

基于大规模预训练 CLIP 模型微调的图文检索方法

摘要

随着互联网的迅速发展，海量数据的涌现使得从大量信息中筛选出有价值内容变得日益重要。信息检索技术因此成为一项关键技术，尤其是在面对真实世界中复杂的多模态数据时。传统的信息检索模型往往只能处理单一模态数据，而现实情况通常更为复杂，涉及文本、图像等多种数据类型。为了解决这一挑战，本文提出了一种结合自然语言处理（NLP）、计算机视觉（CV）技术，基于多模态模型 **CLIP**（**Contrastive Language-Image Pre-training**）的信息检索模型。该模型特别针对中文数据集进行了优化，通过大量公共数据集的预训练，并在特定比赛数据集上进行微调，显著提升了检索精度，超越了直接训练模型的性能。

本文首先对图像与文本数据进行了建模，通过预处理将异构的图像与文本信息转化为结构化的张量形式，使得基于深度学习的模型能更好地对其进行处理。接下来，我们对问题一**图像检索**（Text-to-Image, T2I）任务的与问题二的**文本检索**（Image-to-Text, I2T）任务进行了统一建模，将两大检索问题转化为图像与文本的相似度衡量问题，按照相似度对检索目标进行排序，以具有最高相似度的目标作为检索结果。为了度量图像文本多模态数据的相关程度，我们引入 CLIP 模型对多模态特征进行融合，其基于对比学习（**Contrastive Learning**）方法，训练出能够将图像和文本映射到同一嵌入空间的**图像编码器**（Image Encoder）和**文本编码器**（Text Encoder）。这一映射过程便于计算不同模态数据之间的余弦相似度，从而实现高效的图文互检索功能。

具体而言，为了兼顾模型精度与效率，我们实现的 CLIP 模型采用 **ViT-L/14** 模型作为图像编码器，以及 **RoBERTa** 模型作为文本编码器。同时，这些编码器均经过充分预训练，显著加快了后续训练的收敛速度。接下来，我们收集了一系列高质量的公开数据集，构建了一个包含大约八百万图像文本对的中文多模态预训练数据集，并基于该数据集对所实现的 CLIP 模型进行预训练，得到一个泛化性能强大的基线模型。基于对问题的统一建模，我们的基线模型在问题一的 T2I 任务中达到了 70.92% 的 R@5 精度，在问题二的 I2T 任务中达到了 69.72% 的 R@5 精度。

为了进一步挖掘该模型的潜力，我们针对比赛数据集（容量仅为 5 万），通过**图像裁剪**、**图像翻转**、**文本翻译**等方式进行数据增强，得到一个容量为 40 万的增强数据集。进而，我们使用增强数据集对预训练的 CLIP 基线模型进行微调。经过仅仅 5 个回合的微调，我们的模型便能在问题一的 T2I 任务中达到了 **76.82%** 的 R@5 精度，在问题二的 I2T 任务中达到了 **76.26%** 的 R@5 精度。该结果相较基线模型有着显著地提升，验证了微调策略与数据增强方法的有效性。

综上，本文对文本检索任务与图像检索任务进行了统一建模，基于 CLIP 框架实现了一个兼顾精度与效率的多模态模型，并整合高质量公开数据集进行预训练得到泛化性能强大的基线模型。进而，我们对比赛数据集进行增强，并对基线模型微调，显著提升了特定任务下的性能表现，对中文信息检索领域具有重要的理论和实践意义。

关键词： 多模态特征融合 图文检索 预训练—微调 对比学习 深度学习

目录

一、 问题描述与假设	1
1. 问题背景	1
2. 解决问题	1
(1) 图像检索	2
(2) 文本检索	2
3. 评估指标	2
4. 基本假设	2
二、 问题建模	3
1. 图像建模	3
2. 文本建模	3
3. 图文检索建模	4
(1) 图像检索	4
(2) 文本检索	4
三、 数据分析与处理	5
1. 数据统计	5
(1) 图像数据特征	5
(2) 文本数据特征	6
2. 数据清洗	6
(1) 图像规格统一化	6
(2) 文本异常值处理	6
3. 数据增强	7
(1) 图像数据增强	7
(2) 文本数据增强	7
4. 数据集划分	7
5. 压缩与序列化	8
(1) 图像压缩	8
(2) 图像序列化	8
(3) 文本序列化	9
(4) LMDB 内存索引	10
四、 多模态特征融合方法研究	10
1. 算法选择	11
2. 背景知识	11
(1) 对比学习	11

(2)	卷积神经网络 (CNN)	12
(3)	Transformer	14
(4)	Vision Transformer	14
(5)	词向量模型	16
(6)	BERT	16
3.	模型结构	17
(1)	图像编码器	17
(2)	文本编码器	18
4.	训练目标	19
5.	模型推理	19
(1)	数据预处理	19
(2)	图像文本编码	19
(3)	相似度计算	20
6.	改进策略	20
(1)	梯度累积 (Gradient Accumulation)	20
(2)	掩码特征学习	22
(3)	FlashAttention	22
五、 实验与结果分析		23
1.	模型预训练	23
(1)	数据集构建	23
(2)	模型预训练	24
(3)	“零样本”测试	24
2.	模型微调	25
(1)	数据集构建	25
(2)	模型微调	26
(3)	微调结果测试	26
3.	结果分析	29
(1)	预训练收益	29
(2)	微调有效性	29
4.	模型总结	30
参考文献		33

