

PROBLEM

Most hashing methods are designed to generate binary codes that preserve the Euclidean distance in the original space. Manifold learning techniques, in contrast, are better able to preserve the intrinsic geodesic distance. However, the following problems hinders the use of manifold learning for hashing:

1. Prohibitive computational cost
2. Out-of-sample extension problem – Most manifold learning methods are non-parametric.

Existing methods – All based on Laplacian eigenmaps

- Spectral Hashing: uniform data assumption
- Anchor Graph Hashing: Nyström extension
- Self-Taught Hashing: out-of-sample extension by SVM

CONTRIBUTIONS

We showed how to learn compact binary embeddings on their intrinsic manifolds. The proposed approach here is inspired by Delalleau et al.[2], where they have focused on semi-supervised classification. Our contributions include

1. Make semantic hashing on data manifolds *practical* by an inductive hashing framework
 - *Efficient*: Linear indexing time $O(n)$ and Constant query time $O(1)$
 - *Effective*: Better than L2 scan with t-SNE et al.
2. Connect manifold learning and hashing
 - Any manifold learning methods can be applied in the hashing framework.
 - Evaluation of 9 manifold learning methods for hashing

REFERENCES

- [1] F. Shen, C. Shen, Q. Shi, A. van den Hengel, Z. Tang. Inductive Hashing on Manifolds. In *IEEE Conf. Comp. Vis. Pattern Recogn.*, 2013.
- [2] O. Delalleau, Y. Bengio, and N. Le Roux. Efficient non-parametric function induction in semi-supervised learning. In *Proc. Int. Workshop Artif. Intelli. Stat.*, 2005.

FORMULATION

Denote the training data by $\mathbf{X} := \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ and their manifold embedding by $\mathbf{Y} := \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$. Given a new data point \mathbf{x}_q , we aim to generate an embedding \mathbf{y}_q which preserves the local neighborhood relationships:

$$\min \sum_{i=1}^n w(\mathbf{x}_q, \mathbf{x}_i) \|\mathbf{y}_q - \mathbf{y}_i\|^2, \quad (1)$$

where $w(\mathbf{x}_q, \mathbf{x}_i)$ is the similarity. which is only non-zero for its k nearest neighbors. This results in

$$\mathbf{y}_q^* = \frac{\sum_{i=1}^n w(\mathbf{x}_q, \mathbf{x}_i) \mathbf{y}_i}{\sum_{i=1}^n w(\mathbf{x}_q, \mathbf{x}_i)}. \quad (2)$$

This provides a simple inductive formulation for the embedding of a new data point by a linear combination of the base embeddings.

We developed a prototype algorithm which was able to approximate \mathbf{y}_q using only a small base set with a good bound: m clusters were used to cover \mathbf{Y} . Observing that the cluster centers have the largest overall weight w.r.t the points from their own cluster, i.e., $\sum_{i \in I_j} w(\mathbf{c}_j, \mathbf{x}_i)$, we then approximately select all cluster centers to express $\hat{\mathbf{y}}_q$ for efficiency.

We obtain our general inductive hash function by binarizing the low-dimensional embedding

$$h(\mathbf{x}) = \text{sgn} \left(\frac{\sum_{j=1}^m w(\mathbf{x}, \mathbf{c}_j) \mathbf{y}_j}{\sum_{j=1}^m w(\mathbf{x}, \mathbf{c}_j)} \right), \quad (3)$$

where $\mathbf{Y}_B := \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m\}$ is the embedding for the base set $\mathbf{B} := \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m\}$, which is the cluster centers obtained by K-means. With this, the embedding for the training data becomes

$$\mathbf{Y} = \bar{\mathbf{W}}_{\mathbf{X}\mathbf{B}} \mathbf{Y}_B, \quad (4)$$

where $\bar{\mathbf{W}}_{\mathbf{X}\mathbf{B}}$ is defined such that $\bar{\mathbf{W}}_{ij} = \frac{w(\mathbf{x}_i, \mathbf{c}_j)}{\sum_{i=1}^m w(\mathbf{x}_i, \mathbf{c}_j)}$, for $\mathbf{x}_i \in \mathbf{X}$, $\mathbf{c}_j \in \mathbf{B}$. We term our hashing method *Inductive Manifold-Hashing* (IMH). For IMH, any manifold learning methods can be applied to generate the low dimensional embedding \mathbf{Y}_B as a base.

