无监督对比跨模态哈希算法

Peng Hu, Hongyuan Zhu, Jie Lin, Dezhong Peng, Yin-Ping Zhao, and Xi Peng

摘要 本文研究了如何通过克服两个挑战，使无监督跨模态散列（CMH）从对比学习（CL）中受益。确切地说，i) 为了解决优化散列导致的性能下降问题，我们提出了一种新的动量优化器，该优化器可以执行可在对比学习中学习的散列操作，从而使现成的深度跨模态散列成为可能。说，我们的方法不像大多数现有方法那样涉及二元连续松弛，因此可以获得更好的检索性能；ii) 为了减轻假阴性配对（FNPs）带来的影响，我们提出了一种跨模态排序学习损失（CRL），它利用对所有而非仅对硬阴性配对的判别，其中 FNP 指的是被错误地视为阴性配对类内配对。得益于这种全局策略，CRL 使我们的方法具有更好的性能，因为 CRL 不会过度使用 FNP 而忽略真实的对。据我们所知，所提出的方法可能是首批成功的对比散列方法之一。为了证明所提方法的有效性，我们在五个广泛使用的数据集上进行了实验，并与 13 种最先进的方法进行了比较。代码见 https://github.com/ penghu-cs/UCCH。

索引词条-普通汉明空间、对比散列网络、跨模态检索、无监督跨模态散列

1 导言

Cross-modal 检索系统的目的是从一种模式（如图像）中检索语义相关样本。从另一种模式（如文本）获取查询。跨模态检索的主要挑战是弥合不同模态之间的差距。为了缩小这种所谓的异质性差距，人们提出了多种方法，并取得了可喜的性能[1][2]、[3]、[4]。然而，这些方法的存储和计算成本都很高，因为这些方法学习到的表征是连续值，而连续值对图像的吸引力较小。

- 胡鹏、彭曦，四川大学计算机学院，成都 610065。电子邮箱{penghu.ml, pengx.gm}@gmail.
- Hongyuan Zhu 和 Jie Lin现供职于新加坡科技研究局（A*STAR）信息通信研究所，邮编：138632。电子邮件：hongyuanzhu.cn@gmail.com, lin-j@i2r.a-star.edu.sg。
- 彭德忠，计算机学院，中国成都 610065；成都瑞贝英特信息科技有限公司，中国成都 610094；四川智谦科技有限公司，中国成都 610094。四川智乾科技有限公司，成都 610094。公司，中国成都 610094。电子邮件：pengdz@scu.edu.cn.
- 赵银平，西北工业大学软件，中国西安 710072。电子邮件：zhaoyinping@nwpu.edu.cn.

2021 年 11 月 7 日收到手稿；2022 年 3 月 21 日修订；2022 年 5 月 10 日接受。
出版日期：2022 年 5 月 26 日；当前版本日期：2023 年 2 月 3 日。
本研究部分得到国家自然科学基金62102274、U19A2078、U21B2040、61971296 和62176171的资助，四川省科技计划项目2021YFS0389、2021YFG0317和2021YFG0301的资助，中国博士后科学基金2021M692270的资助，AME项目A18A2b0046的资助、部分资助：职业发展基金（C210812033）；RobotHTPO种子基金（C211518008）；浙江实验室开放研究项目（2021KH0AB02）；教育局OSTIN空间技术与发展计划（S22-19016-STDP）；A*STAR项目（A18A2b0046和A1892b0026）。
(通讯作者：Xi Peng。）由D. Meng 推荐接受。
数字对象标识符编号10.1109/TPAMI.2022.3177356

大规模跨模态检索。因此，如何有效弥合大规模跨模态检索的异质性差距仍是一个有待解决的问题。

为了有效缩小异质性差距并提高检索性能，跨模态散列技术受到了业界的广泛关注 [5][6]、[7]、[8]、[9]、10]。跨模态散列的基本思想是将高维多模态数据投影到紧凑的二进制位中 [11]、[12]、[13]。由于采用了比特相似性测量（即 XOR），散列过程在存储和通信方面比连续值方法 [7][8]、[14] 更为高效。现有的大多数跨模态散列方法可大致分为有监督和无监督两类。更具体地说，有监督方法 [8]、9]、[15]、[16] 通常利用标注的语义信息从多模态数据中学习哈希代码，并取得了良好的性能。然而，这些方法需要大量的标注数据，而且数据标注非常费力[13], [17]。与有监督的方法不同，无监督的跨模态哈希方法 [13], [18], [19] 可以避免密集的数据标注，在实践中更具吸引力。本文主要关注无监督学习范式。

所有现有的无监督跨模态哈希方法都基于浅层或深层模型。简而言之，浅层方法学习单层线性或非线性变换，将不同模态投射到一个共同的汉明空间[13], [18], [20]。然而，这些浅层模型不能很好地捕捉高水平的非线性信息[21]，因此它们的性能不尽如人意。为了解决这个问题，深度神经网络（DNN）[13]、[19]、[22]、[23] 被用来学习散列函数，因为它们在非线性建模方面具有优势。受近期对比学习成功的激励，人们非常期待研究如何为跨模态哈希进行无监督对比学习。虽然这种想法看似简单明了，但由于存在以下两个挑战，因此并非易事。首先，它是一个挑战、

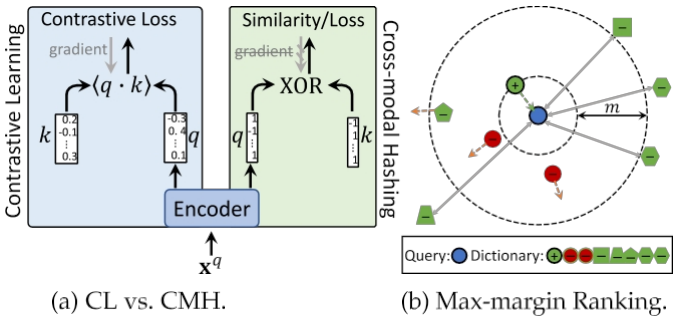


图 1.现有方法的局限性。在本图中，我们以一个查询为例。(a) 显示了对比学习（CL）和跨模态散列（CMH）之间的差异/差距。更具体地说，CL 以可微分的方式对连续值进行优化。相反，CMH 采用二进制编码，无法计算梯度。图中，q 和 k 为查询样本和关键样本。(b) 显示传统的最大边际排序会忽略边际外的交叉模态样本，导致更多关注假阴性样本（红圈）。图中，绿线代表可释放/正的交叉模态相关性；橙线代表无关/负的交叉模态相关性；灰线表示忽略的相关性；蓝色项表示查询样本；绿色项表示真实的负样本，红色项表示假阴性点。

对比学习总是被视为预训练步骤与下游的跨模态哈希检索存在差距。事实上，对比学习通常采用连续值优化策略，这与跨模态散列的二进制输出不一致（见图 1a），因此很可能导致性能下降。其次，为了衔接散列学习和跨模态检索，现有的跨模态散列方法大多采用最大边际排序损失法，其性能在很大程度上取决于已建立的正负对（见图 1b）。然而，在无监督环境下，由于无法获得标签，很难很好地建立正负样本。因此，无监督跨模态哈希算法通常会将共同出现的样本视为正样本，而将其他样本视为负样本。显然，这种方法会导致新的噪声，即一些类内样本被错误地视为负样本。据我们所知，这种假阴性对（FNP）问题迄今为止接触较少，也没有用于跨模态哈希的现有解决方案。

为了解决上述两个问题，我们提出了一种深度无监督跨模态哈希方法，称为无监督对比跨模态哈希（UCCH）。具体来说，UCCH 采用了一种新颖的基于动量的二值化优化器来赋予哈希操作学习能力，从而使现成的深度跨模态哈希成为可能。其次，为了克服 FNP 挑战，我们提出了一种跨模态排序学习损失（CRL），它利用了所有而不是最难的负对（见图 4）。提出这种补救措施是为了避免最大边际损失的特性所导致的性能下降，即传统的最大边际三重损失 [19], [24], [25] 容易过度拟合 FNP 而忽略真负对（TNP），因为 FNP 通常比 TNP 对 DNN 优化更有吸引力，也更难（见第 4.3.7 节）。

与已被广泛研究的对比学习模型不同，我们的 UCCH 是一种指定任务的对比学习方法。更具体地说，几乎所有现有的对比学习方法 [26]、[27]、[28] 旨在以自我监督的方式学习模型，然后对模型进行微调以适应下游任务。受限于这种两阶段策略，对比学习模型与下游任务之间存在性能差距。为了缩小性能差距，我们的 UCCH 专门设计用于以单级方式实现跨模态散列。此外，与最大边际损失不同的是，我们的 CRL 可以从所有负对中获得比硬对更多的区分度，因为前者包含更多的 TNPs（见第 4.3.5 节和第 4.3.7 节），从而获得更好的性能。据我们所知，这可能是关于跨模态散列中的 FNPs 的首批研究之一。

这项工作的主要贡献和新颖之处可概括如下：

- 据我们所知，所提出的 UCCH 可能是第一种赋予对比学习以无监督跨模态哈希算法的方法。
- 我们提出了一种新颖的动量优化器，使二进制内存库成为可学习的，从而缩小了对比学习和哈希算法之间的差距。
- 为了克服 FNP 挑战，我们提出了一种跨模态排序学习损失（Cross-modal Ranking Learning loss，CRL），利用对所有而非硬性负对的判别来克服 FNP 挑战。得益于 CRL，我们的方法对 FNP 具有更好的性能和鲁棒性。
- 在五个广泛使用的基准多模态数据集上进行的大量实验验证了我们的方法与 13 种最先进方法的功效。

2 相关工作

在过去的几十年里，人们已经提出了多种方法来学习共同的哈明空间，以弥合跨模态数据中存在的同质性差距。在本节中，我们将从有监督的跨模态哈希方法、无监督的跨模态哈希方法和对比学习等方面简要回顾一些相关工作。