一、问题描述与假设

1. 问题背景

随着近年来智能终端设备和多媒体社交网络平台的飞速发展，多媒体数据呈现海量增长的趋势，使当今主流的社交网络平台充斥着海量的文本、图像等多模态媒体数据，也使得人们对不同模态数据之间互相检索的需求不断增加。有效的信息检索和分析可以大大提高平台多模态数据的利用率及用户的使用体验，而不同模态间存在显著的语义鸿沟，大大制约了海量多模态数据的分析及有效信息挖掘。因此，在海量的数据中实现跨模态信息的精准检索就成为当今学术界面临的重要挑战。图像和文本作为信息传递过程中常见的两大模态，它们之间的交互检索不仅能有效打破视觉和语言之间的语义鸿沟和分布壁垒，还能促进许多应用的发展，如跨模态检索、图像标注、视觉问答等。

图像文本检索指的是输入某一模态的数据（例如图像），通过训练的模型自动检索出与之最相关的另一模态数据（例如文本），它包括两个方向的检索，即基于文本的图像检索和基于图像的文本检索，如图 1 所示。基于文本的图像检索的目的是从数据库中找到与输入句子相匹配的图像作为输出结果；基于图像的文本检索根据输入图像，模型从数据库中自动检索出能够准确描述图像内容的文字。然而，来自图像和来自文本的特征存在固有的数据分布的差异，也被称为模态间的“异构鸿沟”，使得度量图像和文本之间的语义相关性困难重重。



图 1: 图像文本检索

2. 解决问题

本赛题是利用附件 1 的数据集，选择合适方法进行图像和文本的特征提取，基于提取的特征数据，建立适用于**图像检索**的多模态特征融合模型和算法，以及建立适用于**文本检索**的多模态特征融合模型和算法。基于建立的“多模态特征融合的图像文本检索”模型，完成以下两个任务，并提交相关材料。

(1) 图像检索

基于图像检索的模型和算法，利用附件 2 中“word_test.csv”文件的文本信息，对附件 2 的 ImageData 文件夹的图像进行图像检索，并罗列检索相似度较高的前五张图像，将结果存放在“result1.csv”文件中（模板文件详见附件 4 的 result1.csv）。其中，ImageData 文件夹中的图像 ID 详见附件 2 的“image_data.csv”文件。

(2) 文本检索

基于文本检索的模型和算法，利用附件 3 中“image_test.csv”文件提及的图像 ID，对附件 3 的“word_data.csv”文件进行文本检索，并罗列检索相似度较高的前五条文本，将结果存放在“result2.csv”文件中（模板文件见附件 4 的 result2.csv）。其中，“image_test.csv”文件提及的图像 ID，对应的图像数据可在附件 3 的 ImageData 文件夹中获取。

3. 评估指标

图像文本检索包括两个具体的任务，即文本检索（Image-to-Text，I2T），即针对查询图像找到相关句子；以及图像检索（Text-to-Image，T2I），即给定查询语句检索符合文本描述的图像。为了与现有方法公平地进行比较，在文本检索问题和图像检索问题中都采用了广泛使用的评价指标：召回率 Recall at K (R@K)。R@K 定义为查询结果中真实结果（Ground Truth）排序在前 K 的比率，通常 K 可取值为 1、5 和 10，计算公式如式 (1) 所示。

$$R@K = \frac{\text{Matched}_{\text{top-}K}}{\text{GroundTruth}_{\text{total}}} \quad (1)$$

其中， $\text{GroundTruth}_{\text{total}}$ 表示真实匹配结果出现的总次数， $\text{Matched}_{\text{top-}K}$ 表示在排序前 K 个输出结果中出现匹配样本的次数。 $R@K$ 反映了在图像检索和文本检索中模型输出前 K 个结果中正确结果出现的比例。本赛题的评价标准设定 $K = 5$ ，即评价标准为 R@5。

4. 基本假设

为了构建图像文本双向检索模型，我们做出如下合理的假设：

1. 训练数据集中的图像文本匹配关系正确可靠；
2. 训练集与测试集中的图像文本对的具有一致的数据分布；
3. 任务一的测试集中，每条文本都存在与之匹配的图像；
4. 任务二的测试集中，每幅图像都存在与之匹配的文本。

二、问题建模

多模态图文检索的本质上是对图像和文本两种模态的信息进行压缩编码，压缩编码过程可以利用传统方法也可以利用深度学习方法，但最终会得到图像和文本的压缩编码嵌入 embedding。

在此基础之上，如果得到的 embedding 是空间对齐的，即两个模态的编码在一个语义空间中，那么就可以利用一般的相似度匹配进行图文检索；如果得到的 embedding 是空间不对齐的，那么就需要学习相似度匹配方法来更好地匹配两个图文编码向量的相似度，这样的效率虽然高，但得到的效果显然没有进行向量空间对齐的方法好。

空间对齐指的是公共空间特征学习方法，相似度学习指的是跨模态相似性度量方法。前者为主流方法，并且现在的方法都是基于深度学习模型，同时目前的 SOTA 模型主要为：CLIP、ALBEF、BLIP-2 这些较为成熟的方法模型。

这里，我们选择的是空间对齐的特征学习方法，将图像与文本投影到同构的特征空间，从而利用相似度匹配进行图文检索。下面，我们分别对图像与文本进行建模。

1. 图像建模

在计算机视觉（Computer Vision, CV）领域，常用的图像表示方法是使用张量。张量是多维数组的扩展，可以表示高维数据。对于彩色图像，我们使用三维张量 $x \in \mathbb{R}^{H \times W \times C}$ 描述。其中， H 表示高度， W 表示宽度， C 表示通道数（对于常见的 RGB 图像，其通道数为 3）。

由于题目数据中的图像具有不同的长宽比、分辨率，不利于模型统一处理。于是，我们按照以下规则，对所有图像进行预处理：

1. 对于所有长宽比小于 2:1 的图像，将其拉伸为 1:1，使用双立方插值法（Bicubic Interpolation）下采样至 224×224 分辨率。
2. 对于所有长宽比大于 2:1 的图像，截断其长边，仅保留长宽比小于 2:1 的部分，再按照规则 1. 进行处理。

