# Post Tuned Hashing: A New Approach to Indexing High-dimensional Data

## 背景

传统基于树的索引方法会遭受维度灾难（维度越高、邻居越少）。而哈希是一种将高维数据映射到保留相似性的二进制编码中的有效方法。通过二进制编码，最近邻搜索任务可以有效通过比特级操作完成。

二进制哈希技术可以大致分为**监督（半监督）**方法和**无监督**方法。监督方法将利用标签信息学习二进制编码，使其专门适用于语义搜索；无监督哈希方法通过探索数据的复杂特性学习二进制编码，可被有效用于无标签的数据。本文将关注于无监督哈希。

目前现存的方法遵从一个两阶段范式：在投射阶段会将原始数据映射到低纬度连续空间中；在二进制化阶段会将投射结果二进制化为最终的二进制编码。之前的研究主要关注于这两个阶段的改进与优化，尝试尽可能保留更多的邻居关系。大多数方法目前都想在投射阶段通过探索原始数据的不同属性（比如分布、配对关系、多种结构等）来保留相似性。

尽管有各种尝试，但二进制化阶段不可避免会破坏原始数据的相邻结构，导致巨大的**邻居错误**（原始数据空间中的邻居点索引得到非邻居点、非邻居点索引得到邻居点）。因此，这些方法会出现巨大相似性损失和不尽人意的索引表现。

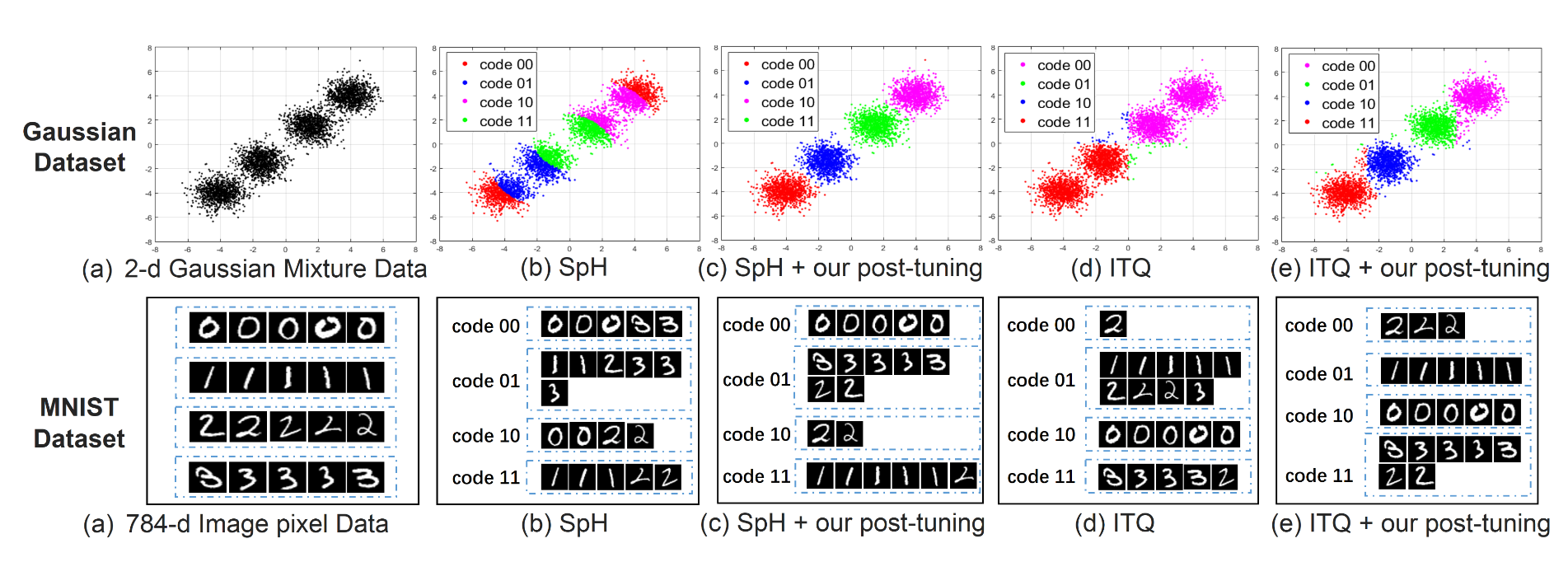


Fig1

## PTH方法

### 无监督二进制哈希

给定一个集合，其中包括个数据点。无监督二进制哈希会首先通过线性或者非线性方程将数据点投射到一个低纬度的连续流形（可理解为空间）。之后将投射结果二进制化，映射到一个二进制值空间（汉明空间）。