2.1 有监督的跨模态哈希方法

通过利用根植于标签判别信息，几乎所有有监督的跨模态方法都能学习不同模态的哈希函数，从而将多模态数据投射到一个共同的汉明空间 [15], [29], [30], [31]。一种典型的方法是利用最大边际排序损失（max-margin rank- ing loss）将排序信息用于跨模态哈希学习 [15]、[32]、[33]。在[32]中，Ding 等人提出了一种新颖的基于排序的哈希框架，通过明确利用最大边际排序损失的排序信息，将不同模态映射到一个共同的汉明空间。为了利用语义排序信息进行跨模态哈希学习，Liu 等人提出了一种基于排序的深度跨模态哈希方法，利用最大边际损失从不同模态中学习统一的哈明表示 [33]。此外，Jiang 等人提出了一种采用跨模态散列（DLFH）的离散潜因模型，通过使用语义信息直接学习二进制散列码[23]。Xu 等人[34] 开发了一种离散跨模态散列（DCH）方法

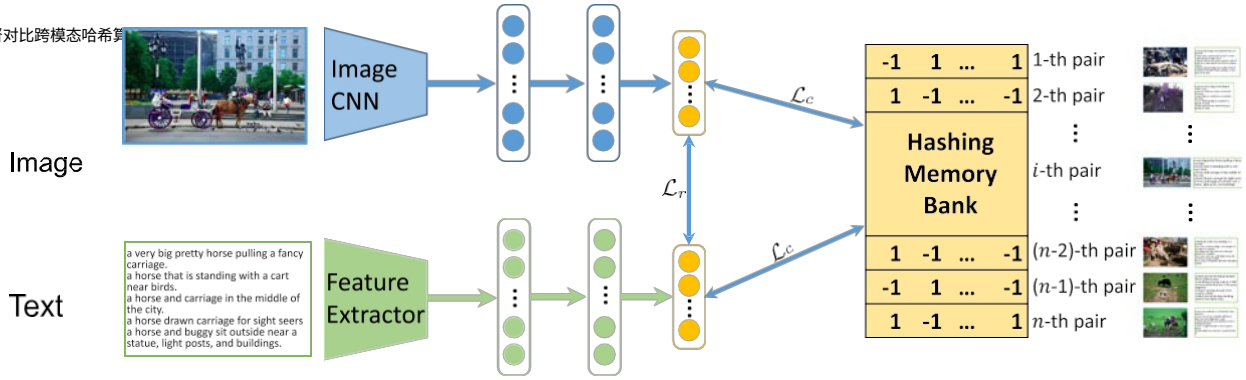


图 2.拟议方法的流水线，我们以双模情况。在这个例子中，两个特定模态网络为不同模态学习统一的二进制表征。网络的输出直接与哈希代码交互，通过使用实例级对比（即对比哈希学习（S_c））来学习潜在判别，而无需连续松弛。跨模态排序损失 S_r 用于将跨模态哈希学习与跨模态检索连接起来。

它通过保留多模态数据的离散约束，直接学习统一的判别二进制代码。得益于标注数据的语义信息，这些有监督的跨模态哈希方法在跨模态检索方面取得了可喜的成绩。然而，要达到理想的性能，这些有监督的方法需要大量的标注数据，成本和时间都很高。

2.2 无监督跨模态散列方法 无监督跨模态散列方法通过最小化跨模态对（如图像-文本对）的相关性来学习统一的二进制编码，这种方法比数据注释更便宜。因此，无监督跨模态哈希方法引起了学术界和工业界研究人员的广泛关注 [13]、[18]、[20]。在 [20] 中，Kumar 等人提出了一种跨视图散列（CVH）方法，利用多模态数据的模态内和模态间相似性来学习通用散列码。文献[35]提出了一种集体矩阵因式分解散列（CMFH）方法，通过使用具有潜在因子模型的集体矩阵因式分解来学习共同哈明空间。Liu 等人[18]提出的融合相似性散列（FSH）方法明确地将不同模态之间基于图的融合相似性嵌入到通用散列表示中。由于上述方法都是浅层方法，因此无法捕捉多模态数据中高度非线性的语义。为了解决这个问题，最近有人提出一些基于 DNN 的方法。文献[19]提出了一种无监督生成对抗跨模态哈希（UGACH）方法，利用 GAN 的能力，以最大边际排序损失来利用跨模态数据的底层流形结构。文献[13]提出了一种无监督方法，称为 "无监督耦合循环生成式对抗哈希网络（UCH）"，通过使用外循环和内循环网络来学习统一的二进制表示。

2.3 无监督对比哈希算法

最近，对比学习 [26]、[27]、[28] 引起了业界的广泛关注。例如，Wu 等人[27] 发现，表观相似性可以在没有明确指导的情况下从数据本身学习到。因此，与其从标签级判别中学习，不如从数据本身学习、对比学习被提出来在实例层面学习辨别能力 [26], [27]。受对比学习巨大成功的启发，一些对比散列方法 [36]、[37]、[38] 被提出来学习单模态数据中的二元表示，并取得了可喜的性能。简而言之，Li 等人[36] 通过使用端到端可微网络、对比损失、散列损失和平衡损失，提出了一种基于双伪协议的自超级散列方法。Qiu 等人[38] 设计了一种通用概率哈希方法，通过最小化对比损失来学习二进制哈希代码，同时减少代码与原始输入数据之间的互信息。此外，Jang 等人[37] 提出了一种基于无监督深度量化的图像检索方法，称为自监督产品量化（SPQ）网络。简而言之，SPQ 通过将对比学习与乘积量化（PQ）编码结合起来，共同学习特征提取器和编码。然而，这些方法都是为处理单模态数据而开发的，据我们所知，在探索如何将对比学习纳入跨模态散列方面投入的精力较少。要使跨模态哈希算法从对比学习中获益，至少面临两个挑战，即离散优化和 FNP 问题（如导言中所述）。

3 建议的方法

在现实世界的许多应用中，一个实例可以用不同的模式来描述，如图像、文本、音频等。在不失一般性的前提下，我们在本文中重点讨论双模态（即图像和文本）散列问题。如图 2 所示，我们的 UCCH 由特征提取模块和哈希学习模块组成。具体来说，特征提取模块旨在使用给定的提取器从原始多媒体输入中提取特征，以重新生成相应的图像/文本样本。该模块的具体实现将在第 4 节中说明。我们的散列学习模块试图将不同的模态投射到一个潜在的共同汉明空间中，在这个空间中，相关的样本被压缩，不相关的样本被分散。在本节中，我们将介绍拟议方法的细节，包括问题模拟和散列学习算法。

3.1 问题的提出

为了便于表述，我们首先给出跨模态哈希问题的一些定义。粗体大写字母（如 X）和粗体小写字母（如 x）表示矩阵，而粗体小写字母（如 x）表示矩阵。
和向量。让 Đ = $\{x_i, y_i\}_{i=1}^{n}$ 表示
包含 n 对图像-文本/实例的跨模态数据集、
其中，$x_i \in R^{d_x \times 1}$ 是图像模态的*第 i 个*样本，$y_i \in R^{d_y \times 1}$ 是与 $x_{(i)}$ 相关的文本模态，d_x 和 d_y 是图像和文本特征的维度。

跨模态散列的目标是将不同的模态投射到一个共同的汉明空间中。在该空间中，图像和文本的统一编码表示为 $S^{(x)} = \{b_i^x\}_{i=1}^{n}$ 为图像模式，$S^{(y)} = \{b_i^{(y)}\}_{i=1}^{n}$ 为文本模态，其中 $b_i \in \{-1, +1\}^L$，$* \in \{x, y\}$ 和 L 是哈希码的长度。汉明距离用于
评估图像和文本样本之间的相似性。更具体地说，如果*第* i *个*图像和*第* j *个*文本相似，则 b^x 和 b^y 之间的汉明距离应该很小。否则，不同样本之间的汉明距离应该很大。为了方便计算汉明距离，我们可以使用内积 $\langle b^{(x)}, b^{(y)} \rangle$ 来计算汉明距离 $d(b^{(x)}, b^{(y)})$，即 $d(b^{(x)}, b^{(y)}) = \frac{1}{2}(L - \langle b^{(x)}, b^{(y)} \rangle)$。因此，*第 i 幅*图像和*第* j *个*文本之间的相似性可以用内积来量化
$\langle b_i^{(x)}, b_j^{(y)} \rangle$ 在 Hamming 空间中。

为了将不同模态转换为统一的二进制编码，我们需要为跨模态输入学习两个特定模态的哈希函数。为此，我们设计了几个特定模态网络。具体来说，对于图像和文本，这两个哈希函数分别表述为 $f^{(x)}(x, Q^{(x)})$ 和 $f^{(y)}(y, Q^{(y)})$，其中 $Q^{(x)}$ 和 $Q^{(y)}$ 是要学习的相应的特定模态网络参数。在我们的 UCCH 中，Q(x)和 Q(y)的输出分别为哈希函数的定义是 $h_i^x = f^{(x)}(x_i)$ 和 $h_i^y = f^{(y)}(y_i)$
分别代表第 i *个*图像点和*第 i* 个文本点。有了
的二进制表示。

样本的计算方法是对 h_i^* 应用符号函数：

$$b_i^* = \mathrm{sgn}(h_i^*), * \in \{x, y\}, \quad (1)$$

其中，$\mathrm{sgn}(x)$ 是符号函数，如果 x50，其值为 1，否则为-1。为了学习哈希函数，我们提出了一种新颖的无监督目标函数来强制网络消除跨模态差异。与有监督的方法不同，我们的 UCCN 采用对比学习法来挖掘图像-文本对之间的明显相似性，而不是标签。我们的 UCCH 的总体目标函数表述如下：

$$\arg\min_{Q^x, Q^y} (bS_c + (1-b)S_{(r)}), \quad (2)$$

其中，b （0< b< 1）是一个权衡超参数，用于平衡对比散列损失 S_c和跨模式排序损失 S_r，下文将详细介绍这两个参数。

3.2 对比式跨模态哈希学习 对比式学习 [26] 的目的是利用相似和不相似的关系来学习判别表征。
的查询对。这也可以看作是
字典查找问题 [28], [39]。与退出
对比学习方法，我们提出了一种针对具体任务的

