Piecewise-Linear Manifolds for Deep Metric Learning

Shubhang Bhatnagar, Narendra Ahuja
University of Illinois Urbana-Champaign

CPAL

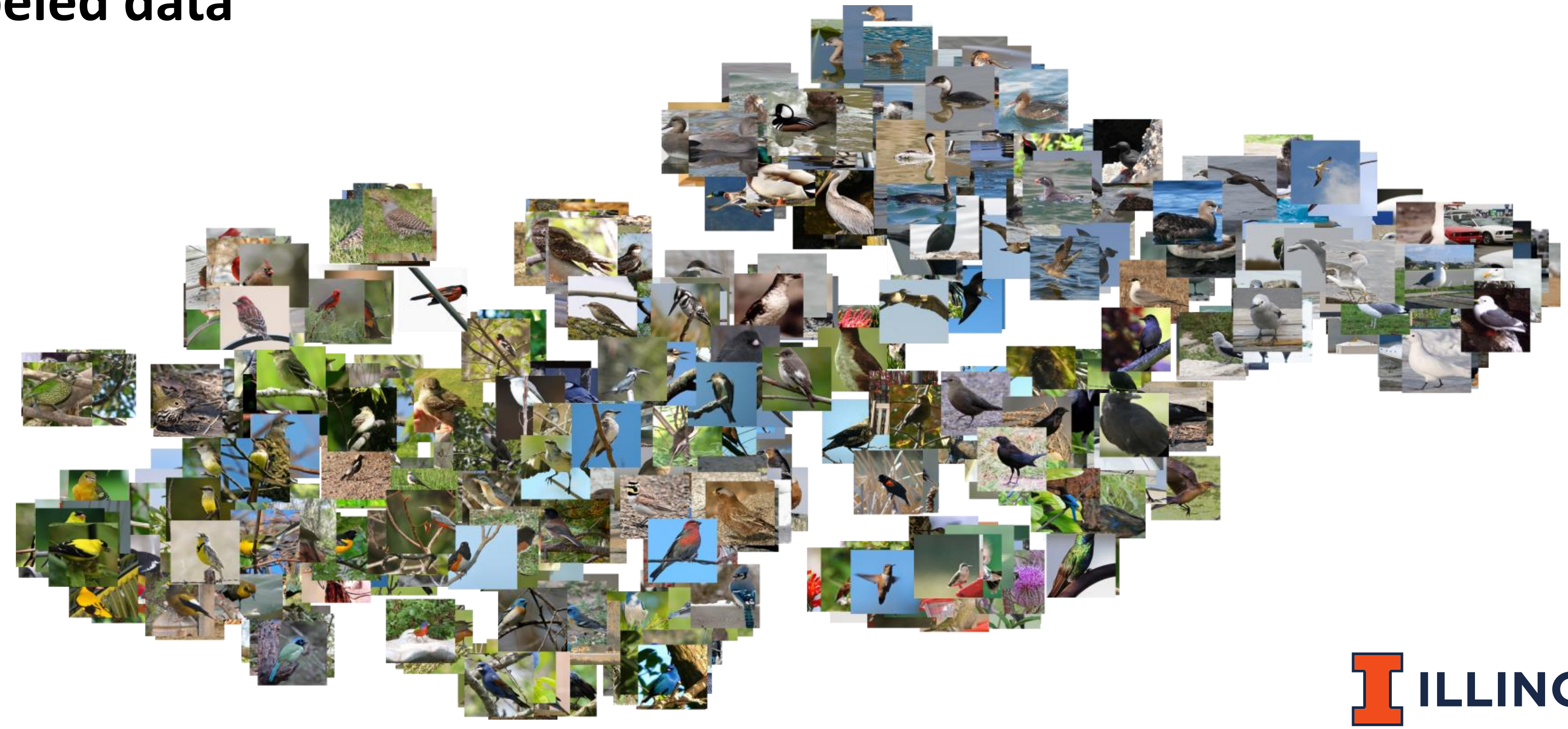
ILLINOIS

Unsupervised Deep Metric Learning: Goal

Learn a low dimensional representation

- **Where semantic similarity encoded as distance**
  (higher distance in representation space = more dissimilarity)
- **Using only unlabeled data**

A t-sne visualization of a semantic representation space learnt using metric learning on the CUB-200 dataset



ILLINOIS

Unsupervised Deep Metric Learning: How?

Extract features using Pre-trained network

Cluster extracted features to identify similar subsets (pseudo-labels)

Fine-tune network using extracted pseudo-labels

Challenges

Large domain gaps (between Pre-training and DML data)

Erroneous Clusters

Poor clusters lead to poor models

Need better models in feature space!