至此，我们可以将所有图像的分辨率处理为 224×224 ，进而使用四维张量 $X \in \mathbb{R}^{N \times H \times W \times C}$ 表示整个数据集，其中 N 为图像数量， $H = W = 224$ ， $C = 3$ 。

2. 文本建模

在自然语言处理（Natural Language Processing, NLP）领域，文本被视作一个由单词组成的序列。为了便于表达，将所有可能出现的单词汇集成一张表，称为词汇表（Vocabulary），其中每个单词对应一个唯一的序号（Index）。

为了使用深度学习模型学习单词的语义，我们需要将每个词语用一个固定长度的向量表示，分为稀疏表示（如 One-hot 编码）和分布式表示（如 Word2Vec）。由于稀疏表示的诸多弊端，这里我们采用单词的分布式表示。分布式表示将词转化为一个定长（设为 D_{emb} ）、稠密并且互相存在语义关系的向量。此处的存在语义关系可以理解为：分布相似的词，是具有相同的语义的。

如此一来，一切文本都能被映射为一个由定长词向量组成的序列。然而，文本中单词的数量或多或少，因此单词序列的长度无法确定，这是不利于语言建模的。为了解决这个问题，常用的方法

是指定一个最大序列长度（设为 L ），然后按以下规则处理不同长度的文本：

1. 对于单词数量小于 L 的文本，在其后方填充若干特殊的单词“<pad>”，使其长度达到 L 。
2. 对于单词数量超过 L 的文本，舍弃第 L 个单词后的内容。

于是，我们可以将所有文本处理为长度为 L 的单词序列，其中每个单词被表示为一个 D_{emb} 维向量。也就是说，一段文本可以被表示为一个形状为 $L \times D_{\text{emb}}$ 的矩阵。进而，我们对数据集中所有文本进行处理，得到一个三维张量 $Y \in \mathbb{R}^{M \times L \times D_{\text{emb}}}$ ，其中 M 是文本数量。

3. 图文检索建模

为了进行图文检索（包括图像检索、文本检索），关键是定义一个匹配度函数。该函数的输入为一幅图像以及一段文本，输出为图像与文本的内容匹配程度，即 $\text{Match}(Image, Text) \in [-1, 1]$ 。其数值大小表示匹配程度，-1 表示完全不匹配，1 表示完全匹配。

对于图像集合 $X \in \mathbb{R}^{N \times H \times W \times C}$ ，以及文本集合 $Y \in \mathbb{R}^{M \times L \times D_{\text{emb}}}$ ，可以得到一个匹配度矩阵 $\text{Score} \in \mathbb{R}^{N \times M}$ 表示所有“图像—文本”对的匹配情况。具体而言， Score 的定义如公式 (2) 所示。

$$\text{Score}[i, j] = \text{Match}(X_i, Y_j), \quad 1 \leq i \leq N, \quad 1 \leq j \leq M. \quad (2)$$

(1) 图像检索

类似的，在图像检索（Text-to-Image，T2I）任务中，我们需要为每段文本寻找与其匹配程度最高的 K 幅图像。而每段文本与 Score 矩阵中的一列所对应，为了实现该目的，我们沿着列方向对 Score 矩阵进行 ArgSort 操作，使每列的图像按照与每段文本的匹配程度排序，并以检索形式呈现。接下来，取检索矩阵的前 K 行，得到矩阵 $\text{ColTop} \in \mathbb{R}^{K \times M}$ ，如公式 (3) 所示，其中 [...] 表示子矩阵检索操作。

$$\text{ColTop} = \text{ArgSort}(\text{Score}, \dim=0)[:, :K] \quad (3)$$

此时， ColTop 的第 j 列对应与文本 Y_j 匹配程度最高的 K 幅图像的位置，则 T2I 任务的结果如公式 (4) 所示，其中 {...} 表示集合。

$$\text{T2I}(Y_j) = \{X_i \mid i \in \text{ColTop}[:, j]\} \quad (4)$$

(2) 文本检索

类似的，在文本检索（Image-to-Text，I2T）任务中，我们需要为每幅图像寻找与其匹配程度最高的 K 段文本。而每幅图像与 Score 矩阵中的一行所对应，为了实现该目的，我们沿着行方向对 Score 矩阵进行 ArgSort 操作，使每行的文本按照与每幅图像的匹配程度排序，并以检索形式呈现。接下来，取检索矩阵的前 K 列，得到矩阵 $\text{RowTop} \in \mathbb{R}^{N \times K}$ ，如公式 (5) 所示。

$$\text{RowTop} = \text{ArgSort}(\text{Score}, \dim=1)[:, :K] \quad (5)$$

此时, RowTop 的第 i 行对应与图像 X_i 匹配程度最高的 K 段文本的位置, 则 I2T 任务的结果如公式 (6) 所示。

$$\text{I2T}(X_i) = \{Y_j \mid j \in \text{RowTop}[i, :]\} \quad (6)$$

根据所建立的模型, 我们只需实现 $\text{Match}(Image, Text)$ 函数, 得到图像与文本的匹配度, 即可完成 I2T 任务与 T2I 任务。

三、数据分析与处理

1. 数据统计

对收集的数据进行统计是数据预处理的关键步骤, 这有助于提前了解数据集的整体特性和潜在的问题。在图像数据中, 我们统计了图像的大小、分辨率、长宽比的分布情况, 同时统计了文本数据的长度范围、并使用词云图对词频进行了可视化。