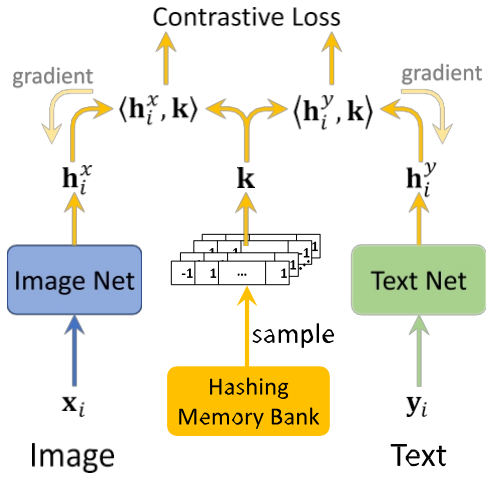


图 3.对比散列学习采用对比损失法，通过将图像-文本查询（x_i, y_i）与从散列记忆库中采样的字典进行匹配来训练特定模态网络。通过对比
利用正键和负键，相关的跨模态对可以直接逼近相应的统一二进制编码，并在不连续松弛的情况下与不相关的对进行分离。记忆库由相应对的动量更新驱动。

通过利用适用于所有模态的统一二进制字典，实现对比式跨模态散列学习方法（CCH）。据我们所知，这样的问题迄今为止触及较少。在没有连续值松弛的情况下，对于给定的查询 h_i^* ($* \in \{x, y\}$)，其目的是从散列点中直接检索相关/正键
字典中的 $\{k_1, k_2, \cdots k_n\}$。此外，字典的第 i 个键 k_i 与*第 i* 对图像和文本相对应。在无监督的跨模态情况下，有一个单一的正键（表示为 k^+），与字典中的

查询 h_i^* ($* \in \{x, y\}$)。对比损失 [26], [28] 评估
计算查询 h_i^* ($* \in \{x, y\}$) 与查询 $h_n^{(*)}$ 之间的相似度。
检索结果 $\{k_i\}_{i=1}$，当 $h_{i)}$ 时，其值较低。 $* \in \{x, y\}$ 与其正密钥 k^+ 相似，与所有的 $_i$
其他键（视为查询的负键）。

实际上，在大型字典上检索是不可行的
的大规模数据集。为了解决这个大规模学习问题，我们从整个词典中随机抽取一部分样本
$\{k_i\}_{i=1}^{K}$（被视为散列记忆库）作为新的小型字典进行检索（图 3）。具体来说，与正密钥相反，我们从散列记忆库中随机抽取 K 个点，构建负密钥集
$\{k^-\}_{j=1}^{K}$，其中 $k^- = \mathrm{sgn}(v_{(r)})$，$v_r$ 是 $k^{(-)}$对应的连续值源密钥（详细定义见下文），r 是对应的随机索引。根据 [27]，我们通过 ℓ_2-normalization 执行 $\|h^{(*)}\|=1$ ($* \in \{x, y\}$ 和 $\|k\|=1$ ($> \in \{+, -\}$)。如前所述，不同哈希点之间的相似性是通过点积来衡量的。利用点积相似性，采用一种有效的对比损失函数（称为 InfoNCE [40]）来最大化实例级消除和最小化跨模态差异：

$$S_c = -\sum_{i=1}^{n} \log P(i|h^{(x)})_i^- \sum_{i=1}^{n} \log P(i|h^{(y)}), \quad (3)$$

其中 $P(i|h^{(x)})$ 和 $P(i|h^{(y)})$ 是 h_i^x 和 h_i^y 的概率
被视为*第* i 个点。公式如下
定义为

$$P(i|\hat{h}_i^*) = \frac{\exp\langle h_i^*, k_i^+\rangle/t}{\exp\langle h_i^*, k_i^+\rangle/t + \sum_{j=1}^{K}\exp\langle h_i^*, k_j^-\rangle/t}, \tag{4}$$

其中 $*\in\{x, y\}$，t 是温度超参数之三[27]。直观地说，这个损失函数也可以看作是 $(K+1)$-way 非参数软最大分类器的负对数概率。与传统的基于软最大值的分类器不同，上述表述的目的是对数据进行分类。

将第 i 个图像-文本对（即 h_i^x 和 h_i^y）视为相应的正密钥（即第 i 个散列点 k_i^+）从记忆库中取出。

字典的另一个挑战是二值优化。因此，式（3）中的散列对比损失要求样本逼近正键的散列代码，并与负键的离散表示区分开来。与现有的跨模态哈希方法不同，我们的 UCCN 直接学习离散表示，无需连续松弛。然而，直接优化离散存储库是一个 NP-困难问题 [20]、[41]、[42]。为了使散列记忆库具有可学习性，我们定义了它的符号幅度为 $\{v_i\}_{i=1}^n$。散列密钥可通过 $k_i = \mathrm{sgn}(v_i)$ 得出。相应的。然后，利用动量机制更新存储库 $\{v_i\}_{i=1}^n$ 具体如下

$$v_{i'} = dv_{i'} + (1-d)\frac{h_i^x + h_i^y}{2}, \tag{5}$$

其中，$d\in[0, 1]$是动量系数，$v_{i'}$是记忆库中第 i'个位置的正值。密钥 k_i^+，该密钥来自采样的多模态对批次 $\{x_{(i)}, y_{(i)}\}$ 中的 $\{x_i, y_{(i)}\}_{j=1}^{N_b}$，其中 N_b 是批次大小。请注意，i' 是迷你批次中第 i 对数据在存储库中的对应位置。

3.3 跨模态排名学习

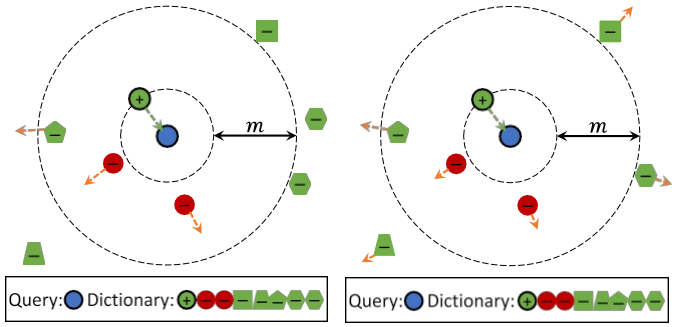
除了从统一的哈希字典中检索，我们还需要将模型训练与下游任务（即跨模态检索）的性能联系起来。为了实现这一目标，我们强制要求相关配对的相似度大于不相关的跨模态样本的相似度。具体来说，一个首先使用图像查询 h_i^x 来检索共同出现的样本。从文本字典 $\{h_j^y\}_{j=1}^n$ 中提取 $h_{j=i}^y$。直观地说，模拟查询点 h_i^x 与相关点 h_i^y 之间的相似性应该是大于 h_i^x 与无关数据之间的相似性。样本 $\{h_{j\neq i}^y\}_{j=1}^n$。这同样适用于文本查询 h_i^y 和相关图像点 h_i^x。为此，一个双向广泛采用最大边际排序损失来强制执行多模态学习中的这一约束 [43], [44]。最大边际排名损失的计算公式如下：

$$S_r' = S_r^{xy} + S_r^{yx}, \tag{6}$$

其中

$$S_r^{*} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j\neq i}^{n}\max\left(0, m + M_{ij}^{*} - M_{ii}^{(*)}\right), \tag{7}$$

$*\in\{xy, yx\}$，$M_{ij}^{xy} = \langle h_i^{(x)}, h_j^{(y)}\rangle$，$M_{ij}^{yx} = \langle h_i^{(y)}, h_j^{(x)}\rangle$，并且 m 是一个正方形。利润率值。



(a) Max-margin ranking loss. (b) Our ranking loss.

图 4最大边际排序损耗与我们的损耗之间的主要区别在于，前者不利用字典之外的样本，而后者则字典之外的样本。具体来说，对于给定的查询，我们的目标是从给定的词典中检索出最相关的样本，其中查询和词典分为两种模式，不同的形状表示不同的类别。由于 FNP 的存在，香草最大边际排序损失可能会导致错误的优化结果，如（a）所示。我们的排序损失并不是只将边缘内的数据对之间的样本推走，而是同时将数据对内部的相似性最大化，而将所有数据对之间的相似性最小化。因此，我们的排序损失可以充分利用所有负样本，从而减轻 FNP 的影响。

公式 (6) 表明，最大边际损失侧重于优化找出相似度不小于 m 的硬负对。比正对（即 $M_{ii}^{*} - M_{ij}^{(*)} < m$），从而忽略更容易。说，香草三连败主要是该方法的重点是较难的负配对，其性能在很大程度上取决于成熟的负样本。然而，在不普遍的情况下，很难保证负对的正确性，因为成对/共同出现的样本总是被用作正对，而其他样本则被视为负对。显然，这种配对结构策略会错误地将一些类内样本视为负样本，而这些 FNP 会导致香草三重损失的优化方向错误。更具体地说，最大边际损失会强调分离 FNP，而忽略真负对 (TNP)，因为由于 FNP 之间的语义相关性，前者比后者更难分离。因此，正如第 4.3.5 和 4.3.7 节所验证的那样，很难获得令人鼓舞的结果。

为了克服 FNP 挑战，我们提出了一种新的学习范式，称为跨模态排序学习（Cross-modal Ranking Learning，CRL），它使用所有负对进行优化。图 4 直观地说明了最大边际损失和最小边际损失之间的差异。

和 CRL。如图所示，可以看出最大边际排序损耗只关注边缘内的对间样本而忽略边距之外的对间样本。如图 4a 所示，在没有语义标签的情况下，它可能会只关注 FNP，而忽略 TNP。与 max-margin loss [19], [24], [33] 不同，我们的 CRL 可以所有负样本，减轻 FNP 的影响，如图 4a 所示。

如图 4b 所示。

为了利用包括被忽略的样本在内的所有负样本对，我们同时考虑边际内和边际外的负样本，构建了以下最大边际的上限。首先，让

$$S_{ij}^{(*)} = \begin{cases} M_{ij}^{*}, & M_{ii}^{*} - M_{ij}^{*} < m \\ M_{ij}^{*} - \xi, & \text{否则}, \end{cases} \tag{8}$$

其中 $\xi > 0$。因此，S_{ij}^{*} 永远不会大于 $M_{ij}^{(*)}$。

定理 1.*对于任何* k∈ ℝ+ *，都成立*

$$\max_{j=1,\ldots,n}(S^*_{ij}) \geq k\log\sum_{j=1}^{n}\exp(S^*_{ij}/k) \tag{9}$$

其中 *∈ {xy, yx}.

