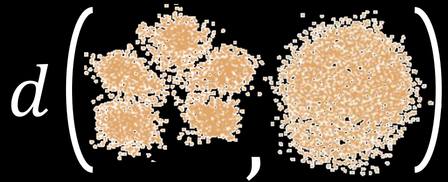
Log-Euclidean Signatures for Intrinsic Distances Between Unaligned Datasets

Tal Shnitzer[1], Mikhail Yurochkin[2], Kristjan Greenewald[2], Justin Solomon[1]



[1] CSAIL, MIT
[2] IBM Research, MIT-IBM Watson AI Lab

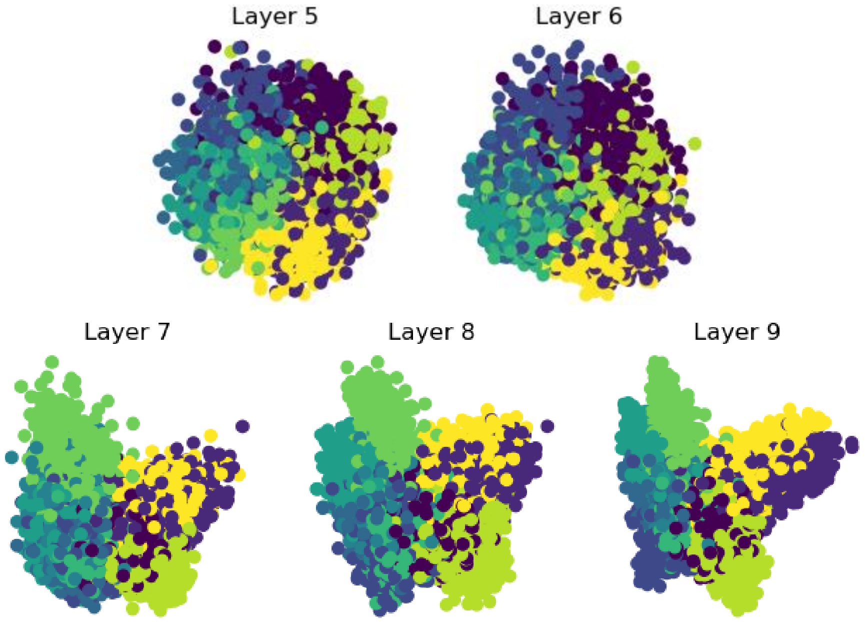
ICML 2022

Comparing Unaligned High Dimensional Point Clouds

Comparing Unaligned High Dimensional Point Clouds

Gene expression data from (Schiebinger et al., 2019)

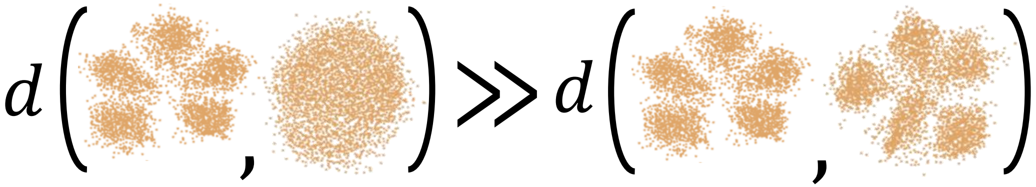
Comparing Unaligned High Dimensional Point Clouds

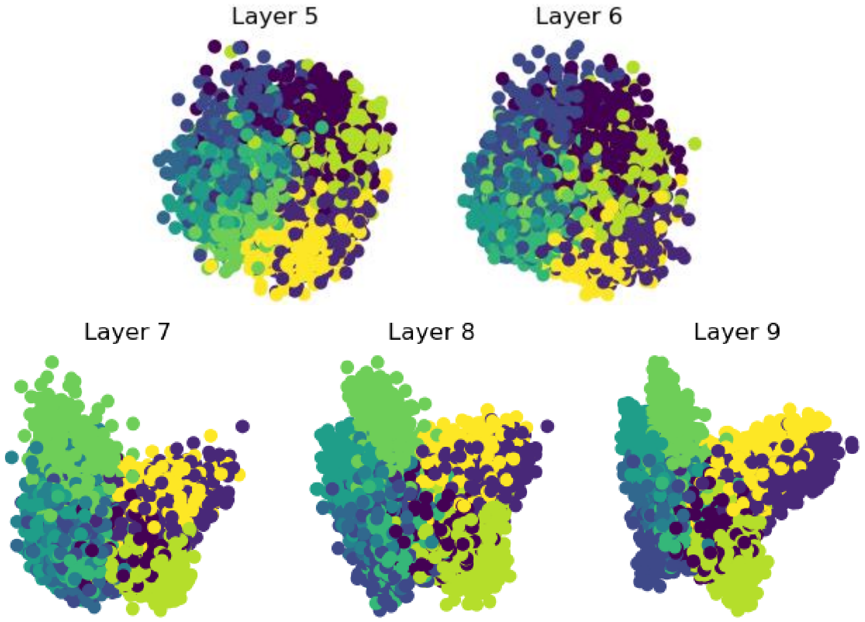


Neural networks layer embeddings

Gene expression data from (Schiebinger et al., 2019)

