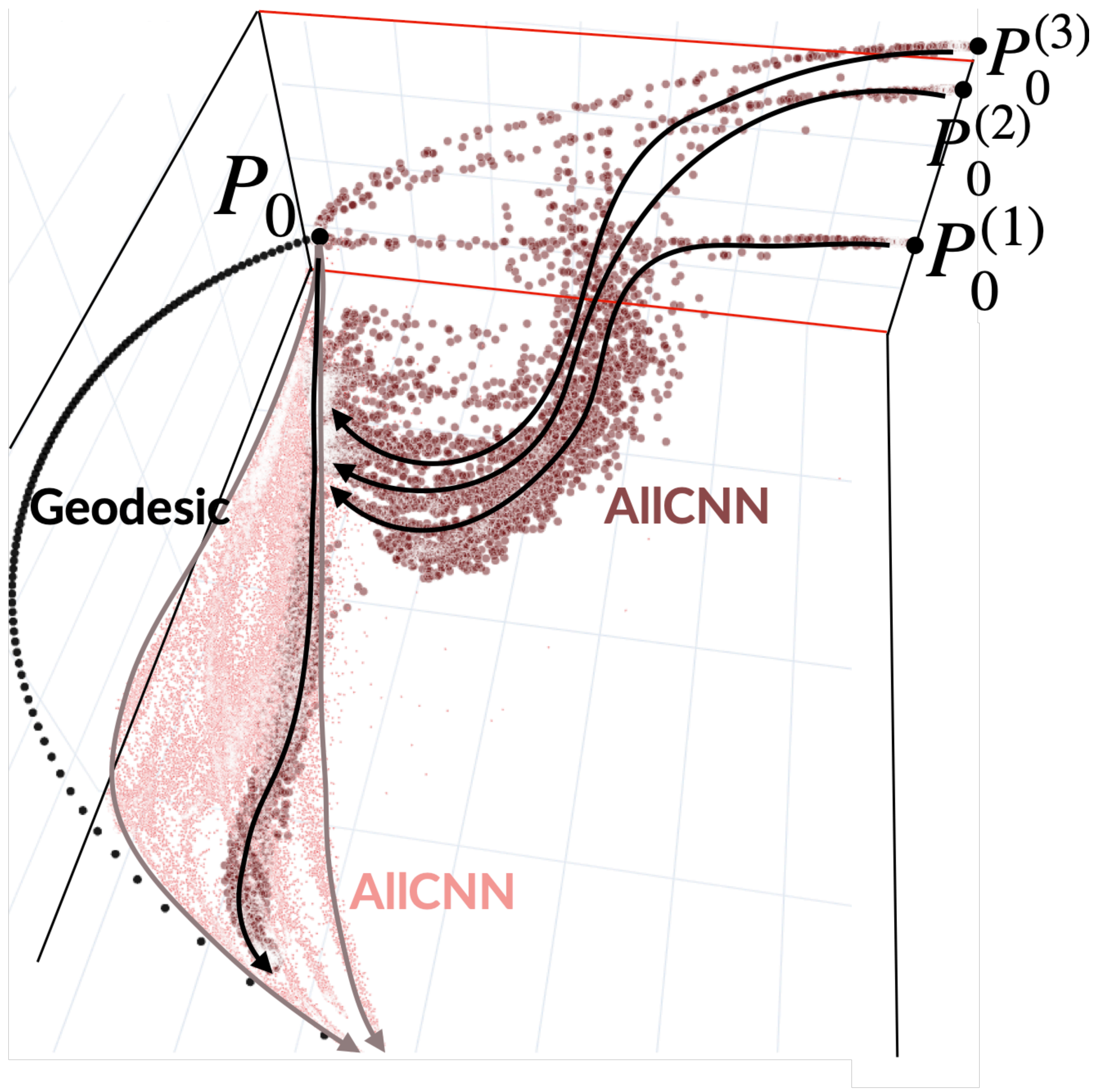
Tangent Embedding

InPCA using 4 points

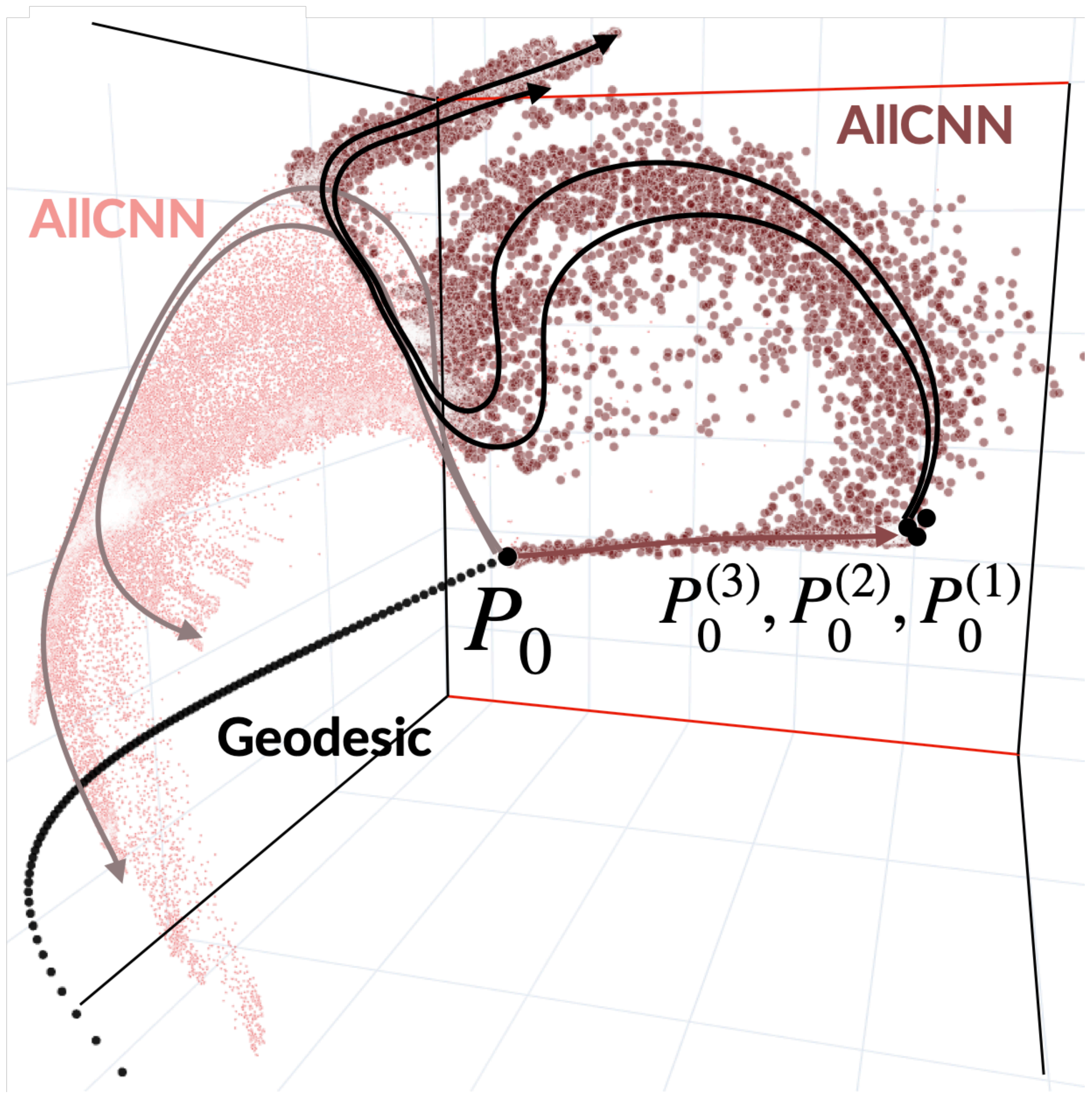


Explained Pairwise Distance