(1) 图像数据特征

表 1: 比赛数据集图像大小分布

大小 (KB)	(0, 50]	(50, 200]	(200, 500]	(500, 1000]	(1000, 5000]	(5000, 10000]	(10000, 15000]
图像数量	51716	47056	4366	1085	727	49	1
占比	49.25%	44.82%	4.16%	1.03%	0.69%	0.05%	9.52E-04%

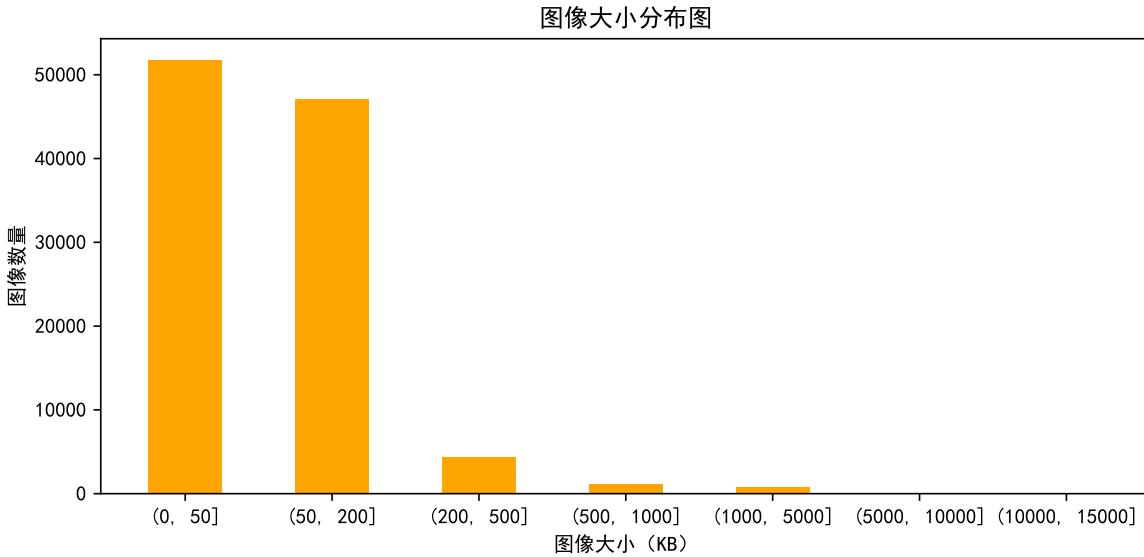


图 2: 比赛数据集图像大小柱状图

从图像大小分布图可以看出数据集中图像大多集中于 $(0, 200]$ 区间,

表 2: 比赛数据集图像分辨率分布

分辨率 (MB)	(0, 0.5]	(0.5, 1]	(1, 5]	(5, 10]	(10, 20]	(20, 40]	(40, 80]
图像数量	62894	30288	10342	869	508	92	7

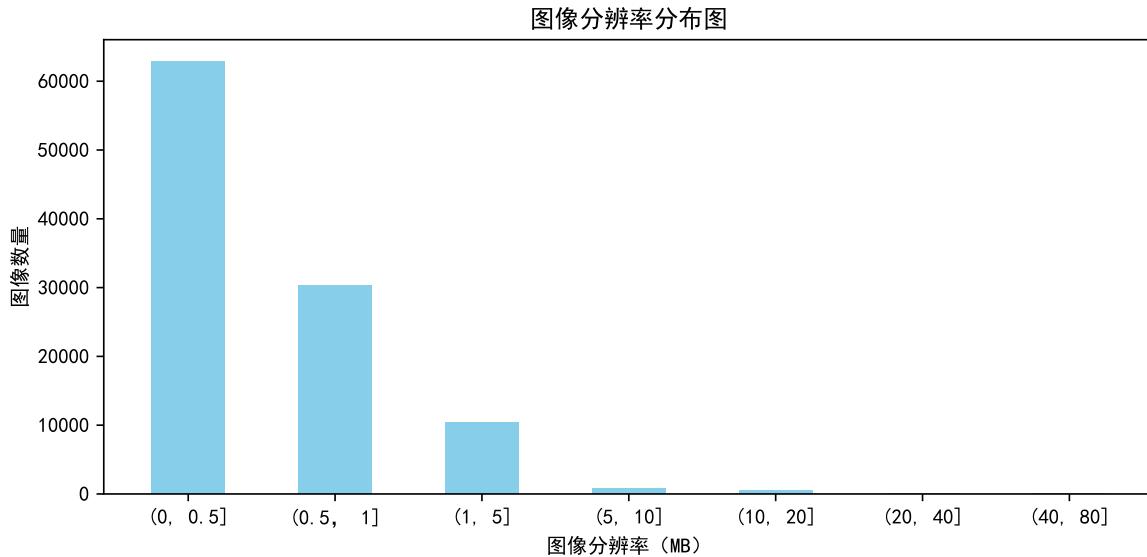


图 3: 比赛数据集图像分辨率柱状图

(2) 文本数据特征

2. 数据清洗

对附件中的数据进行数据清洗，以得到更高质量的数据集，便于模型的预训练。

(1) 图像规格统一化

浏览 ImageData 文件夹中的图像数据可以发现，这些图像的大小不一，而且有些图像的像素很高，不能直接传入神经网络中进行训练，必须把他们处理成统一的大小。

按照章节 [二](#)、中提到的方法，我们对每张图像应用随机裁剪、拉伸、双立方下采样等操作，将所有图像的分辨率处理为统一的 224×224 ，并对数据进行归一化，以便于后续任务的进行。处理后的结果如图 [6](#) 所示。