其中为元素级函数，以0为阈值将投射数据的每个维度二进制化。

但是，二进制化会带来邻居损失，可由如下公式定义此损失：

其中，为Frobenius范数（定义为矩阵各项元素平方的总和开根），分别为原始数据点和二进制编码的邻居关系矩阵。具体而言，中第行第列元素就表示和的邻居关系。当然，巨大的会极度下降二进制编码在近似最近邻居搜索中的性能。

### PTH大致框架

本文提出一种名为**后调哈希**（Post Tune Hash, PTH）的新颖哈希模型，加入了一个后调阶段来改良二进化后的编码。这样，需要优化的邻居错误就变为：

需要说明的是，很多方法都可以被用于生成，因此后调过程可被用于广泛的哈希方法，帮助这些方法改良二进制编码并提升性能。

为了最小化邻居错误，本文需要将其转变为一个计算可解的问题。之前提到的中的每个元素代表原始数据点的邻居关系，这里通过欧几里得距离来判断邻居关系：

其中，为判别邻居关系的欧几里得距离阈值。而中的每个元素表示二进制编码的邻居关系，定义为一个内积：

为一个范围为的离散变量。代表和完全一样；代表和完全不一样。现在，我们可以把邻居损失转化为如下公式：

其中，为的二进制编码。

例子：

PTH将通过一个二进制矩阵作为后调矩阵来改良二进制编码，从而最小化。后调会重构破坏的邻居结构，并因此极大提升索引能力。通过使用，后调阶段的结果很自然也会是二进制的，优化公式如下：

其中，为阿达玛乘积（元素级乘积），代表的第行第列的元素。准确地说，表示矩阵的第行第列的元素是否应该翻转（与互换）来最小化邻居错误。最终的二进制编码就被定义为。

阿达玛乘积：

### PTH优化算法

尽管二进制编码的长度为1（），为上述优化公式找到一个最优的也是NP难问题，朴素的解决时间复杂度为。但本文可以通过一个实际算法来获得一个局部最优解。

本文注意到：目标函数的所有二次项都是常数，最小化就是最小化线性项之和。

因此，本文就从元素级的角度进行优化。首先，对每个的元素计算线性项。令，目标函数可以被转化为：

将中第行看成变量，其他看为常数，线性项为：

令为第行向量，为矩阵第行第列的元素。令为去除第行的矩阵。令为矩阵的第行第列元素。因此上述式子可以转化为：

令为对称矩阵的第行第列元素。将作为变量，其他作为常数，线性项为：

（利用对称性，同时时）

对于元素，最小化就是最小化上述式子。线性参数可被当做的权重。因此，如果线性系数，；否则为。

可以看出，决定一个需要其他中的元素参与，这样会存在干扰。因此本文采用一个贪心算法来求解：将所有元素都设为，并每次迭代会更新最大线性系数的元素，直到最后的增益很小。基本的后调算法就是这样，下面是一些优化：