ALGORITHM

Algorithm 1: Inductive Manifold-Hashing

Input: Training data $\mathbf{X} := \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, code length r , base set size m , neighborhood size k

- 1 Generate the base set \mathbf{B} by random sampling or clustering (e.g., K-means);
- 2 Embed \mathbf{B} into the low dimensional space by any appropriate manifold learning method;
- 3 Obtain the low dimensional embedding \mathbf{Y} for the whole dataset inductively by (4);
- 4 Threshold \mathbf{Y} at zero;

Output: Binary codes

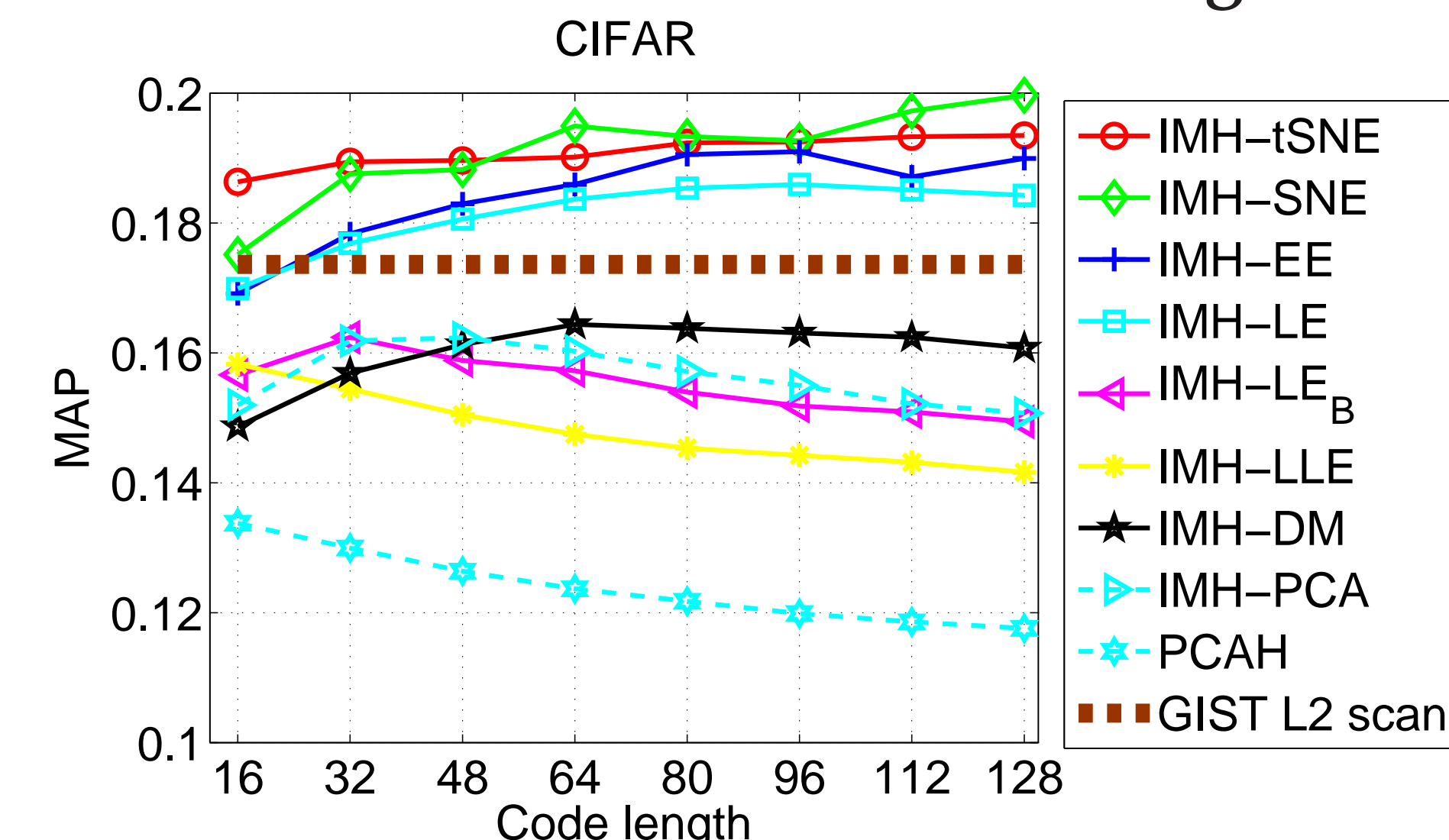
$$\mathbf{Y} := \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\} \in \mathbb{R}^{n \times r}$$