证明为了证明上述 Lemma，我们有以下条件

$$\begin{aligned}\max_{j=1,\ldots,n}(S^*_{ij}) &= \max_{j=1,\ldots,n} k\log\exp(S^*_{ij})^{1/k} \\ &= k\log\max_{j=1,\ldots,n}(\exp(S^*_{ij}))^{1/k} \\ &\leq k\log\sum_{j=1}^{n}\exp(S^*_{ij}/k)\end{aligned} \tag{10}$$

定理 1.*对于任何* k ∈ +，*都成立*

$$\sum_{j=1}^{n}\max(0, m+S^*_{ij}-S^*_{ii}) \leq 4nm + k\log\sum_{j=1}^{n}\exp(S^*_{ij}/k) - S^*_{ii} \tag{11}$$

其中 *∈ {xy, yx}.

证明根据 Lemma 1，我们很容易得出

$$\begin{aligned}&\sum_{j=1}^{n}\max(0, m+S^*_{ij}-S^*_{ii}) \\ &= |A_{(i)m}| + \sum_{S_{ij}\in A_i} S^*_{ij} - |A_i| S^*_{ii} \\ &\leq 4|A|m + \max_{j=1,\ldots,n}(S^*_{ij}) - S^*_{ii} \\ &\leq 4|A_{(i)m}| + k\log\sum_{j=1}^{n}\exp(S^*_{ij}/k) - S^*_{ii} \\ &\leq 4nm + k\log\sum_{j=1}^{n}\exp(S^*_{ii}/k) - S^*_{ii},\end{aligned} \tag{12}$$

其中 $A_i = \{S^*_{ij} | S^*_{ij} - S^*_{ii} < 4m; i \neq j, j = 1, 2, \ldots, n\}$，且 $|A_i|$ 是 A_i 的大小。

因此，我们很容易得到下面的不等式：

$$S^*_r \leq \frac{1}{n}\sum_{i=1}^{n}\left(4m + k\log\sum_{j=1}^{n}\exp(S^*_{ij}/k) - S^{(*)}_{ii}\right). \tag{13}$$

那么，我们就可以将
将公式 (6) 转化为最小化其上界如下：

$$\begin{aligned}S_m &= \frac{1}{n}\sum_{i=1}^{n}\left(k\log\sum_{j=1}^{n}\exp(S^{xy}_{ij}/k) - S^{xy}_{ii}\right) \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}\left(4m + k\log\sum_{j=1}^{n}\exp(S^{yx}_{ij}/k) - S^{yx}_{ii}\right)\end{aligned} \tag{14}$$

同时分散从上到下排名的不相关样本，而不是只关注排名靠前样本。更进一步

更重要的是，我们的方法更关注最重要的不相关因素。
样本（被认为是更困难的点）比机器人的

因此，上部的 "蘑菇 "可以与下部的 "蘑菇 "充分分开。
查询。同时，损失也会压缩正样本（即相关的图像-文本对），以消除跨模态差异。

3.4 优化

学习最佳散列函数的过程是通过联合最小化对比散列损失 S_c 和跨模态排序损失 S_r 来实现的，如公式 (2)。联合损失如下

$$S = bS_c + (1 - b)S_r. \tag{15}$$

我们的 UCCH（公式 (2)）可以逐批迭代优化。通过最小化 S_c，我们的 UCCH
它通过实例级辨别学习来捕捉表观相似性[27]，并将多模态数据编码为二进制代码，而不进行连续松弛。此外，跨模态检索指标被直接注入学习过程，以弥补跨模态差距。我们的 UCCH 整个模型可以通过以下方法进行优化
使用任何一种随机梯度下降优化算法，如 Adam [45]。算法 1 总结了 UCCH 的优化过程。

算法 1.我们的 UCCH 优化过程

输入：训练图像-文本对 Ð = $\{x_i, y\}_{i=1}^{(n)}$ 长度
的哈希码 L、批量大小 N_b、平衡参数 b、
动量系数 d，边际约束参数 m、
负样本数 K 和学习率 a。

1: 随机初始化 Q_x、Q_y。
2: while not converge do
3: 从 Ð 到 Ð 随机抽取 N_b 对图像和文本。
构建一个图像-文本迷你批 $\{x_i, y\}_{i=1}^{Nb}$.
4: 随机采样 K 个负密钥 $\{k^-\}_{j=1}^{K}$ 并选择
对应的正密钥 $k_i = \mathrm{sgn}(v_{i'})$，每对 $^+$
散列存储库中的 $\{x_i, y_i\}$。
5: 计算迷你批次中每个点的代表性
使用相应的哈希函数。
6: 计算散列对比损失和跨模态损失
根据公式 (3) 和公式 (14)，小批量生产的排序损失分别为
7: 通过最小化公式（15）中的 S，更新特定视图哈希网络的参数，
其随机性依次递减。
梯度：
$Q_* = Q_* - a(b\frac{@S_c}{@Q^*} + (1-b)\frac{@(S)_r}{@Q^*})$ (*∈ {x, y})
8: 更新采样的相应正键
迷你批 $\{x_i, y\}_i^{(Nb)}$，通过散列存储库中的
公式 (5)：
$v_i^+ = dv_i^+ + (1-d)\frac{(h)_i + (h)(i=1,\ldots,N)}{2}$
9: 同时结束
输出：优化的 UCCH 模型

如图 4 所示，m 是边际约束条件。通过最小化公式 (14)，跨模态网络被训练成可以扫描所有配对间样本。与传统的最大边际排序损失 $S^{'}$ 不同，我们的跨模态排序损失 S_{r} 可以

4 实验研究

为了验证我们的 UCCH 的有效性，我们在五个广泛使用的多模态数据集上进行了实验、

mirflickr-25k [46]、IAPR TC-12 [47]、NUS-wide [48]、MS-COCO [49] 和 Flickr30K [50]。我们的方法是在一个英伟达 GEFORCE RTX 2080 Ti GPU 上使用 PyTorch [51] 实现的。

4.1 数据集

4.1.1 mirflickr-25k [46]

这是一个广泛使用的跨模态数据集，用于跨模态哈希检索。该数据集由 25,000 对图像-文本组成，其中每一对都包含一幅图像及其对应的多个文本标签，这些标签由人工从 24 个独特的语义类别中标注。在对没有类别信息的配对进行剪枝后，我们的实验还剩下 20,015 对配对。为了进行公平比较，我们完全按照文献[13]的数据分区策略，随机选取 2,000 对图像-文本作为查询集，剩余的作为检索数据库。对于有监督基线，我们从检索数据库中随机选取 5000 对作为训练集。图像和文本样本分别用预训练的 19 层 VGGNet [52] 提取的 4,096 维向量和 1,386 维字袋（BoW）向量表示。

4.1.2 亚太区域中心 TC-12 [47]

该数据集共包含 20,000 对图像-文本，其中标注了 255 个独特语义类别的多重标签。与其他数据集不同，我们的实验使用了整个 IAPR TC-12。每对图像用预训练 CNN-F 提取的 4,096 维向量表示 [53]，每段文字用 2,912 维 BoW 向量表示。与 MIRFLICKR-25 K 一样，我们随机选择 2,000 对图像-文本作为查询集，其余的作为检索数据库。此外，我们还从检索数据库中随机选取 5000 对图像文本作为监督基线的训练集。

4.1.3 NUS-WIDE [48]

该数据集包含 269,498 张网络图片，其文字标签被归类为 81 个概念类别中的一个或多个标签。在该数据集中，我们选择了属于 10 个最常见类别的 186 557 个图像-文本对进行实验。我们按照文献[22]的数据分区策略，随机选取 2100 对图像-文本作为查询集，剩下的作为检索集。每个文本点表示为一个 1,000 维的 BoW 向量。每个图像样本的特征是由预训练的 19 层 VGGNet 提取的 4,096 维向量。此外，还从检索集中选取了 5,000 对图像-文本，用于构建监督基线的训练集。

4.1.4 MS-COCO [49]

该数据集共包含 123 287 幅图像。每幅图像都有五个注释句子，其注释被分为 80 个类别。除去没有任何标签信息的图片对，我们的实验还剩下 122 218 个图片-文本对。与其他数据集不同的是，每对图像的文本都是由经过预训练的 Doc2Vec [54] 提取的 300 个分词向量表示的。每个图像由预训练的 19 层 VGGNet 提取的 4,096 维向量表示。我们随机选择 5000 对图像-文本作为查询集，其余的作为检索集。与其他数据集一样，我们随机选择 5000 对图像-文本作为监督方法的训练集。

4.1.5 Flickr30K [50]

该数据集包含 31,000 张图片，每张图片有五个文本注释。我们使用 [55] 的默认分割方式，即训练集包括 29,000 张图像和 145,000 个文本，验证集包括 1,000 张图像和 5,000 个文本，测试集包括 1,000 张图像和 5,000 个文本。参照文献[55]，每张图片都表示为从预训练的 19 层 VGGNet 的 FC7 中提取的 4,096 个区域向量。与之前的四个数据集不同，Flickr30 K 是一个无标签数据集。因此，我们只能在 Flickr30 K 上进行图像-文本匹配，而不是基于语义的跨模态检索，后者是基于实例的跨模态检索的一种。

4.2 评估规程和基线

4.2.1 评估规程

对于每个数据集，我们按照 [22]、[56]、[57]的方法，从全部数据集中随机抽取一些样本作为查询集并将剩下的样本作为检索数据库。为了评估跨模态哈希方法的性能，我们完成了两个不同的跨模态检索任务：使用图像查询检索相关文本点（Image ！ Text）和使用文本查询检索相关图像点（Text ！ ）。地面实况相关邻域被定义为至少有一个相同语义类别的跨模态点。为了评估检索结果的准确性，实验中使用了广泛使用的汉明排序和哈希查找作为检索协议。评价指标采用了广泛使用的平均精度（MAP），即每个查询的平均精度（AP）得分的平均值，来衡量汉明排序结果的准确度得分。MAP 同时考虑了检索精度和返回结果的排序，因此被广泛用于评估跨模态检索的性能。除 MAP 外，我们还采用精度-检索曲线作为哈希查找协议，以直观地评估跨模态检索的性能。请注意，所有 MAP 分数都是根据实验中所有返回的检索结果计算的（即 MAP@ALL）。除了上述两个类别级指标的比较外，我们还采用了不同 K 值的 Recall@K（R@K，越高越好）来衡量实例级图像-文本匹配的性能[55]。简而言之，R@K 是在 K 个排名结果中至少有一个正确项目的测试查询的百分比[55]。

