TSBP: Tangent Space Belief Propagation for Manifold Learning

Thomas Cohn, Odest Chadwicke Jenkins, Karthik Desingh, Zhen Zeng

IROS 2020

Laboratory for
Perception **RO**botics and **G**rounded **RE**asoning **S**ystem**S**

UNIVERSITY OF MICHIGAN

Robot Sensing Today

Robots now have access to increasingly detailed sensor data

- High resolution images

Robot Sensing Today

Robots now have access to increasingly detailed sensor data

- High resolution images
- Dense point clouds

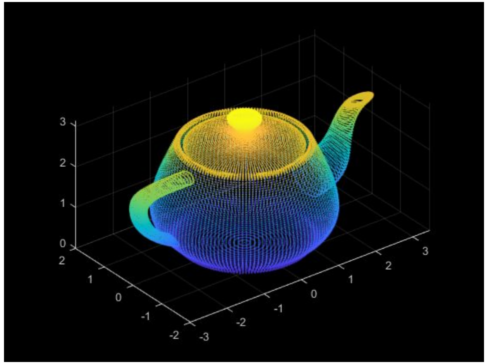


IMAGE: MathWorks

Robot Sensing Today

Robots now have access to increasingly detailed sensor data

- High resolution images
- Dense point clouds
- Motion capture data

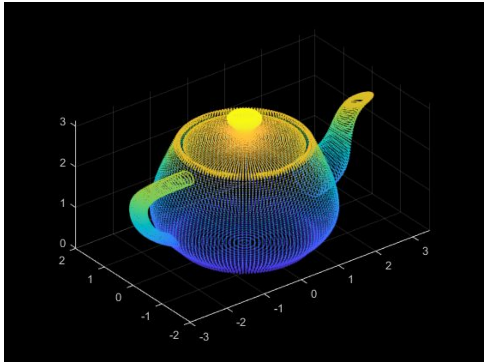


IMAGE: MathWorks

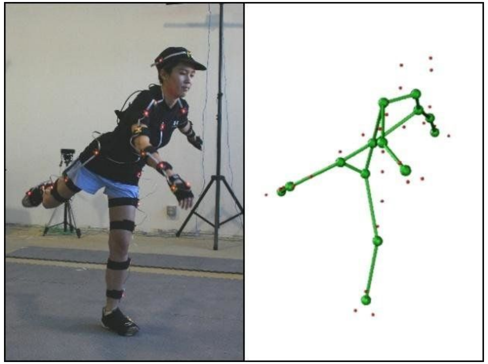


IMAGE: *3D motion capture by computer vision and virtual rendering* (D Jáuregui)

Robot Sensing Today

Robots now have access to increasingly detailed sensor data

- High resolution images
- Dense point clouds
- Motion capture data
- Tactile information

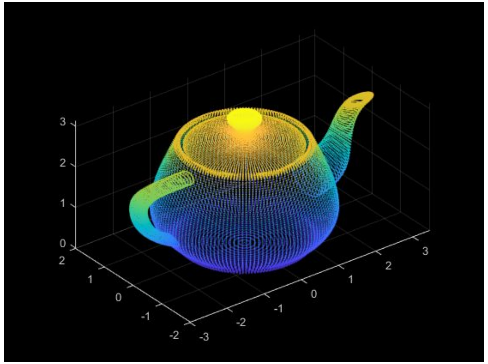


IMAGE: MathWorks

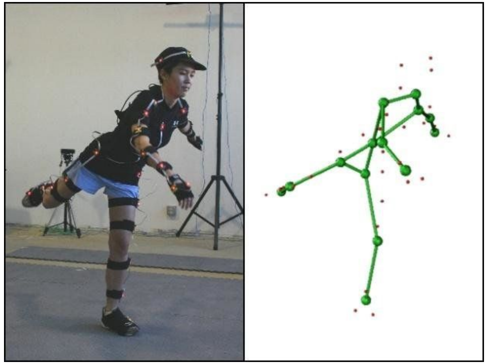


IMAGE: *3D motion capture by computer vision and virtual rendering* (D Jáuregui)

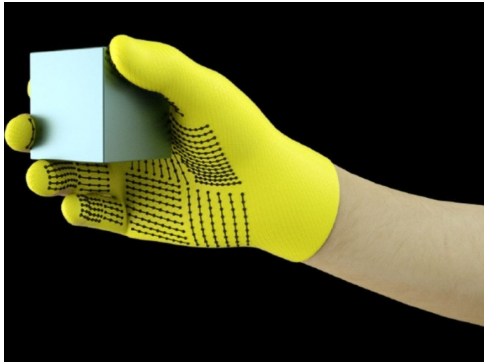


IMAGE: *Learning the signatures of the human grasp using a scalable tactile glove* (S Sundaram et al)

The Challenges of High Dimensional Data

- Some problems and algorithms quickly become intractable
    - Partially Observable Markov Decision Processes (POMDPs)
    - Particle Filters
    - Reinforcement Learning

The Challenges of High Dimensional Data

- Some problems and algorithms quickly become intractable
  - Partially Observable Markov Decision Processes (POMDPs)
  - Particle Filters
  - Reinforcement Learning
- Interfacing with humans
  - Data visualization
  - Robot teleoperation

The Challenges of High Dimensional Data

- Some problems and algorithms quickly become intractable
  - Partially Observable Markov Decision Processes (POMDPs)
  - Particle Filters
  - Reinforcement Learning
- Interfacing with humans
  - Data visualization
  - Robot teleoperation
- Data sparsity grows exponentially with dimension
- Hughes Phenomenon: more features can harm accuracy
- Handling bias-variance tradeoff
- Outlier detection

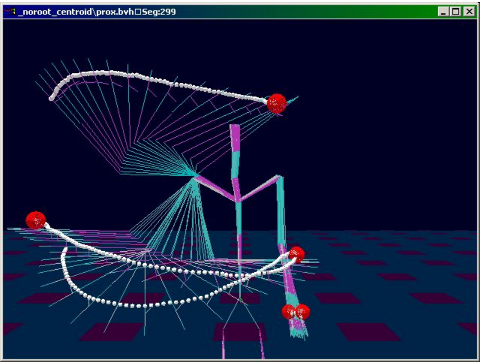
Dimensionality Reduction to the Rescue

Reduce the dimension of the data while preserving latent information

Dimensionality Reduction to the Rescue

Reduce the dimension of the data while preserving latent information

1. *A Spatio-temporal Extension to Isomap Nonlinear Dimension Reduction* (OC Jenkins et al)

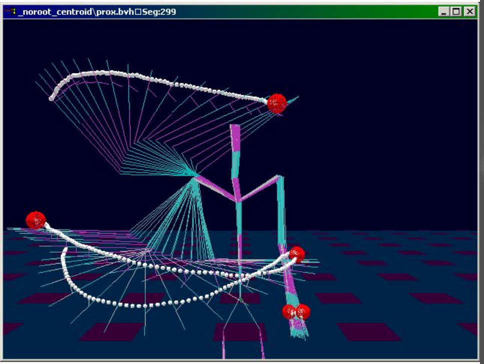


1. Human Motion Capture Data

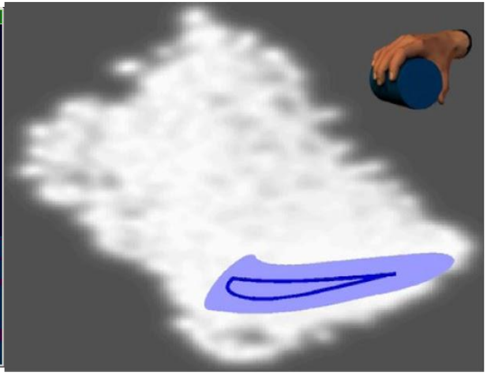
Dimensionality Reduction to the Rescue

Reduce the dimension of the data while preserving latent information

1. *A Spatio-temporal Extension to Isomap Nonlinear Dimension Reduction* (OC Jenkins et al)
2. *Extracting Postural Synergies for Robotic Grasping* (J Romero et al)



1. Human Motion Capture Data

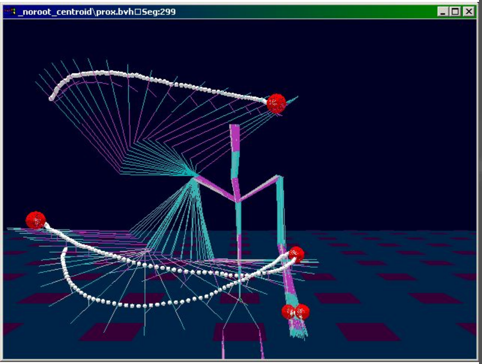


2. Grasp Trajectory

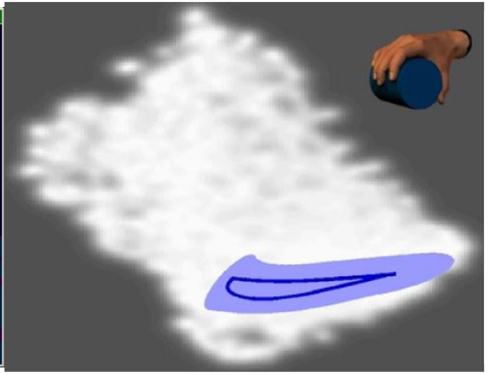
Dimensionality Reduction to the Rescue

Reduce the dimension of the data while preserving latent information

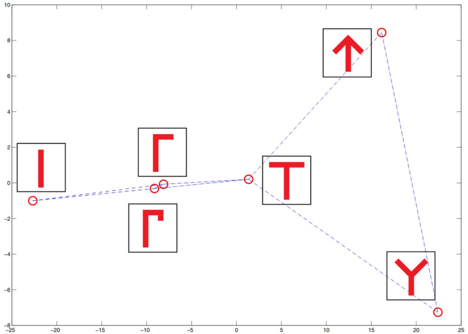
1. *A Spatio-temporal Extension to Isomap Nonlinear Dimension Reduction* (OC Jenkins et al)
2. *Extracting Postural Synergies for Robotic Grasping* (J Romero et al)
3. *Detecting the Functional Similarities Between Tools Using a Hierarchical Representation of Outcomes* (J Sinapov et al)



1. Human Motion Capture Data



2. Grasp Trajectory

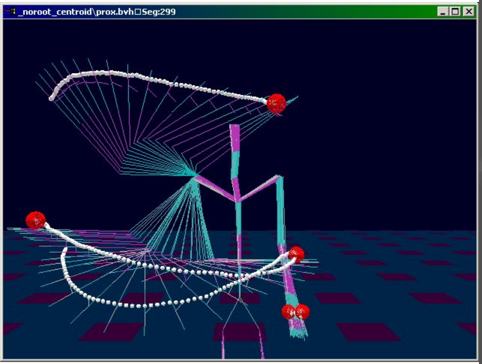


3. Tool Shape Similarities

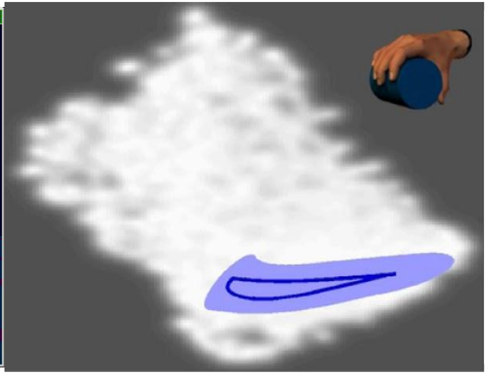
Dimensionality Reduction to the Rescue

Reduce the dimension of the data while preserving latent information

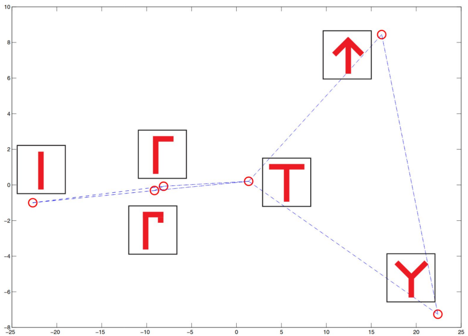
1. *A Spatio-temporal Extension to Isomap Nonlinear Dimension Reduction* (OC Jenkins et al)
2. *Extracting Postural Synergies for Robotic Grasping* (J Romero et al)
3. *Detecting the Functional Similarities Between Tools Using a Hierarchical Representation of Outcomes* (J Sinapov et al)
4. *Learning Kinematic Models for Articulated Objects* (J Sturm et al)



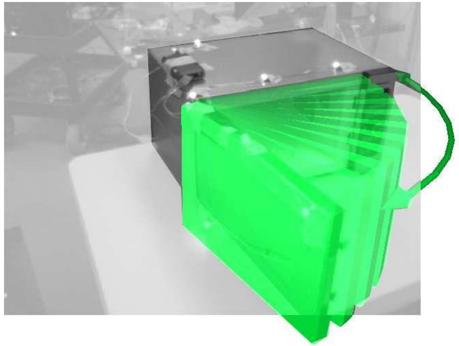
1. Human Motion Capture Data



2. Grasp Trajectory



3. Tool Shape Similarities



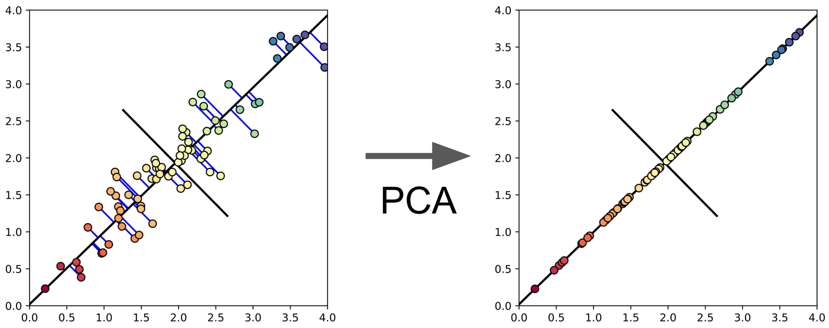
4. Learned Kinematic Model

Dimensionality Reduction

- Transform data to a lower-dimensional space
- Preserve data similarities/dissimilarities

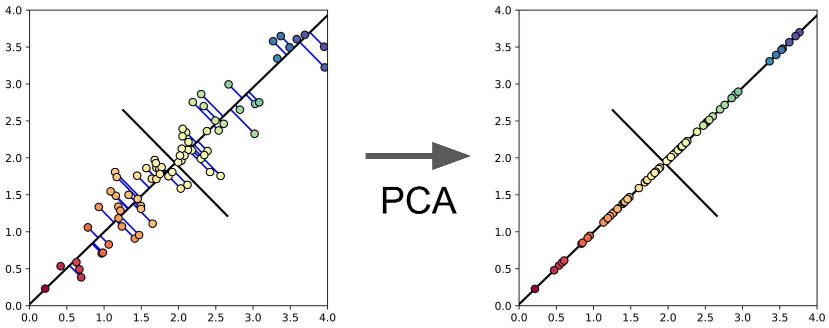
Dimensionality Reduction

- Transform data to a lower-dimensional space
- Preserve data similarities/dissimilarities



Dimensionality Reduction

- Transform data to a lower-dimensional space
- Preserve data similarities/dissimilarities



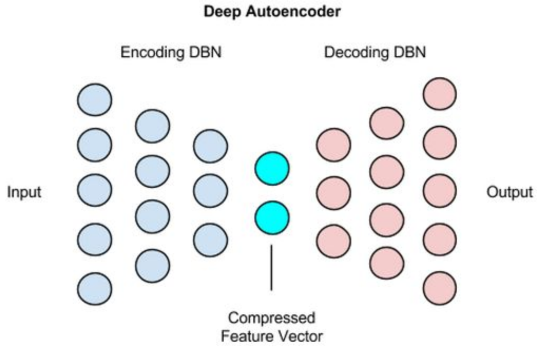
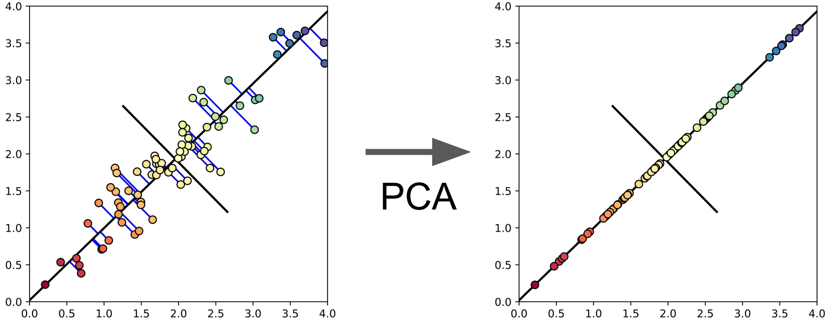


IMAGE: Pathmind

Dimensionality Reduction

- Transform data to a lower-dimensional space
- Preserve data similarities/dissimilarities



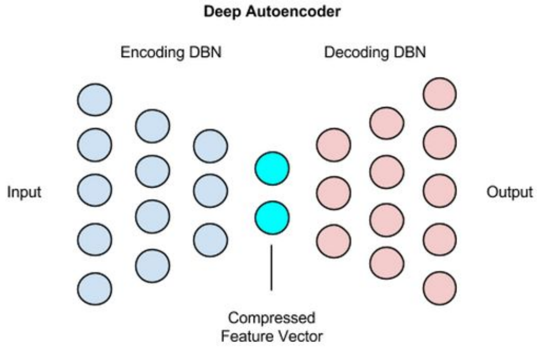


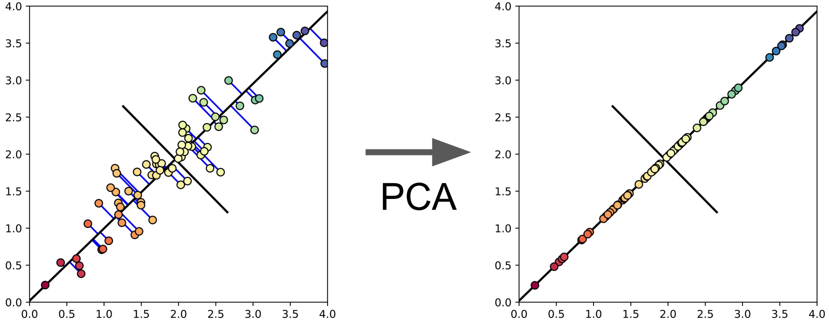
IMAGE: Pathmind

Manifold Learning

- Model data as lying along a manifold
- Learn a low-dimensional embedding

Dimensionality Reduction

- Transform data to a lower-dimensional space
- Preserve data similarities/dissimilarities



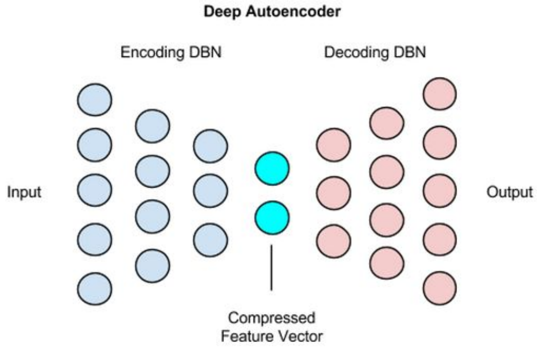


IMAGE: Pathmind

Manifold Learning

- Model data as lying along a manifold
- Learn a low-dimensional embedding

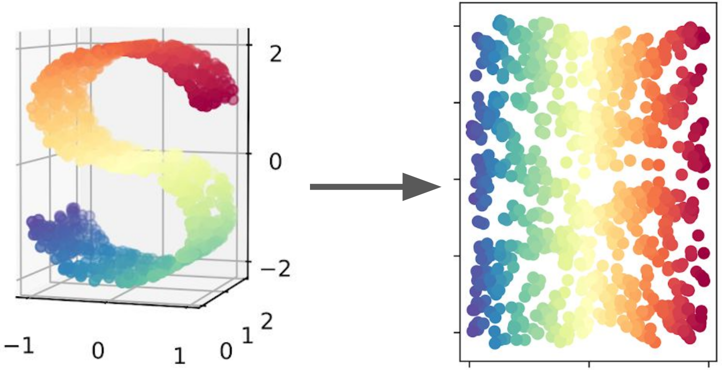


IMAGE: scikit-learn

Nearest-Neighbors

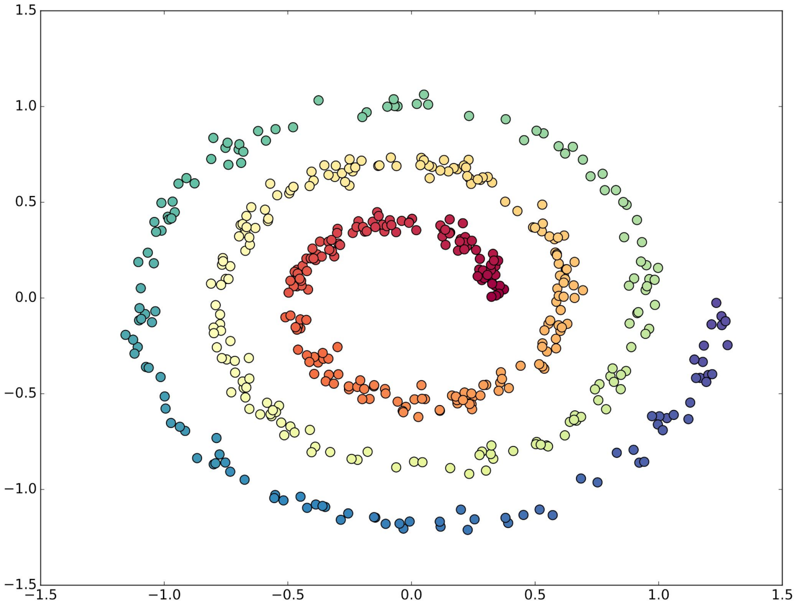
- Common tool to discover local structure
- Used as a starting point by many manifold learning algorithms
  - Isometric Mapping (ISOMAP)
  - Locally Linear Embedding (LLE)
  - Laplacian Eigenmaps (LE)
  - Local Tangent Space Alignment (LTSA)

Nearest-neighbors

- Can be "short-circuited" by false edges, caused by
  - Noisy or sparse data
  - Distinct regions of the manifold passing close to one another
- Makes points falsely appear highly similar