SOURCE CODE

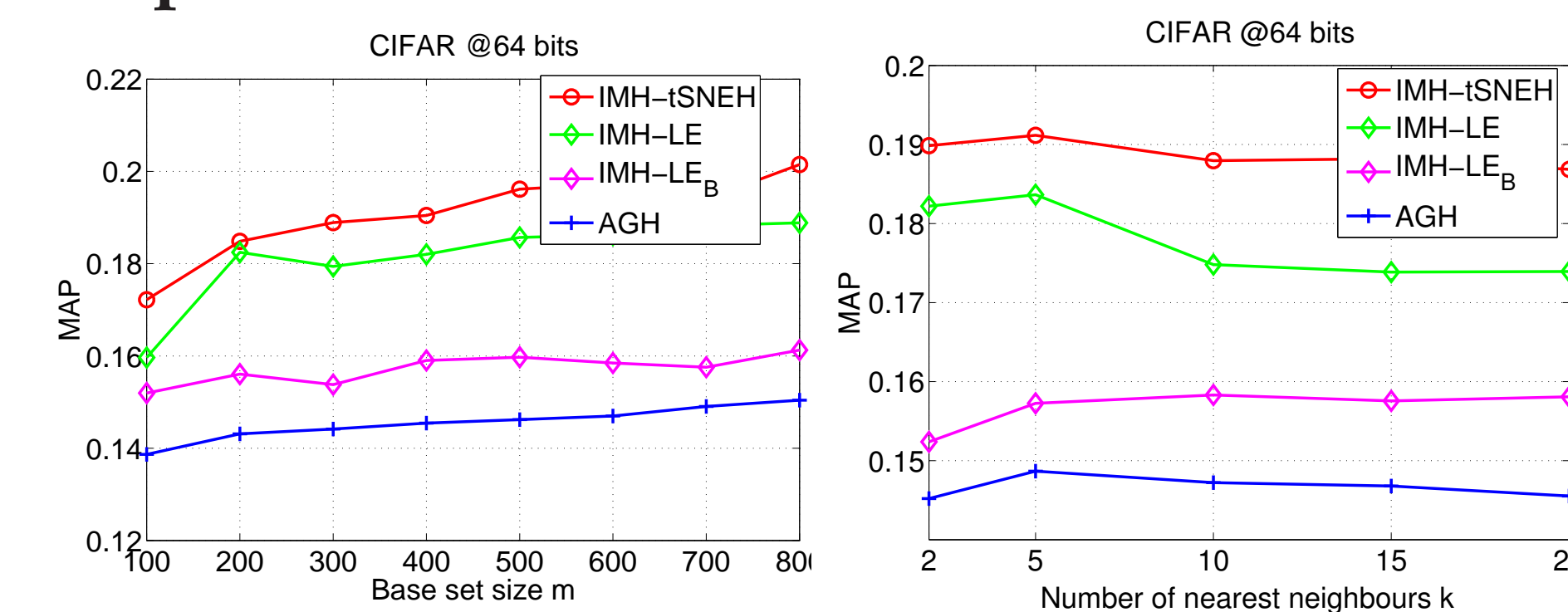
available at: <http://goo.gl/A9IFL>

EVALUATION

Evaluation of manifold learning methods



Impact of m and k

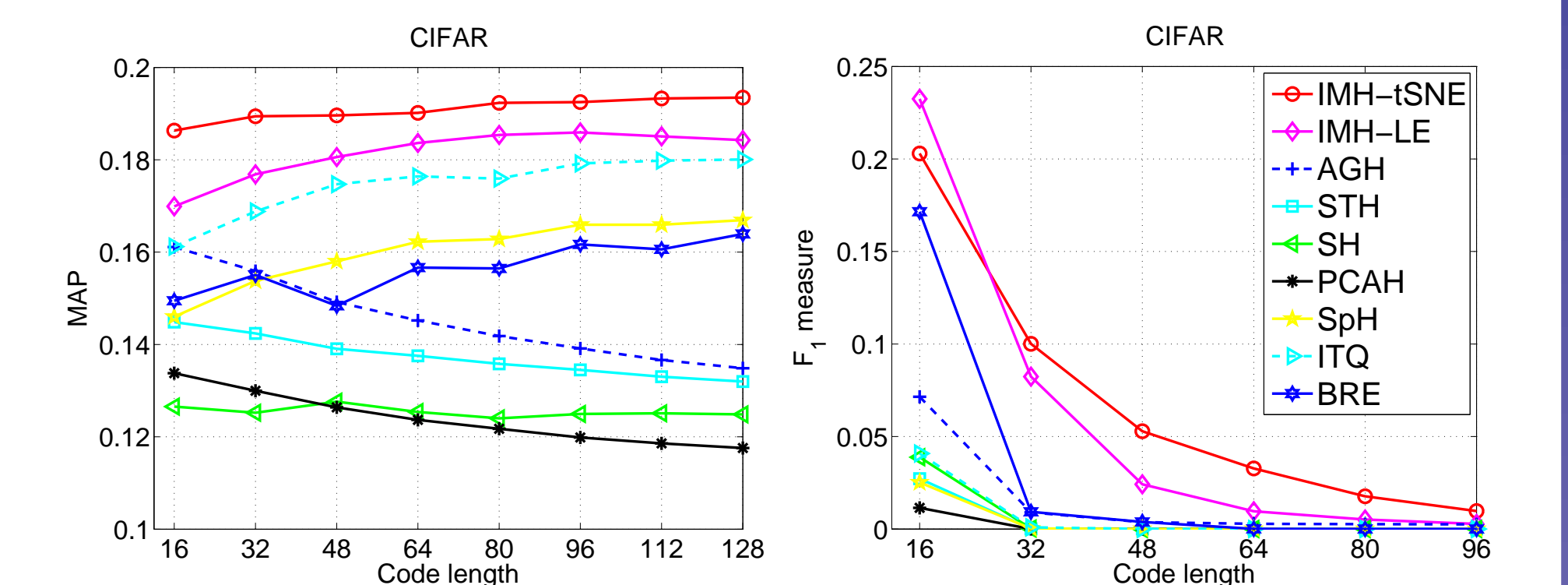