4.2.2 基线

在我们的实验中，使用了 13 种最先进的跨模态哈希方法作为基线，包括四种有监督的跨模态哈希方法（DLFH [23]、MTFH [16]、FOMH [58]、DCH [34]）和九种无监督方法

表 1
MIRFLICKR-25 K 和 IAPR TC-12 数据集的 MAP 分数性能比较

方法	MIRFLICKR-25 K								IAPR TC-12							
	图像 ！ 文本				文本 ！ 图像				图像 ！ 文本				文本 ！ 图像			
	16	32	64	128	16	32	64	128	16	32	64	128	16	32	64	128
CVH [20]	0.620	0.608	0.594	0.583	0.629	0.615	0.599	0.587	0.392	0.378	0.366	0.353	0.398	0.384	0.372	0.360
LSSH [57]	0.597	0.609	0.606	0.605	0.602	0.598	0.598	0.597	0.372	0.386	0.396	0.404	0.367	0.380	0.392	0.401
CMFH [58]	0.557	0.557	0.556	0.557	0.553	0.553	0.553	0.553	0.312	0.314	0.314	0.315	0.306	0.306	0.306	0.306
FSH [18]	0.581	0.612	0.635	0.662	0.576	0.607	0.635	0.660	0.377	0.392	0.417	0.445	0.383	0.399	0.425	0.451
DLFH [23]	0.638	0.658	0.677	0.684	0.675	0.700	0.718	0.725	0.342	0.358	0.374	0.395	0.358	0.380	0.403	0.434
MTFH [16]	0.507	0.512	0.558	0.554	0.514	0.524	0.518	0.581	0.277	0.324	0.303	0.311	0.294	0.337	0.269	0.297
FOMH [56]	0.575	0.640	0.691	0.659	0.585	0.648	0.719	0.688	0.312	0.316	0.317	0.350	0.311	0.315	0.322	0.373
DCH [34]	0.596	0.602	0.626	0.636	0.612	0.623	0.653	0.665	0.336	0.336	0.344	0.352	0.350	0.358	0.374	0.391
UGACH [59]	0.685	0.693	0.704	0.702	0.673	0.676	0.686	0.690	0.462	0.467	0.469	0.480	0.447	0.463	0.468	0.463
DJSRH [60]	0.652	0.697	0.700	0.716	0.662	0.691	0.683	0.695	0.409	0.412	0.470	0.480	0.418	0.436	0.467	0.478
JDSH [61]	0.724	0.734	0.741	0.745	0.710	0.720	0.733	0.720	0.449	0.472	0.478	0.484	0.447	0.477	0.473	0.486
DGCPN [62]	0.711	0.723	0.737	0.748	0.695	0.707	0.725	0.731	0.465	0.485	0.486	0.495	0.467	0.488	0.491	0.497
UCH [13]	0.654	0.669	0.679	／	0.661	0.667	0.668	／	0.447	0.471	0.485	／	0.446	0.469	0.488	／
UCCH	0.739	0.744	0.754	0.760	0.725	0.725	0.743	0.747	0.478	0.491	0.503	0.508	0.474	0.488	0.503	0.508

最高分以黑体显示。

表 2
在 NUS-WIDE 和 MS-COCO 数据集上的 MAP 分数性能比较

方法	NUS-WIDE								MS-COCO							
	图像 ！ 文本				文本 ！ 图像				图像 ！ 文本				文本 ！ 图像			
	16	32	64	128	16	32	64	128	16	32	64	128	16	32	64	128
CVH [20]	0.487	0.495	0.456	0.419	0.470	0.475	0.444	0.412	0.503	0.504	0.471	0.425	0.506	0.508	0.476	0.429
LSSH [57]	0.442	0.457	0.450	0.451	0.473	0.482	0.471	0.457	0.484	0.525	0.542	0.551	0.490	0.522	0.547	0.560
CMFH [58]	0.339	0.338	0.343	0.339	0.306	0.306	0.306	0.306	0.366	0.369	0.370	0.365	0.346	0.346	0.346	0.346
FSH [18]	0.557	0.565	0.598	0.635	0.569	0.604	0.651	0.666	0.539	0.549	0.576	0.587	0.537	0.524	0.564	0.573
DLFH [23]	0.385	0.399	0.443	0.445	0.421	0.421	0.462	0.474	0.522	0.580	0.614	0.631	0.444	0.489	0.513	0.534
MTFH [16]	0.297	0.297	0.272	0.328	0.353	0.314	0.399	0.410	0.399	0.293	0.295	0.395	0.335	0.374	0.300	0.334
FOMH [56]	0.305	0.305	0.306	0.314	0.302	0.304	0.300	0.306	0.378	0.514	0.571	0.601	0.368	0.484	0.559	0.595
DCH [34]	0.392	0.422	0.430	0.436	0.379	0.432	0.444	0.459	0.422	0.420	0.446	0.468	0.421	0.428	0.454	0.471
UGACH [59]	0.613	0.623	0.628	0.631	0.603	0.614	0.640	0.641	0.553	0.599	0.598	0.615	0.581	0.605	0.629	0.635
DJSRH [60]	0.502	0.538	0.527	0.556	0.465	0.532	0.538	0.545	0.501	0.563	0.595	0.615	0.494	0.569	0.604	0.622
JDSH [61]	0.647	0.656	0.679	0.680	0.649	0.669	0.689	0.699	0.579	0.628	0.647	0.662	0.578	0.634	0.659	0.672
DGCPN [62]	0.610	0.614	0.635	0.641	0.617	0.621	0.642	0.647	0.552	0.590	0.602	0.596	0.564	0.590	0.597	0.597
UCH [13]	／	／	／	／	／	／	／	／	0.521	0.534	0.547	／	0.499	0.519	0.545	／
UCCH	0.698	0.708	0.737	0.742	0.701	0.724	0.745	0.750	0.605	0.645	0.655	0.665	0.610	0.655	0.666	0.677

最高分以黑体显示。

(cvh [20], lssh [59], cmfh 60], fsh [18], ugach 61]、DJSRH[62]、JDSH[63]、UCH[13]和 DGCPN64]）。所有这些方法都是浅层跨模态哈希模型，只有 UGACH、DJSRH、JDSH、UCH 和 DGCPN 是最近提出的七种深度哈希方法。为了进行公平比较，所有方法都使用相同的特征来学习哈希代码，并且在训练过程中没有对深度方法的提取器（或骨干）进行微调。我们从检索数据库中随机抽取 2000 个实例作为验证集。其他方法的超参数采用作者提供的默认参数。对于 UCCH，我们利用验证集来选择超参数 b。其他超参数在所有实验中均根据经验设定为固定值，即 d= 0.4、t= 0.9、K= 4096、m= 0.2 和 a= 0.0001。对于 Flickr30 K，我们移植了 DJSRH [62]、JDSH [63] 和我们在 VSE++ [55] 框架上的 UCCH，以进行公平比较。

4.3 实验分析

4.3.1 汉明排序

两个跨模态检索任务（即图像！ 文本和文本
！Image）在五个广泛使用的基准多模态数据集上进行，以评估我们的 UCCH 和其他基线的性能。表 1 和表 2 报告了这些任务的 MAP@ALL/Recall@K 分数，即表 1 是 MIRFLICKR-25 K 和 IAPR TC-12 数据集的结果，表 2 是 NUS-WIDE 和 MS-COCO 数据集的结果，表 4 是 Flickr30 K 数据集的结果。从表中显示的实验结果可以看出，在不同的代码长度（即 16、32、64 和 128）下，我们的 UCCH 优于所有其他基线。从实验结果中，我们可以得出以下结论：

1) 基于 DNN 的跨模态散列方法（UGACH、UCH 和我们的 UCCH）优于大多数其他浅层散列方法。

这表明 DNN 的高度非线性可以提高跨模态检索的性能。

2) 虽然有监督方法可以在标注数据充足的情况下取得很好的性能，但在标注数据不足的情况下，它们无法超越大多数无监督方法，这表明无监督方法在非标注数据量方面有很大的潜力。有监督方法过于依赖昂贵的标记数据，而无监督方法则可以解决这个问题。因此，无监督跨模态哈希方法在处理大规模多模态数据时具有很大的优势。

3) 跨模态散列方法（DLFH 和我们的 UCCH）无需任何连续松弛即可直接学习离散表示，其性能优于基于松弛的连续方法（即 DLFH 的有监督方法和 UCCH 的无监督方法）。因此，不进行连续松弛的散列学习可以提高检索性能。

4) 与类别级跨模态检索相比，实例级图像-文本匹配对散列敏感度要高得多，这可能是因为实例级检索比类别级检索要复杂得多。即便如此，我们的方法仍然可以达到具有竞争力的散列性能，这表明我们的方法可以很好地捕捉到跨模态散列的实例级判别。