Method: Piecewise Linear manifolds

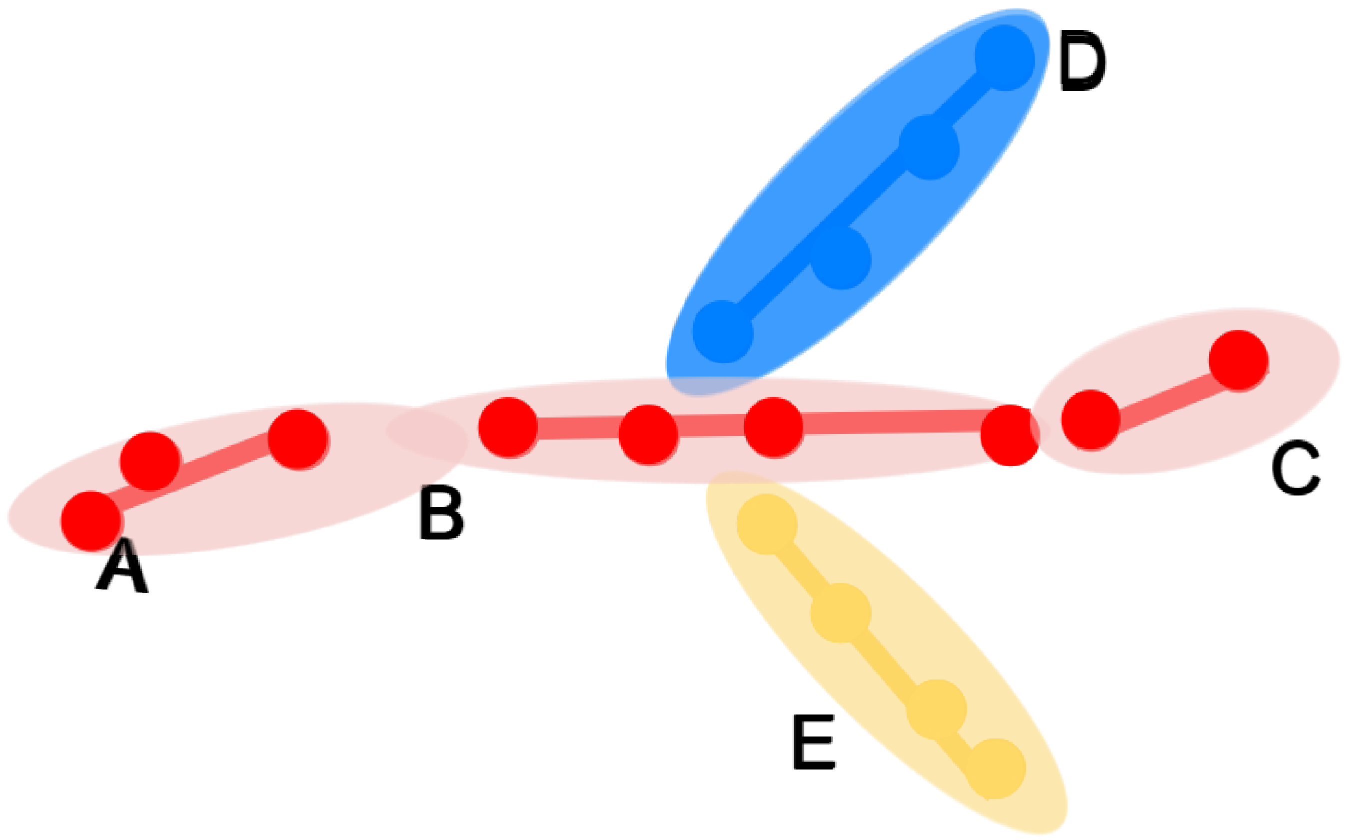
Obtain a Piecewise Linear model
of the data manifold