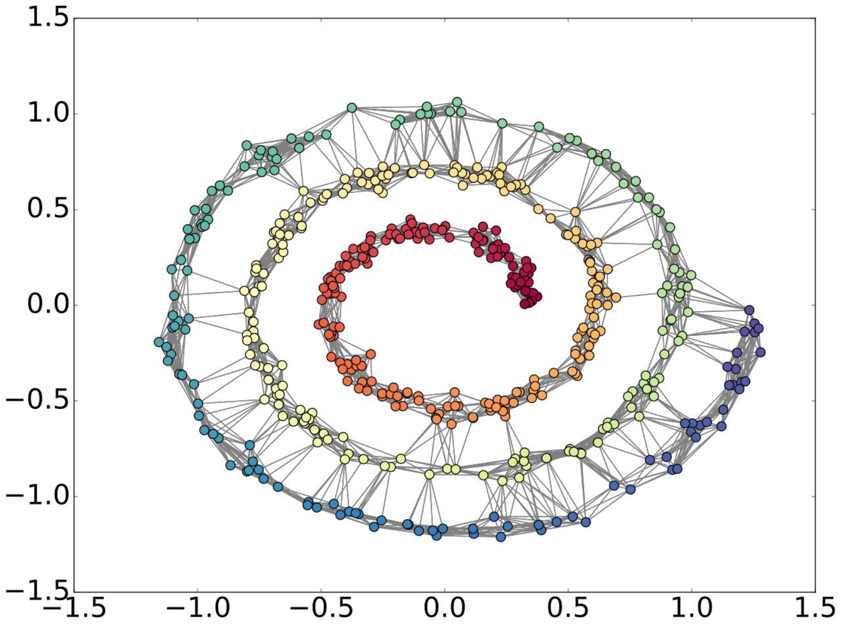
Nearest-neighbors

- Can be "short-circuited" by false edges, caused by
  - Noisy or sparse data
  - Distinct regions of the manifold passing close to one another
- Makes points falsely appear highly similar



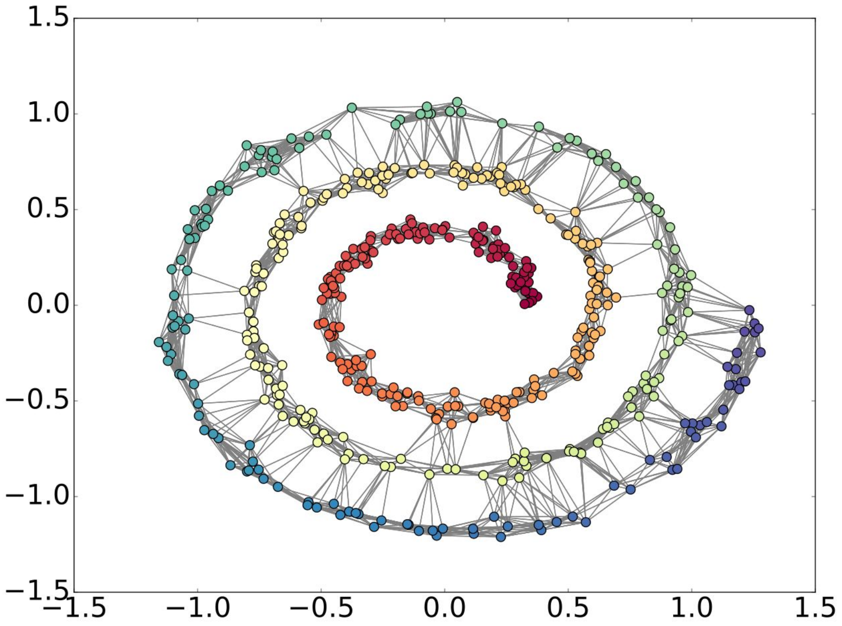
Nearest-neighbors

- Can be "short-circuited" by false edges, caused by
  - Noisy or sparse data
  - Distinct regions of the manifold passing close to one another
- Makes points falsely appear highly similar



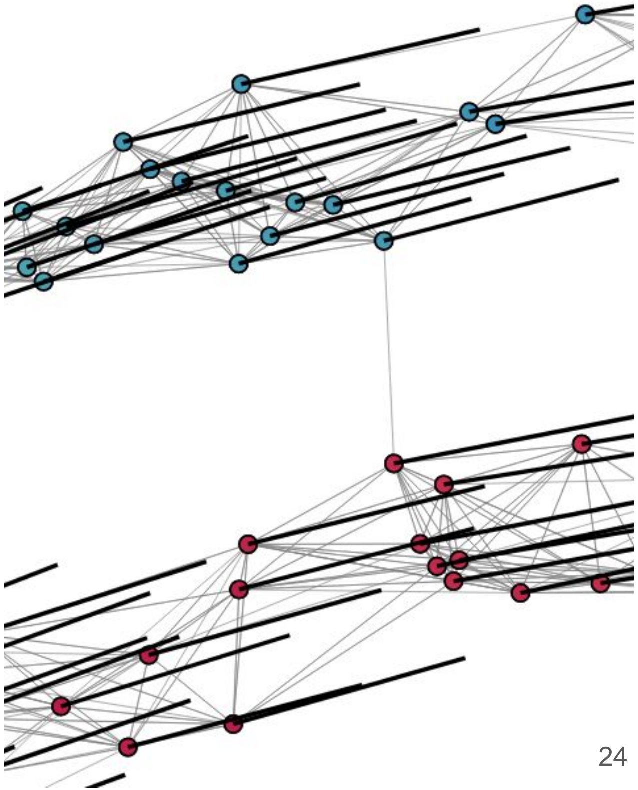
Nearest-neighbors

- Can be "short-circuited" by false edges, caused by
  - Noisy or sparse data
  - Distinct regions of the manifold passing close to one another
- Makes points falsely appear highly similar
- Want to remove false edges



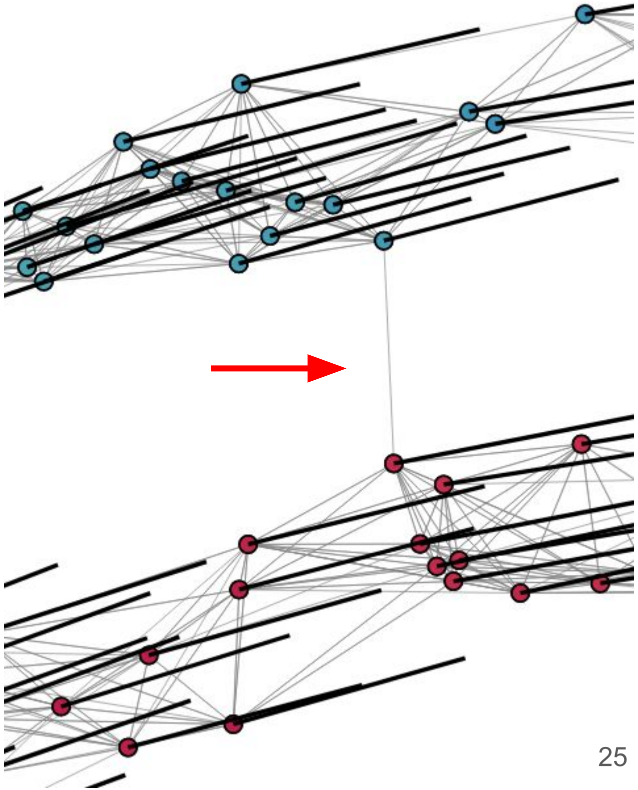
Key Observations from Manifold Geometry

- Red and blue points are from different regions of the manifold



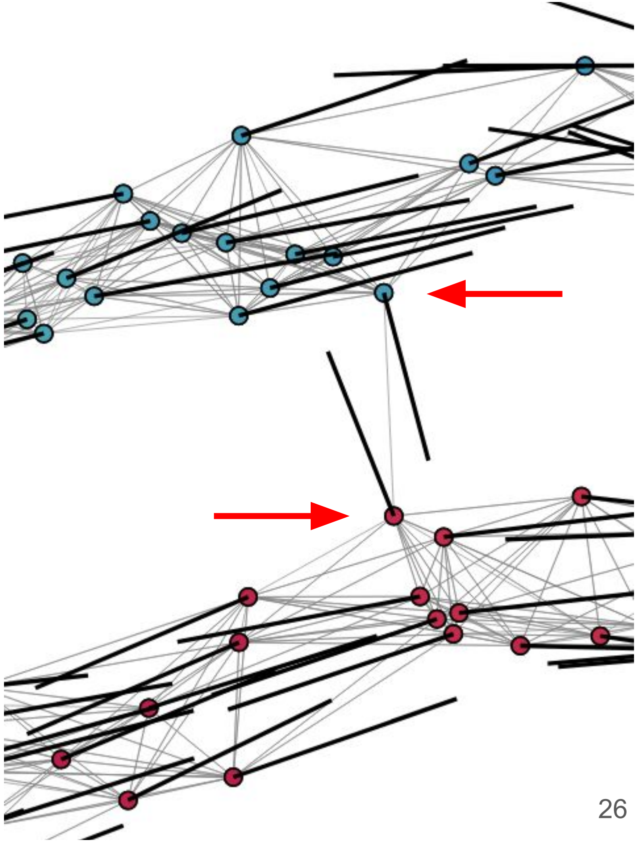
Key Observations from Manifold Geometry

- Red and blue points are from different regions of the manifold
- False edges disagree with the tangent spaces
  - Tangent spaces (thick, black lines) tell us which edges to remove



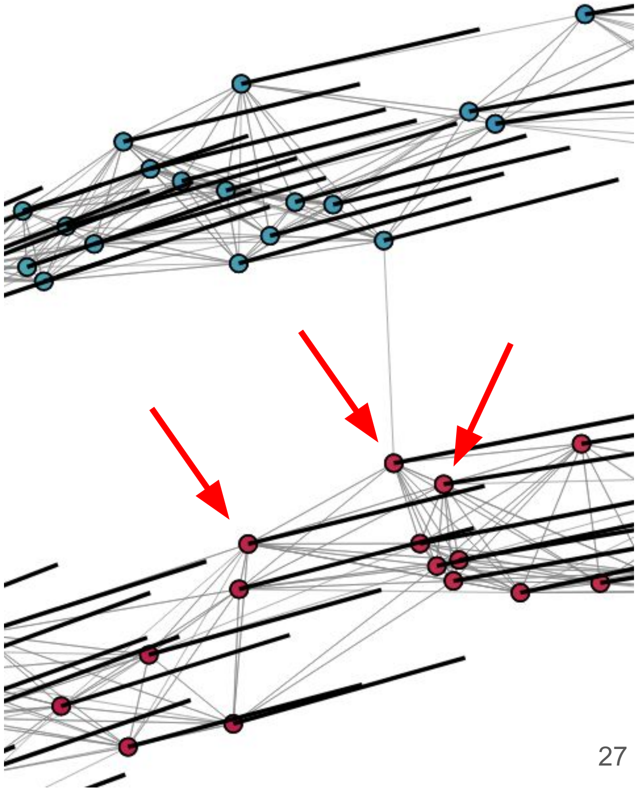
Key Observations from Manifold Geometry

- Red and blue points are from different regions of the manifold
- False edges disagree with the tangent spaces
  - Tangent spaces (thick, black lines) tell us which edges to remove
- Problem: poor local neighborhoods skew tangent space estimates



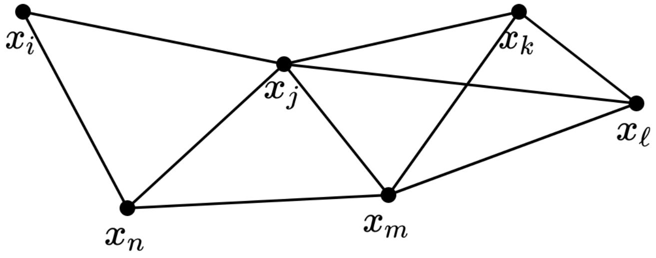
Key Observations from Manifold Geometry

- Red and blue points are from different regions of the manifold
- False edges disagree with the tangent spaces
  - Tangent spaces (thick, black lines) tell us which edges to remove
- Problem: poor local neighborhoods skew tangent space estimates
- Nearby points have similar tangent spaces
  - Can we use this to obtain a better estimate?

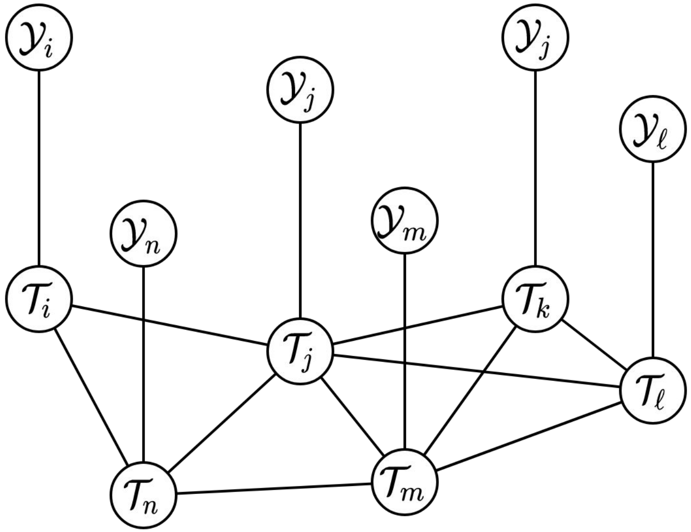


Tangent Space Estimation via Belief Propagation

- Formulate as inference on a Markov Random Field (MRF)
- For each point
  - Latent variable represents its tangent space
  - Observation from PCA on local neighborhood