**更新策略**：本文将更新策略改为“如果系数大于一个固定阈值就更新系数”。这样就可以按序处理元素，极大减少算法复杂度。实验中，被设为所有系数的均值。另外，本文为更新一行的提供相同的矩阵，因为被所有行所决定，不会对一行数据的改变敏感。

**削减策略**：本文只会微调那些映射结果接近于0，或者小于阈值的元素。因为这些元素有更高概率会被二进制为错误的值。这个策略会减少后调的复杂型，并有效防止过拟合。实验中，被设为投射结果的绝对值均值。

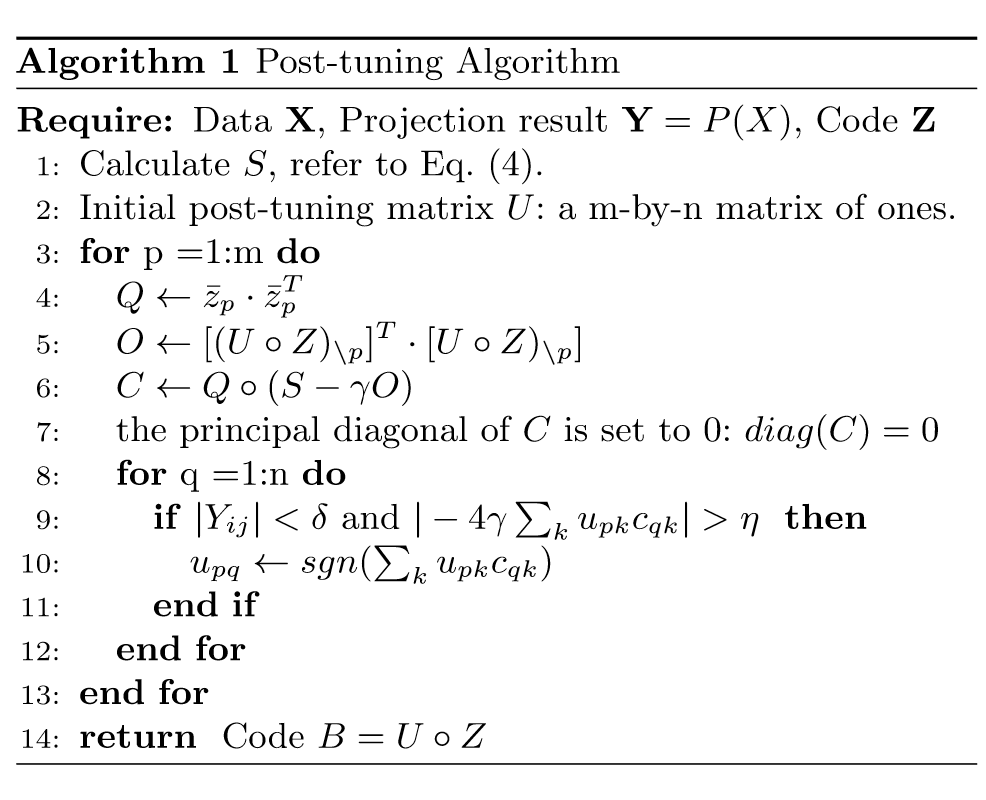


Fig2

整个算法可以迭代次，通过查看目标函数的值的趋势，本文将设为。

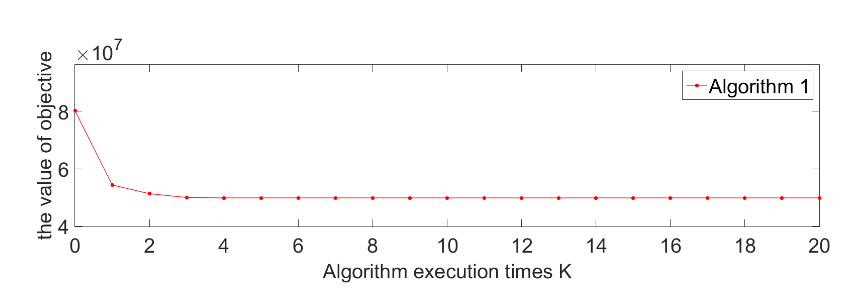


Fig3

### 样本外后调

目前，后调算法可以对已知数据进行二进制编码改良。下一个关键问题自然就是泛化后调算法，使其能处理新数据（）。当然，样本外数据必须是和保持一致的。因此，本文将作为骨架点（决定了后调的重新编码规则）。那么整个后调包括两步：后调骨架点和后调与一致的样本外点。

本文后调任何骨架点的目的是减少编码与其他中点的编码之间的不正确关系。考虑到一致性，对样本外的点进行微调也应该减小和之间的不正确关系。令为的原始二进制编码，样本外后调的目标函数可被定义为：