Comparing Unaligned High Dimensional Point Clouds



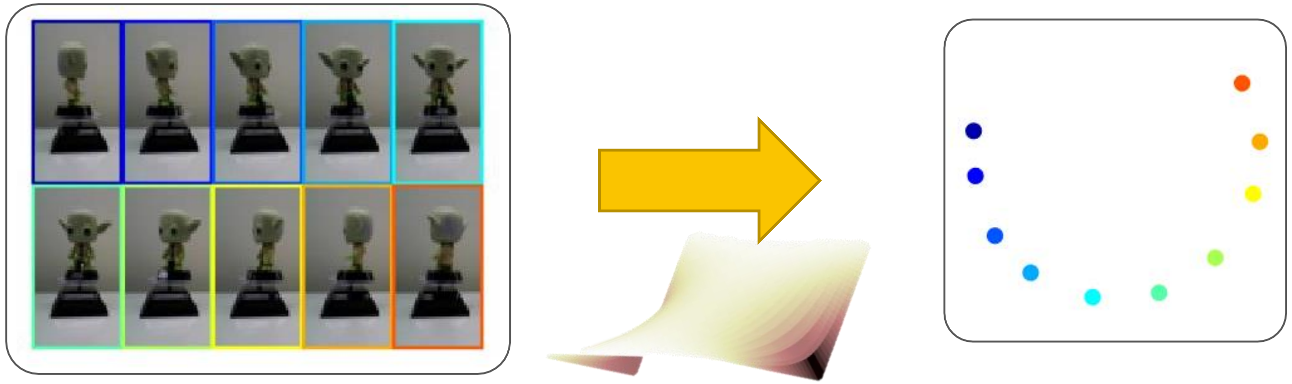


Neural networks layer embeddings

Gene expression data from (Schiebinger et al., 2019)

Representations of High Dimensional Point Clouds

- Common approach: manifold learning – representing the underlying manifold of the data.



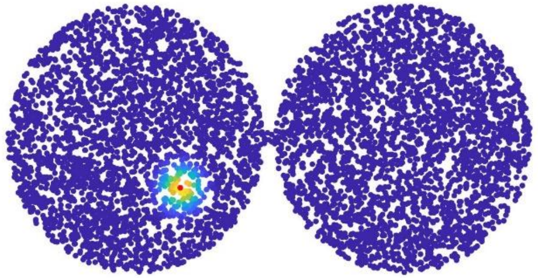
Representations of High Dimensional Point Clouds

- Common approach: manifold learning – representing the underlying manifold of the data.
- Diffusion maps (Coifman and Lafon, 2006): $\{x_i \in \mathbb{R}^d\}_{i=1}^N$

$$K(i,j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\epsilon}\right)$$

$$\widehat{K}(i,j) = \frac{K(i,j)}{\sum_\ell K(i,\ell) \cdot \sum_\ell K(\ell,j)}$$

$$W(i,j) = \frac{\widehat{K}(i,j)}{\sqrt{\sum_\ell \widehat{K}(i,\ell) \sum_\ell \widehat{K}(\ell,j)}}$$



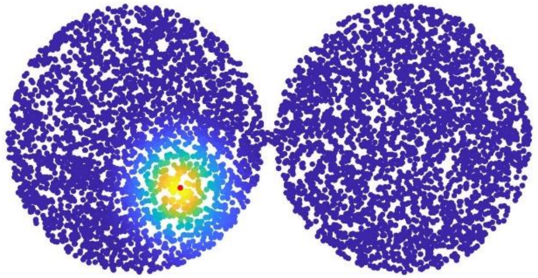
Relates to the heat kernel of the data manifold

Representations of High Dimensional Point Clouds

- Common approach: manifold learning – representing the underlying manifold of the data.
- Diffusion maps (Coifman and Lafon, 2006): $\{x_i \in \mathbb{R}^d\}_{i=1}^N$

$$K(i,j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{\epsilon}\right)$$

$$\widehat{K}(i,j) = \frac{K(i,j)}{\sum_\ell K(i,\ell) \cdot \sum_\ell K(\ell,j)}$$



$$W(i,j) = \frac{\widehat{K}(i,j)}{\sqrt{\sum_\ell \widehat{K}(i,\ell) \sum_\ell \widehat{K}(\ell,j)}}$$

← Relates to the heat kernel of the data manifold

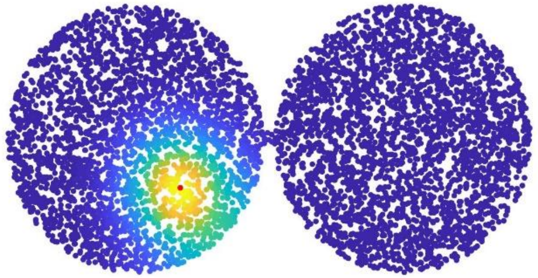
Representations of High Dimensional Point Clouds