Train network using semantic similarity
information extracted from model

Update Embeddings
to improve Piecewise Linear model

Better Class models than clustering

Method: Intuition



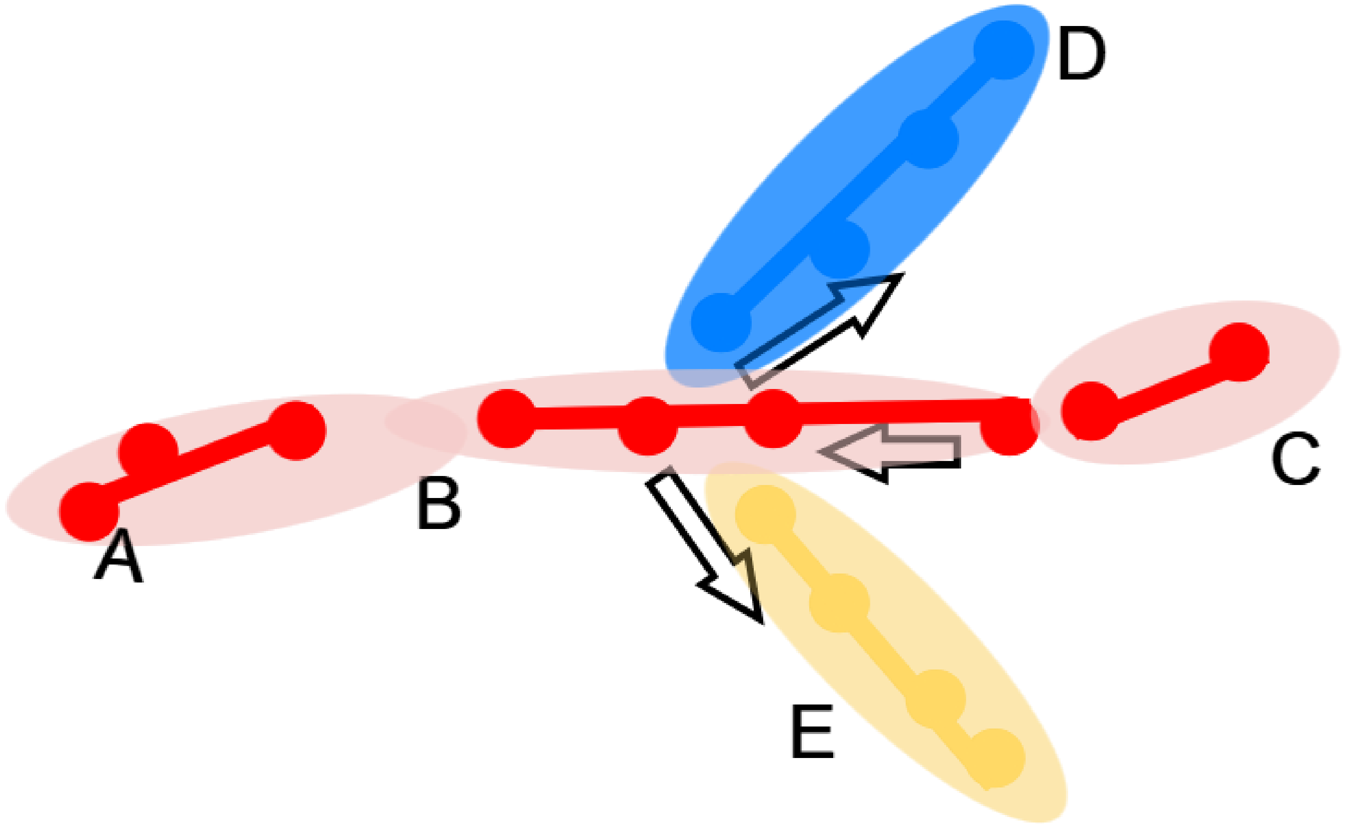
2D visualization of features extracted from a toy dataset

Our method uses a 3 step process

Step 1

- Grow neighborhood around each point
- Approximate it with a linear model
- Add all points that do not breach error threshold