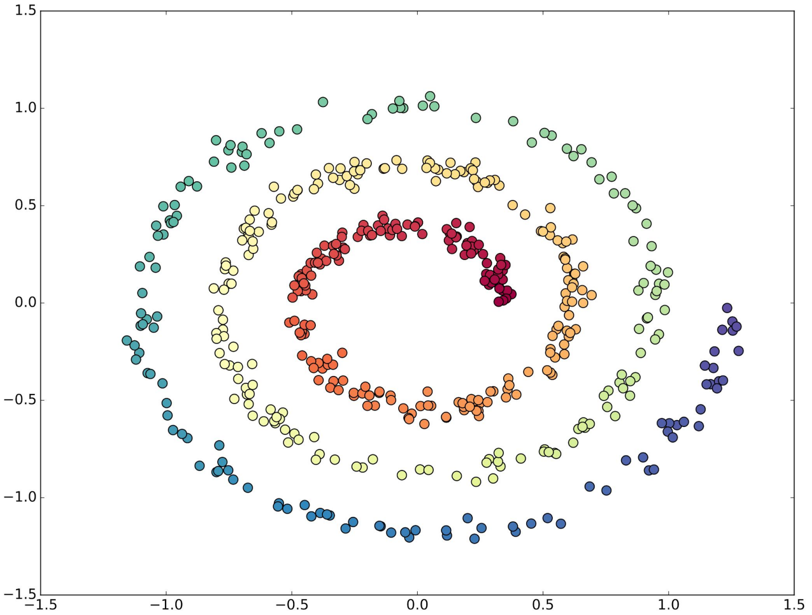


Neighborhood Graph

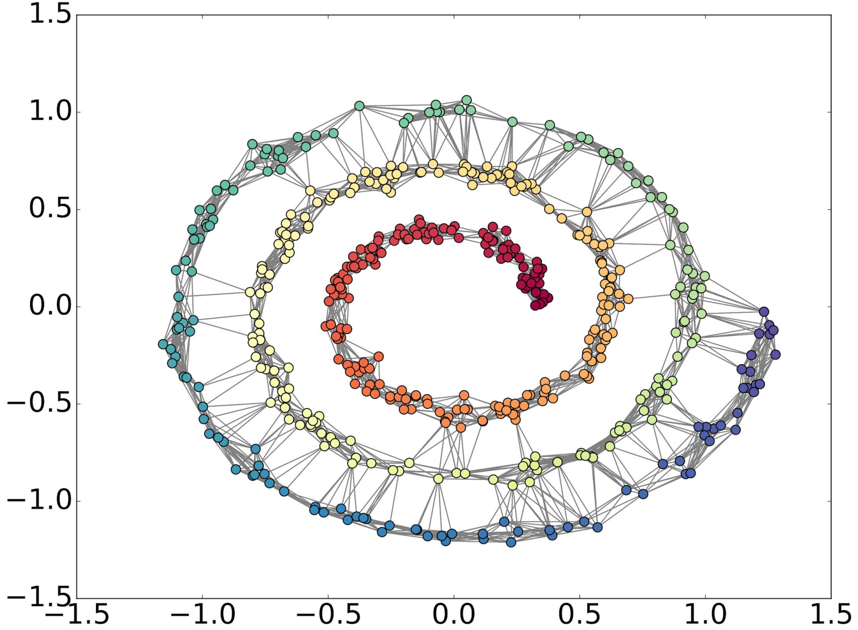


MRF

Spiral Experiment

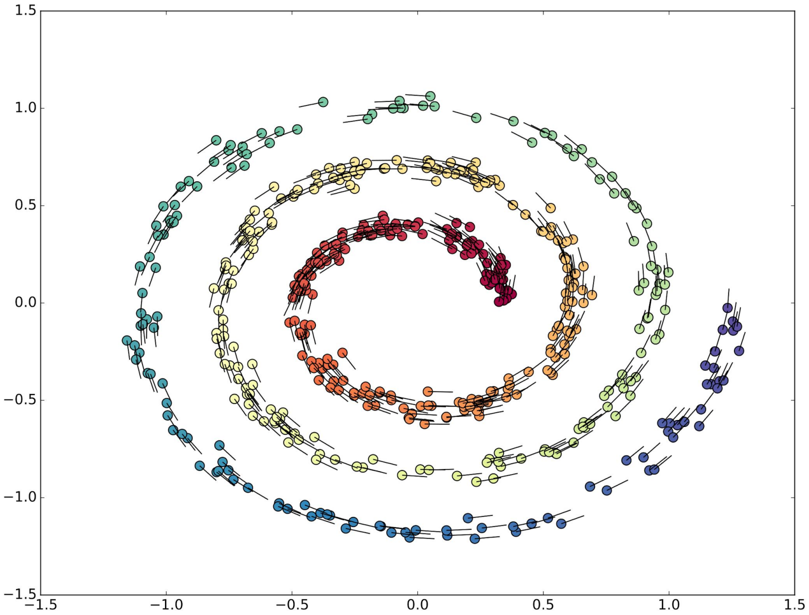


Dataset

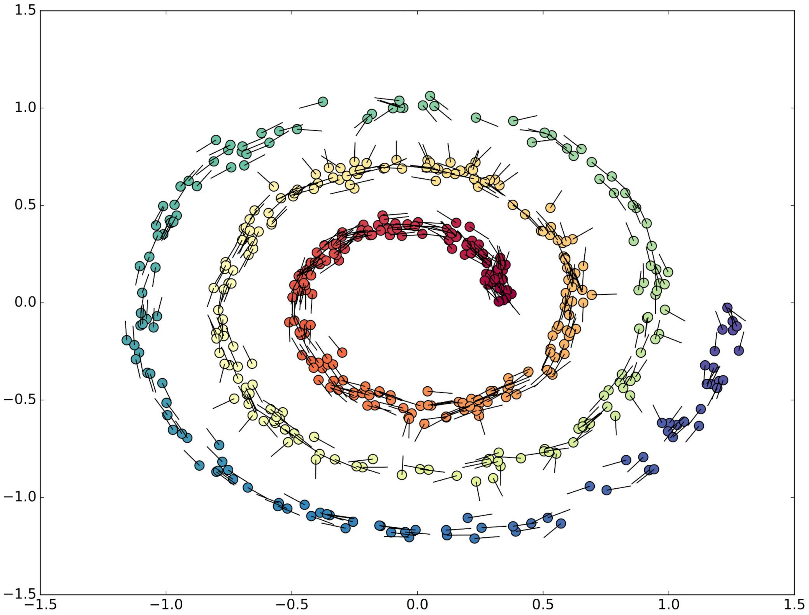


k-Nearest Neighbors

Spiral Experiment Tangent Estimation

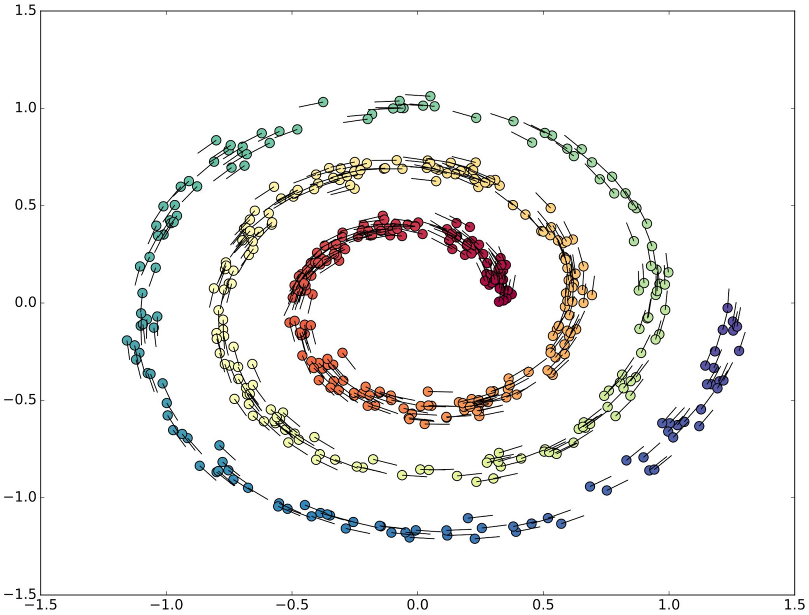


Ground Truth

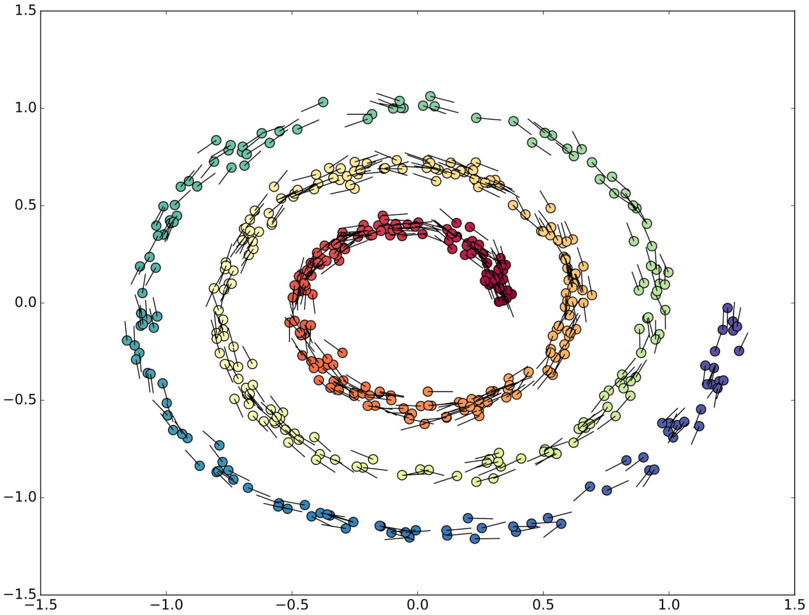


PCA

Spiral Experiment Tangent Estimation

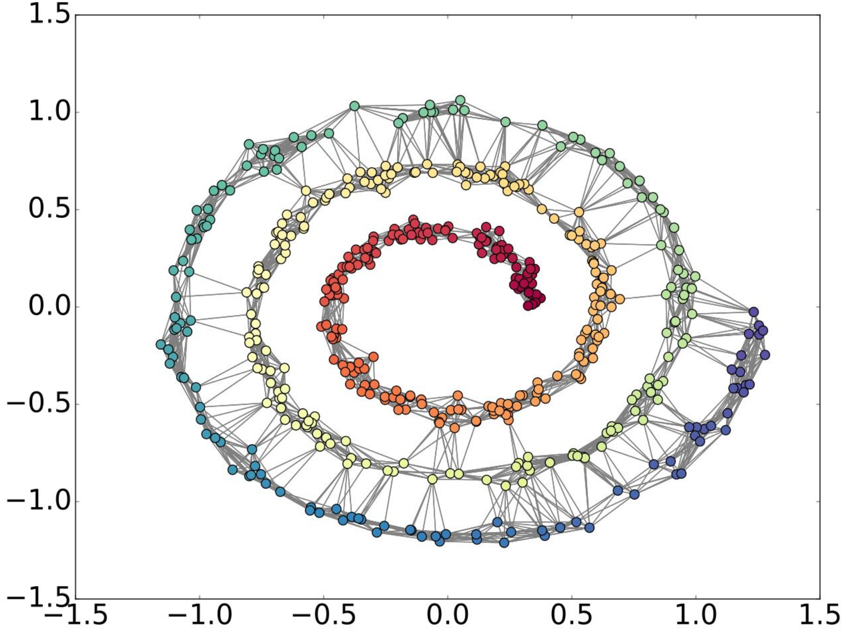


Ground Truth

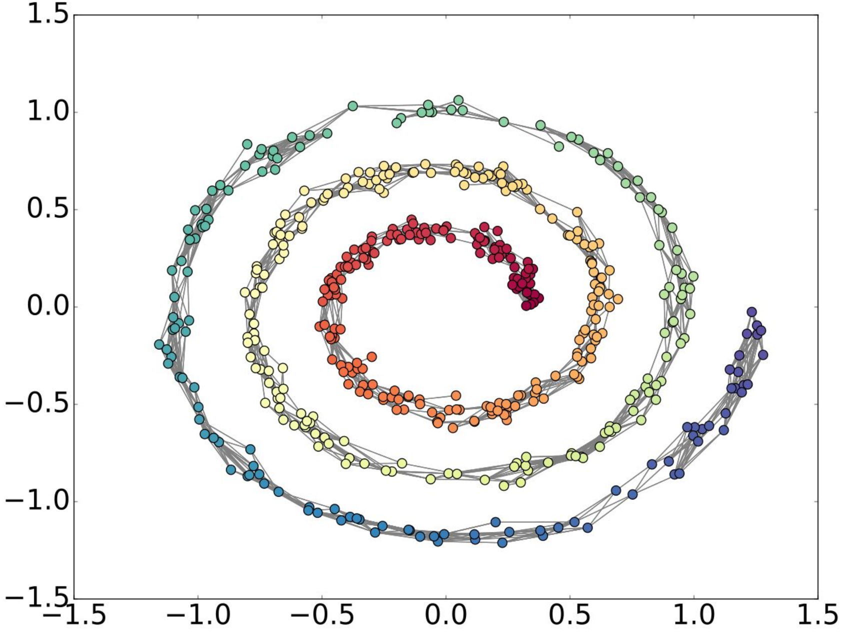


TSBP

Spiral Experiment Nearest Neighbors

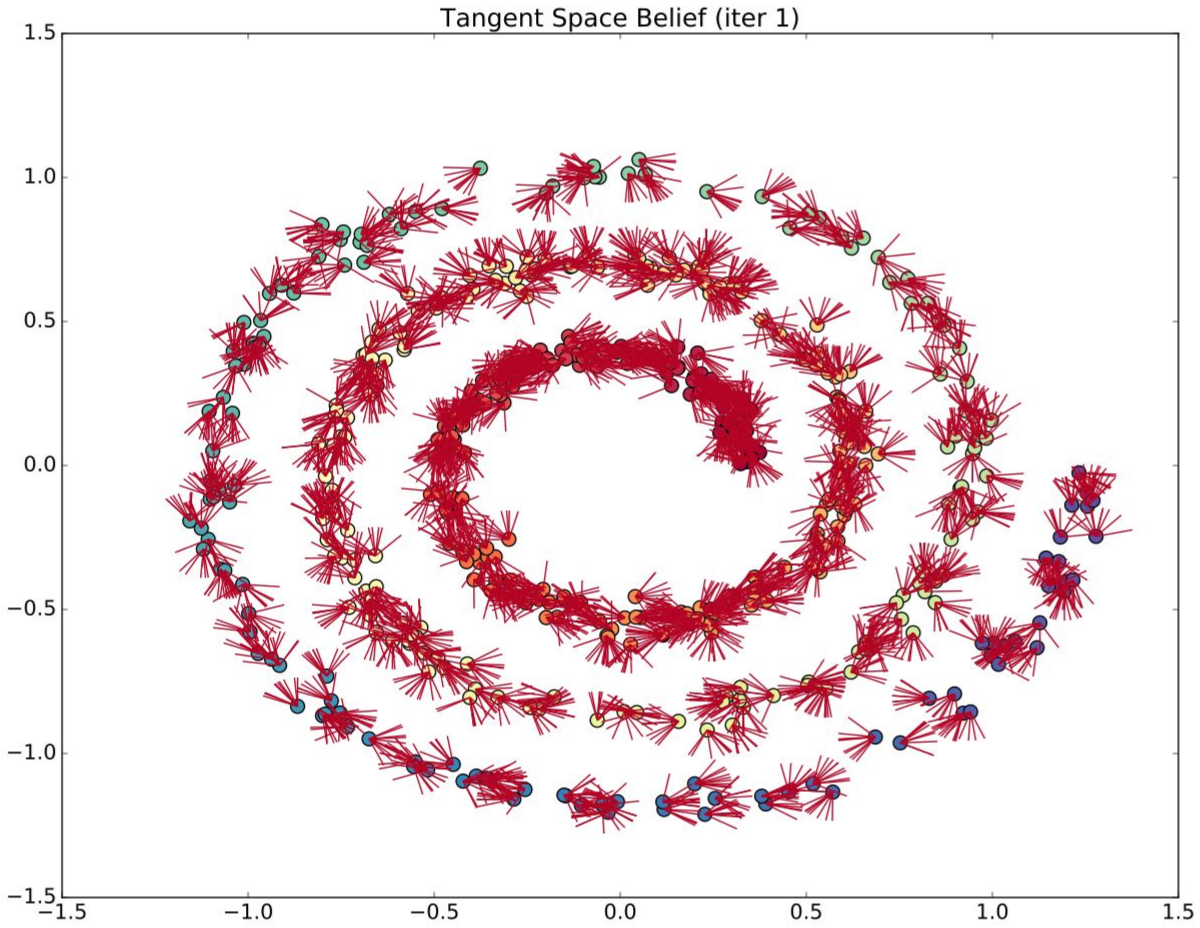


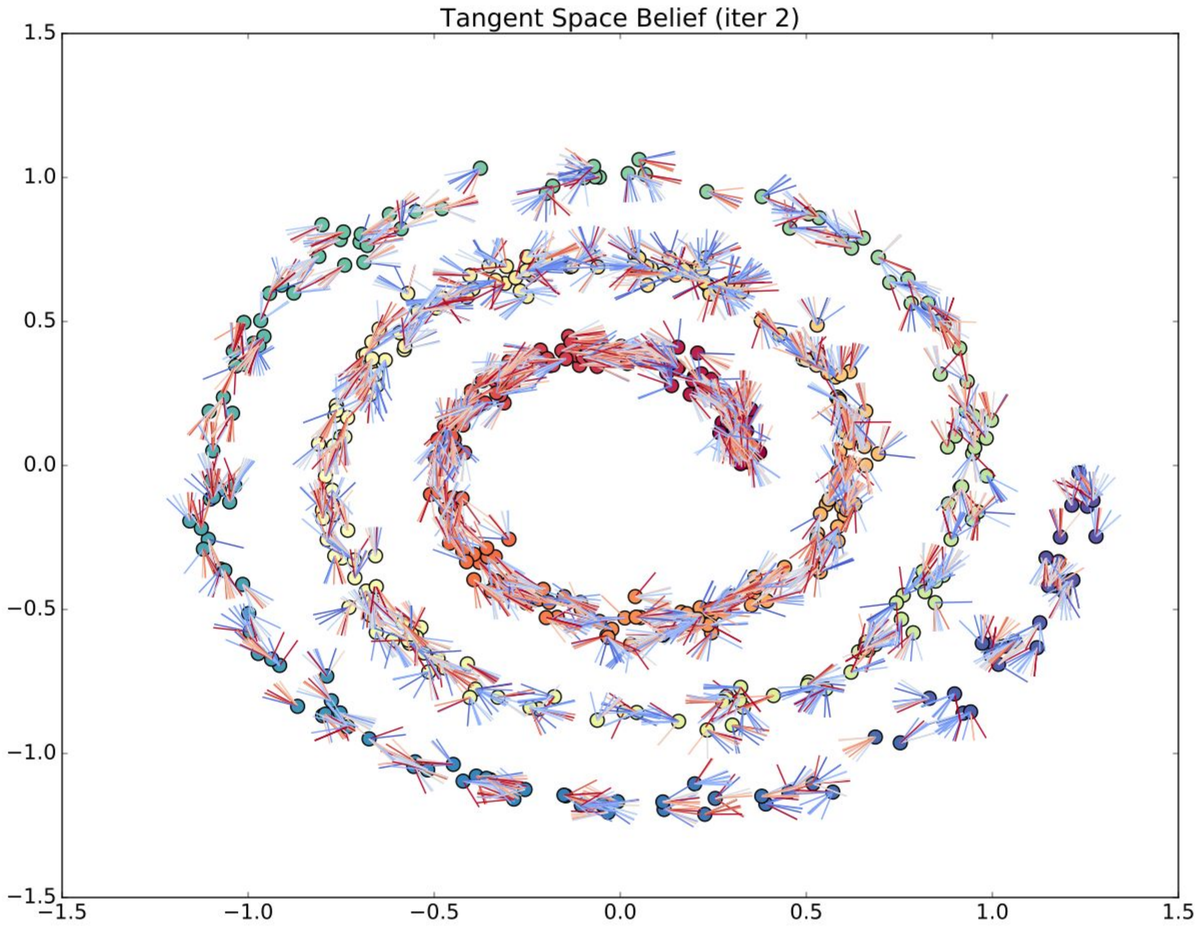
k-Nearest Neighbors



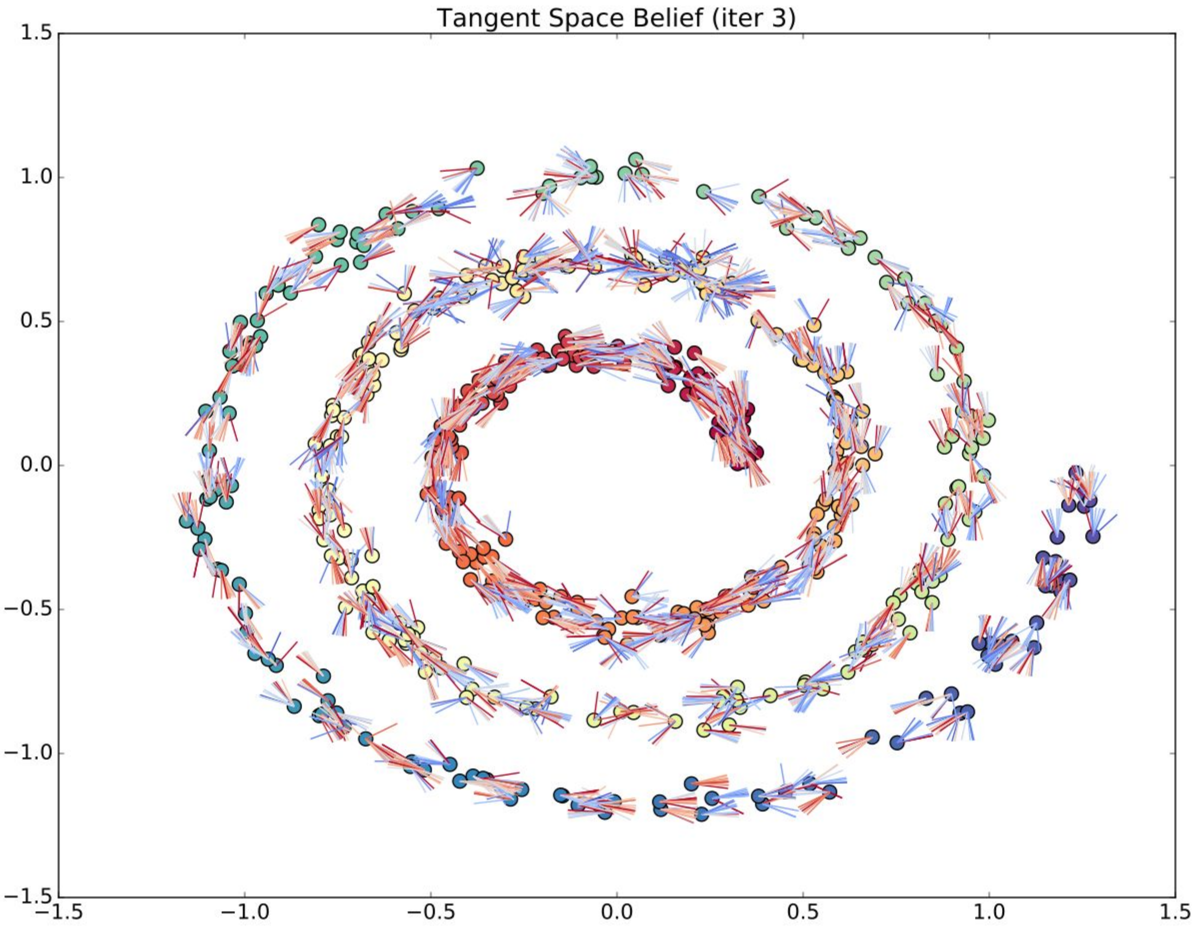
TSBP

Spiral Experiment Belief Propagation

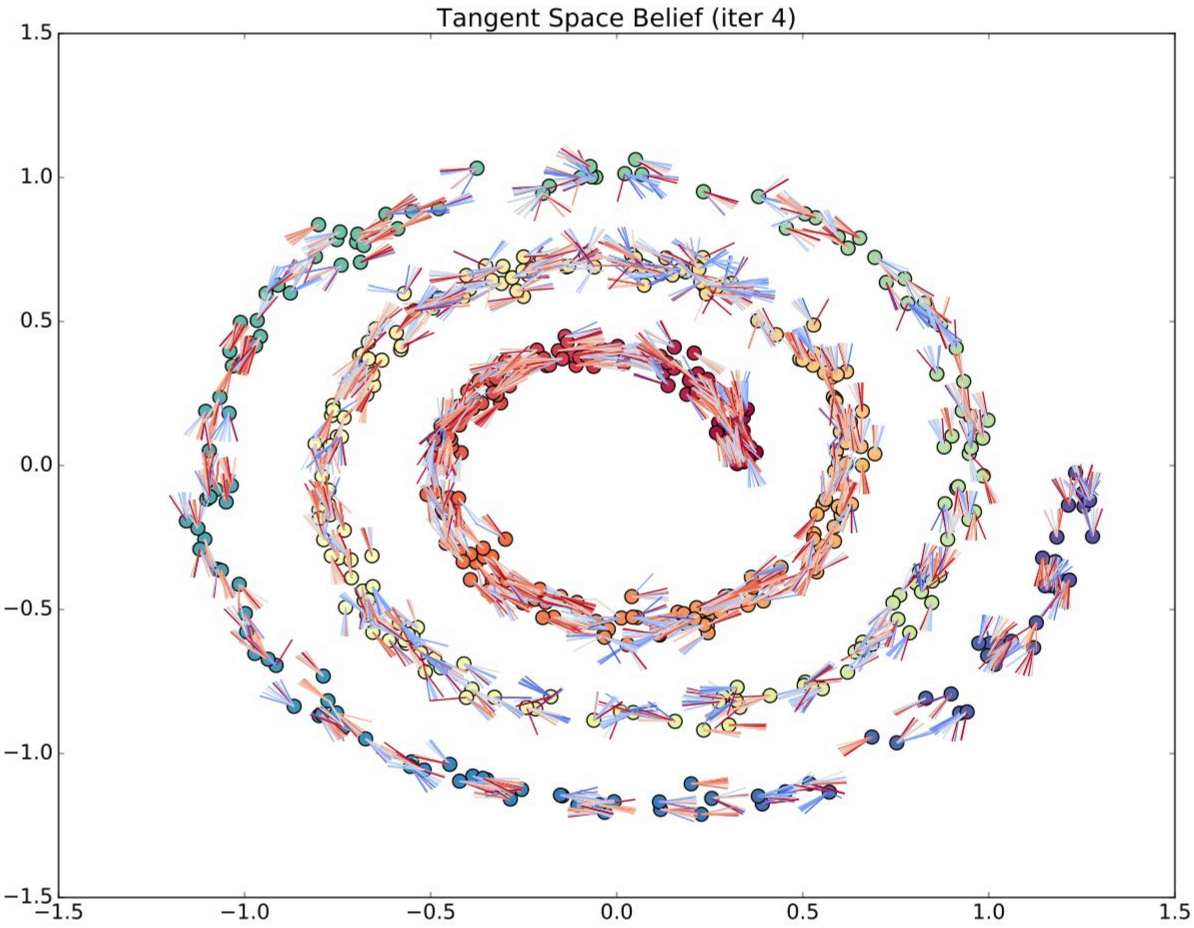

Tangent Space Belief (iter 1)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

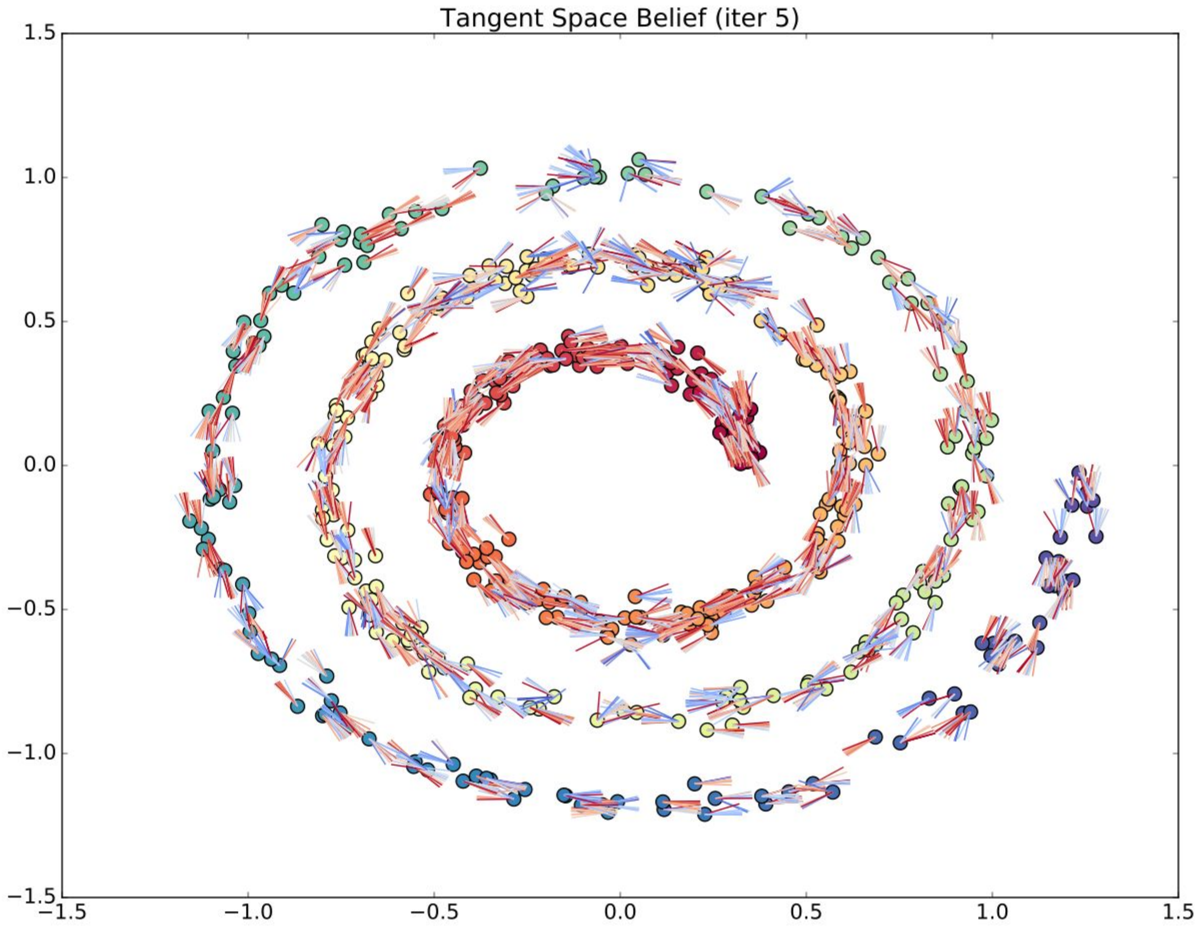


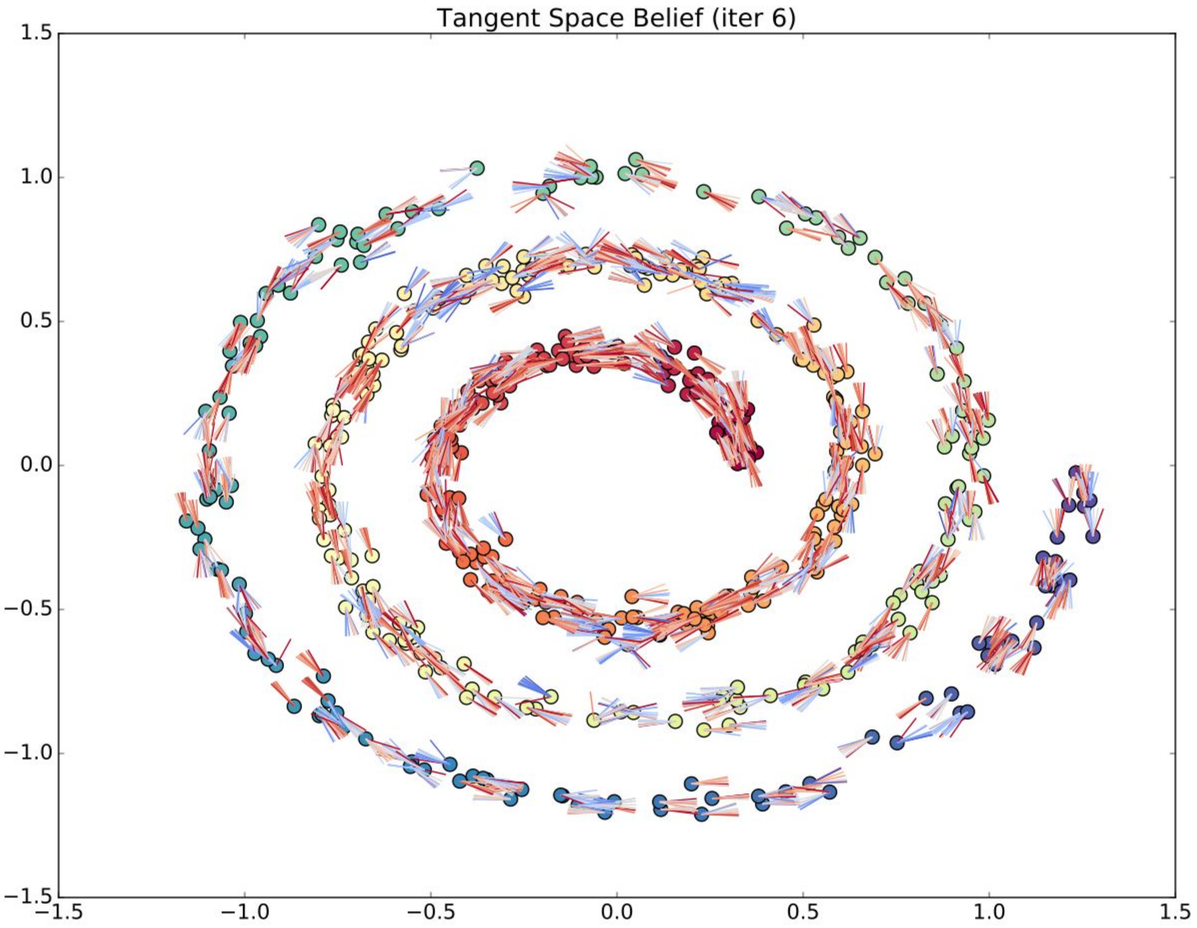
Tangent Space Belief (iter 2)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