Random sampling vs. K-means on CIFAR-10

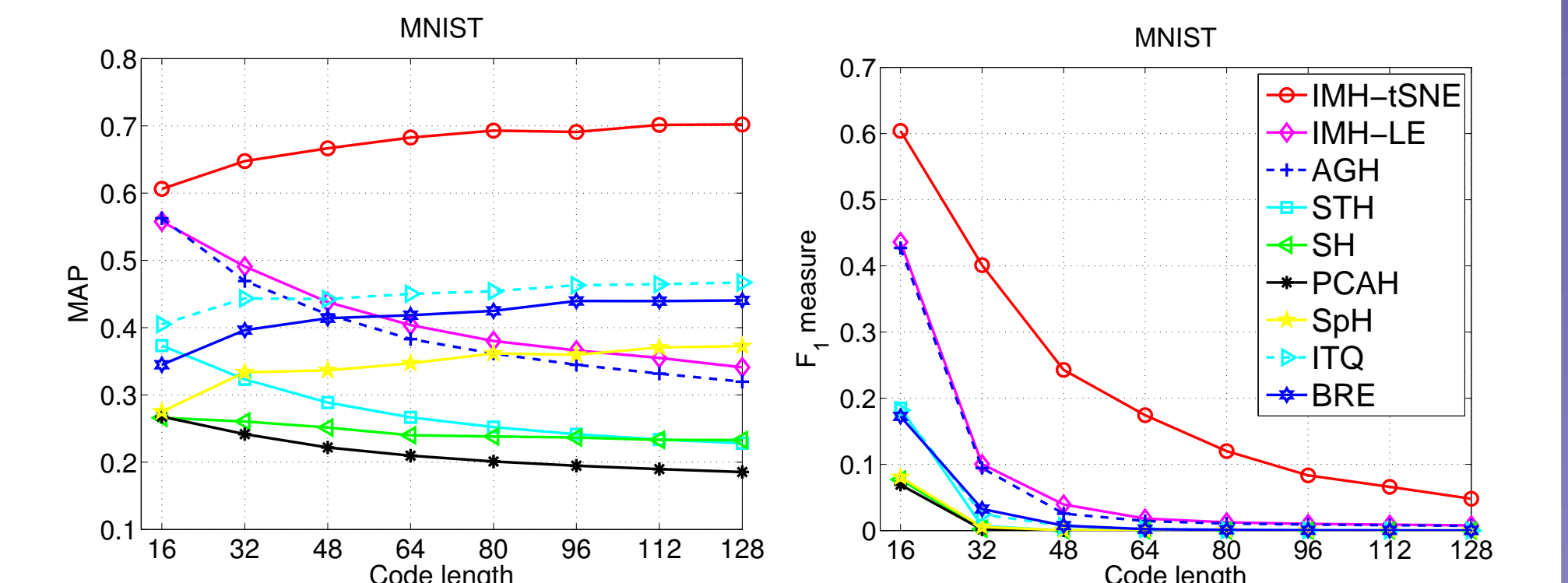
bits		IMH-LE	IMH-tSNE
32	Random	16.20	17.26
	K-means	17.48	18.38
64	Random	16.98	16.93
	K-means	18.20	19.04
96	Random	17.02	17.21
	K-means	18.56	19.41