其中，为和真实的邻居关系（欧几里得距离阈值）。为的后调编码。为需要学习的后调向量。为的后调编码。那么上述后调算法少加改动就可以解决这个问题。通过以骨架点为基准，样本外点的邻居结构就可以被重建。由于后调是一个二进制后的独立学习过程，因此可以与哈希训练集不同。实验中，本文在哈希训练集中选择了一小部分点作为。

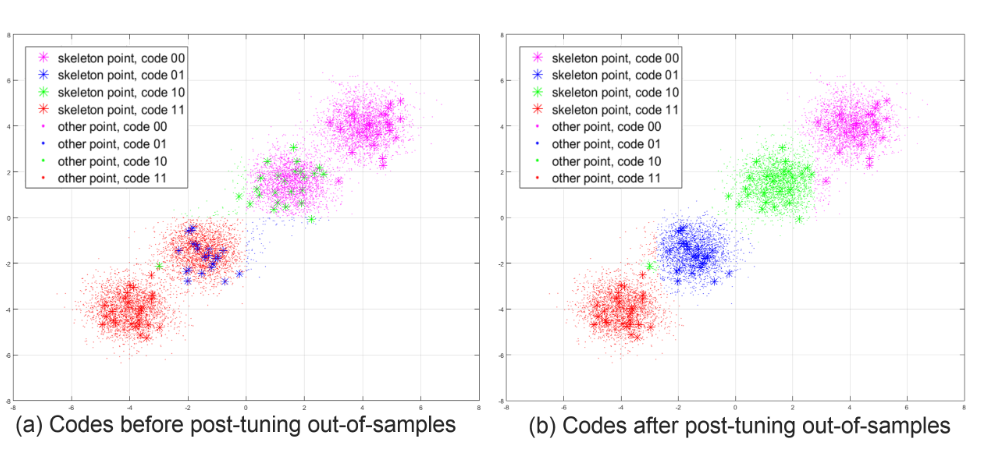


Fig4

### 复杂度

**空间复杂度**：PTH的空间复杂度为，其中为数据集的大小、为二进制编码的维度、是骨架点的数量、是原始特征的空间。第一项是用来存储二进制编码的（所有哈希方法都会使用）；第二项为PTH的额外花销。由于骨架点较少，因此PTH的额外空间花销不大。