(2) 文本异常值处理

观察给出的文本数据，可以发现这些文本的信息比较杂乱，质量参差不齐，其中包含了很多特殊字符、无用的空白空格和换行、连续出现的标点符号等。

我们使用正则表达式 (regex) 来识别、替换或删除这些文本中存在的噪声。对文本数据进行清洗有助于确保数据准确、可靠和一致，以用于分析和建模。清洗后的文本样例如图 [7](#) 所示。

表 3: 比赛数据集图像长宽比分布

长宽比	(0, 0.5]	(0.5, 1]	(1, 1.5]	(1.5, 2]	(2, 3]
图像数量	1113	22120	42117	34900	4750

表 4: 比赛数据集文本长度分布

最大长度	最小长度	平均长度	1
38	2	23	

3. 数据增强

(1) 图像数据增强

图像数据增强方法主要分为两类，一种类型的增强涉及数据的空间/几何变换，如裁剪和调整大小、旋转和翻转。另一种类型的增强涉及外观变换，例如颜色失真（包括颜色下降、亮度、对比度、饱和度、色调）、高斯模糊和 Sobel 过滤。

SimCLR 用实验证明了数据增强操作的组合对学习好的表征是至关重要的，而且无监督的对比学习受益于比监督学习更强的数据增强。因此，我们所使用的数据增强操作是这几种方法的组合：

1. 随机裁剪和调整大小；
2. 随机旋转；
3. 随机翻转；
4.

(2) 文本数据增强

为了能在不改变原文语义的情况下，生成一定数量的训练语料文本，同时提升模型的泛化性能、干扰波动的能力，我们主要使用以下几种文本数据增强方法：

1. **同义词替换**: 在这种方法中，我们从句子中随机取出一个词，将其替换为对应的同义词。图 8 演示了该方法的流程。
2. **翻译互转**: 将文本翻译成另外一种语言，然后再翻译回来。同时，我们也可以翻译成多个语言，从而得到多条回译样本。图 9 演示了该方法的流程。

4. 数据集划分

为了进行训练与测试，我们对“附件 1”中的训练集进行了划分，将其中的 90% 作为真正的训练集，剩下的 10% 作为验证集，用来评估模型的训练效果。根据验证集上的评估指标，我们可以更准确地掌握模型的性能表现，并量化其泛化能力，避免对训练集数据过拟合。具体而言，表 5 详细说明了数据集的划分情况，以及各个子集的组成。



图 4: 文本数据词云图

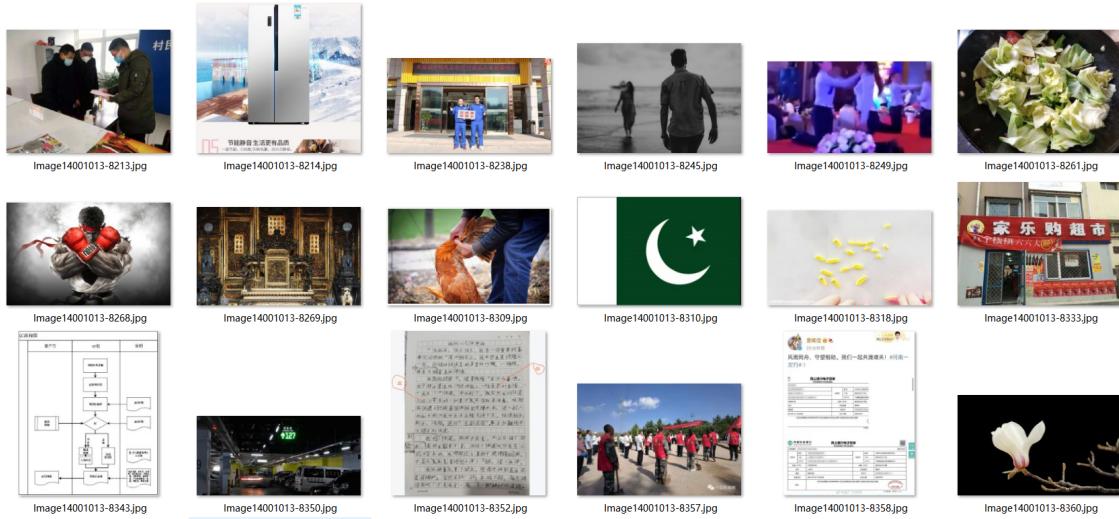


图 5: 原始图像文件概览。

5. 压缩与序列化

(1) 图像压缩

为了加快文件读取速度，我们对数据增强后的图像进行 JPEG 压缩，在保证 90% 以上的图像质量前提下，极大地减小了图像的体积。经统计，图像的体积相比未压缩能减少 75%。

(2) 图像序列化

为保证文件处理效率，我们不是将图片以大量的小文件方式存放，而是将训练/验证/测试图片先经过压缩，再将其字节码序列化为 base64 格式，分别存放在 tsv 文件中，每行表示一张图片，包