- Common approach: manifold learning – representing the underlying manifold of the data.
- Diffusion maps (Coifman and Lafon, 2006): $\{x_i \in \mathbb{R}^d\}_{i=1}^N$

$$K(i,j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\epsilon}\right)$$

$$\widehat{K}(i,j) = \frac{K(i,j)}{\sum_\ell K(i,\ell) \cdot \sum_\ell K(\ell,j)}$$

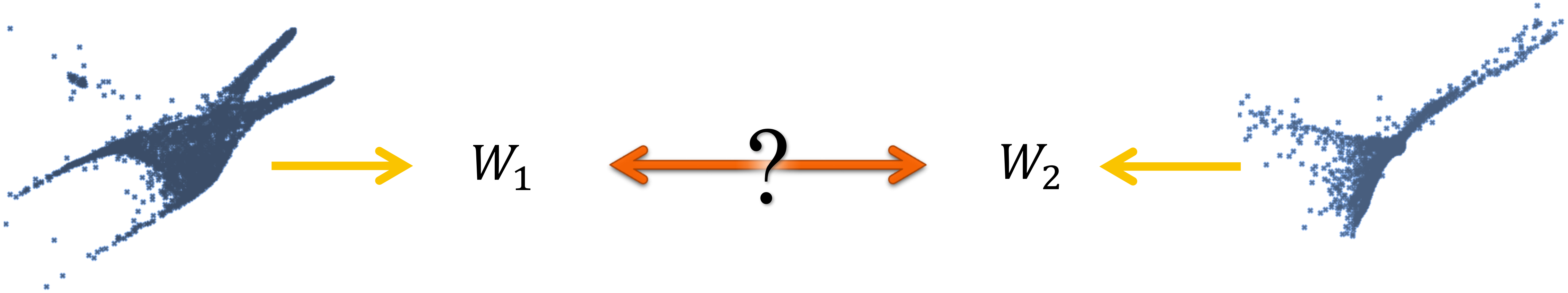
$$W(i,j) = \frac{\widehat{K}(i,j)}{\sqrt{\sum_\ell \widehat{K}(i,\ell) \sum_\ell \widehat{K}(\ell,j)}}$$



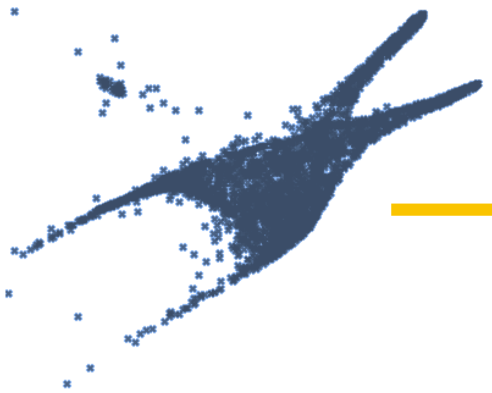
← Relates to the heat kernel of the data manifold