RESULTS

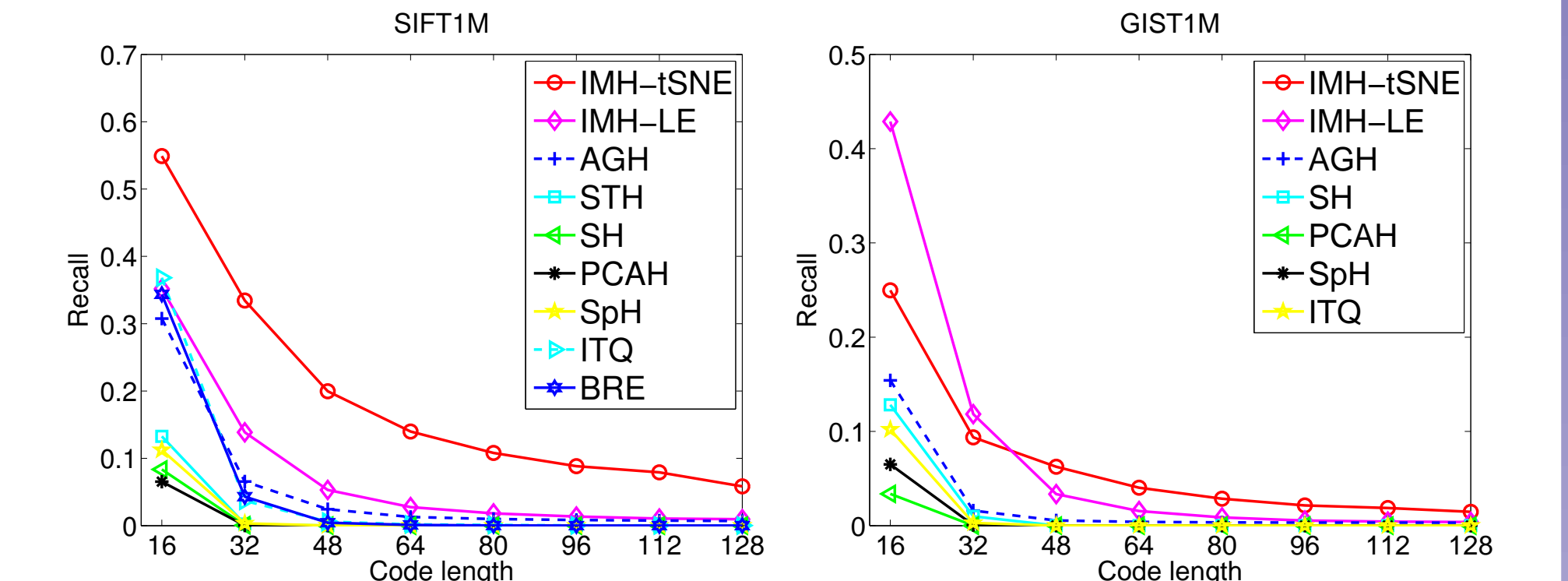
Retrieval results on CIFAR-10 (60K)



Retrieval results on MNIST (70K)



Retrieval results on SIFT1M and GIST1M



Computational times (seconds) on MNIST

Method	Train time		Test time	
	64-bits	128-bits	64-bits	128-bits
IMH-LE	9.9	9.9	5.1×10^{-5}	3.8×10^{-5}
IMH-tSNE	16.7	20.2	2.8×10^{-5}	3.1×10^{-5}
SH	6.8	16.2	5.8×10^{-5}	1.8×10^{-4}
STH	266.1	485.4	1.8×10^{-3}	3.6×10^{-3}
AGH	9.5	9.5	4.7×10^{-5}	5.5×10^{-5}
PCAH	3.8	4.1	5.7×10^{-6}	1.2×10^{-5}
SpH	19.7	41.0	1.3×10^{-5}	2.0×10^{-5}
ITQ	10.4	20.3	6.9×10^{-6}	1.1×10^{-5}
BRE	418.9	1731.9	1.2×10^{-5}	2.4×10^{-5}

Classification accuracy with linear SVM