4.3.2 *哈希查询*

除了汉明排序外，精确度和召回率也是通过返回结果的汉明距离计算得出的。

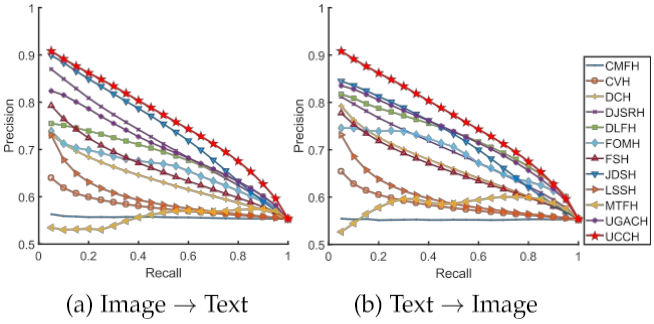


图 5.MIRFLICKR-25 K 数据集的精度-召回曲线。代码长度为 128。

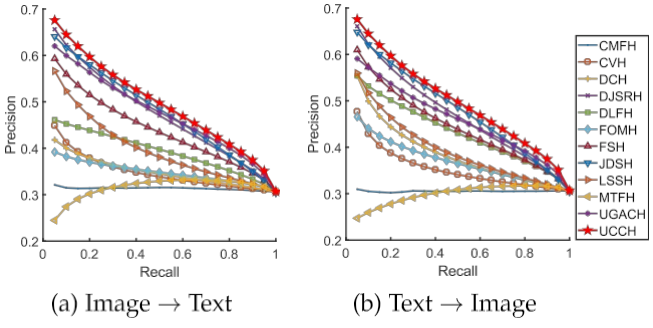


图 6.IAPR TC-12 数据集的精度-召回曲线。代码长度为 128。

[9]、[13]和[61]。如图 5、图 6、图 7 和图 8 所示，在 MIRFLICKR-25 K、IAPR TC- 12、NUS-WIDE 和 MS-COCO 数据集上绘制了码长为 128 的精度-调用曲线，以评估跨模态哈希方法的性能。可以看出，这些图中的精度-召回评估与汉明排序的 MAP 分数一致，我们的 UCCH 优于所有跨模态哈希方法。此外，在其他代码长度（如 16、32、64）的情况下，所提出的 UCCH 也优于其他方法，篇幅所限，我们省略了这些曲线。总之，与这些跨模态哈希方法相比，我们的 UCCH 在跨模态哈希检索方面能达到最佳性能。

4.3.3 *对参数的敏感性*

为了研究超参数 b 的影响，图 9 绘制了跨模态检索的 MAP 分数与不同的

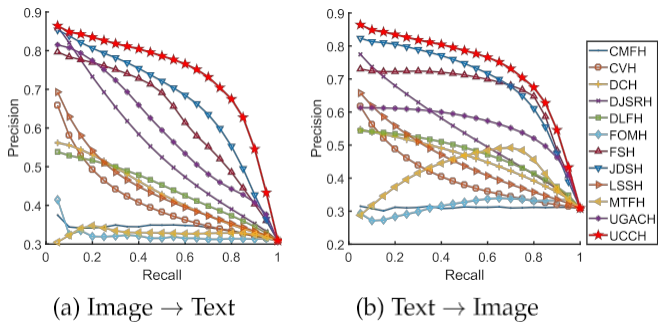


图 7.NUS-WIDE 数据集的精度-召回曲线。代码长度为 128。

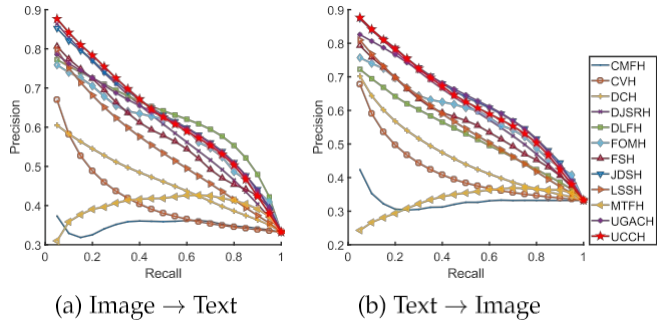


图 8.MS-COCO 数据集的精度-召回曲线。代码长度为 128。

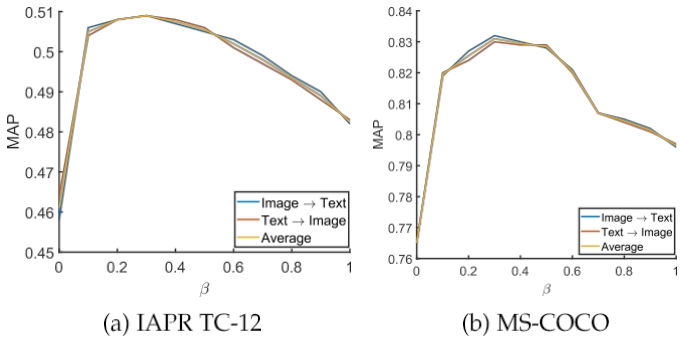
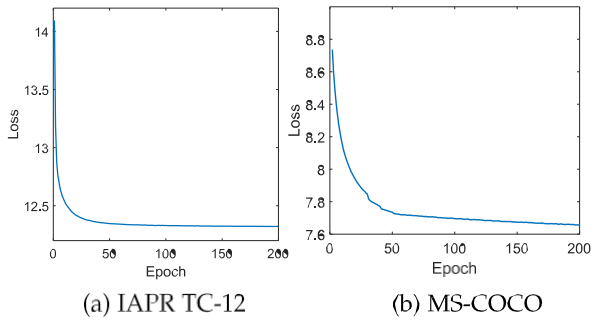


图 9.我们的 UCCH 在 IAPR TC-12 和 MS-COCO 数据集验证集上的跨模态检索性能（MAP 分数与不同 b 值的关系）。代码长度为 128。



(a) IAPR TC-12 (b) MS-COCO

图 10.我们的 UCCH 在 IAPR TC-12 和 MS-COCO 验证集上的收敛曲线。代码长度为 128。

b 在 IAPR TC-12 和 MS-COCO 数据集上的结果。从图中可以看出，对比哈希损失（S_c）和跨模态排序损失（S_r）都有助于利用多模态数据的辨别能力。更多的消融评估见第 4.3.5 节。从实验结果中可以发现，所提出的方法对参数的选择是稳健的。请注意，参数值是由其他评估部分中相应验证集的检索性能决定的。

4.3.4 趋同分析

图 10 是我们的 UCCH 在 IAPR TC-12 和 MS-COCO 数据集上的收敛曲线，其中 x 轴表示损失函数（即 S）的值，y 轴表示不同的历元数。从图中可以看出，我们的 UCCH 从第 50 个到第 100 个 epoch 快速收敛，损失在前 20 个 epoch 显著减少。因此，UCCH

我们的 UCCH 在所有数据集上的最大历时均设为 20。

4.3.5 消融研究

在本节中，我们将研究不同组件（即 S_c 和 $S_{(r)}$）对跨模态哈希检索的贡献。为了全面评估每个组件的性能，我们在 IAPR TC-12 和 MS-COCO 数据集上比较了我们的 UCCH 及其三个变体，即 UCCH

表 4
Flickr30 K 上 Recall@k 分数的性能比较

(比特)	(方法)	图像 ！ 文本			文本 ！ 图像		
		R@1	R@5	R@10	R@1	R@5	R@10
64	VSE++ [55]	10.7	28.0	39.2	8.3	25.4	37.1
	DJSRH [60]	3.6	14.4	22.1	3.4	11.6	18.5
	JDSH [61]	10.0	28.6	39.3	8.0	23.6	34.5
	UCCH	14.5	37.6	50.8	10.9	32.3	44.0
128	VSE++ [55]	11.3	31.1	42.6	9.2	27.7	40.4
	DJSRH [60]	7.7	27.2	37.8	5.9	19.9	30.3
	JDSH [61]	10.7	30.0	42.5	8.2	25.6	37.3
	UCCH	17.9	44.9	55.4	14.0	37.0	50.1
512	VSE++ [55]	13.5	34.7	48.2	10.8	31.1	43.6
	DJSRH [60]	17.9	43.5	56.3	13.3	36.3	48.9
	JDSH [61]	13.6	35.6	49.4	9.8	29.1	42.6
	UCCH	22.8	48.1	61.0	16.9	41.8	54.9

最高分以黑体显示。

只有 S_c的 UCCH、只有 S'的 UCCH 和只有 $S_{(r)}$的 UCCH。为了进行公平比较，所有变体都使用与我们的 UCCH 相同的设置进行训练。实验结果如表 3 所示。从表中可以看出，在两个数据库，没有 S_c或 S_r的 UCCH 性能比我们的 UCCH 差。因此，S_c和$S_{(r)}$对检索性能都有贡献，S_c和S_r的相互配合可以提高检索性能。从表中还可以看出，我们提出的跨模态排序损失 S_r（即公式 (14)）有效地改善了传统的最大边际排序损失 S'（即公式 (6)），这证明了考虑所有样本的有效性。更多

从 S' 之间的比较来看 $S^{'}_{r,m=0.1}$，$S^{'}_{r,m=0.5}$ 和 $S^{'}_{r,m=0.9}$ 可以看出，使用了更多的负配对、性能就会变得更好。这验证了我们关于最大边际损耗的说法，即它会忽略 TNP。此外，我们还在图 11 中绘制了 MAP 曲线，以显示不同变化的性能。从结果可以看出，由于使用了所有负对，CRL 比最大边际损耗更稳定。

表 3
不同数据集的消融研究

(数据集)	方法	图像 ！ 文本				文本 ！ 图像			
		16	32	64	128	16	32	64	128
IAPR TC-12	UCCH （仅限 S_c	0.457	0.469	0.478	0.482	0.447	0.469	0.483	0.486
	UCCH （仅限 $S^{'}_{R,M=0.1}$ 仅）	0.410	0.426	0.432	0.438	0.421	0.434	0.461	0.460
	UCCH （仅限 $S^{'}_{R,M=0.5}$ 仅）	0.423	0.446	0.463	0.470	0.434	0.450	0.471	0.479
	UCCH （仅限 $S^{'}_{R,M=0.9}$ 仅）	0.444	0.460	0.472	0.480	0.450	0.472	0.469	0.476
	UCCH （仅有 $S_{(r)}$	0.461	0.482	0.496	0.495	0.457	0.476	0.492	0.488
	完整的 UCCH	0.478	0.491	0.503	0.508	0.474	0.488	0.503	0.508
MS-COCO	UCCH （仅限 $S_{R,M=0.1}$	0.577	0.605	0.621	0.624	0.579	0.610	0.626	0.627
	UCCH （仅限 $S^{'}_{R,M=0.5}$ 仅）	0.495	0.512	0.548	0.555	0.483	0.503	0.534	0.549
	UCCH （仅限 $S^{'}_{R,M=0.9}$ 仅）	0.499	0.525	0.554	0.579	0.498	0.527	0.546	0.566
	UCCH （仅限 S' 仅）	0.529	0.535	0.554	0.558	0.525	0.545	0.546	0.560
	UCCH （仅有 $S_{(r)}$	0.563	0.574	0.599	0.602	0.563	0.576	0.606	0.609
最高分以黑体显示。	完整的 UCCH	0.605	0.645	0.655	0.665	0.610	0.655	0.666	0.677