Comparing the Representations

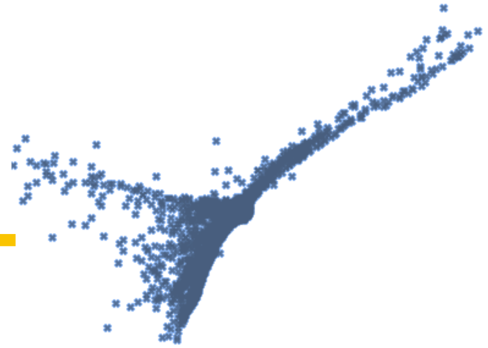
Comparing the Representations



Comparing the Representations

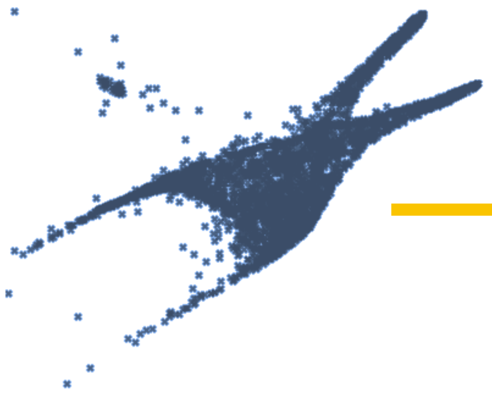


W_1 ? W_2

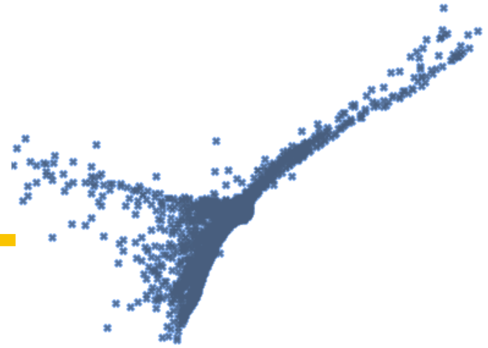


$W_1 - W_2$

Comparing the Representations



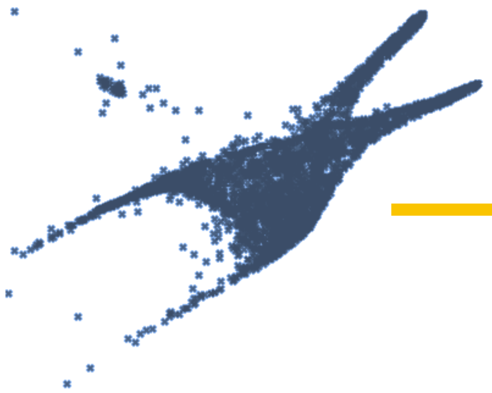
W_1 ⟷ ? ⟷ W_2



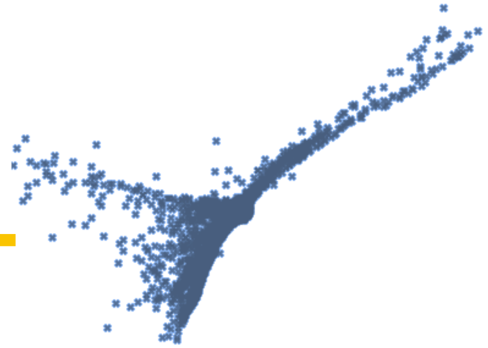
$W_1 - W_2$

$W_1 W_2 - W_2 W_1$

Comparing the Representations



W_1 ⟷ ? ⟷ W_2



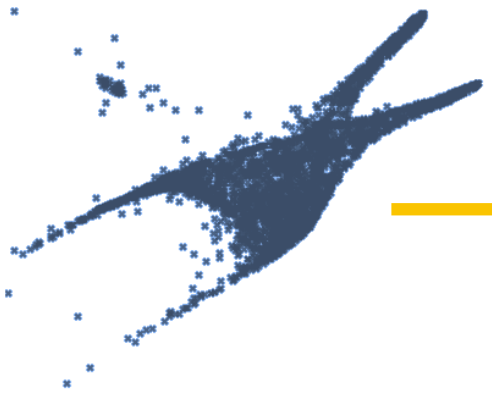
$W_1 - W_2$

$W_1 W_2 - W_2 W_1$

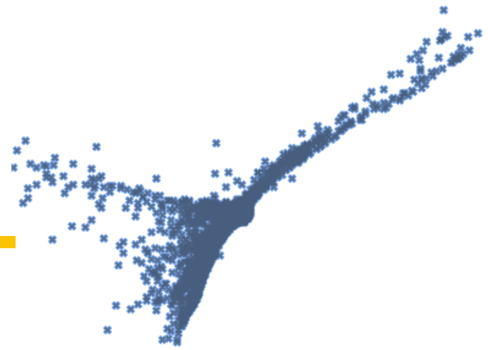
$Log_{W_1} W_2$

⋮

Comparing the Representations



W_1 ⟷ ? ⟷ W_2



$W_1 - W_2$

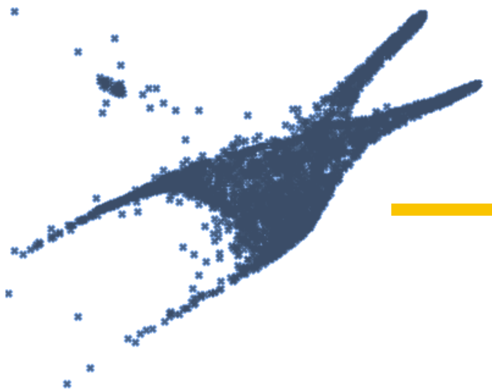
$W_1 W_2 - W_2 W_1$

$Log_{W_1} W_2$

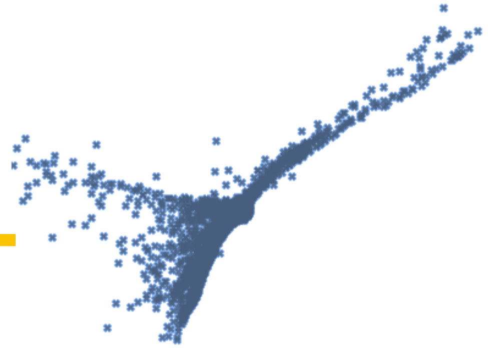
⋮

Require costly pointwise alignment

Comparing the Representations



W_1 ⟷ ? ⟷ W_2



$W_1 - W_2$

$W_1 W_2 - W_2 W_1$

$Log_{W_1} W_2$

⋮

~~*Require costly pointwise alignment*~~

Compare spectral properties!

Comparing *Unaligned* Representations

$$d_{LES}^{2}(W_1, W_2) = \sum_{i=1}^{K} \left(\log\left(\lambda_i^{(W_1)} + \gamma\right) - \log\left(\lambda_i^{(W_2)} + \gamma\right) \right)^2$$

$K \ll N$

Comparing *Unaligned* Representations

- Accounting for the symmetric positive definiteness of W_ℓ, we use the Log-Euclidean metric (Arsigny et al., 2006).

$$d^2_{LES}(W_1, W_2) = \sum_{i=1}^{K} \left(\log\left(\lambda_i^{(W_1)} + \gamma\right) - \log\left(\lambda_i^{(W_2)} + \gamma\right) \right)^2$$

$K \ll N$

Comparing *Unaligned* Representations

- Accounting for the symmetric positive definiteness of W_ℓ, we use the Log-Euclidean metric (Arsigny et al., 2006).
- Define a pseudo-metric by lower bounding the Log-Euclidean metric:

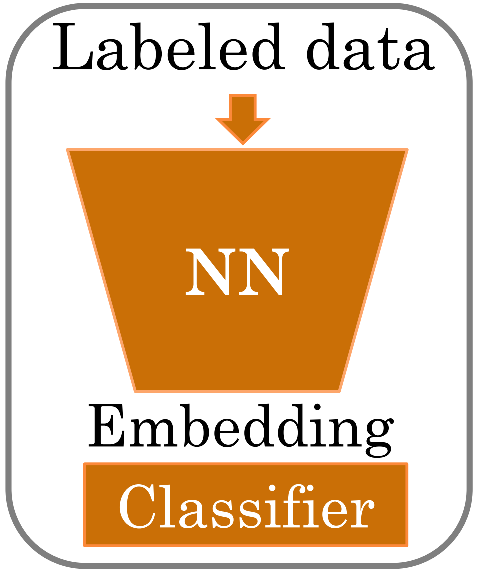
$$\|\log(W_1) - \log(W_2)\|_F^2 \geq \sum_i \left(\log \lambda_i^{(W_1)\downarrow} - \log \lambda_i^{(W_2)\downarrow}\right)^2$$

$$d_{LES}^2(W_1, W_2) = \sum_{i=1}^{K} \left(\log\left(\lambda_i^{(W_1)} + \gamma\right) - \log\left(\lambda_i^{(W_2)} + \gamma\right)\right)^2$$