Experiment 2: Initialization in multiple corners

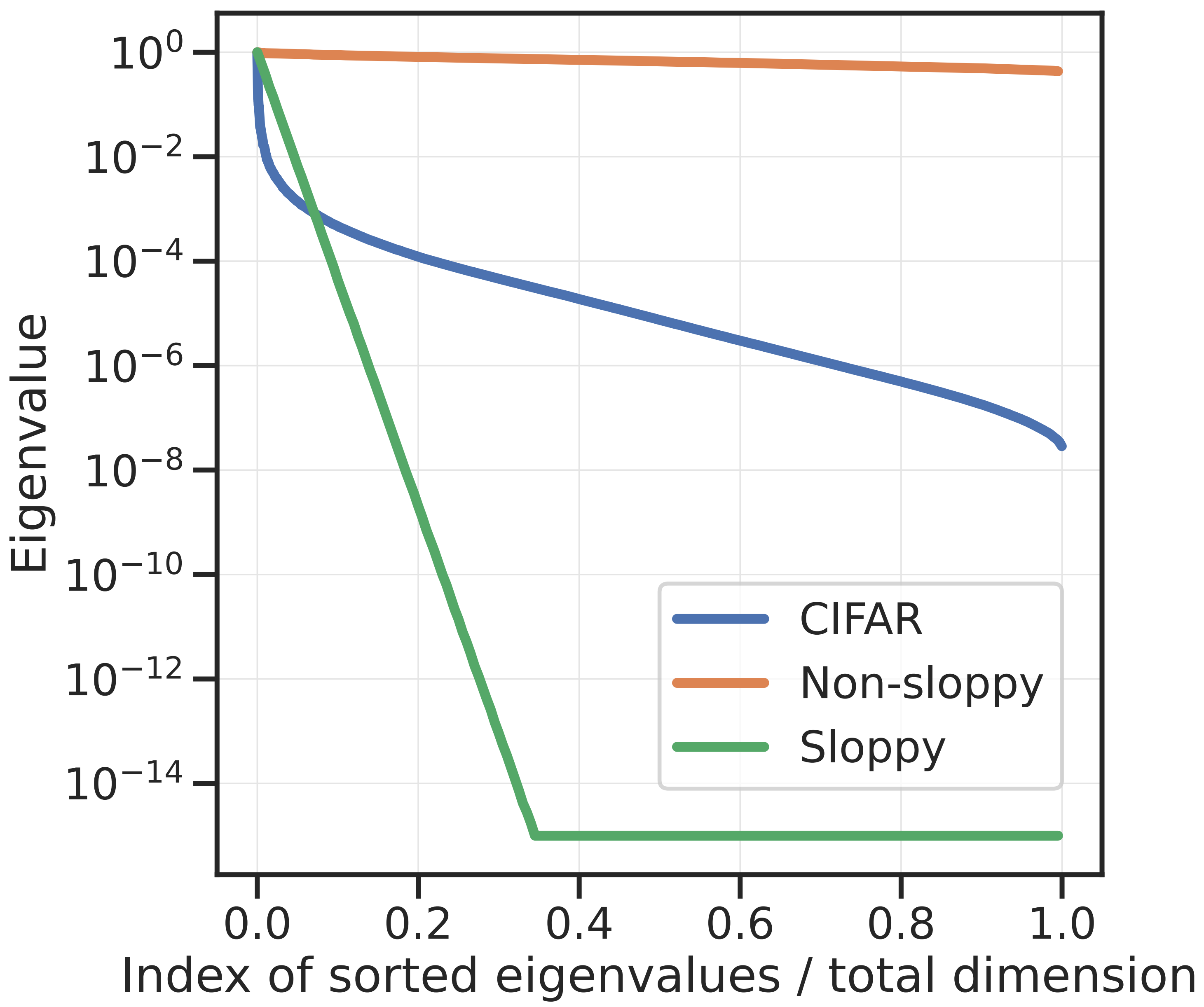


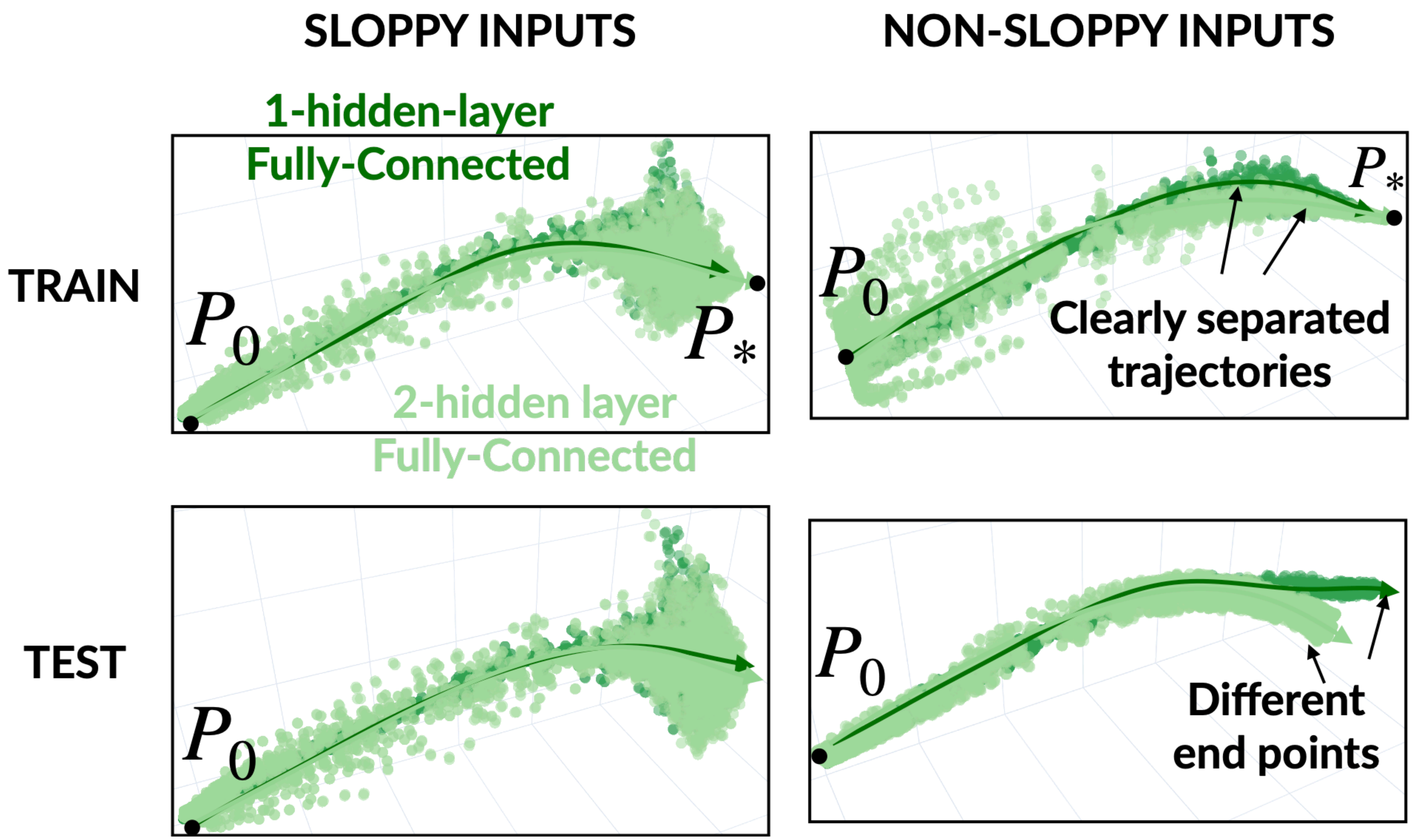