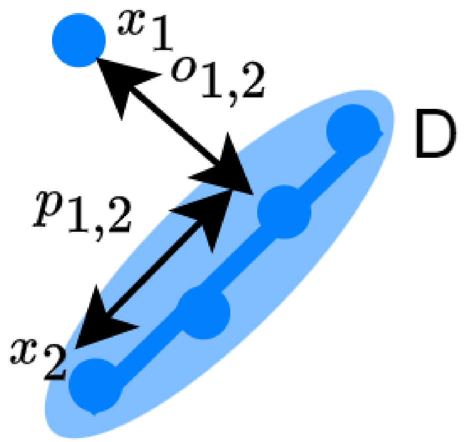
Method: Intuition



Step 2

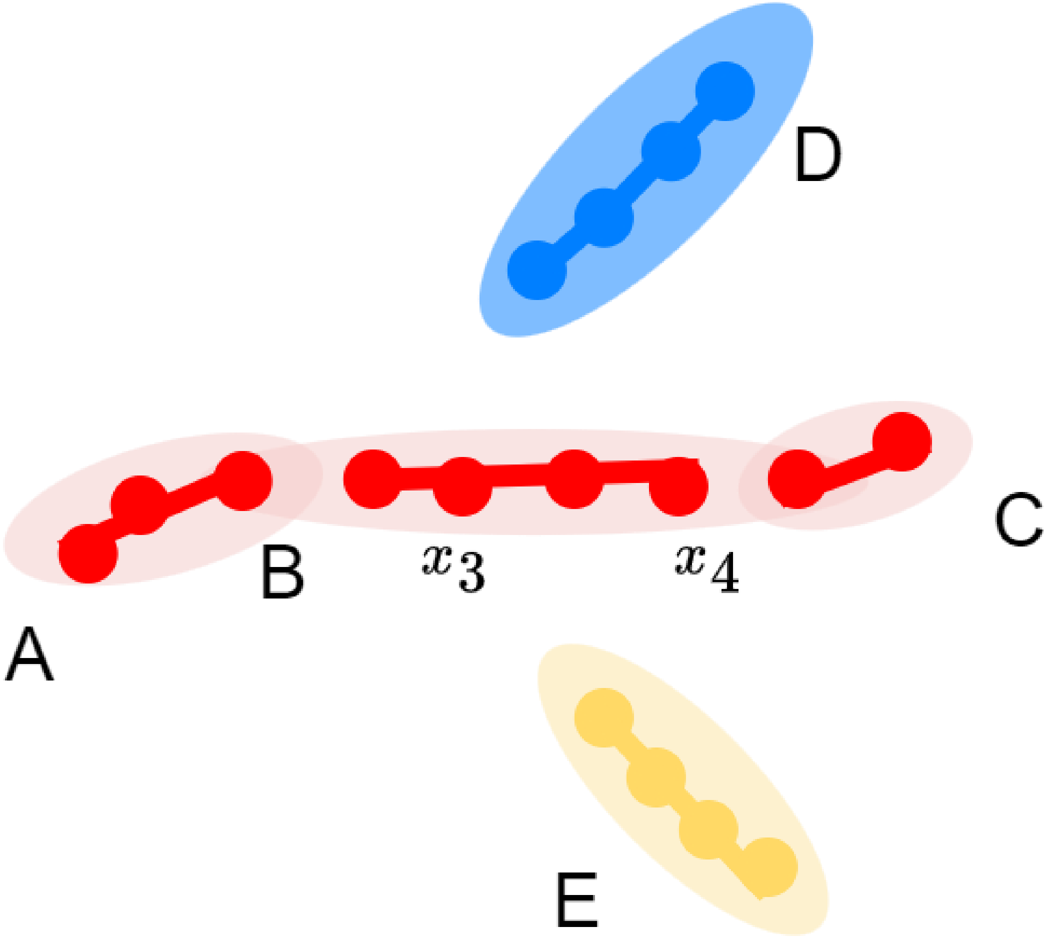
Similarity Calculation between x_1, x_2

- Similarity decays both along D and perpendicular to it
- Faster decay in similarity perpendicular to D