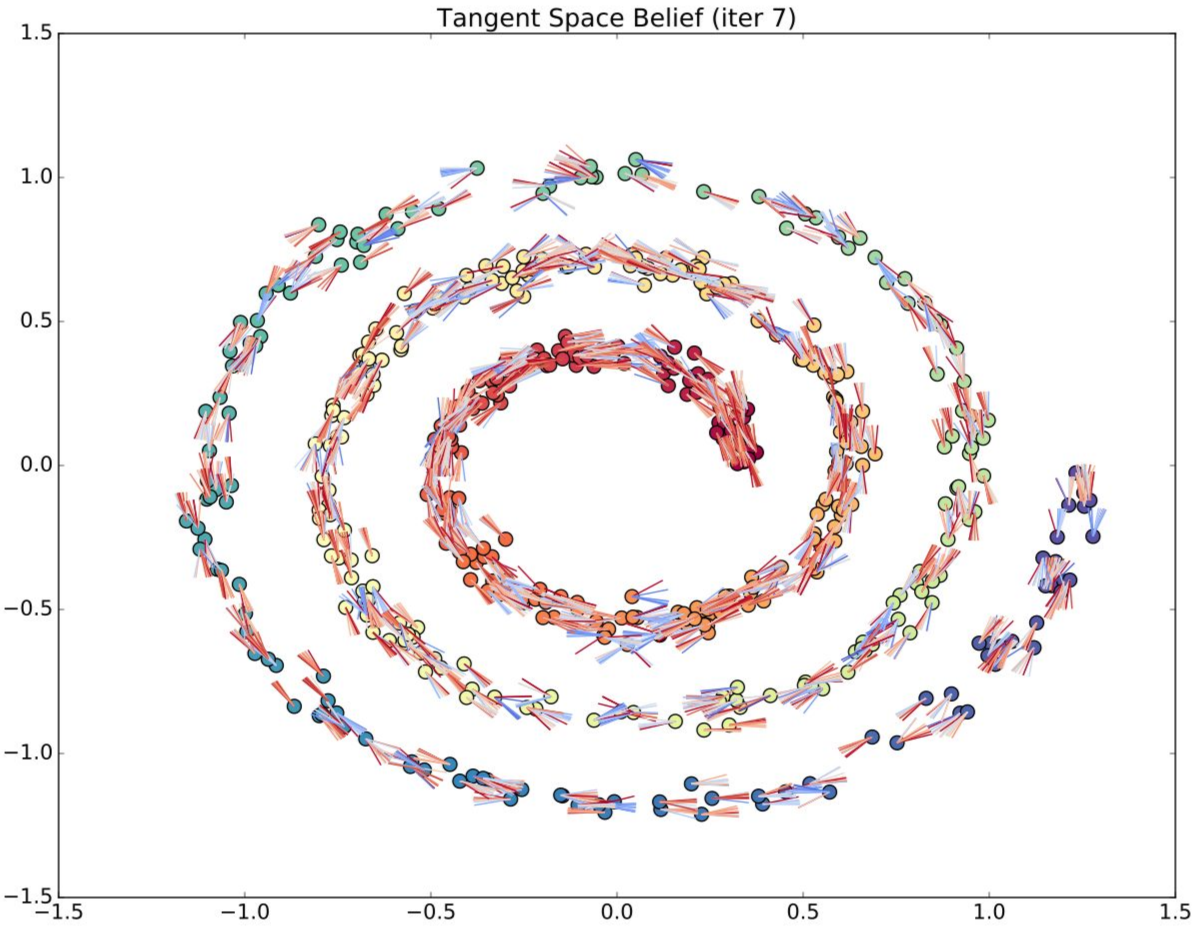


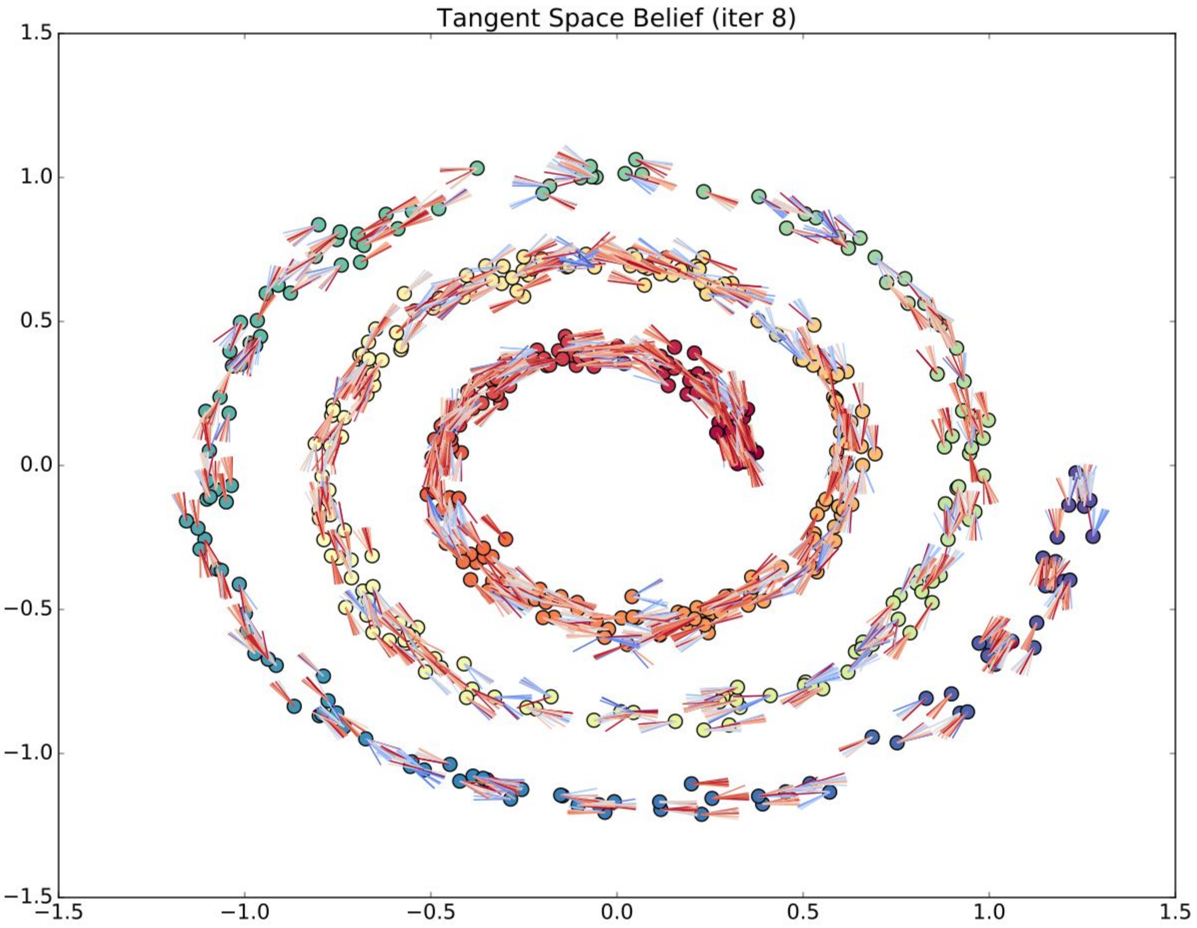
Tangent Space Belief (iter 3)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

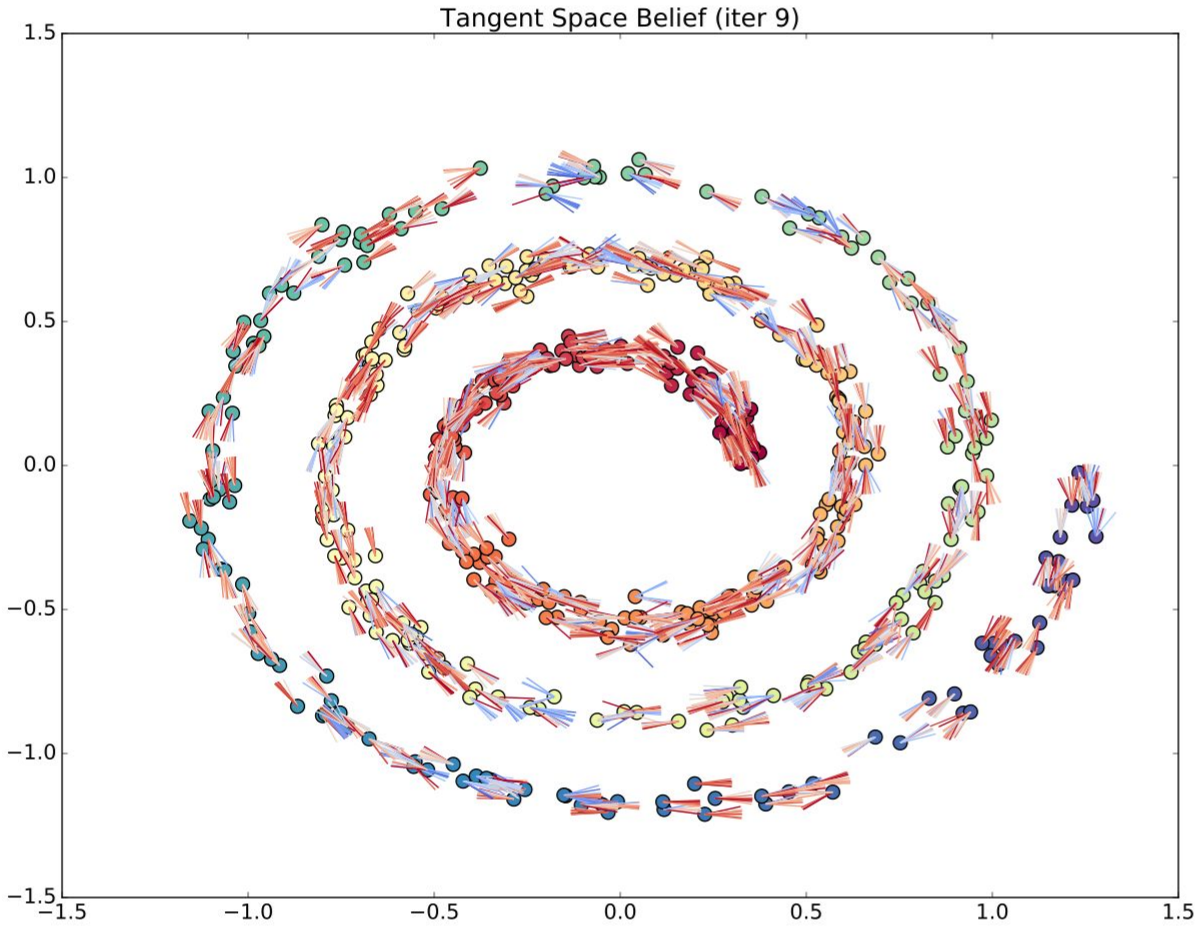

Tangent Space Belief (iter 4)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

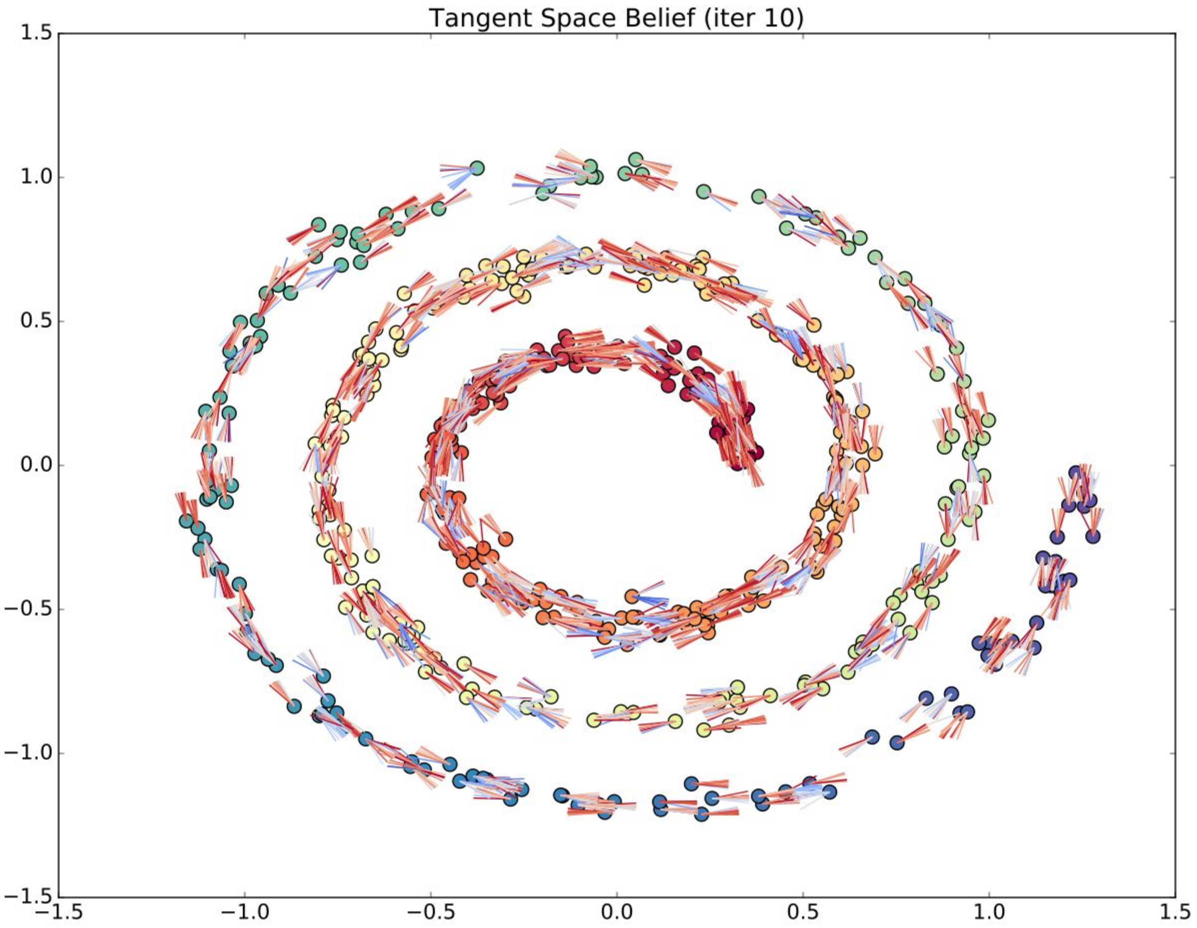

Tangent Space Belief (iter 5)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

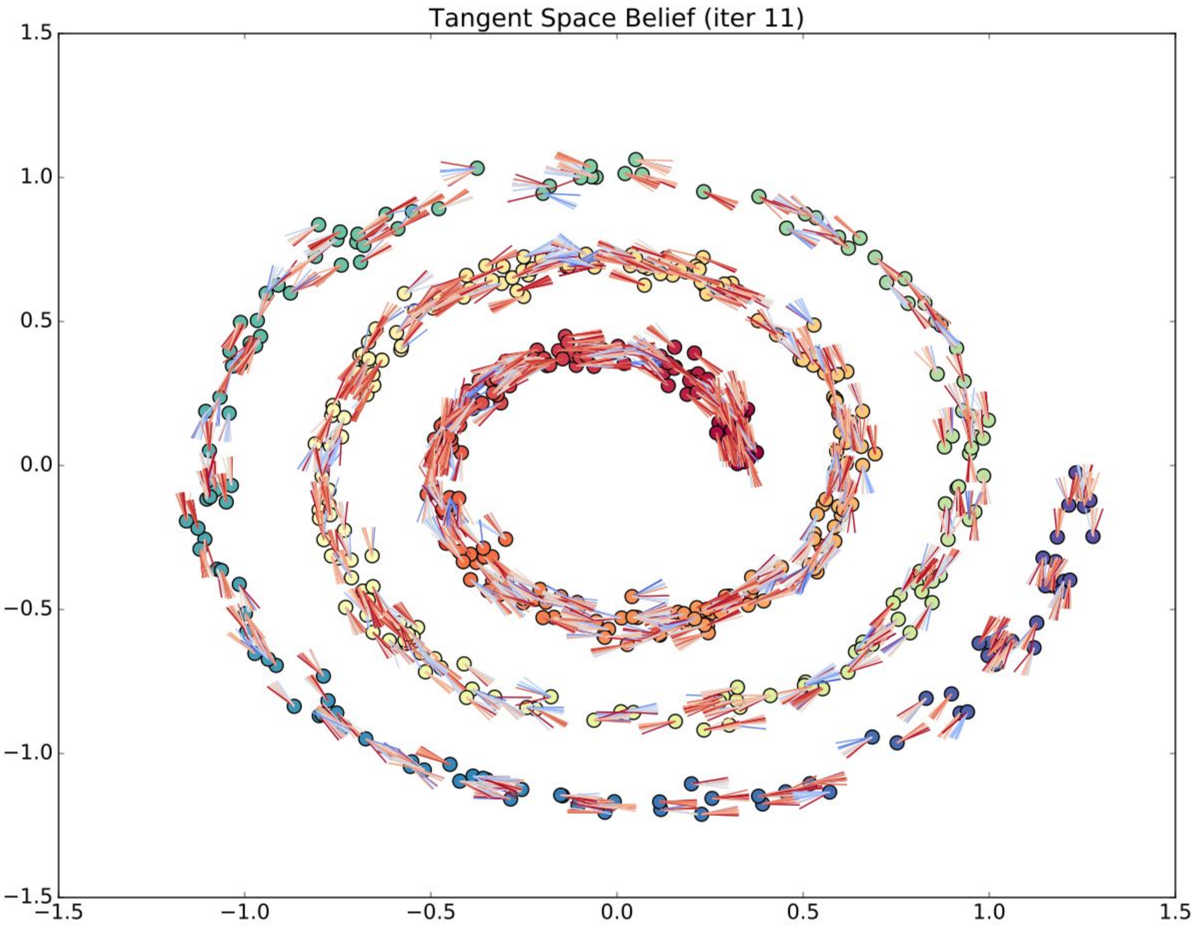

Tangent Space Belief (iter 6)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Tangent Space Belief (iter 7)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

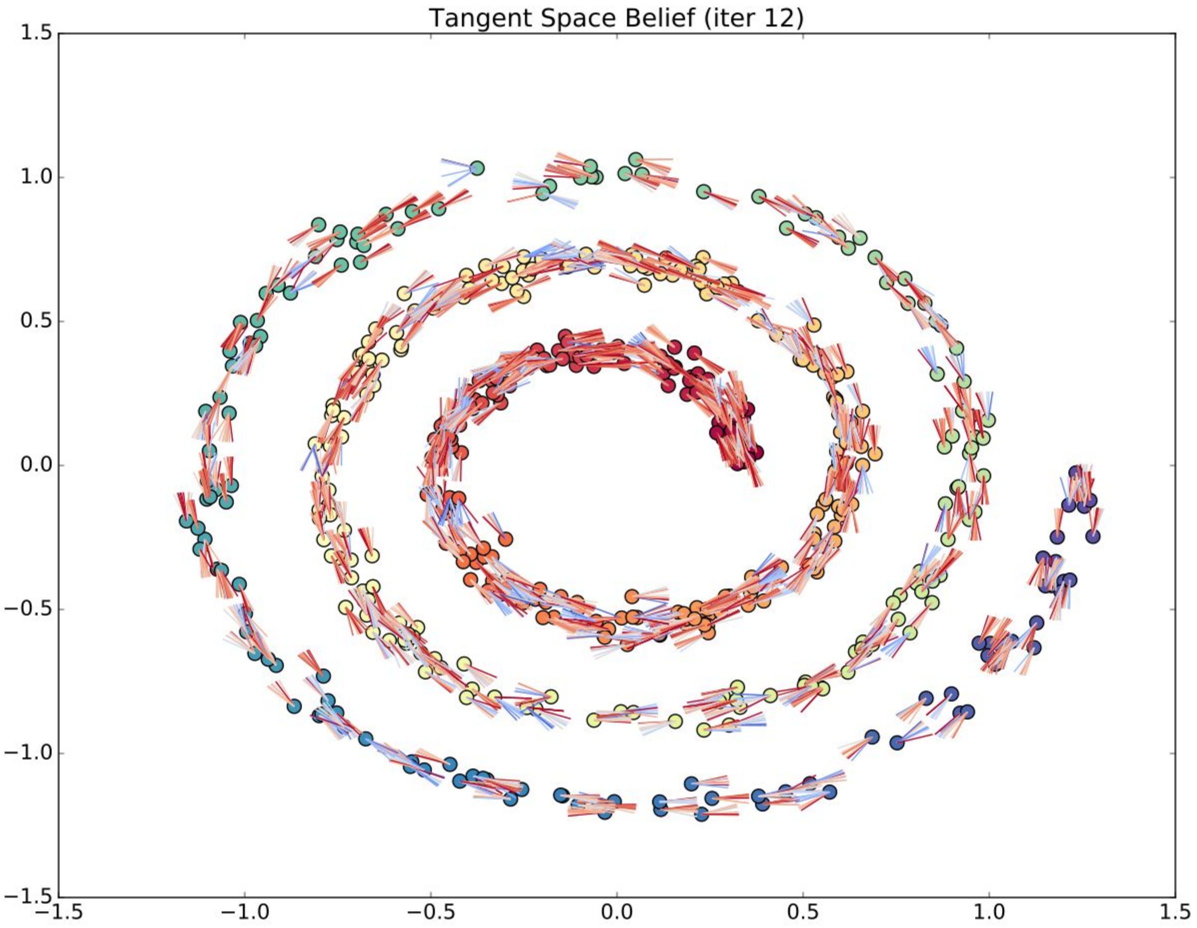

Tangent Space Belief (iter 8)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Tangent Space Belief (iter 9)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