Training Data



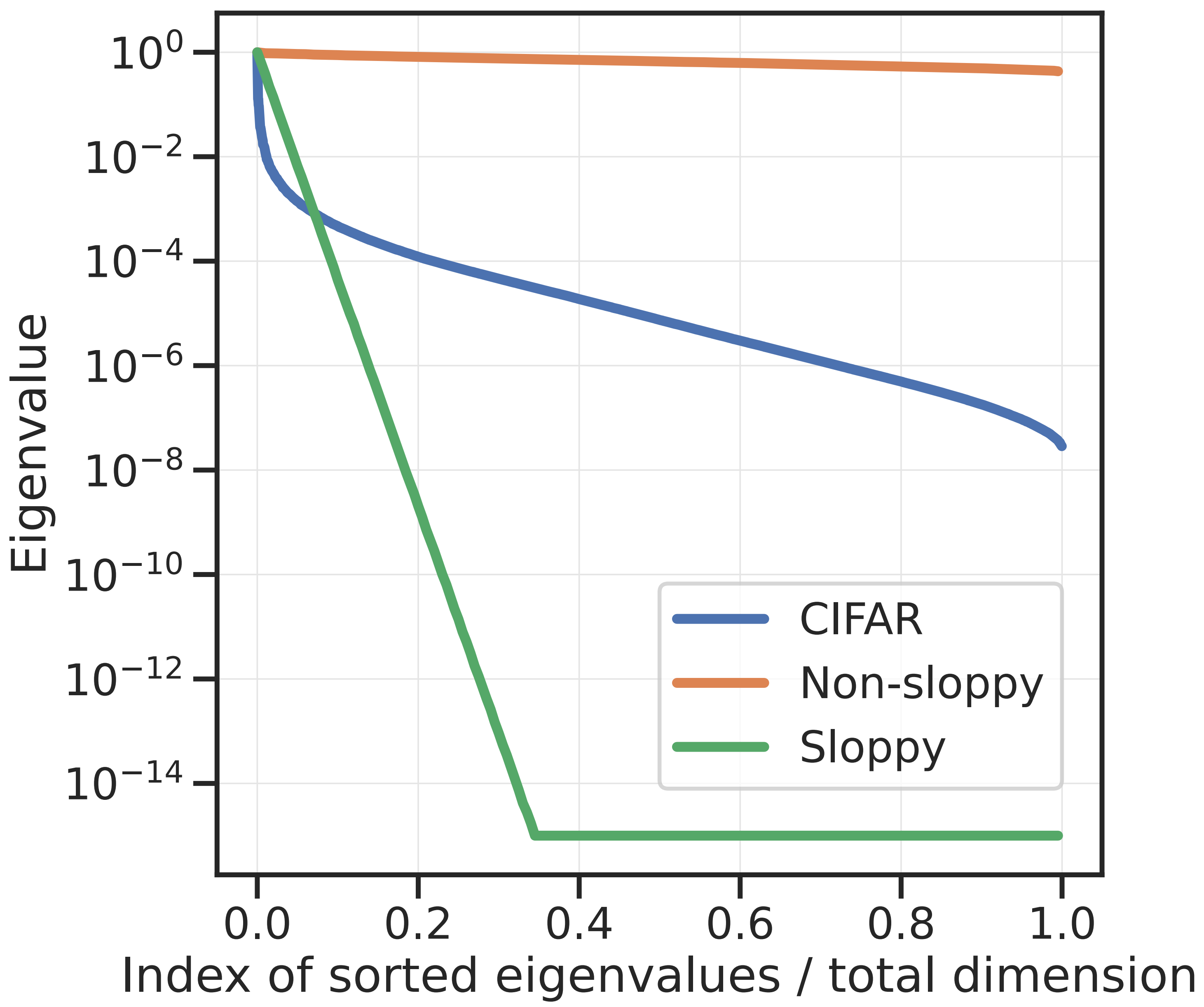
Test Data

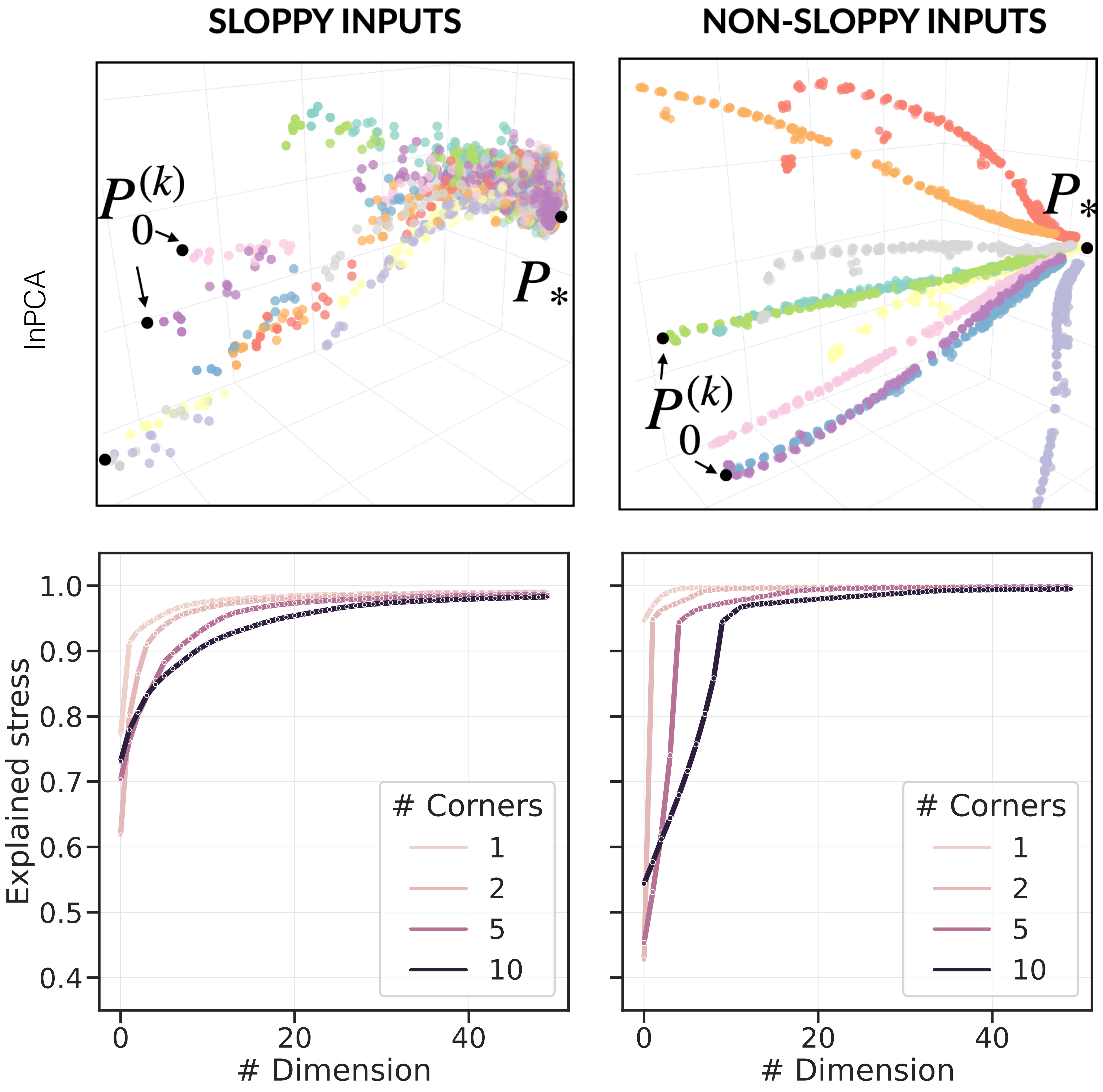
Experiment 3: Synthetic