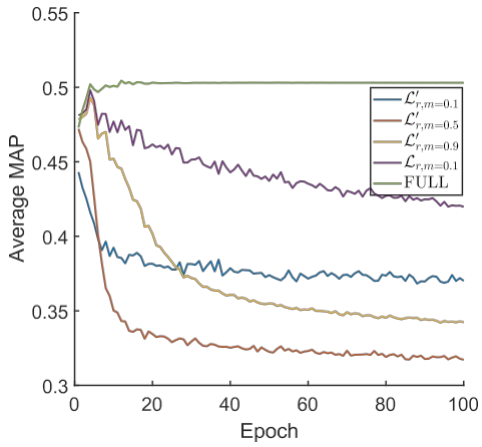
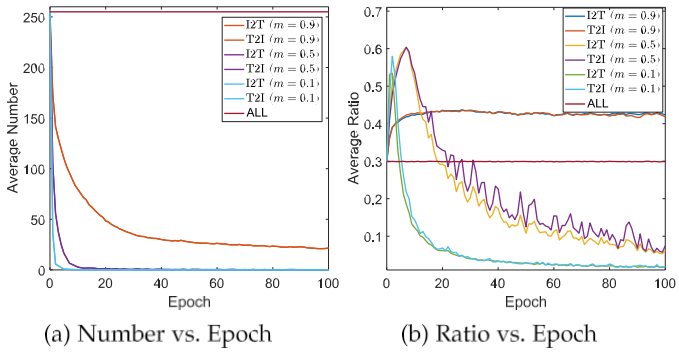


图 11.在 IAPR TC-12 验证集上使用不同变体进行跨模态检索的不同历时与平均 MAP 分数对比。代码长度为 128。



(a) Number vs. Epoch (b) Ratio vs. Epoch

图 13.IAPR TC-12 的假阴性样本分析。在不同的边距值（m= 0.1、0.5、0.9）下，(a) 显示了有效阴性对的数量与不同历时的对比。(b) 显示了 FNPs 的比例与不同时间的对比。负对分别来自图像查询文本（I2T）和文本查询图像（T2I）。"ALL "表示所有负对。

4.3.6 *基于动量的二值化的有效性*

在本节中，我们将拟议的 CCH 与无散列强制学习（CL w/o Hashing）[65] 进行比较。以研究我们的二值化技术的有效性。

图 12图 12 显示了它们在 IAPR TC-12 验证集上跨模态检索的平均 MAP 分数，随着历时的增加而增加。从结果可以看出，CL 在跨模态哈希检索中的表现并不稳定。在我们的二值化策略的帮助下，CCH 实现了更好、更稳定的性能。因此，我们可以得出结论，在 CL 的基础上开发跨模态哈希算法并非难事。

4.3.7 *假阴性配对分析*

为了进一步研究 FNP 在训练过程中的影响，我们采用最大边际损失来训练 DNN 模型，其结果是 $\mathcal{L}'_{r,m=0.1}$, $\mathcal{L}'_{r,m=0.5}$ 不同的边际值（即 $\mathcal{L}'_{r,m=0.1}$ 和 $\mathcal{L}'_{r,m=0.5}$）关于 IAPR TC-12.因此，图 13 显示了以下演变曲线

图 13a 显示了每个纪元所有批次的有效阴性对的平均数量。从图中可以看出，一些真阴性和假阴性样本会被挤出边缘。对于较低的边缘，几乎所有的阴性样本都会被挤出边缘。

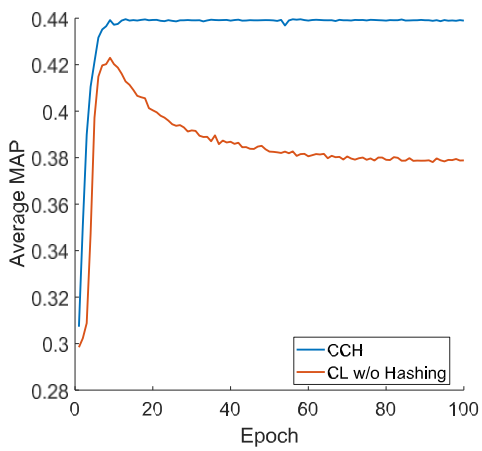


图 12.在 IAPR TC-12 验证集上进行跨模态检索时，CCH 和 CL w/o Hashing 在平均 MAP 分数方面的性能变化。

例如，m= 0.1 和 0.5。换句话说，几乎没有负数对参与进一步的训练。此外，图 13b 显示，对于较低的边际（即 m= 0.1 和 0.5），FNP 率在最初的几个历时内会增加，并随着进一步的训练而迅速下降。这可能是因为 TNP 比 FNP 更容易分离。对于较大的边缘（即 m= 0.9），更多的负样本会留在边缘内，如图 13a 所示。图 13b 显示，在最初的几个历元中，模型将使用大量负样本。因此，TNPs 会被迅速挤出边缘，而 FNPs 则会在负样本对中占主导地位，从而导致性能下降，如图 12 所示。相比之下，我们的方法不会遇到这个问题，因此性能更好，如表 3 所示。

4.3.8 *与最新技术的效率比较*

方法

在本节中，我们将在 MIRFLICKR-25 K 数据集上评估所提出的方法与一些最先进的无监督跨模态哈希方法的效率。测试平台采用英特尔 i9-10900X CPU@3.70 GHz 和 GeForce RTX 2080Ti GPU。从表 5 中可以看出，与浅层方法（如 CVH、FSH、CMFH 等）相比，我们基于 DNN 的 UCCH 可能需要更多的训练时间，这是深度方法中常见的现象，因为需要对多层神经网络进行迭代优化。然而，推理时间远远少于训练时间，与浅层方法不相上下。此外，与基于 DNN 的跨模态哈希方法（即 UGACH）相比，我们在训练和推理阶段所需的时间都要少得多。此外，在训练阶段，有对比学习（S_c）的 UCCH 比没有对比学习（$S_{(c)}$）的 UCCH 需要花费更多的时间和内存，但都在可接受的范围内（每历时约 0.67 秒，内存成本增加 9.64%）。此外，由于在推理阶段没有对比学习，它们的推理效率相同。为了进行公平比较，浅层方法使用了作者提供的默认超参数。对于基于 DNN 的方法，最大训练历元设置为 20。如图 10 所示，我们的 UCCH 可以在 20 个历元内接近收敛，但 UGACH 需要更多的历元。

表 5
使用 128 码长的 MIRFLICKR-25 K 的效率比较

方法	推理时间	训练时间	训练记忆
CVH [20]	0.15 s	12.61 s	3.11 G
FSH [18]	0.28 s	78.03 s	3.80 G
CMFH [58]	0.23 s	5.76 s	5.05 G
LSSH [57]	7.78 s	180.99 s	5.68 G
UGACH [59]	26.59 s	>12 h	14.22 G
不含 S_c的 UCCH	0.41 s	64.75 s	3.94 G
UCCH	0.41 s	78.18 s	4.32 G

推理时间是哈希整个数据集（不包括检索过程）的总时间成本。训练时间/内存是在训练集上完成相应方法训练的时间/内存成本。

与基于图的方法 UGACH [61]相比，我们的方法显著减少了训练时间和内存消耗，即训练时间从> 12 h 减少到 78.18 s，内存消耗减少了 69.62%。

5 结论

在本文中，我们提出了一种新颖的跨模态散列方法，称为无监督对比跨模态散列（UCCH），它将不同模态投射到一个共同的汉明空间中。UCCH 包括两个特定任务学习部分，即对比跨模态散列（CCH）和跨模态排序学习（CRL）。，CCH 通过一种新颖的基于动量的二值化优化器强制不同模态适应统一的二值化表示。借助该优化器，CCH 可以在无监督的情况下进行跨模态哈希（cross modal hashing）对比学习。另一方面，CRL 利用的是从所有负对而不是最难的负对中消除，这将减轻 FNP 的影响，从而促进跨模态检索。在五个广泛使用的基准数据集上的大量实验结果和综合分析表明，与 13 种最先进的方法相比，所提出的方法是有效和高效的。今后，我们计划研究如何利用少量标记数据进一步提高我们方法的性能。

参考资料

[1] Z.Yue, H. Yong, D. Meng, Q. Zhao, Y. Leung, and L. Zhang, "Robust multiview subspace learning with nonindependently and nonidentically distributed complex noise," *IEEE Trans.Neural Netw.Learn.Syst*.4, pp.

[2] X.Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: Multi-view clustering without parameter selection," in *Proc.Conf.Mach.Learn.*, 2019, pp.

[3] P.Hu, L. Zhen, D. Peng, and P. Liu, "Scalable deep multimodal learning for cross-modal retrieval," in *Proc.ACM SIGIR Conf.Res.开发。Inf.Retrieval*, 2019, pp.

[4] F.Ma, D. Meng, X. Dong, and Y. Yang, "Self-paced multi-view co- training," *J. Mach.Learn.Res.*, vol. 21, pp.

[5] J.Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *Proc.ACM SIGMOD Int.Conf.Manage.Data*, 2013、
pp.785-796.

[6] Y.Shen, L. Liu, L. Shao, and J. Song, "Deep binaries：为高效的文本-视觉交叉检索编码语义丰富的线索"，*Proc. IEEE Int.Conf.Comput.Vis.*，2017 年，第 4097-4106 页。

[7] E.Yang, T. Liu, C. Deng, and D. Tao, "Adversarial examples for hamming space search," *IEEE Trans.Cybern*.4,
pp.1473-1484, April.

[8] C.Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Trans.图像处理*》，第 27 卷第 8 期，第 3893-3903 页，2018 年 8 月。

[9] P.Hu, X. Wang, L. *Zhen*, and D. Peng, "Separated variational hashing networks for cross-modal retrieval," in *.Multimedia, 2019, pp.Multimedia*, 2019, pp.

[10] J.Song, X. Zhu, L. Gao, X.-S.Xu, W. Liu, and H. T. Shen, "Deep recurrent quantization for generating binary codes," in *Proc.Joint Conf.Artif.Intell.*, 2019pp.

[11] F.Shen *等人*，"通过检索进行分类：Binarizing data and clas- sifiers," in *Proc.ACM SIGIR Conf.Res.开发。Inf.Retrieval*, 2017, pp.