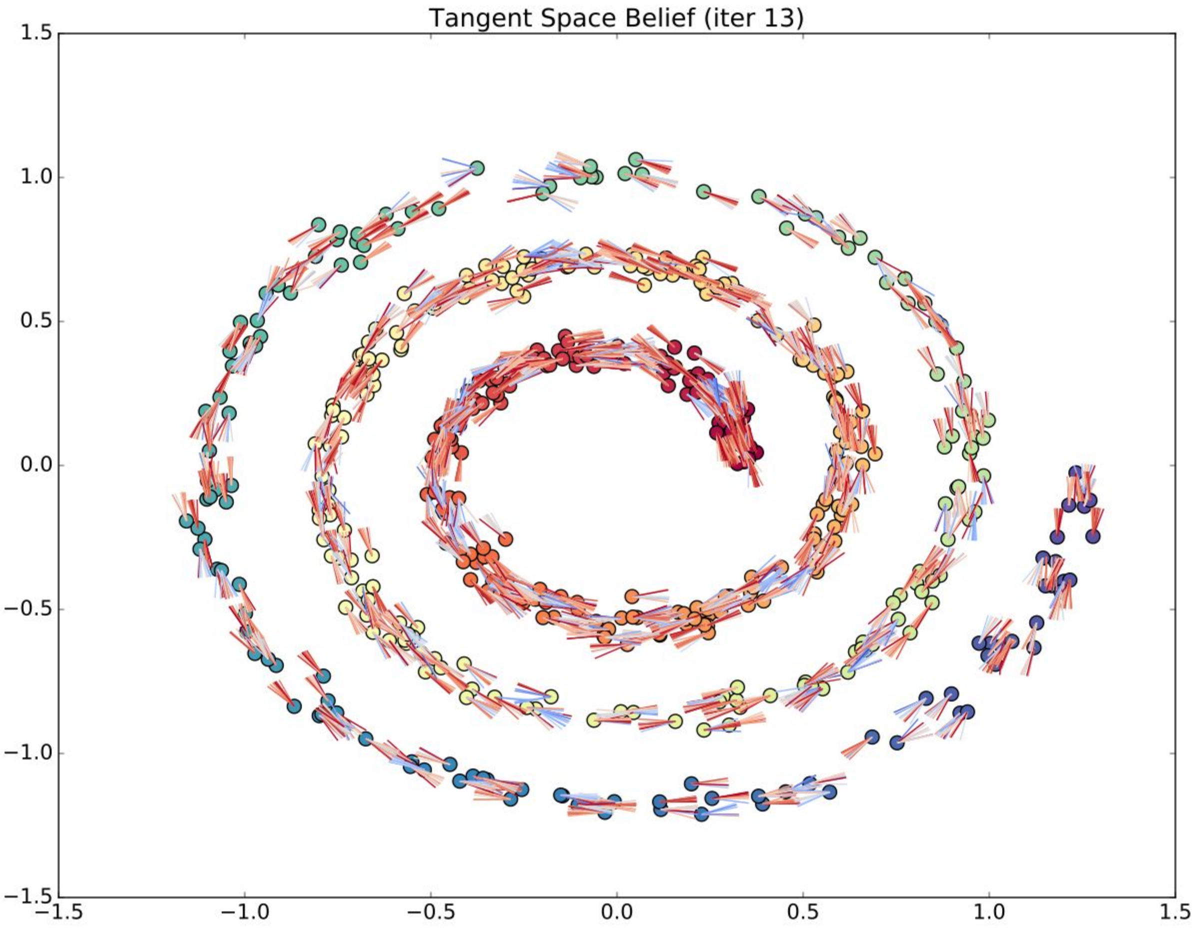

Tangent Space Belief (iter 10)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5



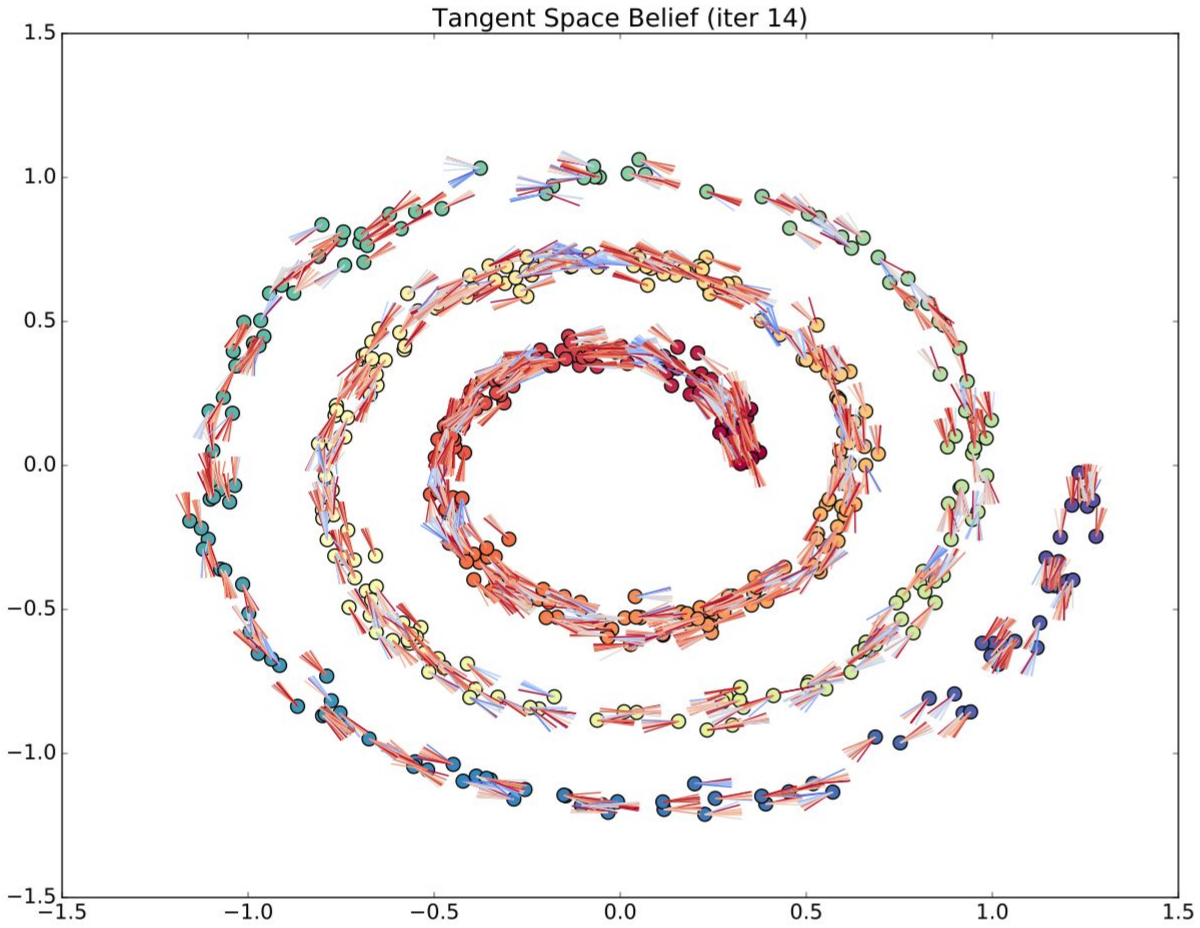
Tangent Space Belief (iter 11)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

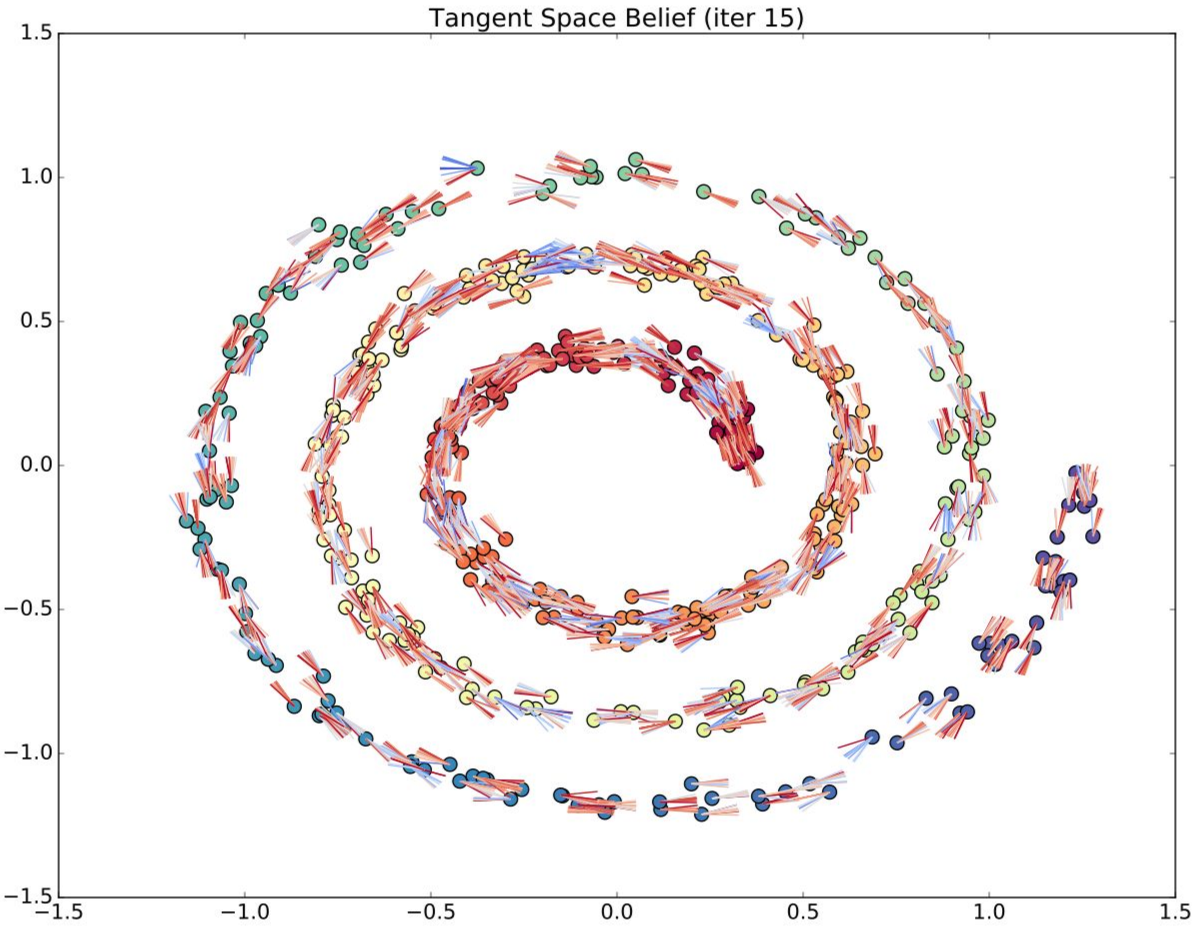


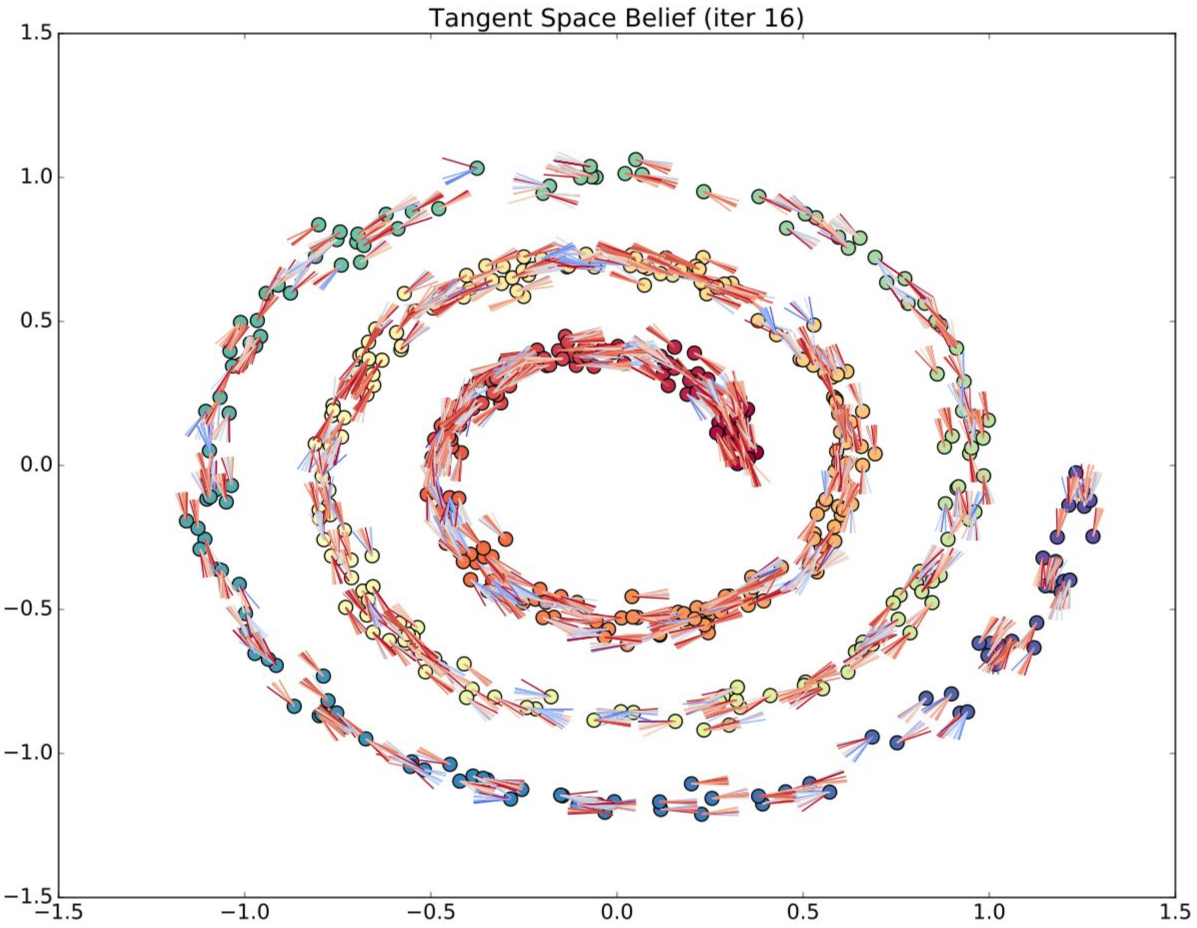
Tangent Space Belief (iter 12)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

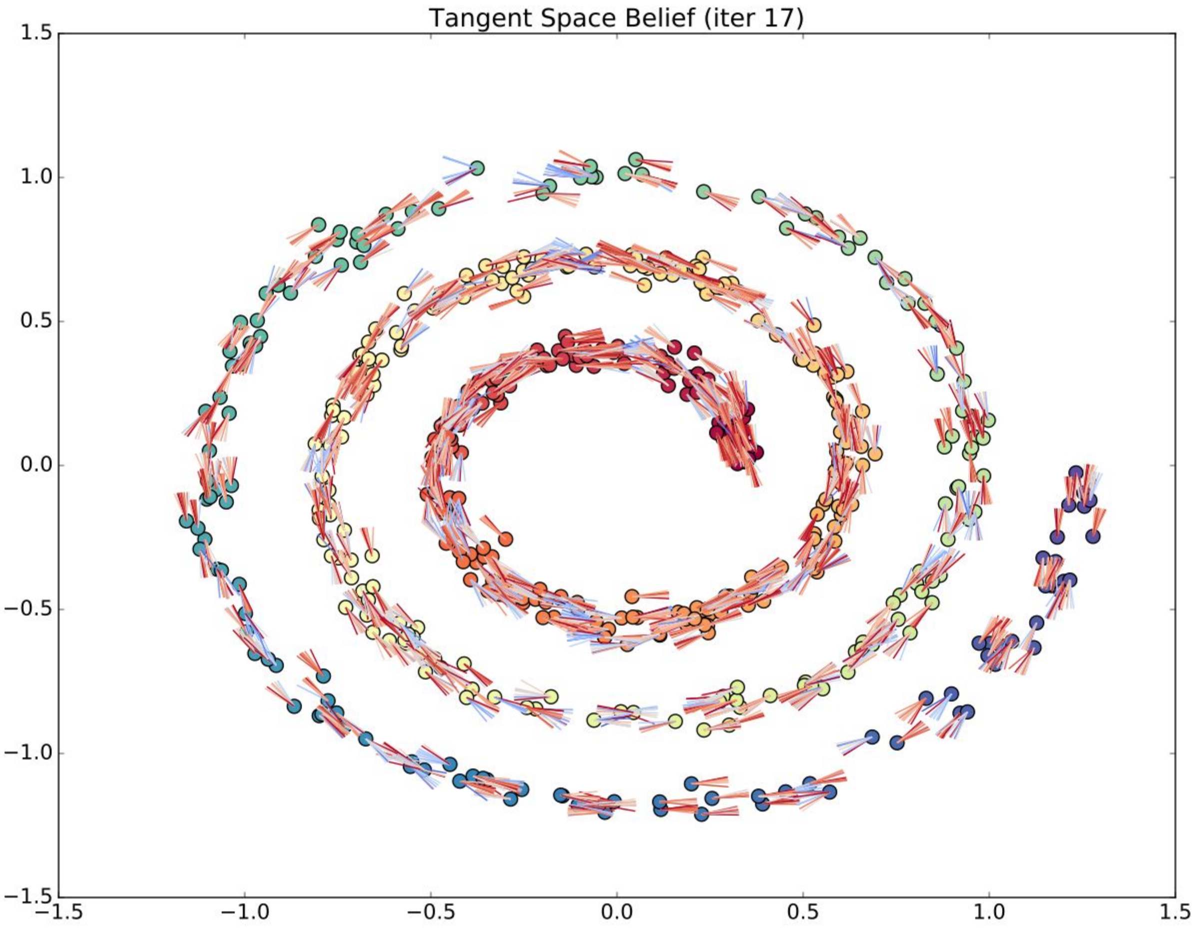


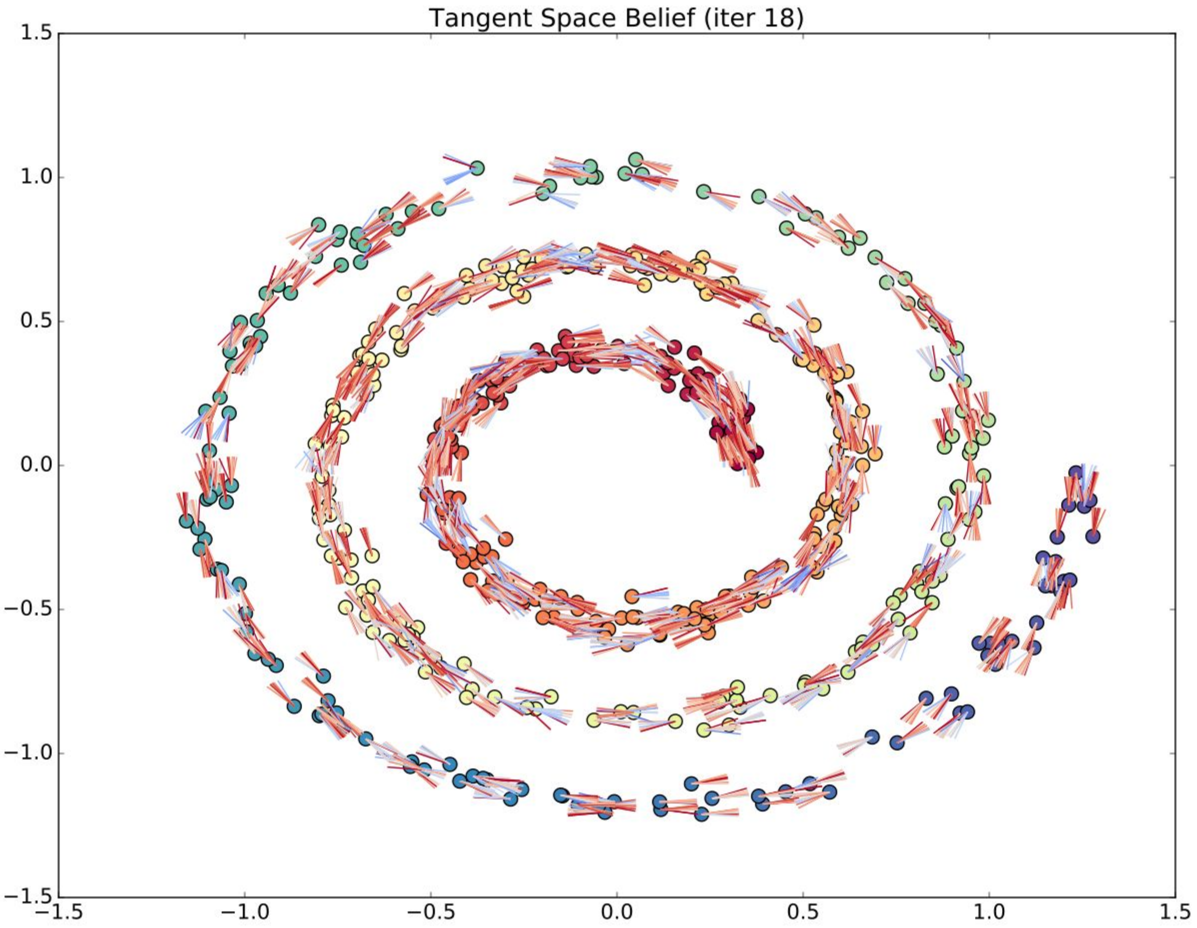
Tangent Space Belief (iter 13)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