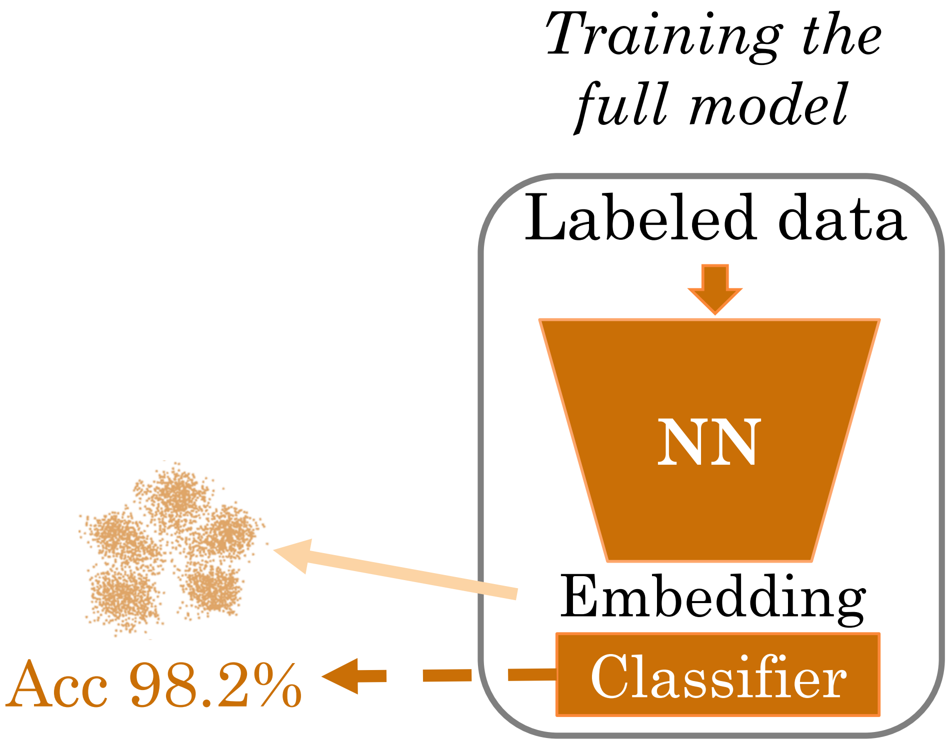
$K \ll N$

Predicting Success of NN Embedding in Few-Shot Learning

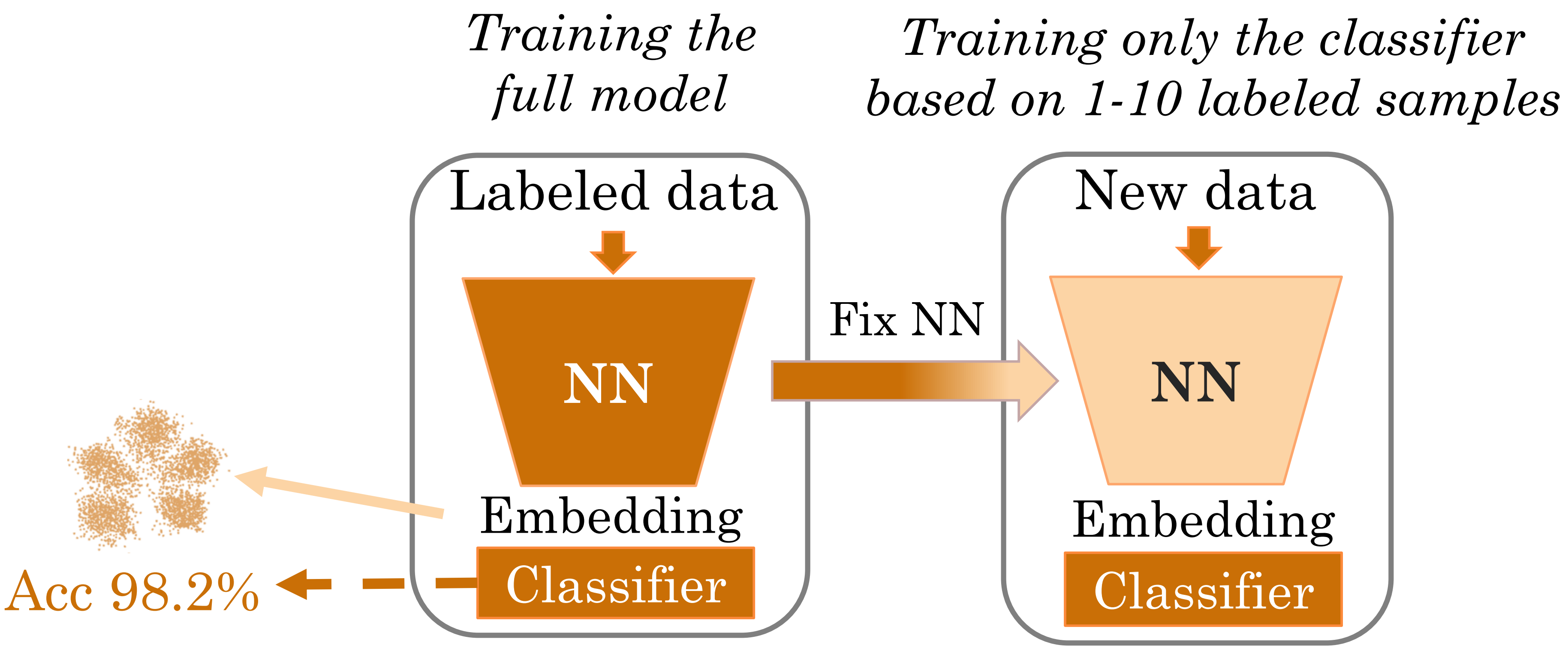
Training the full model



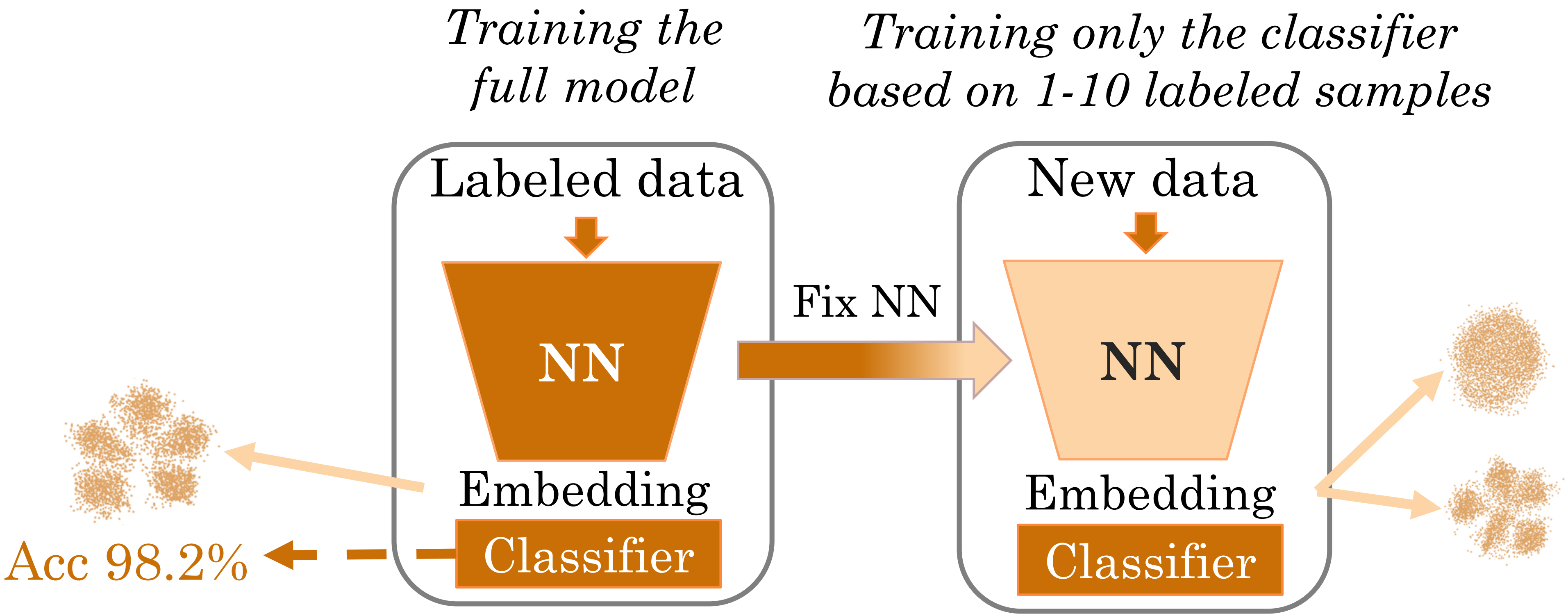
Predicting Success of NN Embedding in Few-Shot Learning