data with varying dimensionality





Experiment 3.2: Synthetic data with varying dimensionality & initialization



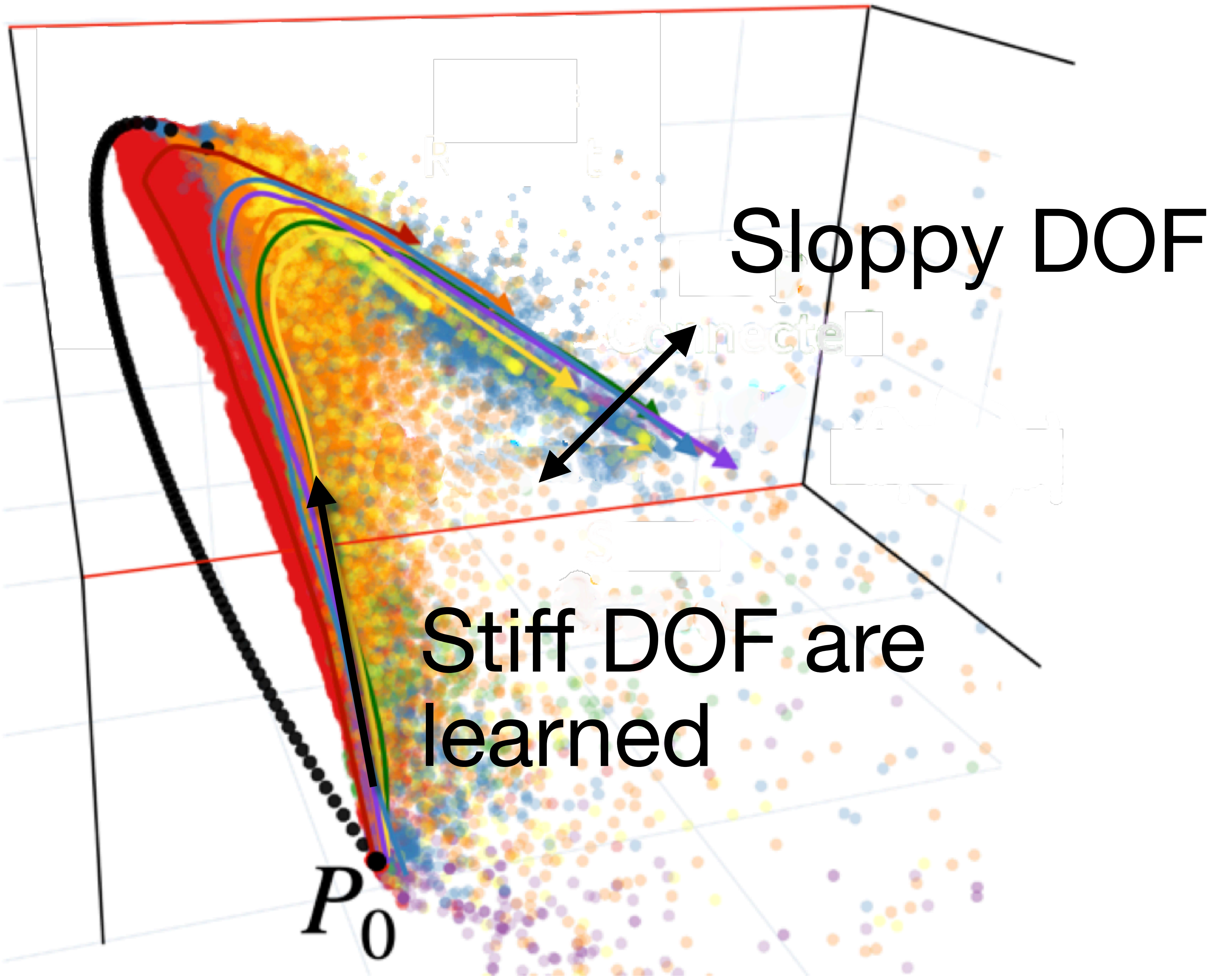


Why Are The Training Manifolds Low Dimensional?
a hint to why training neural nets is easy, and why they generalize well