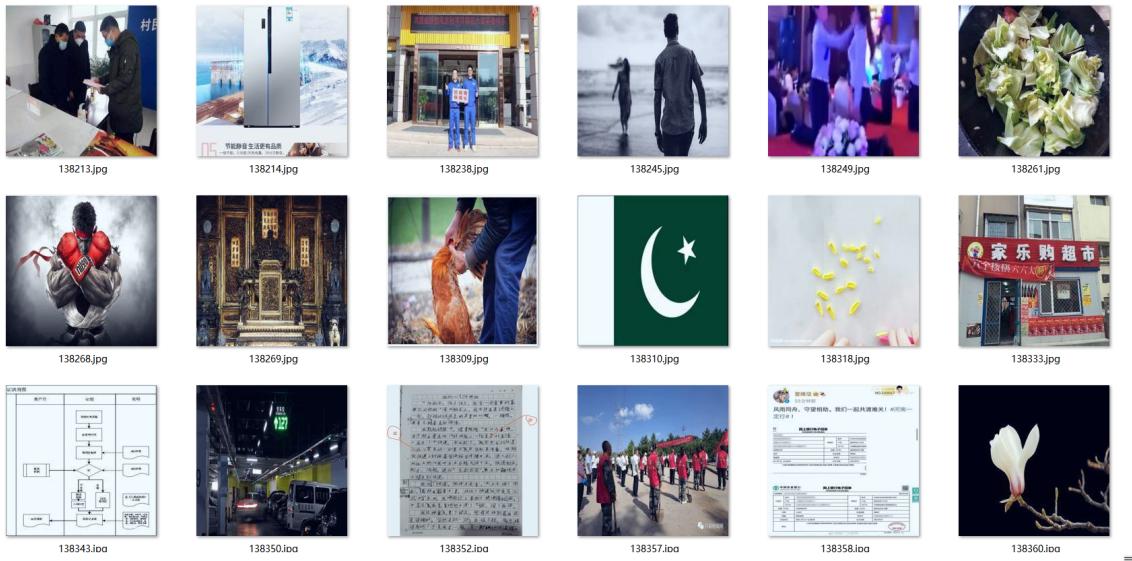


图 6: 处理后的图像文件概览。

1	image_id	caption
2	Image14001001-0000.jpg	《绿色北京》摄影大赛胡子<人名>作品
3	Image14001001-0001.jpg	只看阵容对手就已经吓尿了，巴西队黄金一代神挡杀神，佛挡杀佛！
4	Image14001001-0002.jpg	招聘计划学校现有教职工1500余人。
5	Image14001001-0003.jpg	从没买过游戏机的新手买任天堂switch要注意什么？
6	Image14001001-0004.jpg	老人养狗，为的是有个伴儿，年轻人养狗，图的是有个宠物。
7	Image14001001-0005.jpg	增涂金属膜或增反介质膜；2. 以临界角或掠入射。
8	Image14001001-0006.jpg	石景山区纪委监委机关党委各党支部召开专题组织生活会
9	Image14001001-0007.jpg	外地客户自行送修的们会加急为您的机器排除故障，争取当天完成维修。
10	Image14001001-0010.jpg	rng夺冠后, 1o1插画师发文称赞galia, 只因用了jk1皮肤而
11	Image14001001-0012.jpg	秋天来了来一碗麻辣鸭血麻辣鲜香软滑细嫩美味无比
12	Image14001001-0013.jpg	青海省重大科技专项中期进展检查会在金诃藏药隆重召开
13	Image14001001-0014.jpg	校长<人名>应邀出席部省共建在粤“双一流”高校协议签约仪式
14	Image14001001-0015.jpg	网大又迎来一部“烂片”，剧情魔改女主深v, 差点以为在看
15	Image14001001-0016.jpg	而摆在桌上以及拿在战士们手中的轻武器，则是足以令人眼前一亮。
16	Image14001001-0017.jpg	药书手抄日记本3本合售 包邮挂

图 7: 处理后的文本示例。

含图片 id 和图片的 base64 编码。形式如下：

“138213 /9j/4AAQSkZJRgABAQAAAQABAAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQ”

这样，便可以避免单独存储大量的图片小文件，而是直接存储一个文本文件。

(3) 文本序列化

我们将文本信息和图文对的匹配关系保存在 JSONL 文件中。每行是一个 JSON 对象，包含文本 id、文本内容和匹配的图片 id 列表。形式如下：

“{“text_id”: 8428, “text”: ”高级感托特包斜挎”, “image_ids”: [10345, 17602]}”



图 8: 同义词替换示意图。



图 9: 翻译互转示意图。

这样，就可以方便地将文本和图片进行匹配和关联。

(4) LMDB 内存索引

LMDB 全称为 Lightning Memory-Mapped Database，就是非常快的内存映射型数据库，LMDB 使用内存映射文件，可以提供更好的输入/输出性能，对于用于神经网络的大型数据集（比如 ImageNet），可以将其存储在 LMDB 中。为了进一步提高模型训练时随机读取数据的效率，我们还需要将 tsv 和 jsonl 文件一起序列化，转换为内存索引的 LMDB 数据库文件，方便训练时的随机读取，从而减少每个训练 Batch 的额外时间消耗。

四、多模态特征融合方法研究

模态（Modal）是事情经历和发生的方式，我们生活在一个由多种模态（Multimodal）信息构成的世界，包括视觉信息、听觉信息、文本信息、嗅觉信息等等，当研究的问题或者数据集包含多种这样的模态信息时我们称之为多模态问题，研究多模态问题是推动人工智能更好的了解和认知我们周围世界的关键。

每一种信息的来源或者形式，都可以称为一种模态。例如，人有触觉，听觉，视觉，嗅觉；信息的媒介，有语音、视频、文字等；多种多样的传感器，如雷达、红外、加速度计等。以上的每一种都可以称为一种模态。相较于图像、语音、文本等多媒体（Multi-media）数据划分形式，“模态”是一个更为细粒度的概念，同一媒介下可能存在不同的模态。比如我们可以把两种不同的语言当做是两种模态，甚至在两种不同情况下采集到的数据集，亦可认为是两种模态。

在本题目中，我们需要对——图像与文本，这两种模态的数据特征进行融合。

表 5: 比赛数据集划分明细

数据来源	“附件 1”		“附件 2”	“附件 3”
数据类型	训练集	验证集	T2I 测试集	I2T 测试集
图像数量	45000	5000	50000 (搜索空间)	5000 (样本)
文本数量	45000	5000	5000 (样本)	50000 (搜索空间)