**时间复杂度**：给定一个请求，需要花费来生成二进制编码。为原始特征的维度。是为了计算二进制编码的点积（矩阵），可以通过比特级操作来加速。另外，PTH需要的时间为后调骨架点进行离线训练。

## 实验结果

5个图片benchmark实验结果现实本文的PTH模型在平均精确度均值上相比之前的SOTA方法提升了13-58%。

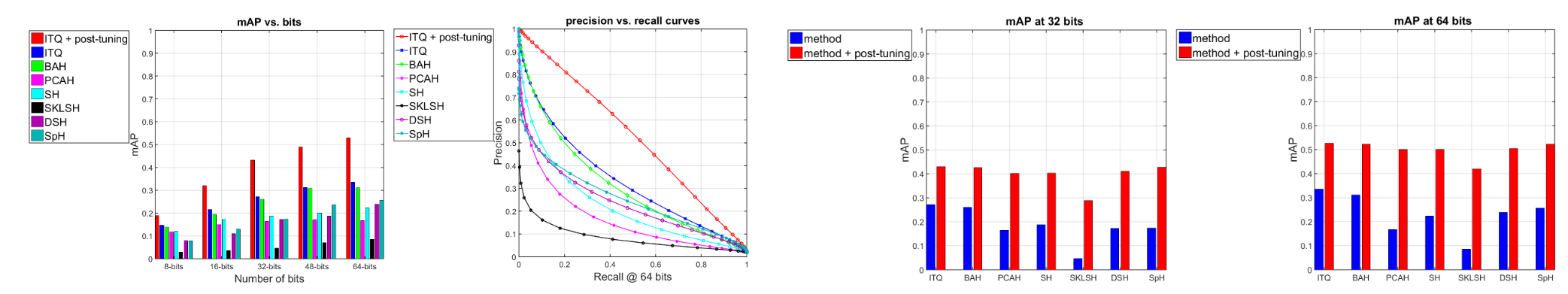


Fig5

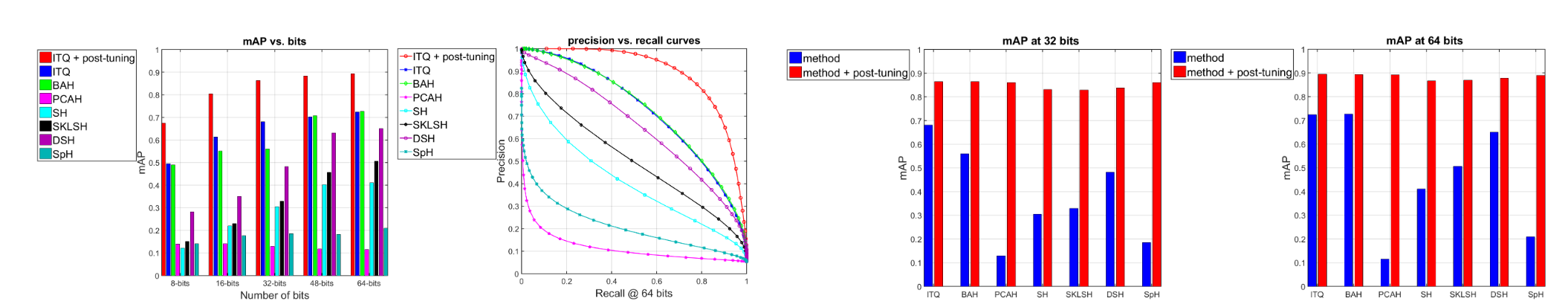


Fig6

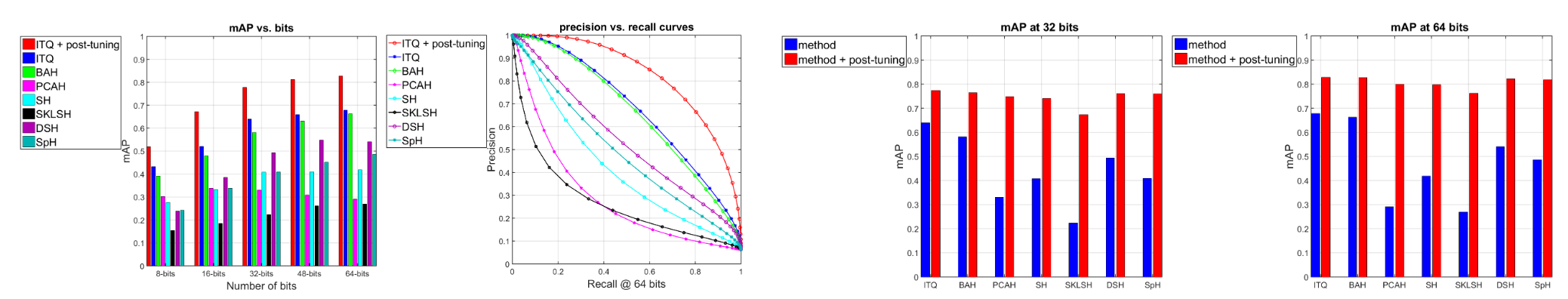