1. Data is Structured - Hard/ Easy images are common

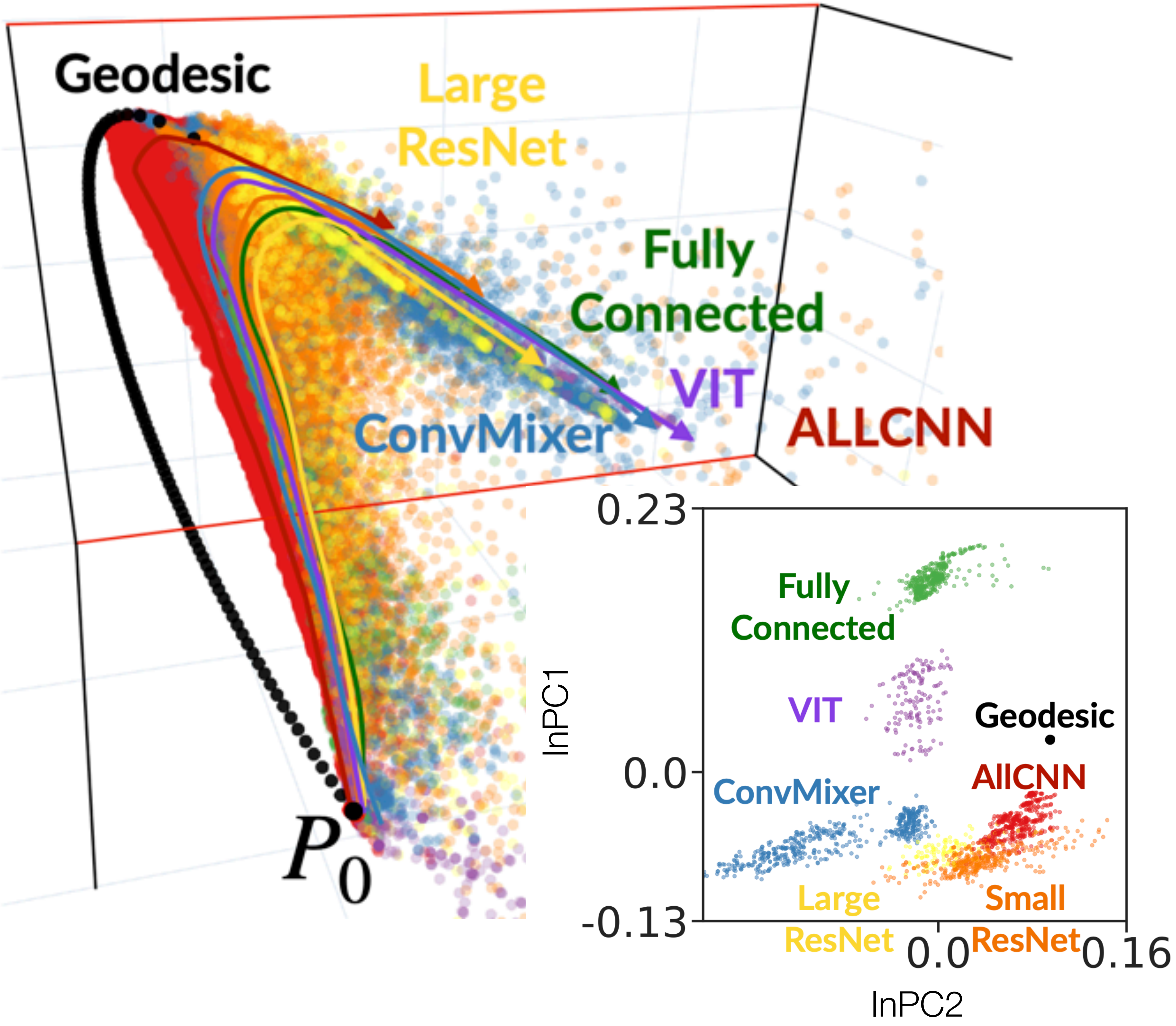
2. Data is Sloppy

3. Weights Initialized at ignorance P_0 explore few hypothesis