ILLINOIS

Method: Intuition

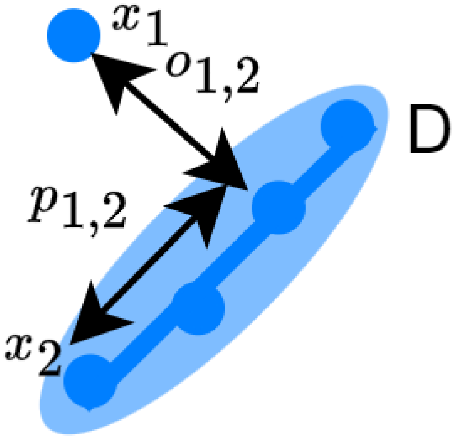


Step 3
Update features using our algorithm

Step 2

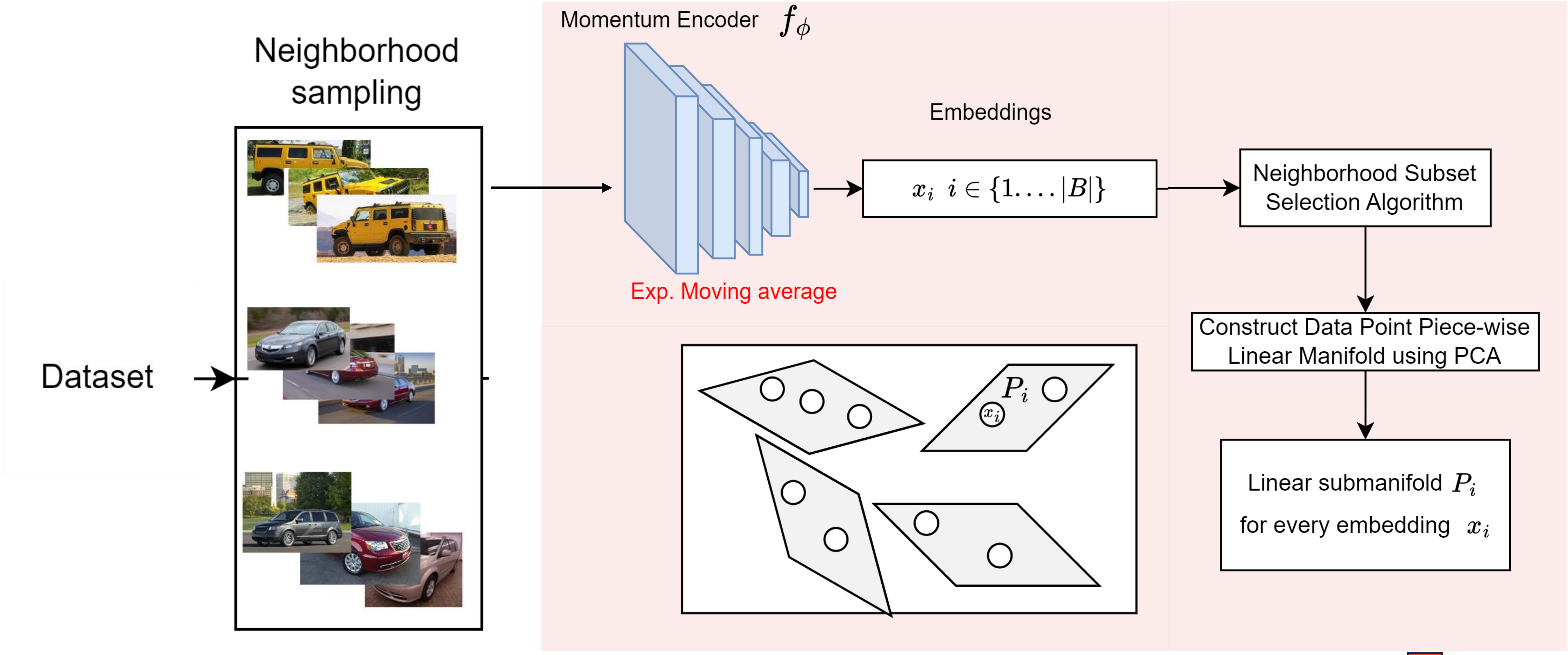
Similarity Calculation between x_1, x_2

- Similarity decays both along D and perpendicular to it
- Faster decay in similarity perpendicular to D



ILLINOIS

Method: Implementation



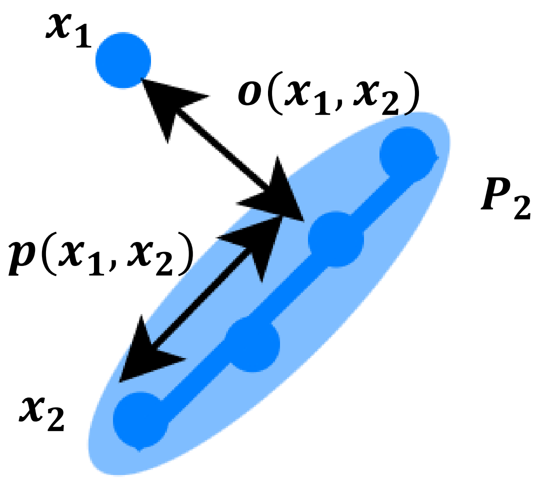
Method: Estimating Similarity

- Calculate similarity between each pair of points using the piecewise linear manifold model

- Similarity $s'(x_1, x_2)$ is a **decreasing** function of
  1. Orthogonal distance of x_1 from linear submanifold P_2
  2. Distance of x_2 from the projection of x_1 on P_2

- Functional form of the similarity on distance is chosen to be reciprocal (≈other forms)
- Steeper decrease with orthogonal distance

- Normalized to (0,1)



$$s'(\mathbf{x}_i, \mathbf{x}_j) = \alpha(\mathbf{x}_i, \mathbf{x}_j)\beta(\mathbf{x}_i, \mathbf{x}_j)$$