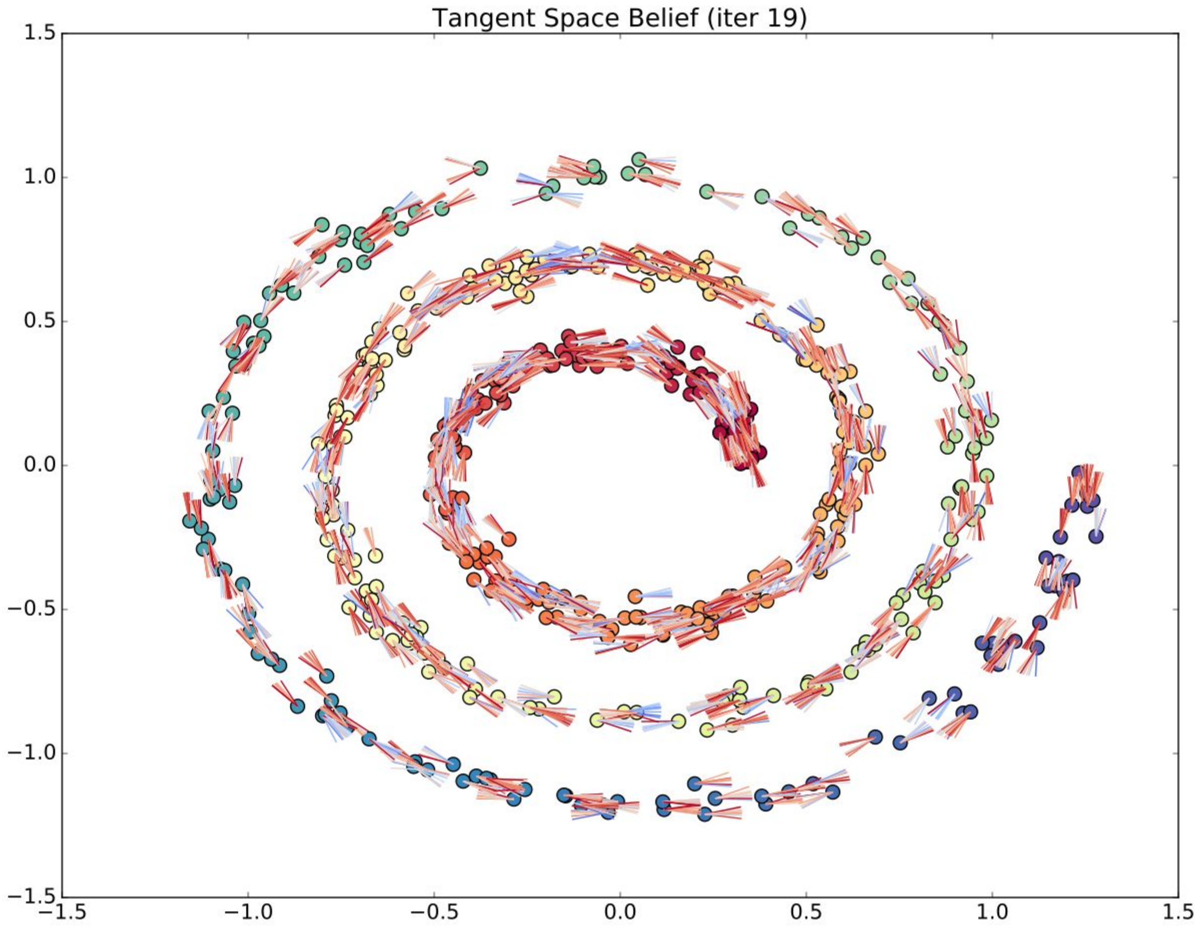

Tangent Space Belief (iter 14)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

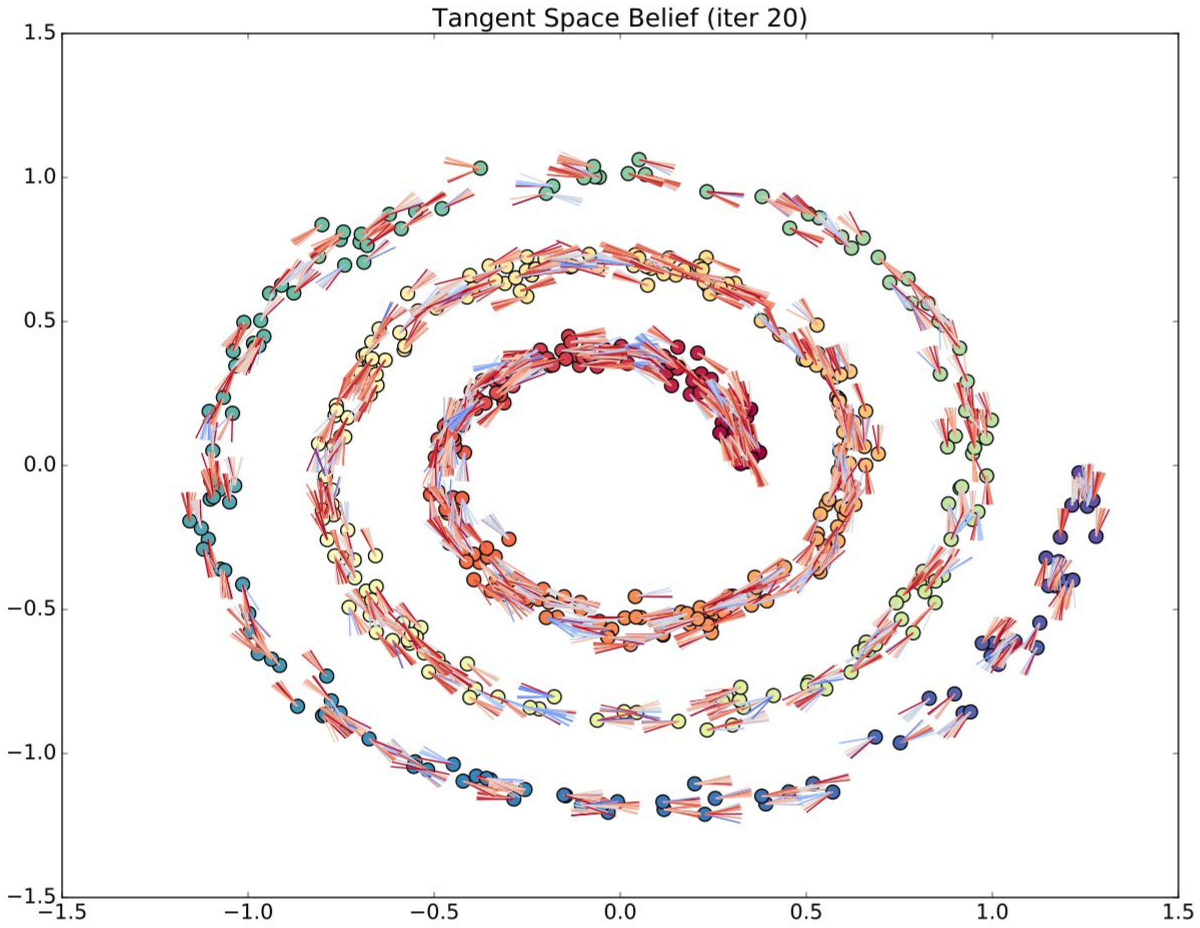

Tangent Space Belief (iter 15)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

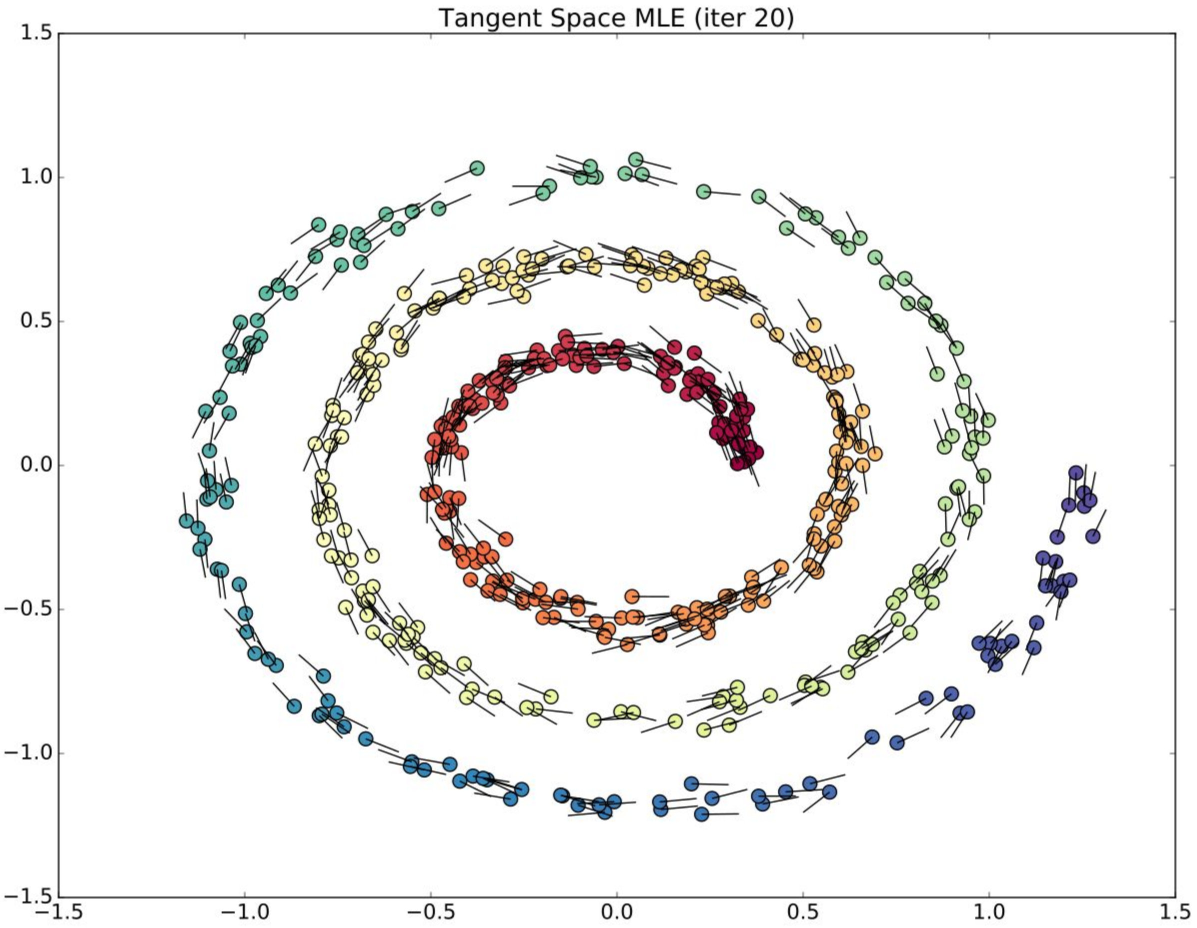

Tangent Space Belief (iter 16)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Tangent Space Belief (iter 17)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

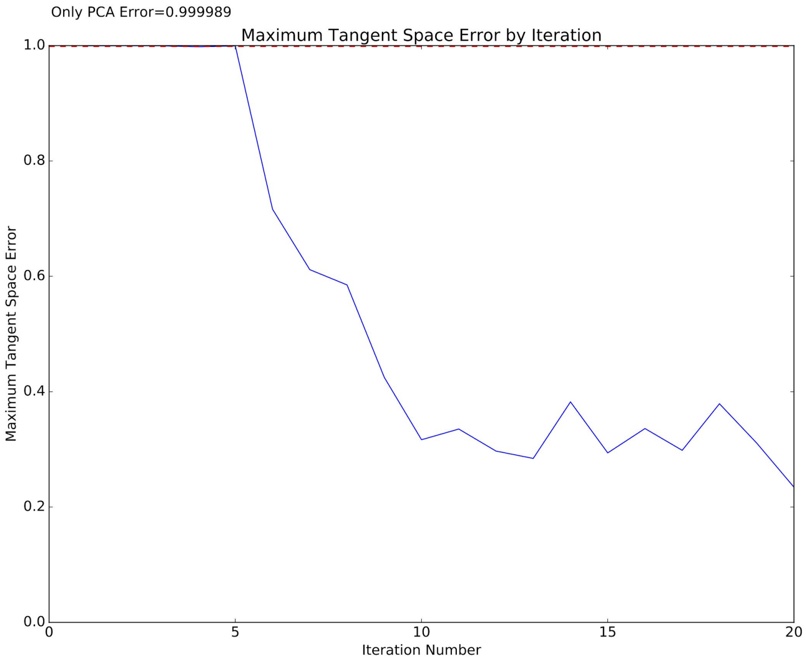

Tangent Space Belief (iter 18)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

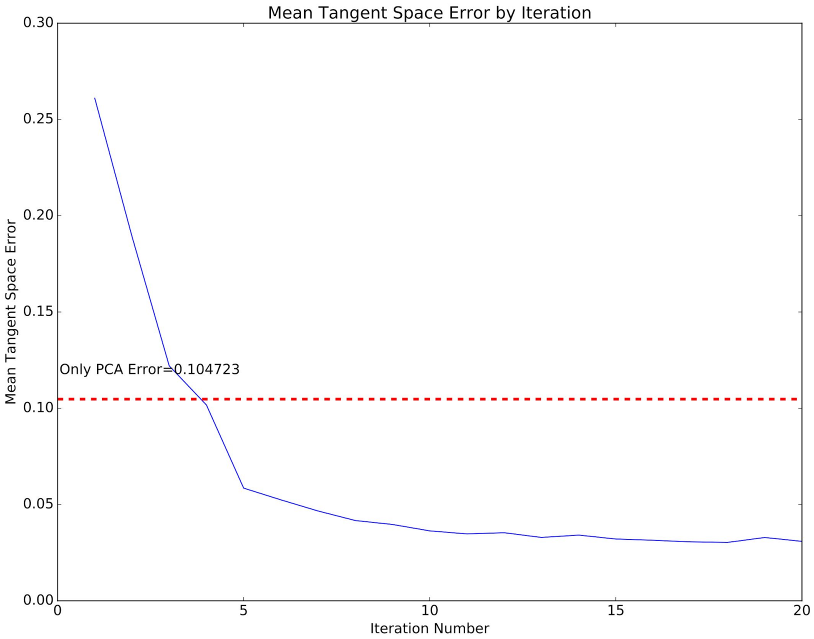

Tangent Space Belief (iter 19)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Tangent Space Belief (iter 20)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Tangent Space MLE (iter 20)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

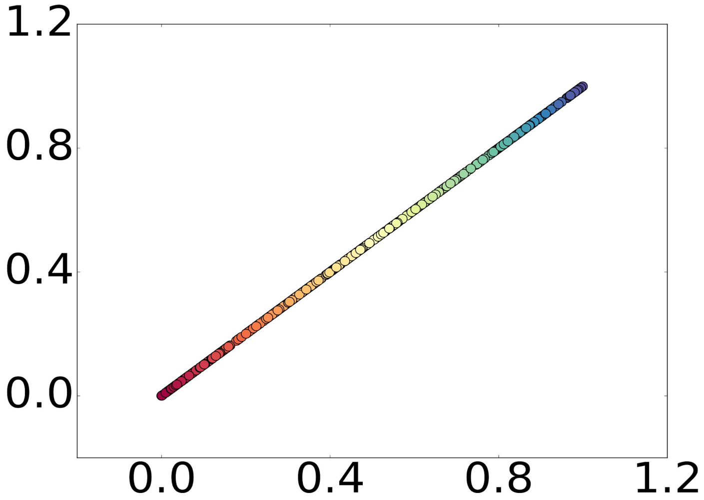
Spiral Experiment Error

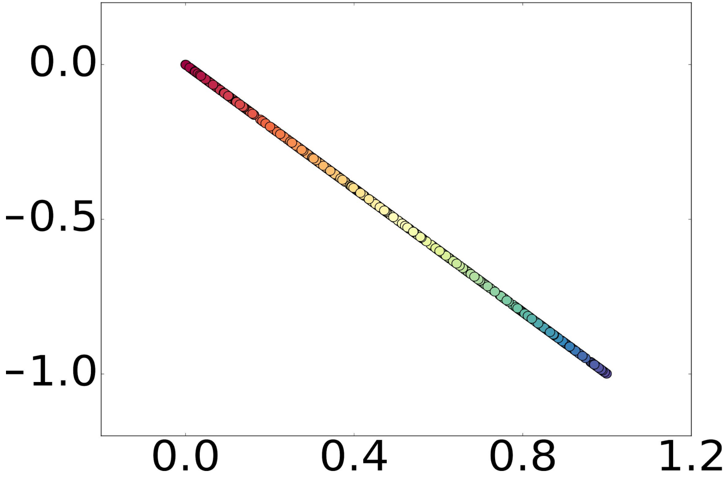




Embedding Evaluation

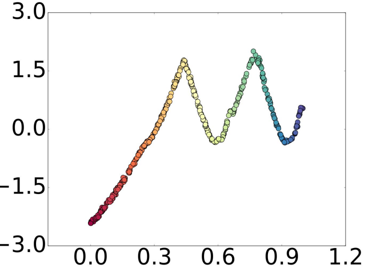
- Plot the original data parameters versus the embedding
- Compare nearest-neighbors and TSBP
- Expect a continuous, monotonic relationship
- Ideal embedding:



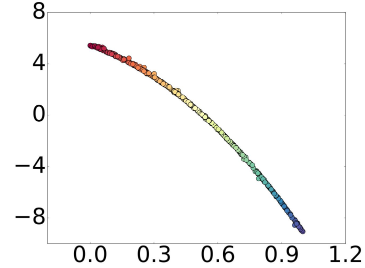


Embedding Evaluation

ISOMAP

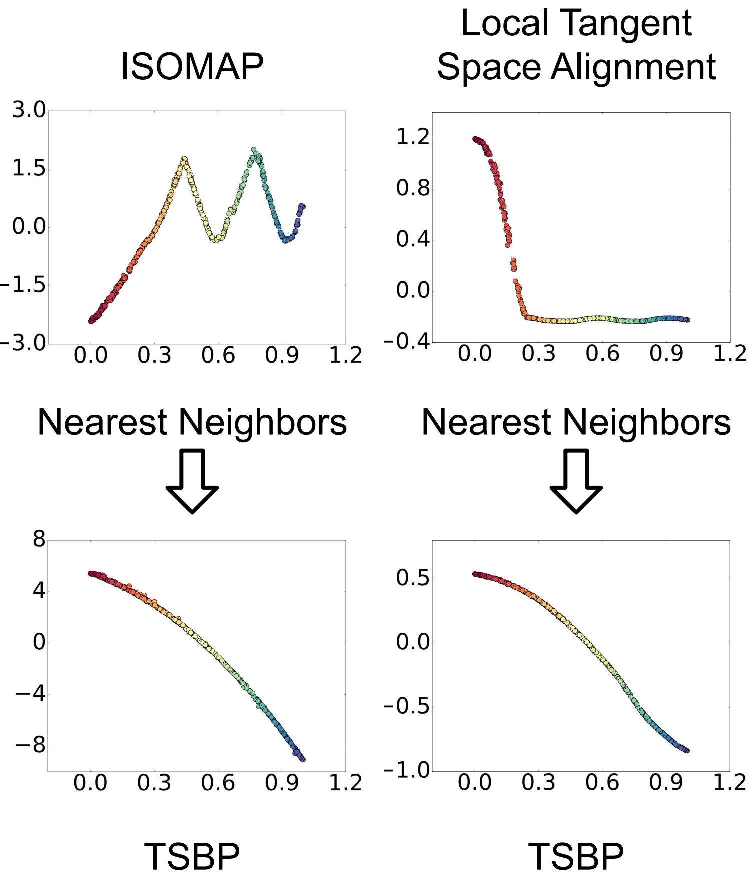


Nearest Neighbors



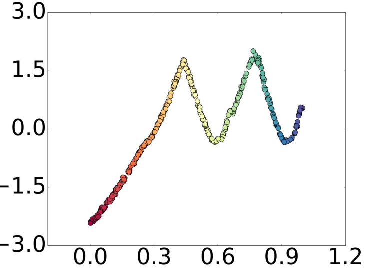
TSBP

Embedding Evaluation

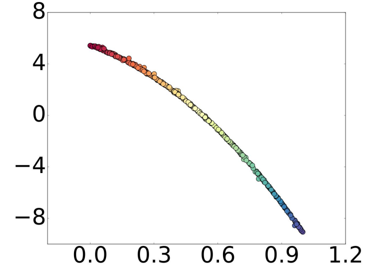


Embedding Evaluation

ISOMAP

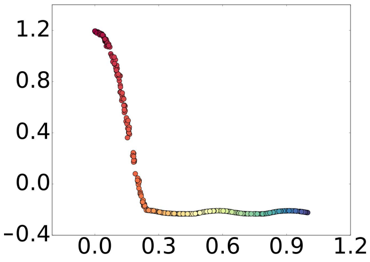


Nearest Neighbors

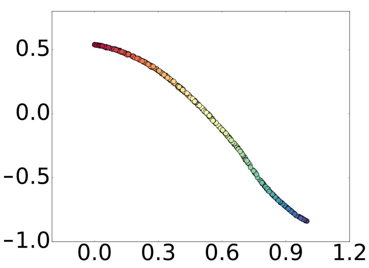


TSBP

Local Tangent Space Alignment

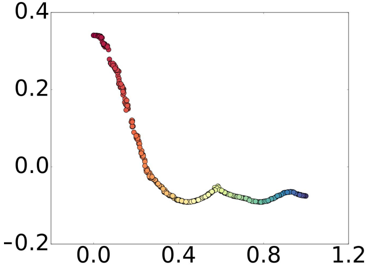


Nearest Neighbors

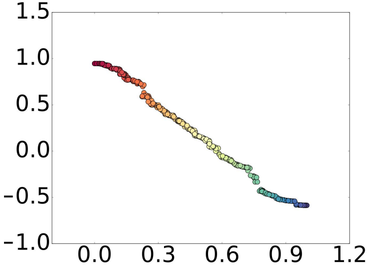


TSBP

Laplacian Eigenmaps



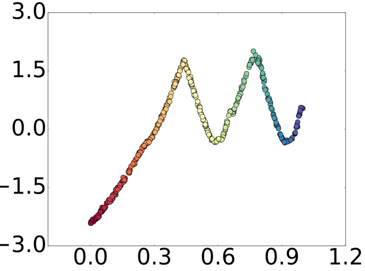
Nearest Neighbors



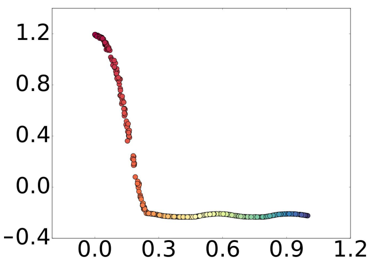
TSBP

Embedding Evaluation

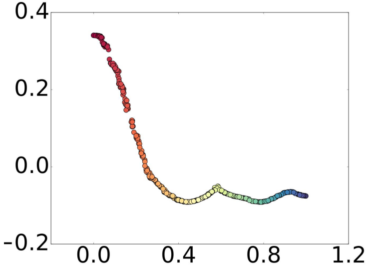
ISOMAP



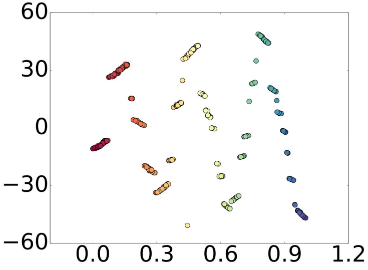
Local Tangent Space Alignment



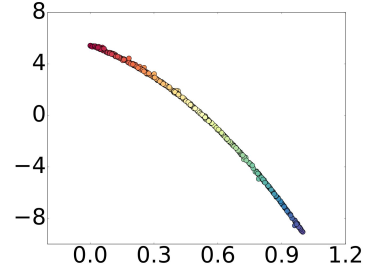
Laplacian Eigenmaps



t-Distributed Stochastic Neighbor Embedding

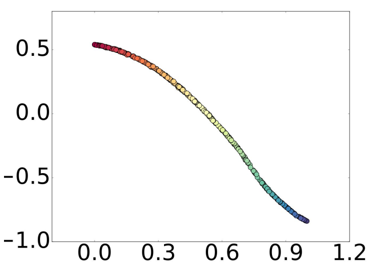


Nearest Neighbors ⇩



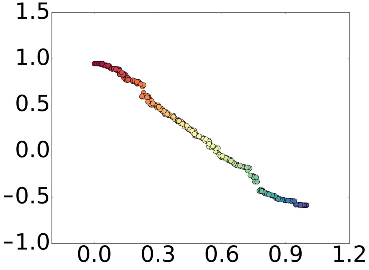
TSBP

Nearest Neighbors ⇩



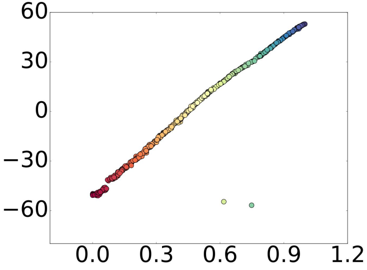
TSBP

Nearest Neighbors ⇩



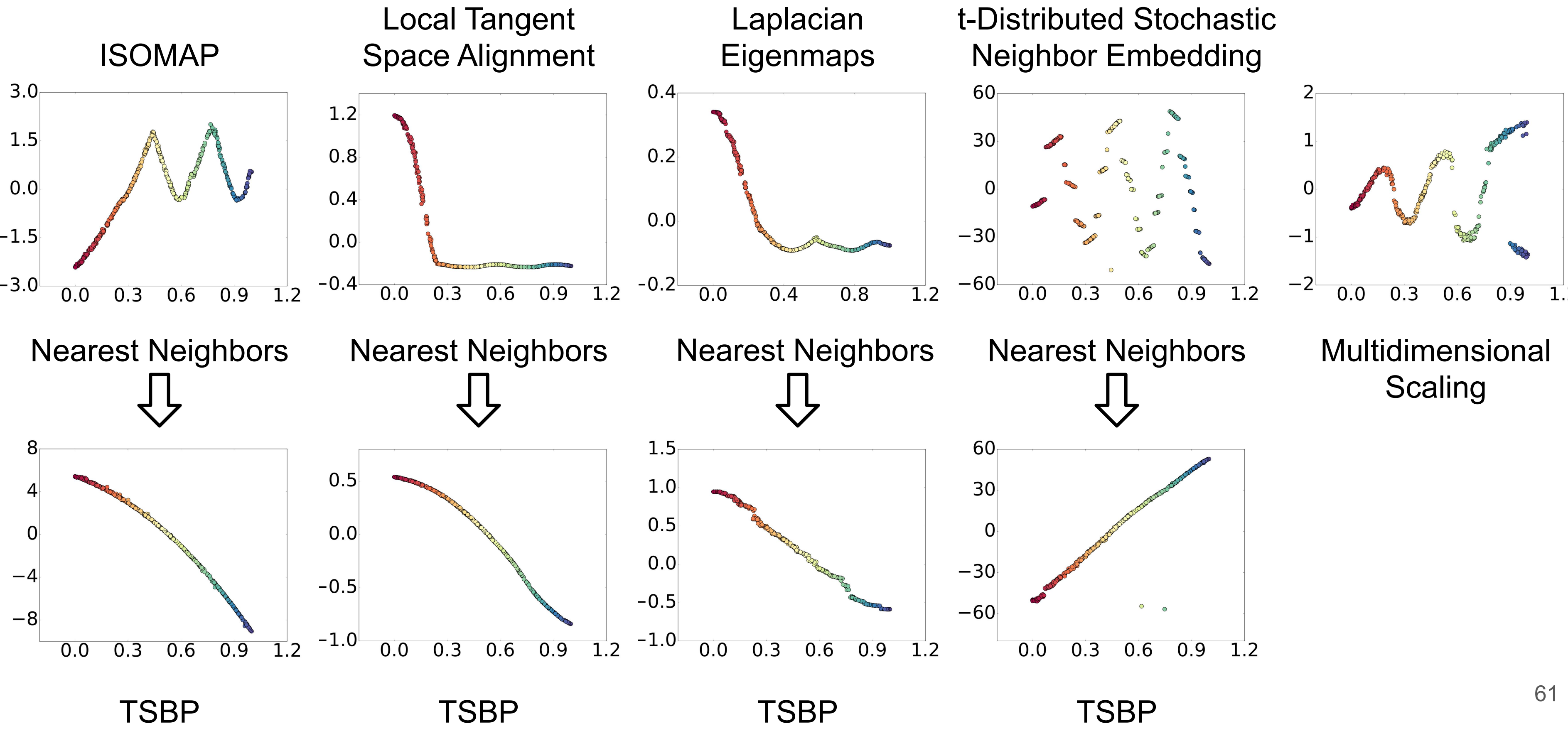
TSBP

Nearest Neighbors ⇩



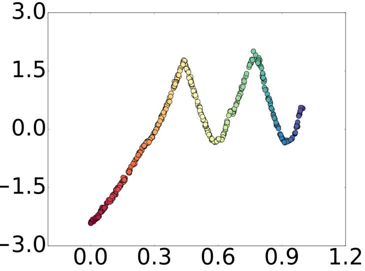
TSBP

Embedding Evaluation

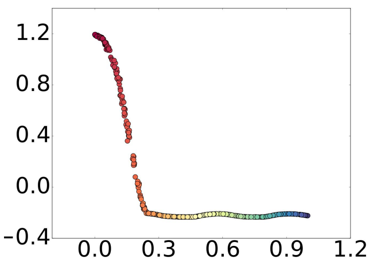


Embedding Evaluation

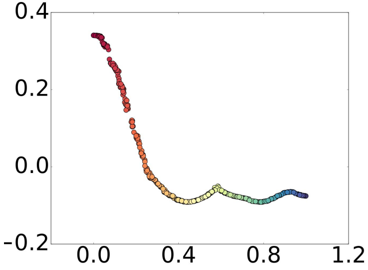
ISOMAP



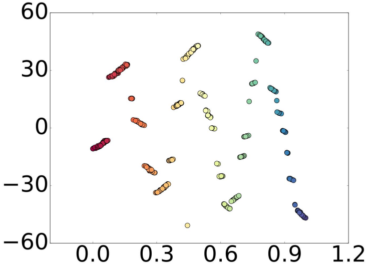
Local Tangent Space Alignment

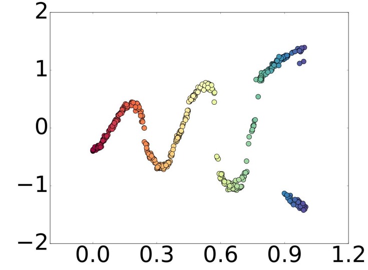


Laplacian Eigenmaps

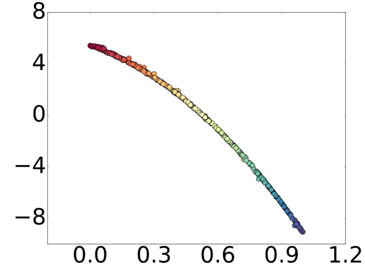


t-Distributed Stochastic Neighbor Embedding



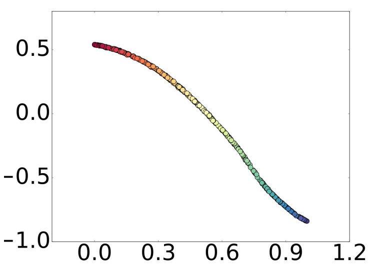


Nearest Neighbors ⇩



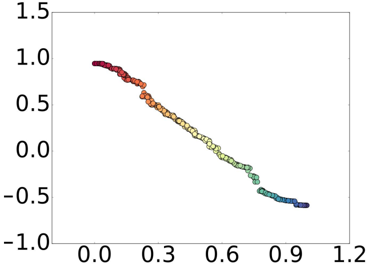
TSBP

Nearest Neighbors ⇩



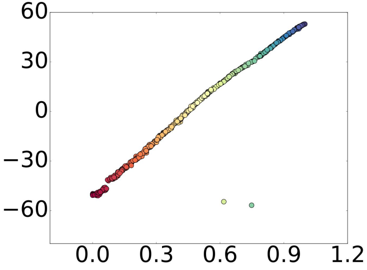
TSBP

Nearest Neighbors ⇩



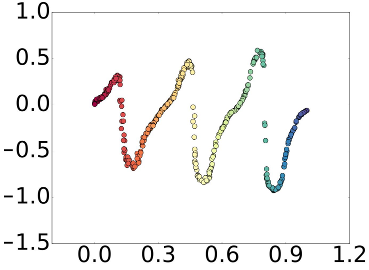
TSBP

Nearest Neighbors ⇩



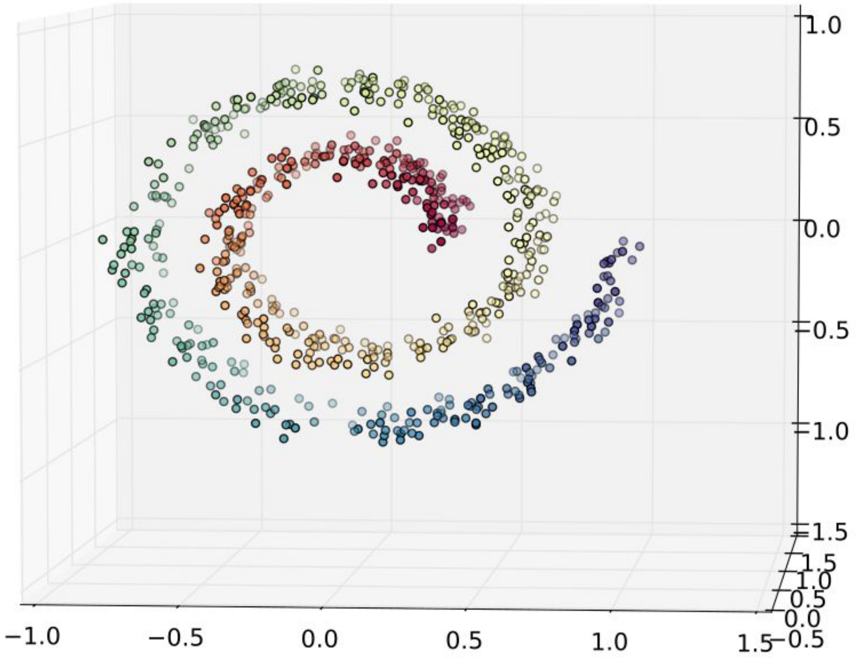
TSBP

Multidimensional Scaling

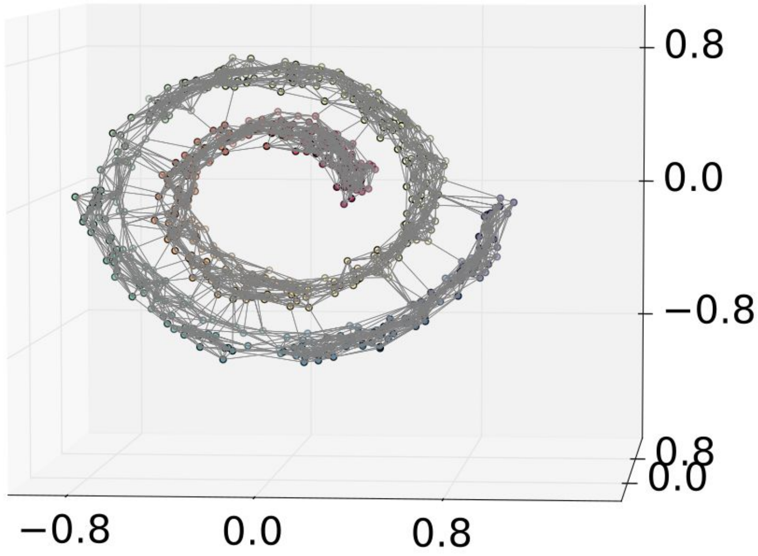


Autoencoder

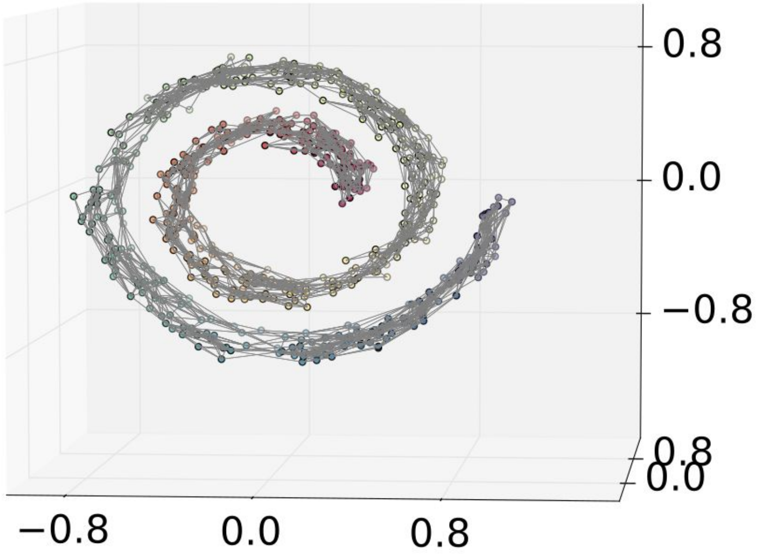
"Swiss Roll" Experiment



"Swiss Roll" Experiment Neighbors



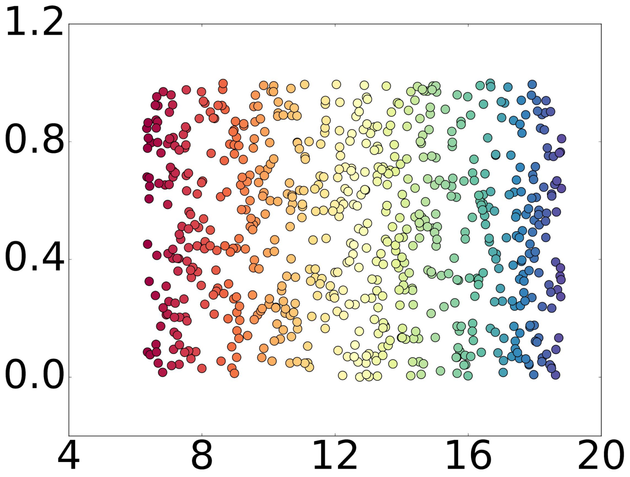
k-Nearest Neighbors



TSBP

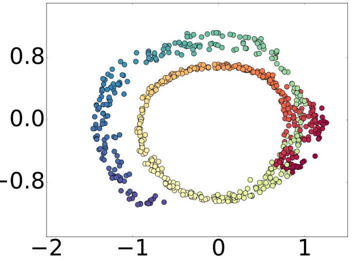
Embedding Evaluation

- Directly plot the 2D embedding
- Compare nearest-neighbors and TSBP
- Expect a rectangular embedding, with no data overlaps
- Ideal embedding:

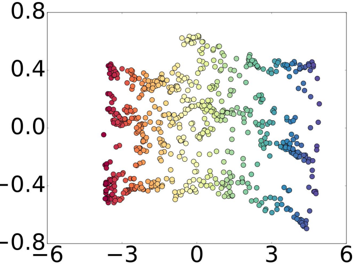


Embedding Evaluation

ISOMAP



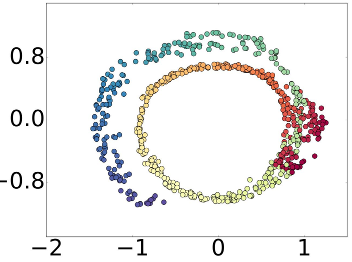
Nearest Neighbors



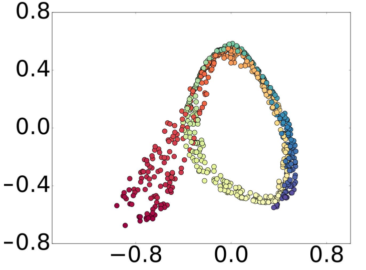
TSBP

Embedding Evaluation

ISOMAP

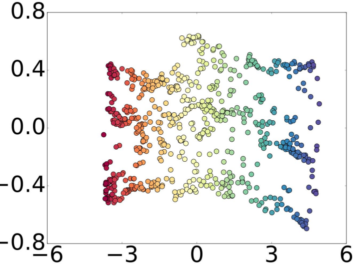


Local Tangent Space Alignment

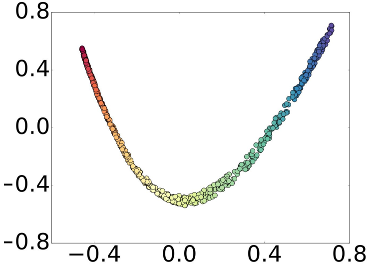


Nearest Neighbors

Nearest Neighbors

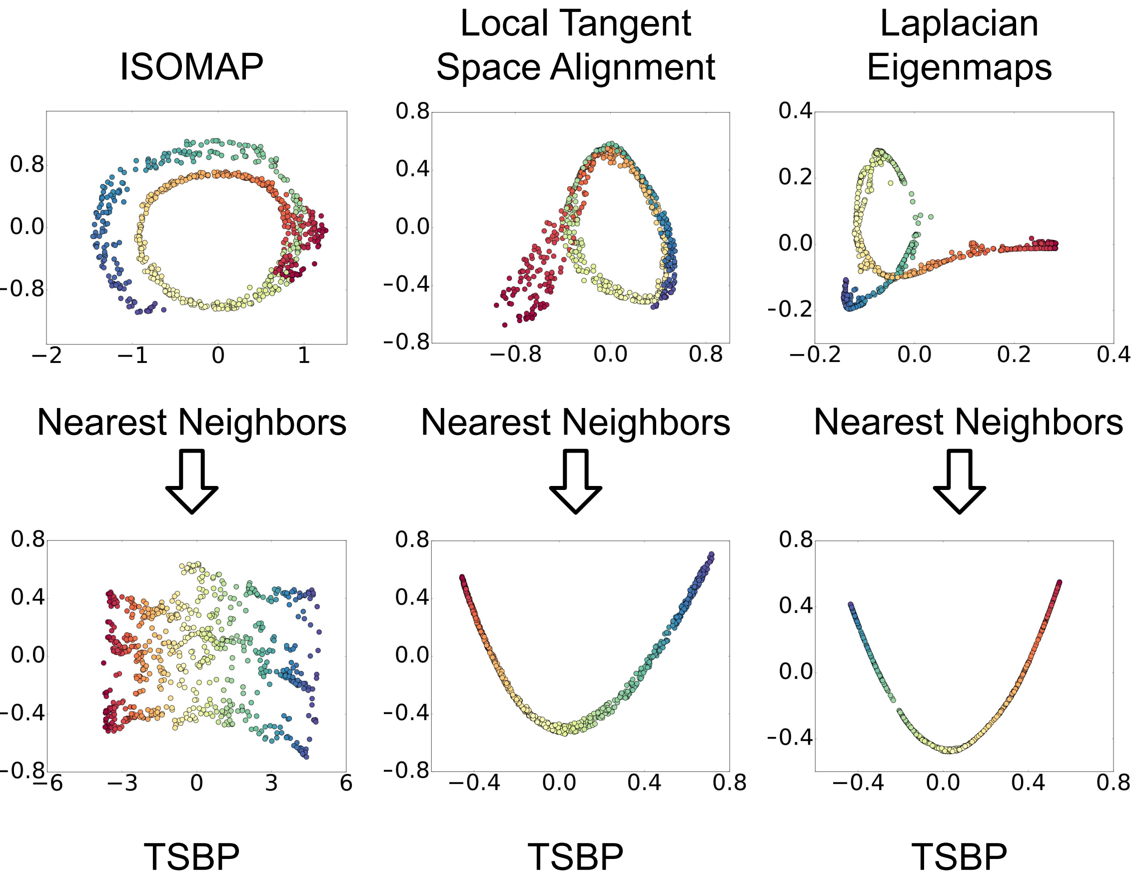


TSBP

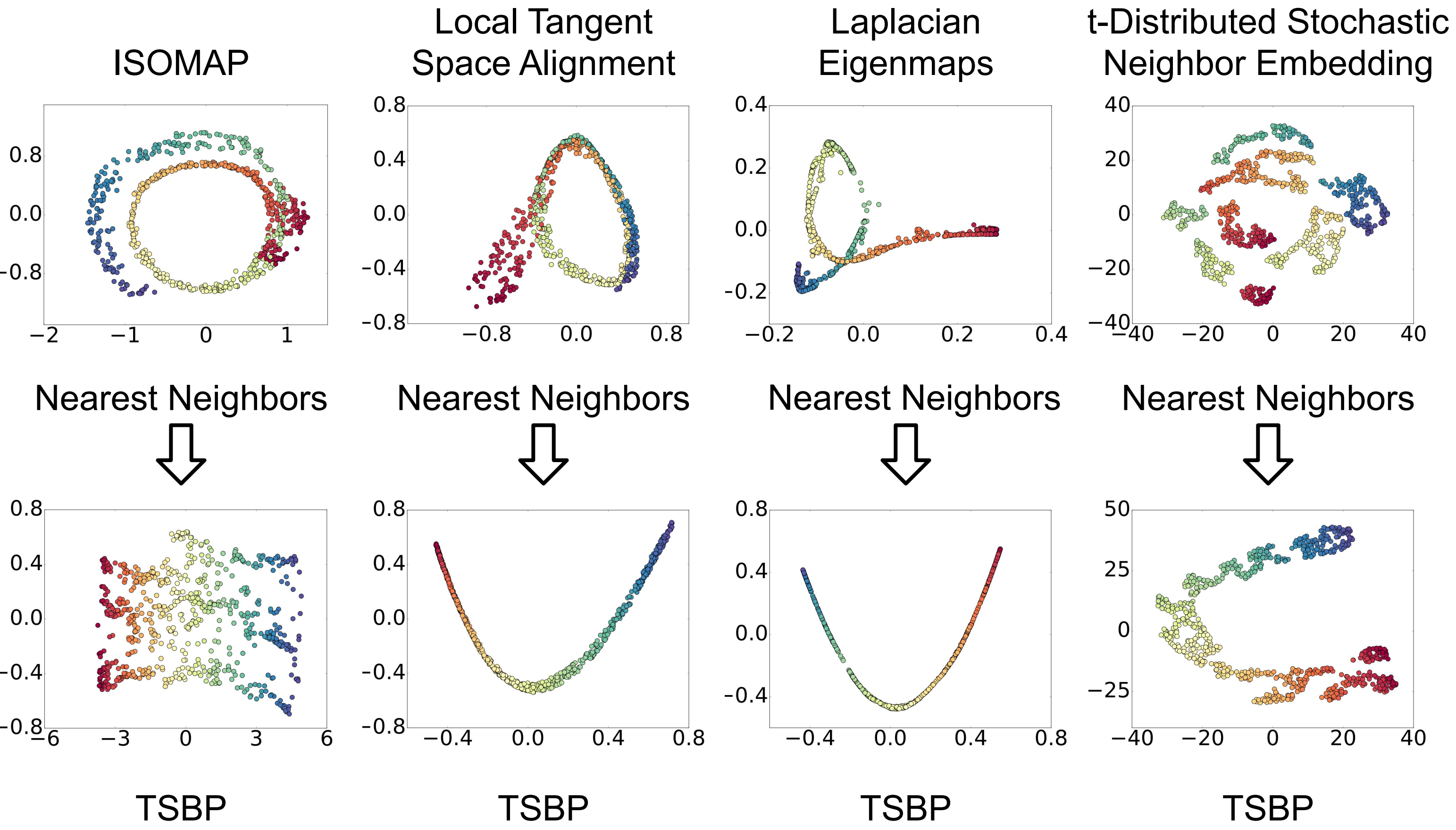


TSBP

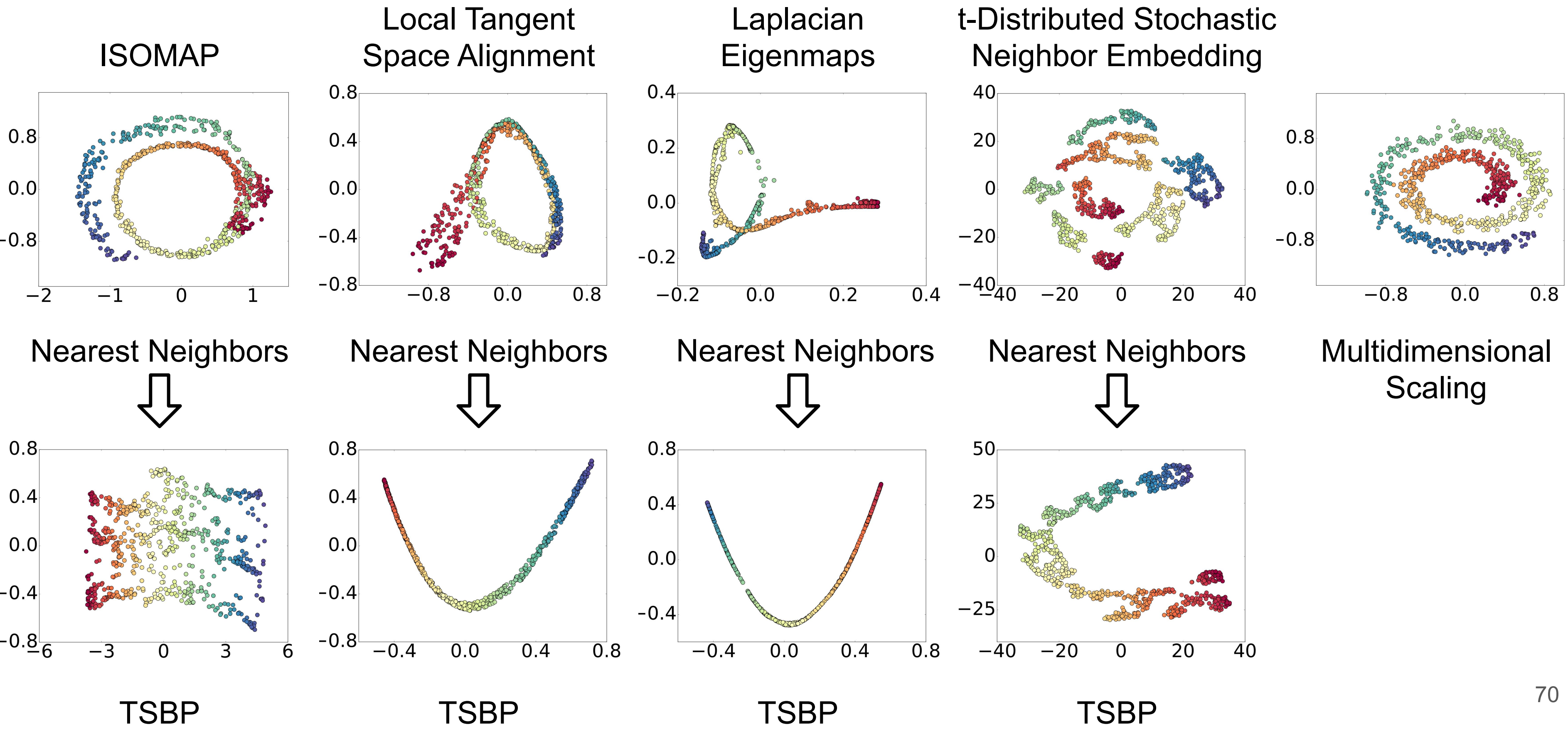
Embedding Evaluation



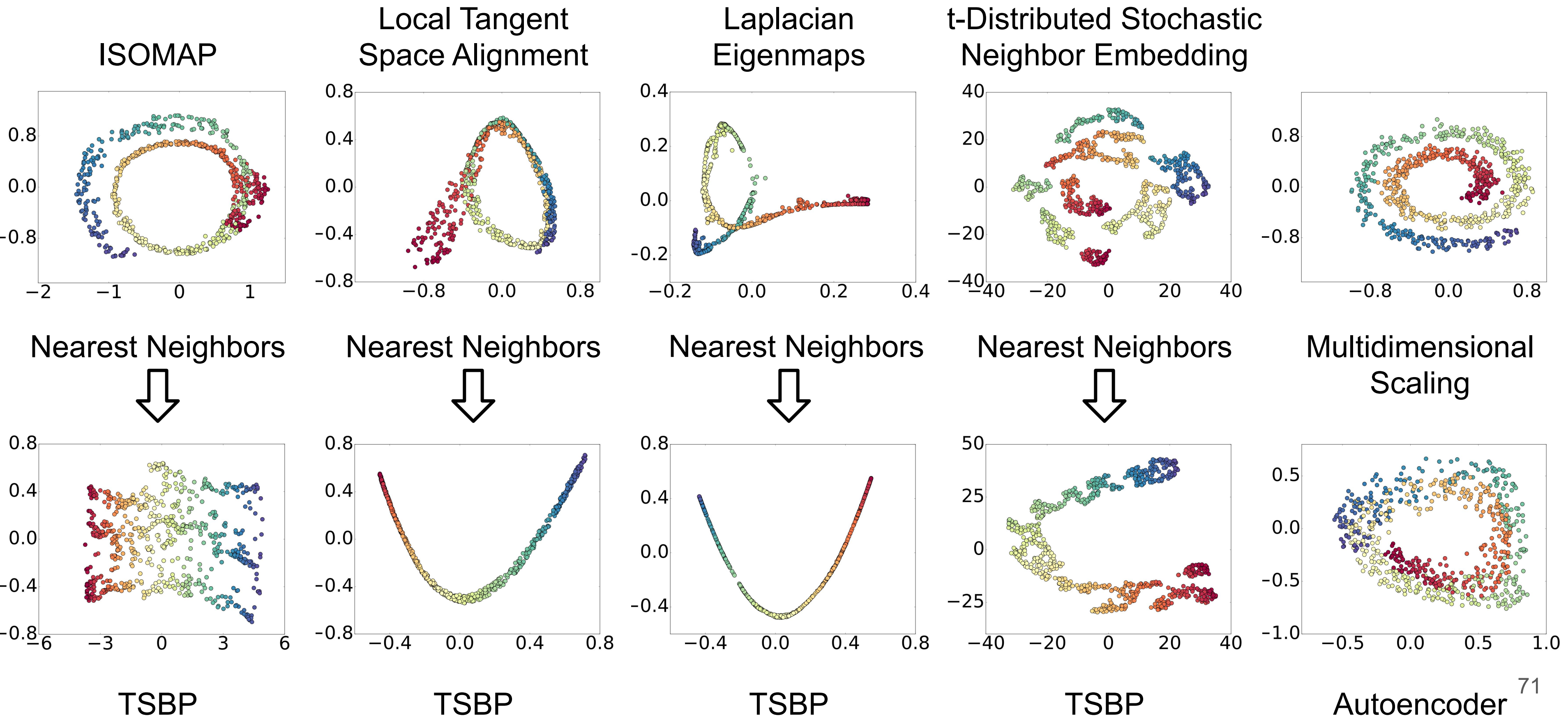
Embedding Evaluation



Embedding Evaluation



Embedding Evaluation



High-Dimensional Tactile Data

- Tactile sensing can be useful for intricate grasping and manipulation
- Scalable Tactile Glove (STAG) gathers detailed information from human grasp
- 548 force sensors (high dimensional data)
- Perform object classification with a Deep Convolutional Neural Network (Deep CNN)

IMAGE: *Learning the signatures of the human grasp using a scalable tactile glove* (S Sundaram et al)

Dimensionality Reduction and Classification

- Want to perform object classification without large amounts of training data
  - 26 objects (plus empty hand)
  - 24 samples per class
  - 50/50 train-test split
- Use a Gaussian Process Classifier (GPC) with RBF Kernel
  - Kernel relies on Euclidean metric, which performs poorly in high dimensional spaces
- Perform dimensionality reduction with t-SNE

Tactile Data Classification Results

	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370						
Top-3 Accuracy	0.1111						

Tactile Data Classification Results

	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370	0.0402					
Top-3 Accuracy	0.1111	0.1140					

Tactile Data Classification Results

	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370	0.0402	0.0421				
Top-3 Accuracy	0.1111	0.1140	0.1156				

Tactile Data Classification Results

	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370	0.0402	0.0421	0.0380			
Top-3 Accuracy	0.1111	0.1140	0.1156	0.1120			

Tactile Data Classification Results

	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370	0.0402	0.0421	0.0380	0.0612		
Top-3 Accuracy	0.1111	0.1140	0.1156	0.1120	0.1551		

Tactile Data Classification Results

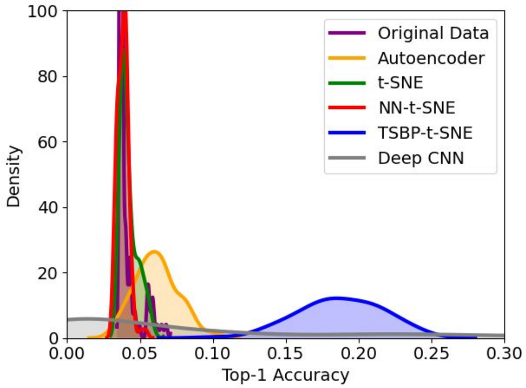
	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370	0.0402	0.0421	0.0380	0.0612	0.0897	
Top-3 Accuracy	0.1111	0.1140	0.1156	0.1120	0.1551	0.1888	

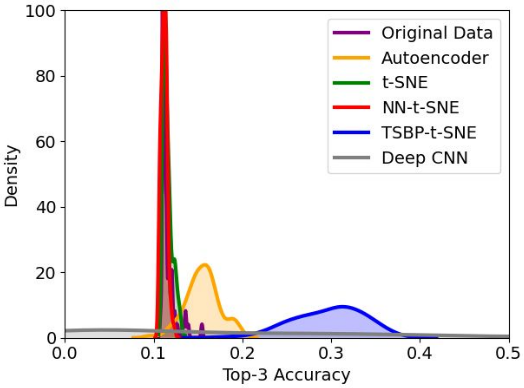
Tactile Data Classification Results

	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370	0.0402	0.0421	0.0380	0.0612	0.0897	**0.1874**
Top-3 Accuracy	0.1111	0.1140	0.1156	0.1120	0.1551	0.1888	**0.2972**

Tactile Data Classification Results

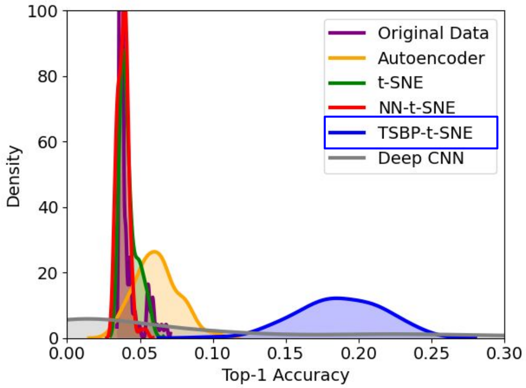
	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370	0.0402	0.0421	0.0380	0.0612	0.0897	**0.1874**
Top-3 Accuracy	0.1111	0.1140	0.1156	0.1120	0.1551	0.1888	**0.2972**

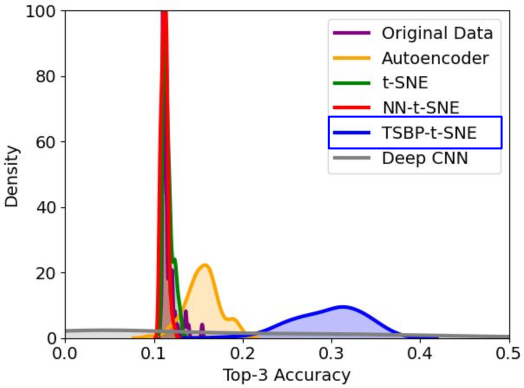




Tactile Data Classification Results

	Random Chance	GPC with Original Data	GPC with Ordinary t-SNE	GPC with Nearest-Neighbors t-SNE	GPC with Autoencoder Embedding	Deep CNN	GPC with TSBP t-SNE
Top-1 Accuracy	0.0370	0.0402	0.0421	0.0380	0.0612	0.0897	**0.1874**
Top-3 Accuracy	0.1111	0.1140	0.1156	0.1120	0.1551	0.1888	**0.2972**





Conclusion

- High dimensional data is inherently difficult to work with
- Manifold learning is one solution to this "Curse of Dimensionality"
- Sparse and noisy data can cause manifold learning to fail

Conclusion

- High dimensional data is inherently difficult to work with
- Manifold learning is one solution to this "Curse of Dimensionality"
- Sparse and noisy data can cause manifold learning to fail
- TSBP makes manifold learning more robust
  - Accurate tangent space estimates are obtained with belief propagation
  - False edges can be identified by comparing with manifold tangents
  - These edges are removed to produce a denoised neighborhood graph
  - This allows existing manifold learning algorithms to produce a more accurate embedding

TSBP: Tangent Space Belief Propagation for Manifold Learning

Thomas Cohn, Odest Chadwicke Jenkins, Karthik Desingh, Zhen Zeng

IROS 2020

Laboratory for
Perception **RO**botics and **G**rounded **RE**asoning **S**ystem**S**

UNIVERSITY OF MICHIGAN

Potential Functions

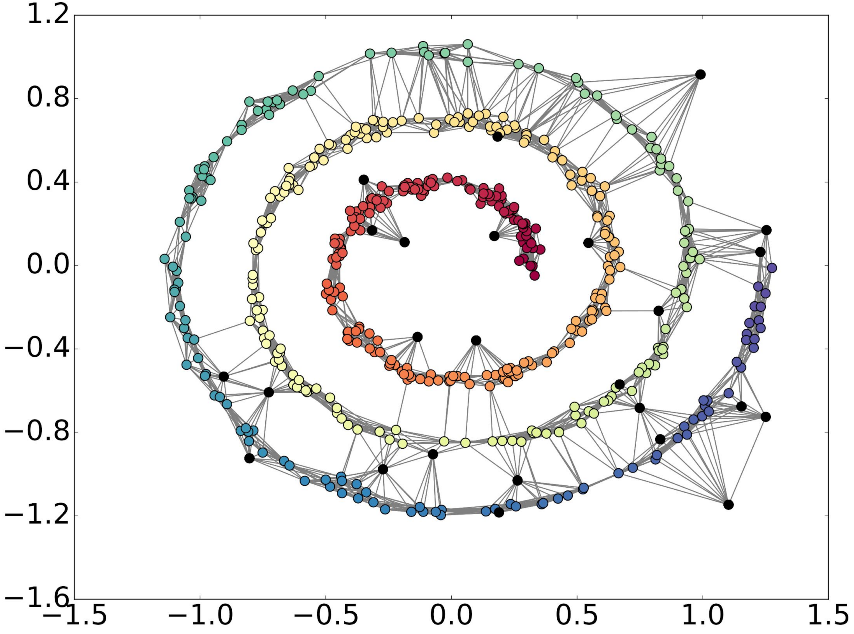
Given bases: $U = (u_1, \ldots, u_k),\ V = (v_1, \ldots, v_k)$

Vector subspace dissimilarity: $$\Gamma(U, V) = \sum_{i=1}^{k} ||u_i - \mathrm{proj}_V\, u_i||^2$$

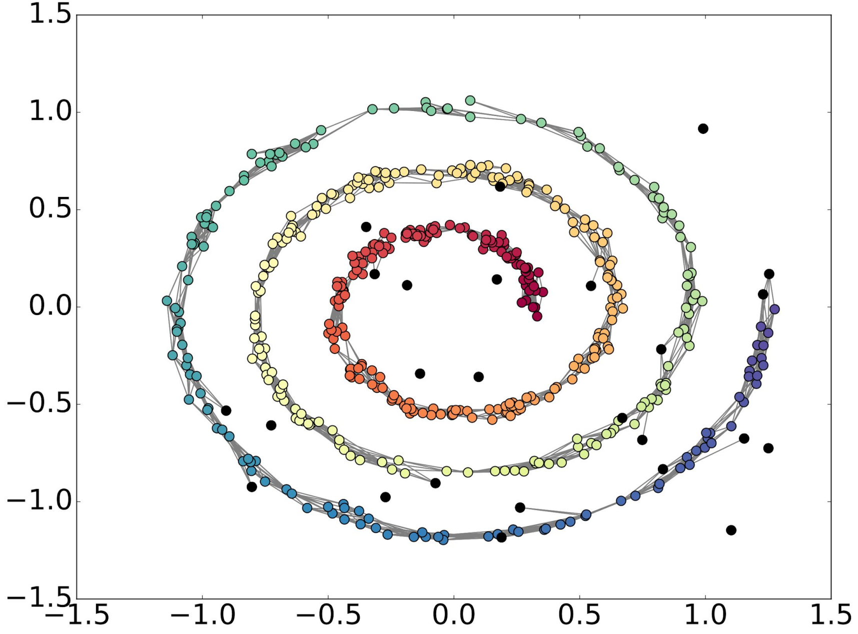
Unary Potential: $\phi(\mathcal{T}_i, \mathcal{Y}_i) = (1 + \Gamma(\mathcal{T}_i, \mathcal{Y}_i))^{-1}$

Pairwise Potential: $\psi(\mathcal{T}_i, \mathcal{T}_j) = (1 + \Gamma(\mathcal{T}_i, \mathcal{T}_j))^{-1}$

Outliers

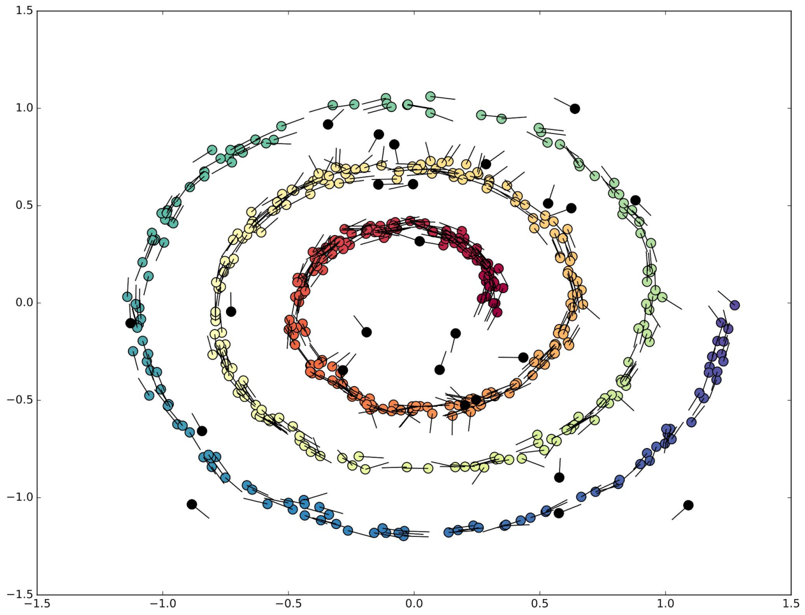


k-Nearest Neighbors

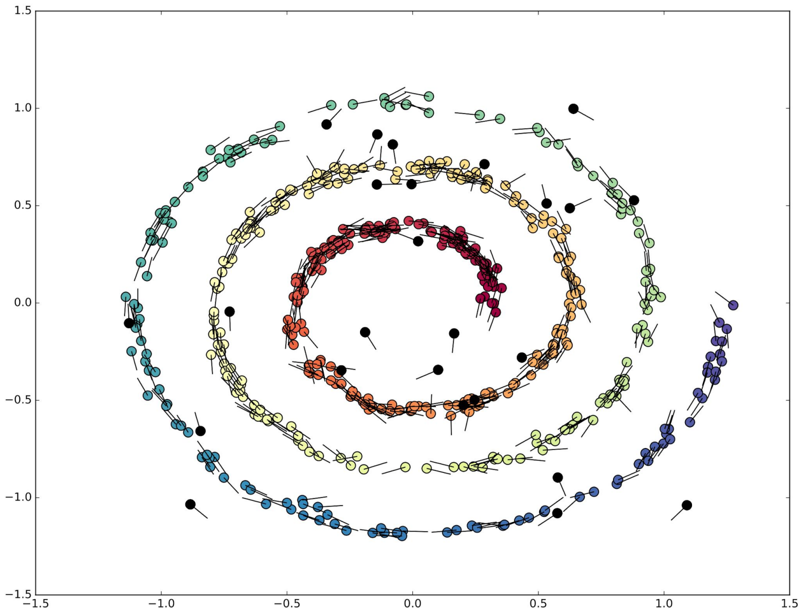


TSBP

L2-L1 PCA Comparison



L2-PCA



L1-PCA