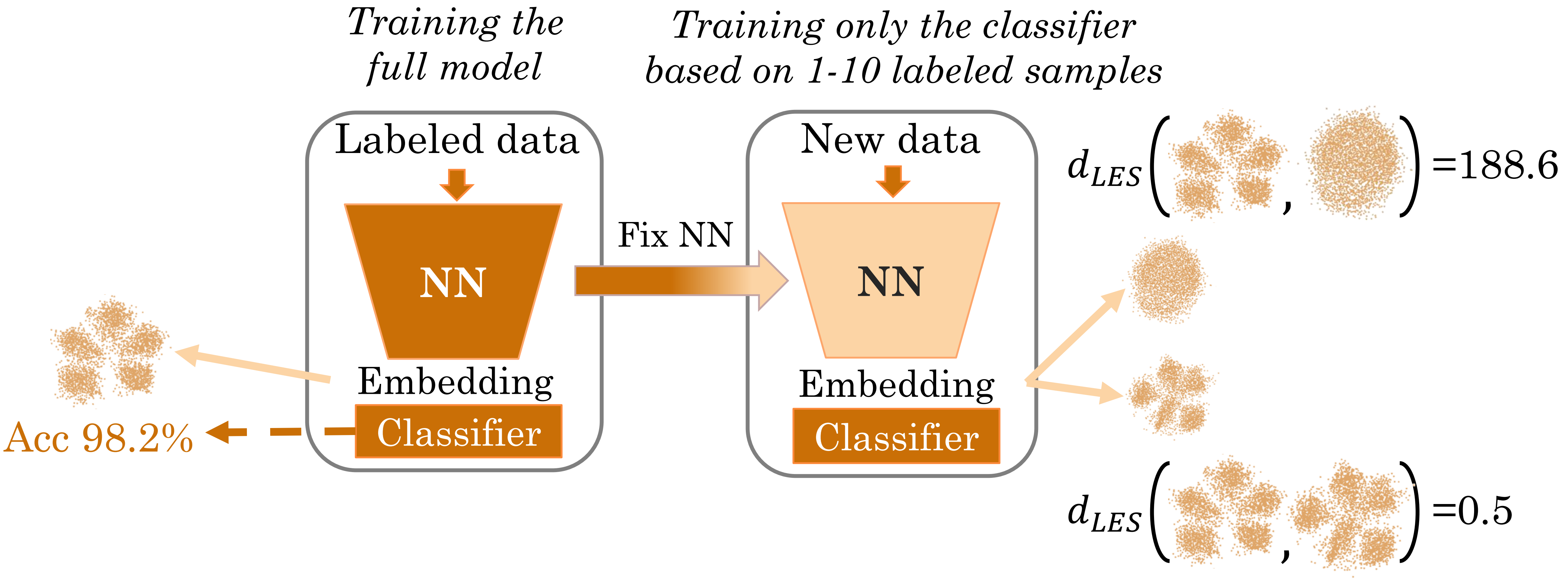
Predicting Success of NN Embedding in Few-Shot Learning



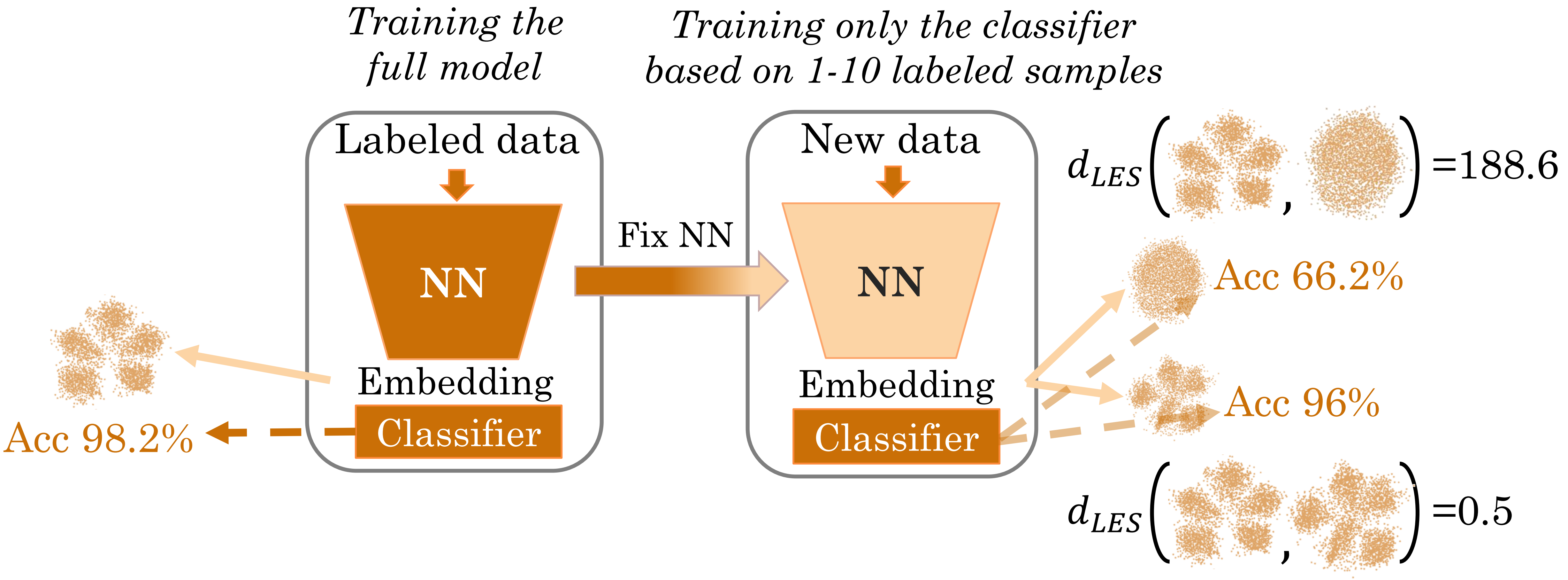
Predicting Success of NN Embedding in Few-Shot Learning