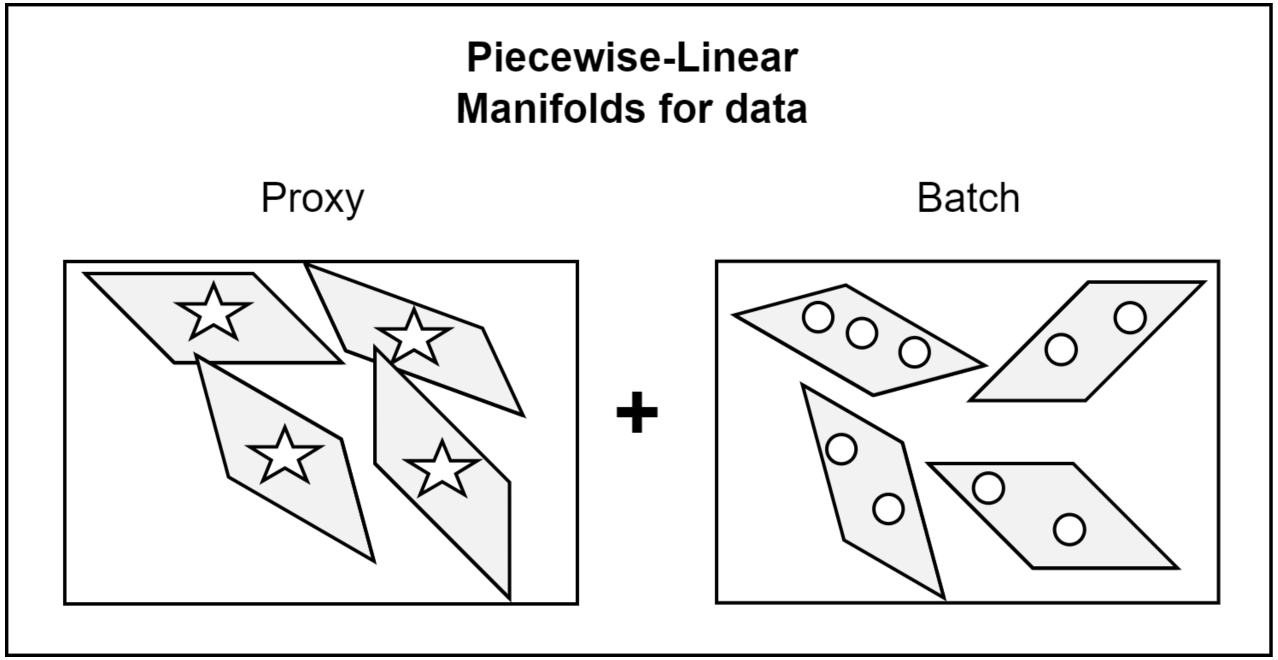
$$\alpha(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{(1 + \frac{o(\mathbf{x}_i, \mathbf{x}_j)}{2})^{N_\alpha}}$$

$$\beta(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{(1 + p(\mathbf{x}_i, \mathbf{x}_j))^{N_\beta}}$$

ILLINOIS

Method: Proxy Manifold

- Batch may not represent complete data manifold
- Proxies help cover the gaps.
- Proxies
  - Represent linear approximation of local manifold
  - Are treated like data embeddings
  - Learnt using backpropagation
- First work demonstrating their use & benefits in an unsupervised metric learning



Method: Training

- Loss has 2 main functions
  1. **Updates embedding network:**

Matches

Dissimilarity Euclidean Distance

$$\left(\ \delta \times (1 - s(\mathbf{x}_i, \mathbf{x}_j)\ ) - \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2\right)^2$$

(between point embeddings)

  2. **Updates Proxy manifold**

Aligns proxy manifold towards most similar data points in current batch

$$\mathcal{L} = \boxed{\mathcal{L}_{point} + \mathcal{L}_{proxy}} \boxed{+ \mathcal{L}_{neighborhood}}$$

ILLINOIS

Results: Image Retrieval

- Embedding learnt tested on zero shot image retrieval
  - Cars-196
  - CUB-200-2011
  - Stanford Online Products