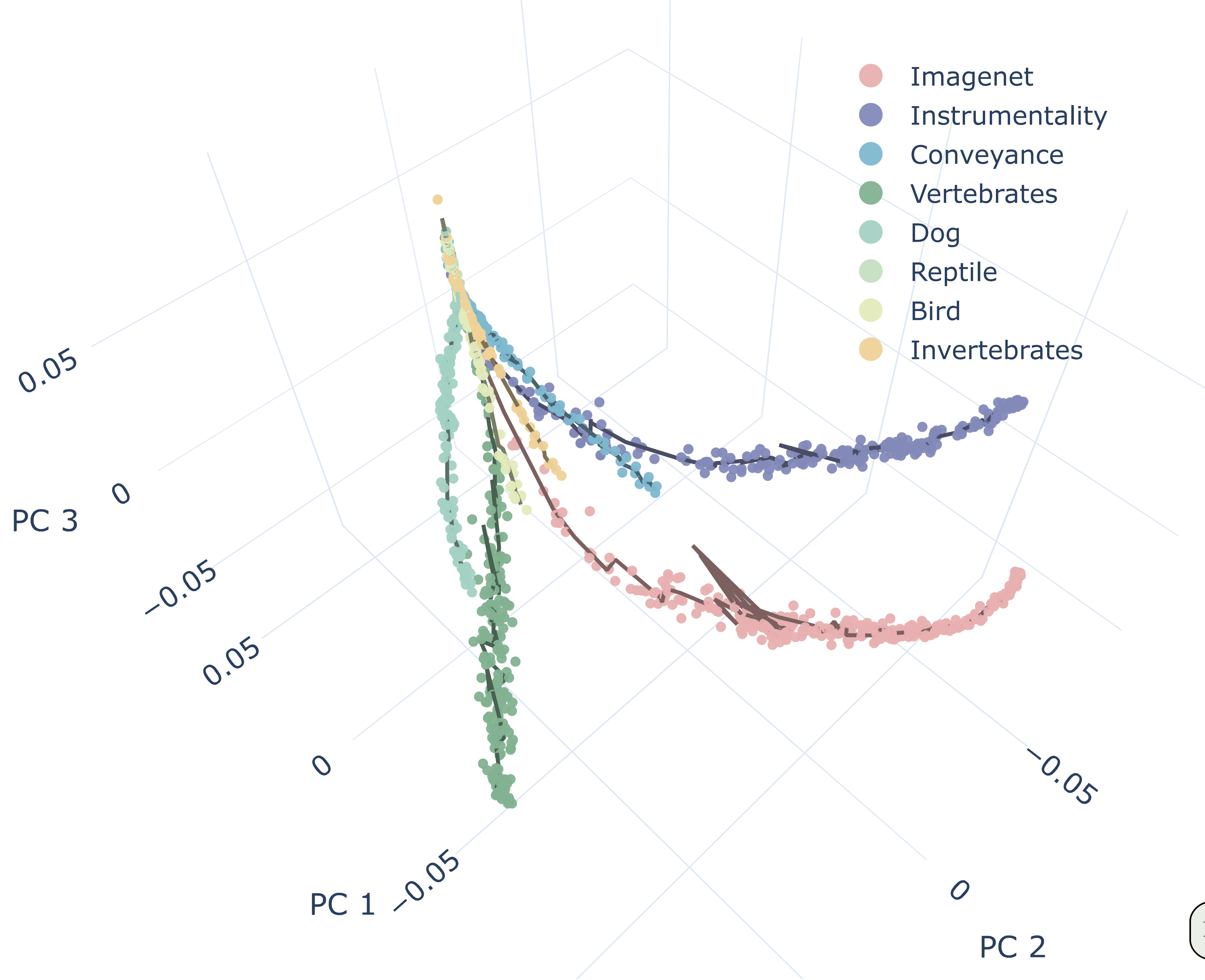


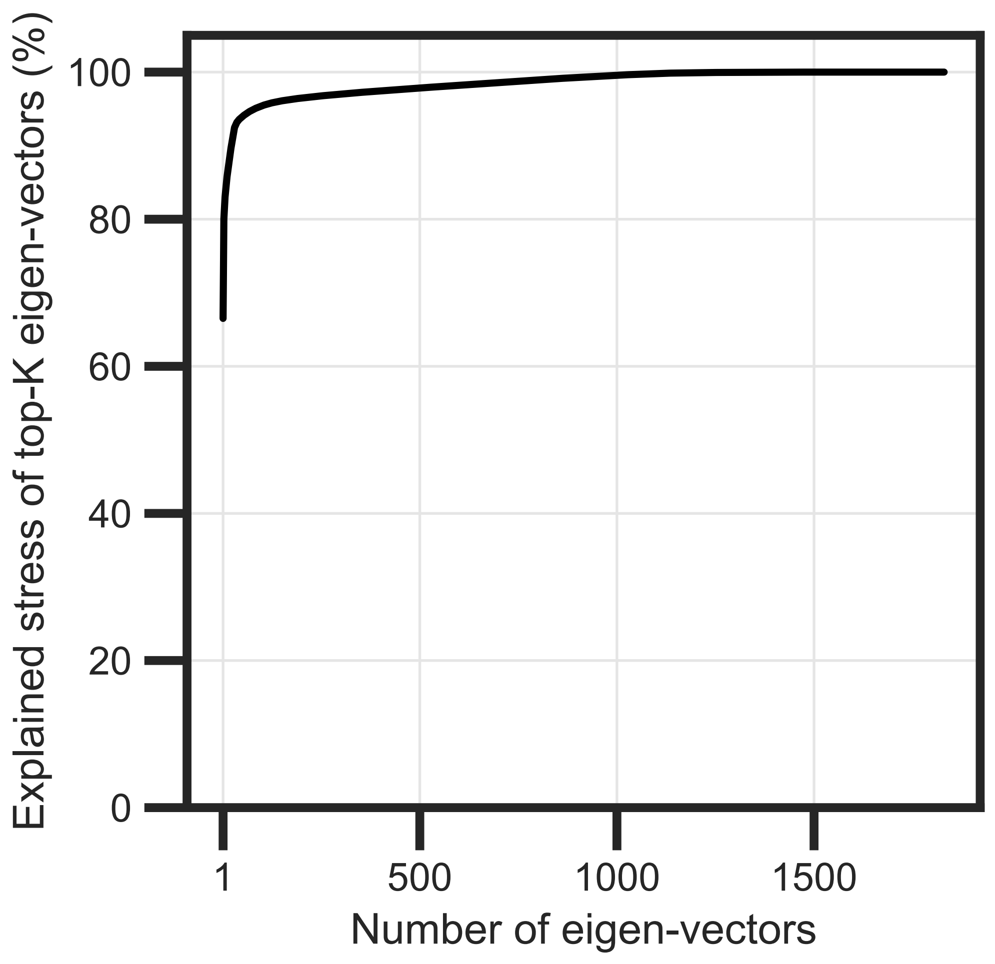
Summary: Intensive Embeddings Uncover that Neural Networks Learn in The Same Way