1. 算法选择

多模态检索任务的核心是对多模态数据相似度的度量。在同一维度空间下，人们可以通过欧氏距离、闵式距离等方式来计算两个向量之间的距离。因此，为了比较异构数据的相似度，一个通用的方法是将它们映射到同一个公共表示空间进行相似度的学习。我们首先可以确定任务的解决方案是基于深度学习的多模态特征融合，不同于传统图像处理和传统机器学习算法与简单神经网络的实现，近些年来基于深度学习的多模态模型针对各种复杂应用场景表现性能更好。

在多模态领域，当前的主流模型是基于对比学习的大规模预训练模型。其代表便是 CLIP 模型。CLIP (Contrastive Language-Image Pre-training) 是一个跨模态学习模型 [1]，由 OpenAI 在 2021 年提出。CLIP 模型的核心思想是通过对比学习的方式，将图像和文本映射到同一个嵌入空间中，使得语义上相关的图像和文本在该空间中更接近。

为了更好地理解所使用的改进 CLIP 模型，下面对相关模型以及使用到的技术进行分别介绍。

2. 背景知识

(1) 对比学习

对比学习 (Contrastive Learning) 是一种自监督学习方法 [2]，它通过学习数据的相似性和差异性来学习特征的一般表示。其采用的具体思想是将样例和与它语义相似的样本（正样本）及与它语义不相似的样本（负样本）进行对比，通过设计模型结构和对比损失，使语义相近的样本对应的表示在表示空间更接近，语义不相近的例子对应的表示距离更远，以达到类似聚类的效果，如图 10 所示。

对比学习的实施通常包括以下几个方面：

1. 正负样本的定义：在对比学习中，图像特征和文本特征构成特征矩阵，该矩阵中图文相匹配为正样本，不匹配为负样本，因此特征矩阵的对角线元素均为正样本，其他元素为负样本。
2. 相似度计算：使用余弦相似度来表示特征之间的相似度，A、B 矩阵的余弦相似度可由公式 (7) 描述。

$$\text{Cosine-Similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|} = \frac{\mathbf{A}}{\|\mathbf{A}\|} \cdot \frac{\mathbf{B}}{\|\mathbf{B}\|} \quad (7)$$

3. 损失函数：定义一个损失函数来训练模型，使得正样本对的相似度高于负样本对的相似度。典

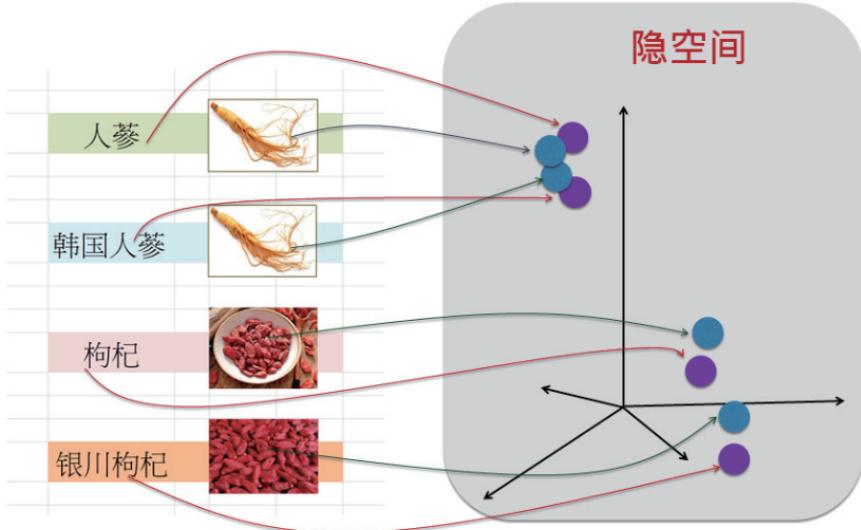


图 10: 对比学习示意图。

型的损失函数包括三元组损失（Triplet Loss）、对比损失（Contrastive Loss）和交叉熵损失等。公式 (8) 展示了常用的 InfoNCE 损失函数的表达式。

$$L_I = L_{\text{InfoNCE}}(z_i^I, z_i^T) = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\text{sim}(z_i^I, z_i^T)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(z_i^I, z_j^T)/\tau)} \quad (8)$$

(2) 卷积神经网络 (CNN)

卷积神经网络（Convolutional Neural Networks, CNN）这个概念的提出可以追溯到二十世纪 80-90 年代，但是有那么一段时间这个概念被“雪藏”了，因为当时的硬件和软件技术比较落后，而随着各种深度学习理论相继被提出以及数值计算设备的高速发展，卷积神经网络得到了快速发展。

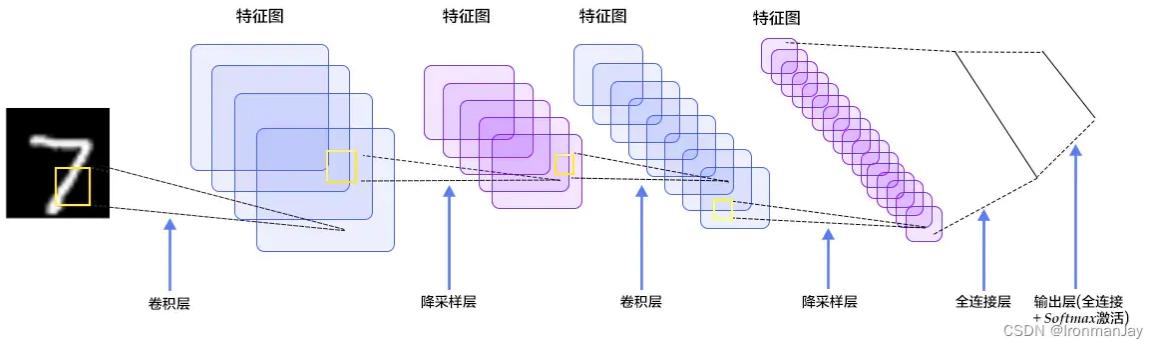


图 11: 卷积神经网络前向过程示意图。

卷积神经网络是一种带有卷积结构的深度神经网络，卷积结构可以减少深层网络占用的内存量，其三个关键的操作，其一是局部感受野，其二是权值共享，其三是 pooling 层，有效的减少了网络的参数个数，缓解了模型的过拟合问题。图 11 形象地演示了卷积神经网络识别手写数字的推理过程。