- GoogleNet backbone with
  - 128 dim embedding
  - 512 dim embedding

- Performance measured using Recall@K

Results: Image Retrieval using R@K

Benchmarks →	**CUB-200-2011**				**Cars-196**				**SOP**		
Methods ↓	R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8	R@1	R@10	R@100
GoogleNet (128 dim)											
Examplar [38]	38.2	50.3	62.8	75.0	36.5	48.1	59.2	71.0	45.0	60.3	75.2
NCE [25]	39.2	51.4	63.7	75.8	37.5	48.7	59.8	71.5	46.6	62.3	76.8
DeepCluster [17]	42.9	54.1	65.6	76.2	32.6	43.8	57.0	69.5	34.6	52.6	66.8
MOM [24]	45.3	57.8	68.6	78.4	35.5	48.2	60.6	72.4	43.3	57.2	73.2
AND [39]	47.3	59.4	71.0	80.0	38.4	49.6	60.2	72.9	47.4	62.6	77.1
ISIF [26]	46.2	59.0	70.1	80.2	41.3	52.3	63.6	74.9	48.9	64.0	78.0
sSUML [40]	43.5	56.2	68.3	79.1	42.0	54.3	66.0	77.2	47.8	63.6	78.3
Ortho [41]	47.1	59.7	72.1	82.8	45.0	56.2	66.7	76.6	45.5	61.6	77.1
PSLR [42]	48.1	60.1	71.8	81.6	43.7	54.8	66.1	76.2	51.1	66.5	79.8
ROUL [36]	56.7	68.4	78.3	86.3	45.0	56.9	68.4	78.6	53.4	68.8	81.7
SAN [43]	55.9	68.0	78.6	86.8	44.2	55.5	66.8	76.9	58.7	73.1	84.6
STML* [19]	57.7	69.8	80.1	87.1	48.0	58.7	69.5	79.5	63.8	77.8	88.9
Ours	**60.6 ± 0.3**	**71.1 ± 0.2**	**81.1 ± 0.1**	**87.8 ± 0.1**	**49.5 ± 0.3**	**60.6 ± 0.3**	**72.1 ± 0.2**	**80.9 ± 0.2**	**65.1 ± 0.3**	**80.4 ± 0.2**	**90.2 ± 0.1**
GoogleNet (512 dim)											
UDML-SS [23]	54.7	66.9	77.4	86.1	45.1	56.1	66.5	75.7	63.5	78.0	88.6
TAC-CCL [16]	57.5	68.8	78.8	87.2	46.1	56.9	67.5	76.7	63.9	77.6	87.8
UHML [15]	58.9	70.6	80.4	87.7	47.7	58.9	70.3	80.3	65.1	78.2	88.3
STML* [19]	58.6	70.2	80.9	87.9	48.6	60.4	71.3	80.8	65.1	79.7	89.1
Ours	**61.7 ± 0.3**	**72.5 ± 0.2**	**82.2 ± 0.2**	**88.3 ± 0.1**	**51.2 ± 0.2**	**62.2 ± 0.2**	**72.1 ± 0.2**	**81.0 ± 0.1**	**66.4 ± 0.2**	**81.1 ± 0.1**	**90.6 ± 0.1**

Results: Quality of Extracted Similarity

- Compare quality of similarity estimated by our method with previous methods using
  1) Correlation of similarity with ground truth
  2) Purity of labels within the submanifolds

Metric →	**Label Purity**			**Correlation with Ground truth**		
Methods ↓	CUB200	Cars196	SOP	CUB200	Cars196	SOP
K-Means [17]	0.38	0.29	0.32	0.37	0.21	0.32
Hierarchical Clustering [15]	0.49	0.33	0.39	0.52	0.36	0.49
Random Walk [24]	-	-	-	0.45	0.26	0.42
Ours	**0.67**	**0.45**	**0.62**	**0.61**	**0.45**	**0.67**

Results: Ablation

- Subset selection for constructing linear submanifolds improves performance!

Method	R@1	R@2
Ours without subset selection	58.4	70.8
Ours	**61.7**	**72.5**

- Using a continuous valued similarity improves performance

Method	R@1	R@2
Ours with binary similarity	54.2	66.5
Ours	**61.7**	**72.5**

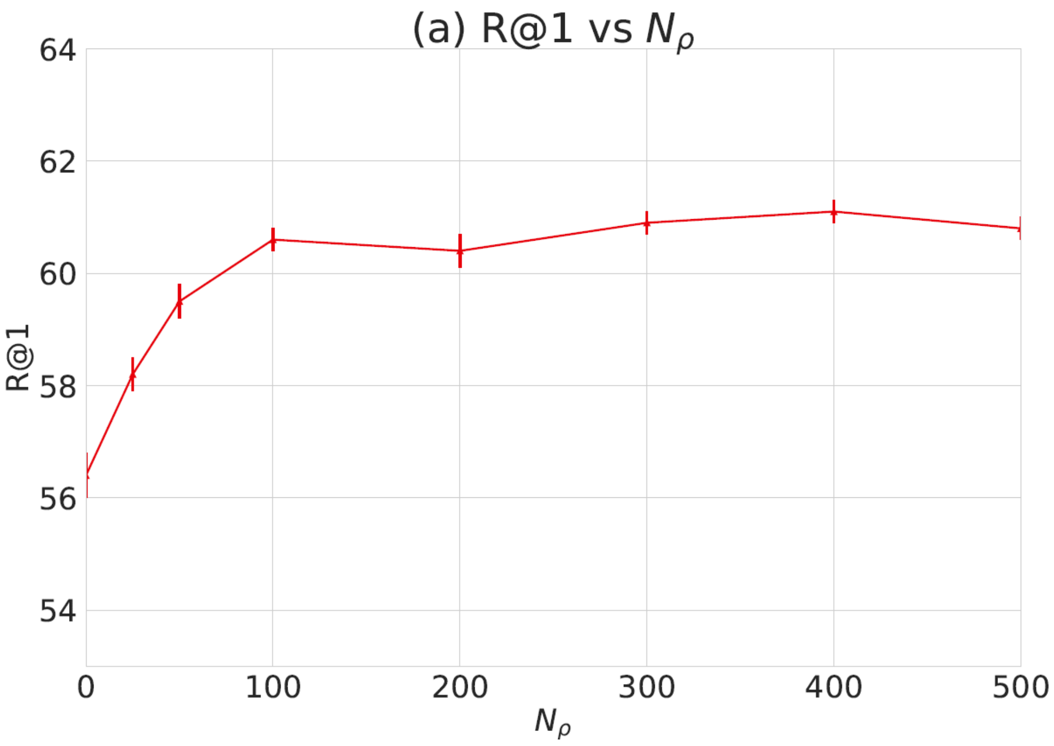
Recall@K reported on the CUB-200-2011 dataset averaged over 5 runs