Predicting Success of NN Embedding in Few-Shot Learning

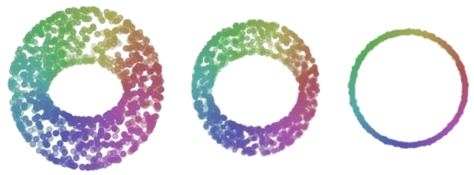


Predicting Success of NN Embedding in Few-Shot Learning



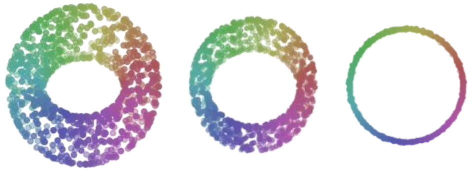
Additional Applications

- Geometric shape comparison



Additional Applications

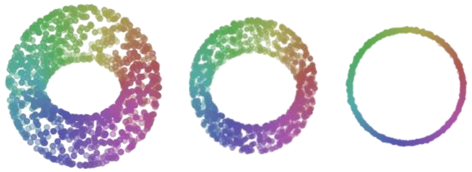
- Geometric shape comparison



- NN layer embedding analysis

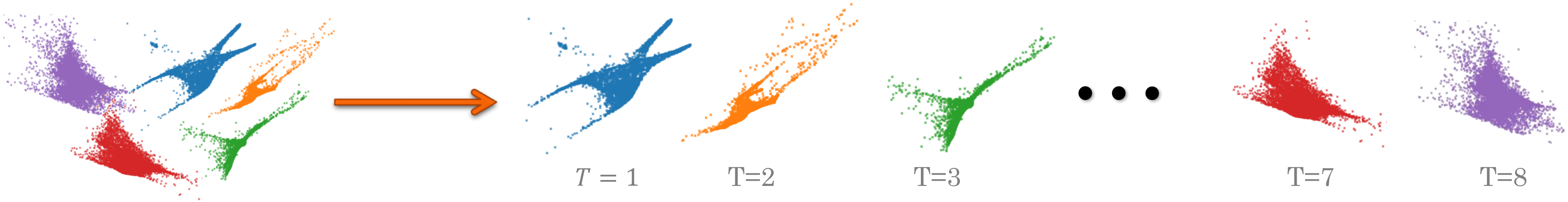
Additional Applications

- Geometric shape comparison



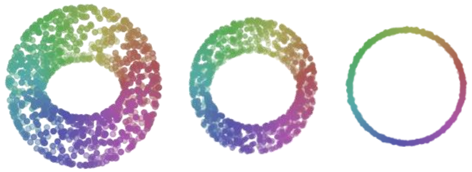
- NN layer embedding analysis

- Evaluation of gene-expression data



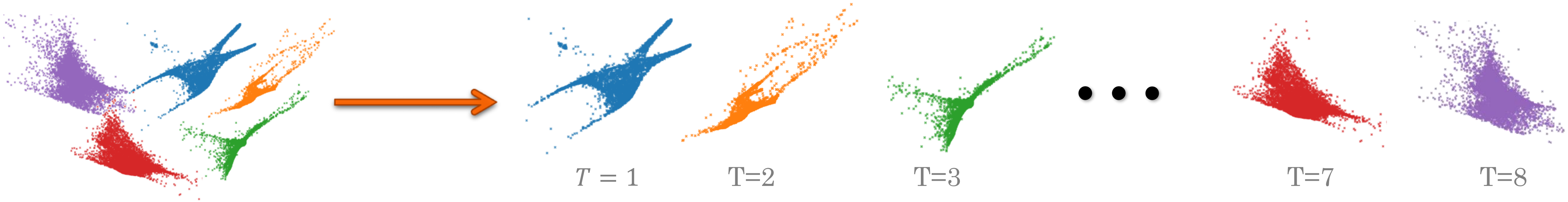
Additional Applications

- Geometric shape comparison



- NN layer embedding analysis

- Evaluation of gene-expression data



Thank You!