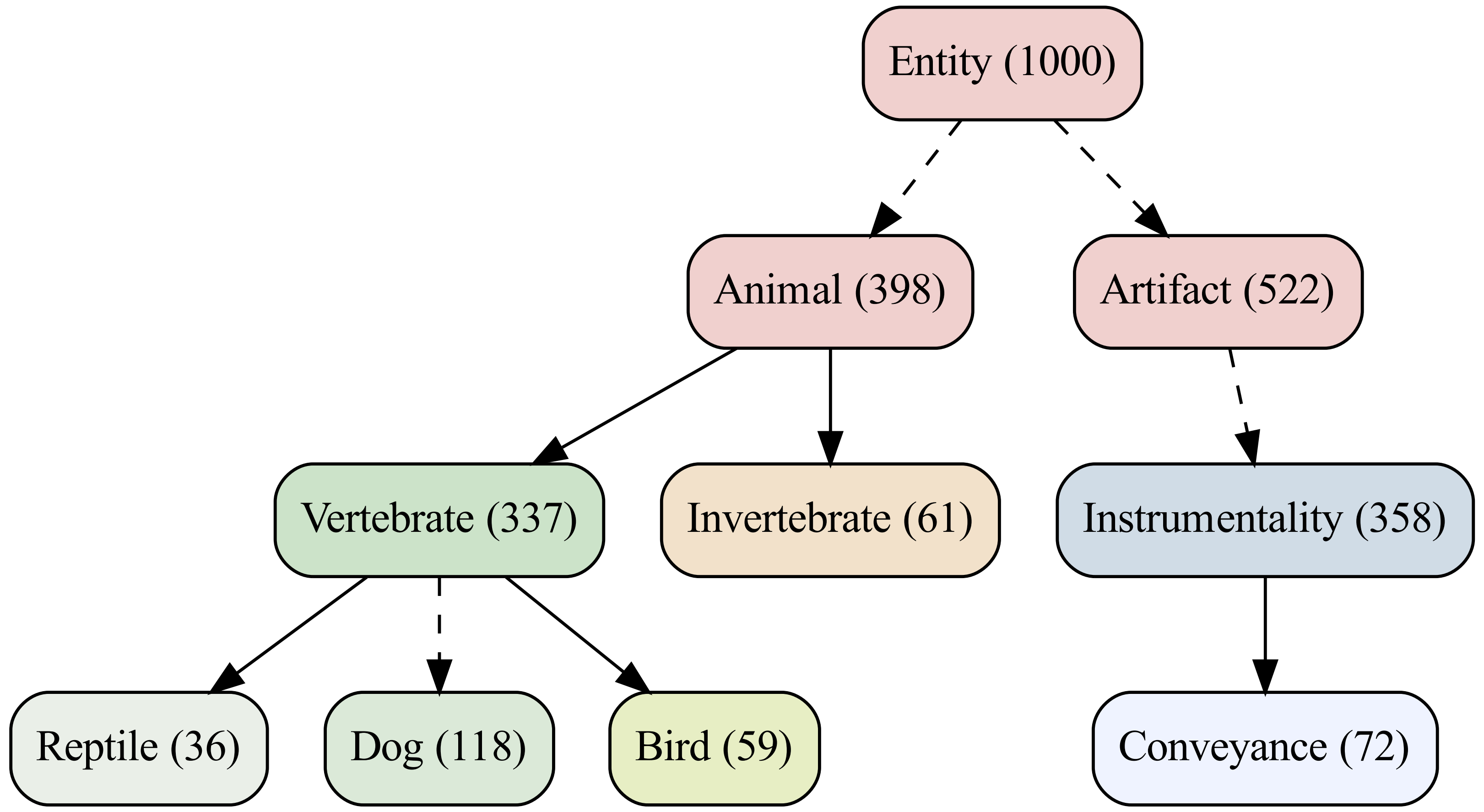
1. InPCA ⇒ Computationally feasible distance in high D.
2. Sloppiness ⇒ Low D visualization
3. The Training of Neural Networks Explores the Same Low D Manifold
4. Configuration distance ⇒ Variation in the path of learning is mostly due to architecture, not optimization technique



The Manifold of Typical Learnable Tasks is Also Low Dimensional







Ramesh et al. ICML 2023