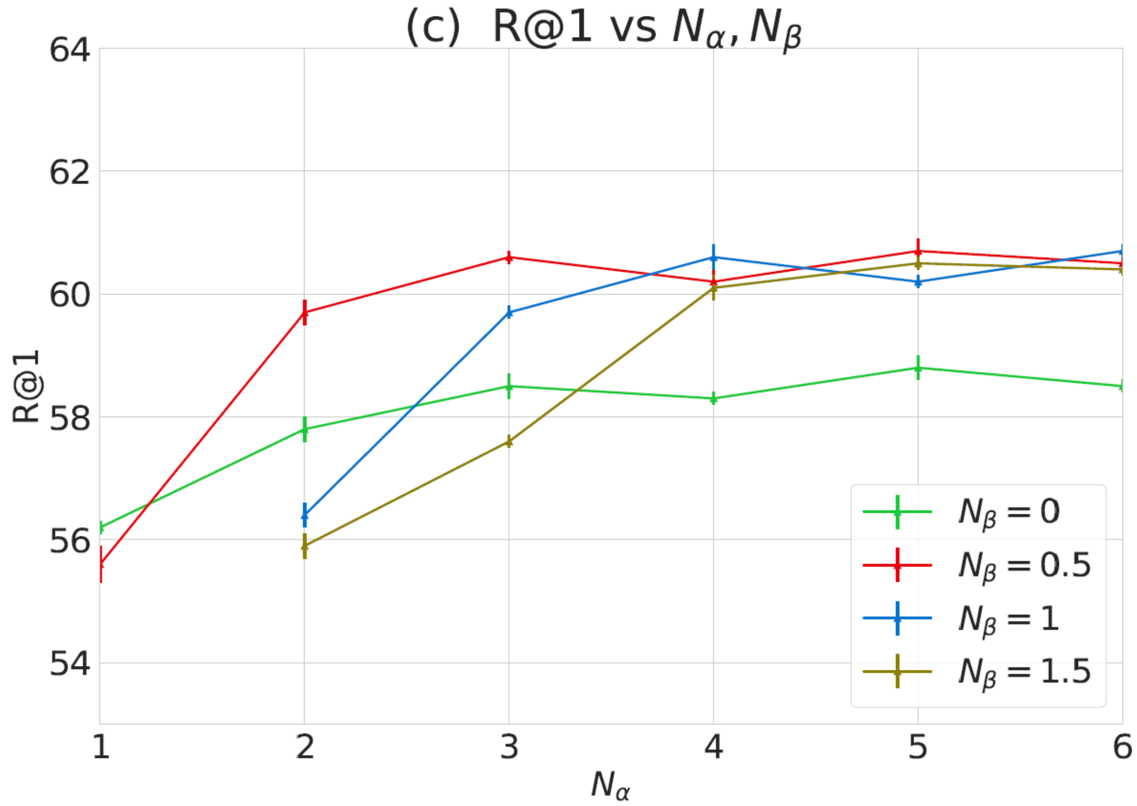
ILLINOIS

Results: Ablation

More Proxies -> better performance (till saturation)

- Similarity decay in both directions ($N_\alpha, N_\beta > 0$) outperforms decay in one
- Faster decay in orthogonal direction ($N_\alpha > N_\beta$) helps





Recall@1 reported on the CUB-200-2011 dataset averaged over 5 runs

ILLINOIS

Conclusion

- Proposed a novel method for unsupervised deep metric learning which
  - Uses a Piecewise Linear model of the data manifold
  - Estimates a continuous valued similarity
  - Makes use of proxies to augment model
  - Achieves state-of-the-art performance for zero-shot image retrieval
- Shows the importance of better modelling the structure of data in the feature space

ILLINOIS

Thank You!
Questions ?

ILLINOIS