Fig7

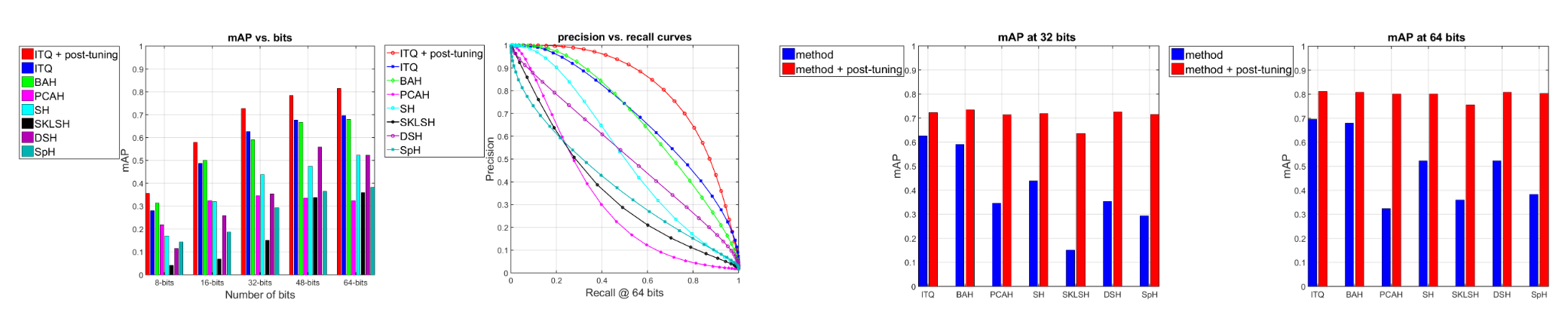


Fig8

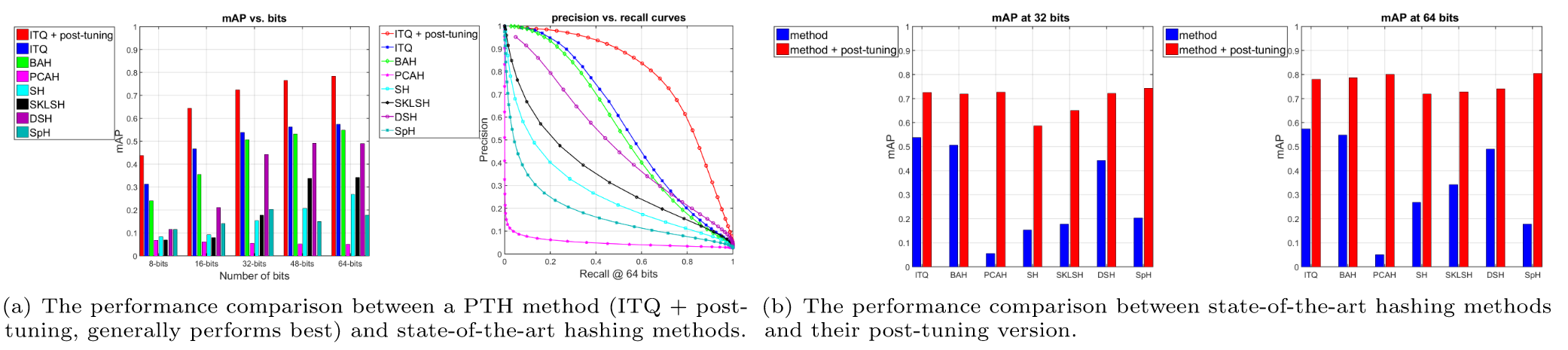


Fig9

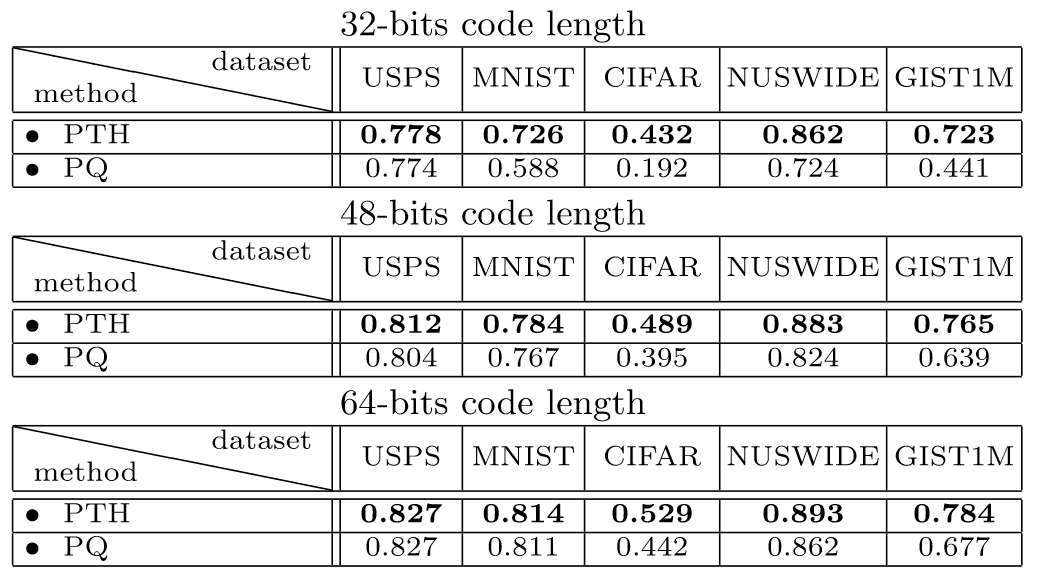


Fig10

骨架点的数量太多对效果的提升没有太大的帮助。另外，随着数据集大小的增加，骨架点的数量保持在100左右就可以有很好的效果。

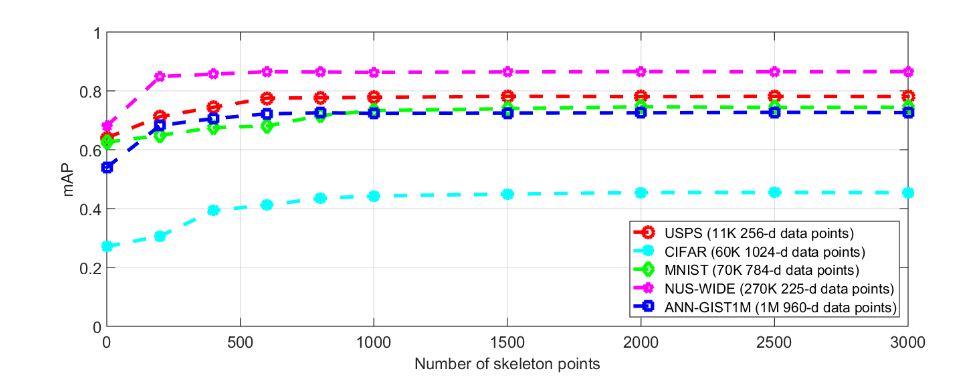


Fig11

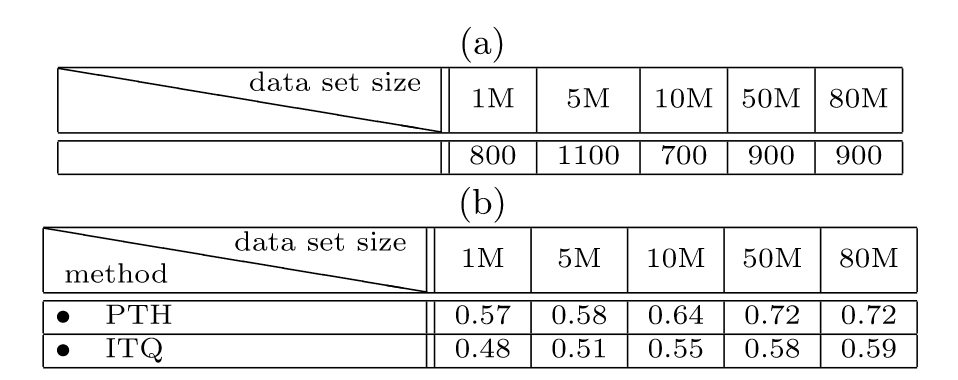


Fig12

PTH所用的训练计算时间较长，而单步请求时间并没有太多增加。

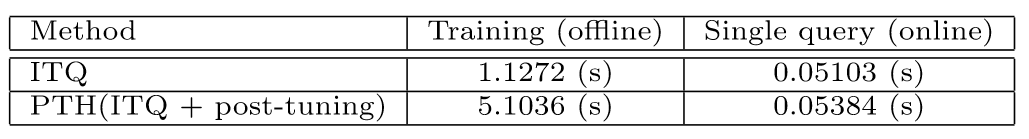


Fig13

## 个人感想

论文特色：

* 后处理的思想，打破预处理的思维枷锁。
* 将一个NP难问题转化为一个可求局部最优解的方法。
* 算法的各方面设定都具备合理性，贴近实际使用。
* 灵活使用离散值-1、1的性质。

创新方向：

* 持续哈希：数据按序送过来，如何逐渐构建数据间的邻居关系？
* 视频哈希：对视频每一帧进行哈希，看看能不能把帧的顺序排列成功。
* 哈希二进制编码与深度学习编码器结合，提高可解释性。