[12] C.Li、C. Deng、N. Li、W. Liu、X. Gao 和 D. Tao，"用于跨模式检索的自监督对抗散列网络"，《*Proc. IEEE Conf.Comput.Vis.Pattern Recognit.*, 2018, pp.

[13] C.Li, C. Deng, L. Wang, D. Xie, and X. Liu, "Coupled cyclegan：用于跨模态检索的无监督散列网络"，*Proc.AAAI Conf.Artif.Intell.*, 2019, pp.

[14] X.Zhou *等人*，"图卷积网络散列"，《*IEEE Trans.*50卷，第4期，第1460-1472页，2020年4月。4, pp.

[15] K.Li, G.-J. Qi, J. Ye, and K. A. Hua, "Linear subspace ranking hashing for cross-modal retrieval," *IEEE Trans.Pattern Anal.机器。Intell.*, vol. 39, no. 9, pp.

[16] X.Liu、Z. Hu、H. Ling 和 Y.-m.Cheung, "MTFH: A matrix tri- factorization hashing framework for efficient cross-modal retrieval," *IEEE Trans.Pattern Anal.Mach.Intell*.3,
pp.964-981, Mar. 2019.

[17] P.Hu, H. Zhu, X. Peng, and J. Lin, "Semi-supervised multi-modal learning with balanced spectral *decomposition*," in *Proc.Artif.Intell.*, , 2020, pp.

[18] H.Liu, R. Ji, Y. Wu, F. Huang, and B. Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *Proc. IEEE Conf.Comput.Vis.Pattern Recognit.*, 2017, pp.

[19] J.Zhang, Y. Peng, and M. Yuan, "Unsupervised generative adver-arial cross-modal hashing," in *Proc.AAAI Conf.Artif.Intell*、
pp.539-546.

[20] S.Kumar and R. Udupa, "Learning hash functions for crossview similarity search,"
in *Proc.Joint Conf.Artif.Intell*、
pp.1360-1365.

[21] X.Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in Int.Yau, and Z. Yi, "Deep subspace clustering with sparsity prior,"
in *Int.Joint Conf.Artif.Intell*、
pp.1925-1931 .

[22] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in Proc.Jiang and W.-J. Li, "Deep cross-modal hashing," in *.Comput.Pattern Recognit.Pattern Recognit.*, 2017, pp.

[23] Q.-Y. Jiang 和 W.-J.Jiang and W.-J. Li, "Discrete latent factor model for cross-modal hashing," *IEEE Trans.图像处理*》，第 28 卷，第 7 期，第 3490-3501 页，2019 年 7 月。

[24] S.Cheng, L. Wang, and A. Du, "Deep semantic-preserving recon-struction hashing for unsupervised cross-modal retrieval," *Entropy*, vol. 22, no.

[25] M.Li and H. Wang, "Unsupervised deep cross-modal hashing by knowledge distillation for large-scale cross-modal retrieval," in *Proc.Conf.多媒体检索*》，2021 年，第 183-191 页。

[26] R.Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an *invariant* mapping," in *Proc.Soc.Comput.Vis.Pattern Recognit.*, 2006, pp.

[27] Z.Wu, Y. Xiong, X. Yu Stella, and D. Lin, "*Unsupervised* feature learning via non-parametric instance discrimination," in *Proc.Comput.Vis.Pattern Recognit.*, 2018, pp.

[28] K.He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/ CVF Conf.Comput.Vis.Pattern Recognit.*, 2020, pp.

[29] Z.Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hash- ing for crossview retrieval," in *Proc.Comput.Pat- tern Recognit.Pat- tern Recognit.*, 2015, pp.

[30] Z.Lin, G. Ding, J. Han, and J. Wang, "Cross-view retrieval via probability-based semantics-preserving hashing," *IEEE Trans.网络*》，第 47 卷，第 12 期，第 4342-4355 页，2017 年 12 月。

[31] D.Mandal, K. N. Chaudhury, and S. Biswas, "Generalized seman- tic preserving hashing for N-label cross-modal retrieval," in *.Comput.Vis.Pattern Recognit.*, 2017, pp.

[32] K.Ding, B. Fan, C. Huo, S. Xiang, and C. Pan, "Cross-modal hash- ing via rank-order preserving," *IEEE Trans.Multimedia*, vol. 19, no.3, pp.

[33] X.Liu, G. Yu, C. Domeniconi, J. Wang, Y. Ren, and M. Guo, "Ranking-based deep cross-modal hashing," in *Proc.AAAI Conf.Artif.Intell.*, 2019, pp.

[34] X.Xu, F. Shen, Y. Yang, H.T. Shen, and X. Li, "Learning discrimi- native binary codes for large-scale cross-modal retrieval," *IEEE Trans.Image Process*.5，第 2494-2507 页，2017 年 5 月。

[35] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc.Comput.Pattern Recognit.Pattern Recognit.*, 2014, pp.

[36] Y.Li, Y. Wang, Z. Miao, J. Wang, and R. Zhang, "Contrastive self supervised hashing with dual pseudo agreement," *IEEE Access*, vol. 8, pp.

[37] Y.K. Jang 和 N. I. Cho, "Self-supervised product quantization for deep unsupervised image retrieval," in Proc.Cho, "Self-supervised product quantization for deep unsupervised image retrieval," in *Proc.Conf.Comput.Vis.*，2021 年，第 12065-12074 页。

[38] Z.Qiu, Q. Su, Z. Ou, J. Yu, and C. Chen, "Unsupervised hashing with contrastive information bottleneck," 2021, *arXiv:2105.06138*.

[39] X.Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[40] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[41] F.Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc.Comput.Vis*. *Pattern Recognit*、pp.37-45.

[42] J.Lu, V. E. Liong, X. Zhou, and J. Zhou, "Learning compact binary face descriptor for face recognition," *IEEE Trans.Pattern Anal.IEEE Trans.Intell.*》，第 37 卷，第 10 期，第 2041-2056 页，2015 年 10 月。

[43] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching" in Proc.Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proc.Eur.Conf.Comput.Vis.*, 2018, pp.

[44] H.Ham、J.-W.Ha, and J. Kim, "Dual attention networks for multi- modal reasoning and matching," in *Proc.Comput.Vis.Pattern Recognit.*, 2017, pp.

[45] P.Diederik Kingma 和 J. Ba, "Adam: A method for stochastic opti- mization," in *Proc.Conf.Learn.Representations*, 2015, pp.

[46] J.Mark Huiskes 和 M. S. Lew，"MIR flickr 检索评估"，*Proc.ACM Int.Conf.Multimedia Inf.Retrieval*, 2008、pp.39-43 .

[47] H.J. Escalante *等人*，"The segmented and annotated IAPR TC-12 benchmark," *Comput.Vis.Image Understanding*, vol. 114, no.4, pp.419-428, 2010.

[48] N.Rasiwasia *等人*，"A. new approach to cross-modal multimedia retrieval," in *Proc.Conf.多媒体*，2010 年，第 251-260 页。

[49] T.-Y. Lin et al.Lin *等人*，"Microsoft coco：上下文中的通用对象"，《微软 *Proc.Eur.Proc.Comput.*2014, pp.

[50] P.Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations： New similarity metrics for semantic inference over event descriptions," *Trans.Assoc.语言学*》，第 2 卷，第 67-78 页，2014 年。

[51] A.Paszke *等人*，"PyTorch：一个命令式的高性能深度学习库"，*第 33 届 Adv.Process.Syst.*, 2019, pp.

[52] K.Simonyan and A. Zisserman, "Very deep convolutional net- works for large-scale image recognition," in *Proc.Conf.Learn.Representations*, 2015.

[53] K.Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details：深入卷积网"，2014 年，*arXiv:1405.3531*。

[54] H.Jey Lau 和 T. Baldwin，"doc2vec 的经验评估与文档嵌入生成的实用见解"，《*Proc.Workshop Representation Learn.NLP*》，2016 年，第 78-86 页。

[55] F.Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "VSE++: Improving visual-semantic embeddings with hard negatives," in *Proc. Brit.Mach.Vis.Conf.*, 2018.

[56] R.-C.Tu *et al.*, "Deep cross-modal hashing with hashing functions and unified hash codes jointly learning," *IEEE Trans.Knowl.数据工程*》，第 34 卷，第 2 期，第 560-572 页，2022 年 2 月。

[57] X.Wang, X. Zou, E. M. Bakker, and S. Wu, "Self-constraining and attention-based hashing network for bit-scalable cross-modal retrieval," *Neurocomputing*, vol. 400, pp.

[58] X.Lu, L. Zhu, Z. Cheng, J. Li, X. Nie, and H. Zhang, "Flexible online multi-modal hashing for large-scale multimedia retrieval," in *Proc.Multimedia, 2019, pp.多媒体*》，2019 年，第 1129-1137 页。

[59] J.Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *Proc.ACM SIGIR Conf.Res.开发。Inf.Retrieval*, 2014, pp.

[60] G. Ding, Y. Guo, J. Zhou, and Y. Gao, "Large-scale cross-modality search via collective matrix factorization hashing," *IEEE Trans.图像处理*》，第 25 卷，第 11 期，第 5427-5440 页，2016 年 11 月。

[61] J.Zhang, Y. Peng, and M. Yuan, "Unsupervised generative adver- sarial cross-modal hashing," in *Proc.Artif.Intell.*, 2018, pp.

[62] S.Su, Z. Zhong, and C. Zhang, "Deep joint-semantics reconstruct- ing hashing for large-scale unsupervised cross-modal ," in *Proc.Conf.Comput.Vis.*, 2019, pp.

[63] S.Liu, S. Qian, Y. Guan, J. Zhan, and L. Ying, "Joint-modal distri- bution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval," in *Proc. 43rd Int.ACM SIGIR Conf.Res.Develop.Inf.Retrieval*, 2020, pp.

[64] J.Yu, H. Zhou, Y. Zhan, and D. Tao, "Deep graph-neighbor coher- ence preserving network for unsupervised cross-modal hashing," in *Proc.AAAI Conf.Artif.Intell.*, 2021, pp.

[65] Y.Tian, D. Krishnan, and P. Isola, "Contrastive Multiview Coding," in *Proc.Comput.Vis.Conf.*，2020 年，第 776-794 页。