卷积神经网络的整体架构可以总结如下：

1. 卷积神经网络是一种多层的监督学习神经网络，隐含层的卷积层和池采样层是实现卷积神经网络特征提取功能的核心模块。该网络模型通过采用梯度下降法最小化损失函数对网络中的权重参数逐层反向调节，通过频繁的迭代训练提高网络的精度。
2. 卷积神经网络的低隐层是由卷积层和最大池采样层交替组成，高层是全连接层对应传统多层次感知器的隐含层和逻辑回归分类器。
3. 第一个全连接层的输入是由卷积层和子采样层进行特征提取得到的特征图像。
4. 最后一层输出层是一个分类器，可以采用逻辑回归，Softmax 回归甚至是支持向量机对输入图像进行分类。

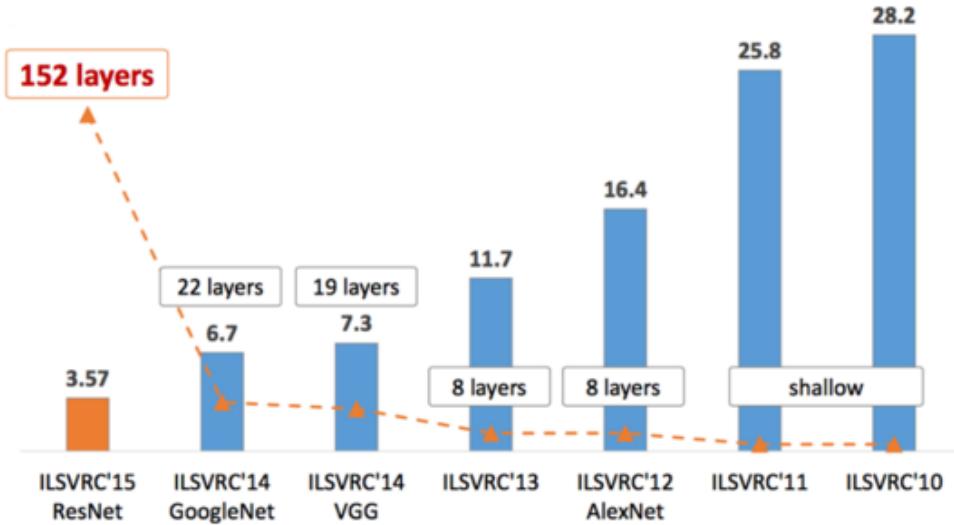


图 12: ImageNet 历届的冠军模型。

卷积神经网络再次兴起后，在几年时间内不断地打破 ImageNet 比赛的最高记录，如图 12 所示。具体而言，输入图像统计和滤波器进行卷积之后，提取该局部特征，一旦该局部特征被提取出来之后，它与其他特征的位置关系也随之确定下来了，每个神经元的输入和前一层的局部感受野相连，每个特征提取层都紧跟一个用来求局部平均与二次提取的计算层，也叫特征映射层，网络的每个计算层由多个特征映射平面组成，平面上所有的神经元的权重相等。通常将输入层到隐藏层的映射称为一个特征映射，也就是通过卷积层得到特征提取层，经过 pooling 之后得到特征映射层。

卷积神经网络的核心思想就是局部感受野、是权值共享和 pooling 层，以此来达到简化网络参数并使得网络具有一定程度的位移、尺度、缩放、非线性形变稳定性。pooling 层对图像进行下采样，可以减少数据处理量同时保留有用信息，采样可以混淆特征的具体位置，因为某个特征找出来之后，它的位置已经不重要了，我们只需要这个特征和其他特征的相对位置，可以应对形变和扭曲带来的同类物体的变化。

然而，正因为卷积神经网络拥有极佳的局部感受野，更倾向于提取局部特征，因此会产生极大的归纳偏置。在小规模数据集上，CNN 的归纳偏置能帮助模型更快地收敛至一个相对不错的结果。但当面对更大规模的数据集、更复杂的任务时，这种归纳偏置会使得模型难以达到更高的精度。

(3) Transformer

Transformer 是一种“编码器—解码器”架构，由编码器 (encoder) 和解码器 (decoder) 组成，其都是多头自注意力模块的叠加。其中，input sequence 分成两部分，分别为源 (input) 输入序列和目标 (output) 输出序列。前者输入编码器，后者输入解码器，两个序列均需进行 embedding 表示并加入位置信息。

(4) Vision Transformer

ViT [3] 是 Google 团队提出的将 Transformer [4] 应用在图像分类的模型，因为其模型“简单”且效果好，可扩展性强，于是成为了 Transformer 在 CV 领域应用的里程碑著作，也引爆了后续相关研究。通过在多个基准数据集上的实验对比，表明了 ViT 可以获得与当时最优卷积神经网络相美的结果。

与传统的卷积神经网络不同，ViT 使用自注意力机制代替卷积操作来提取图像特征。由于 Transformer 需要的是类似于单词序列的一维输入信号，而图像本身为二维信号，因此在 ViT 中，图像被划分成一组固定大小的图像块 (patch)，每个图像块都被看作是一个序列中的一个元素，然后这些序列元素会被输入到 Transformer 模型中，进行特征提取和分类，如图 13 所示。

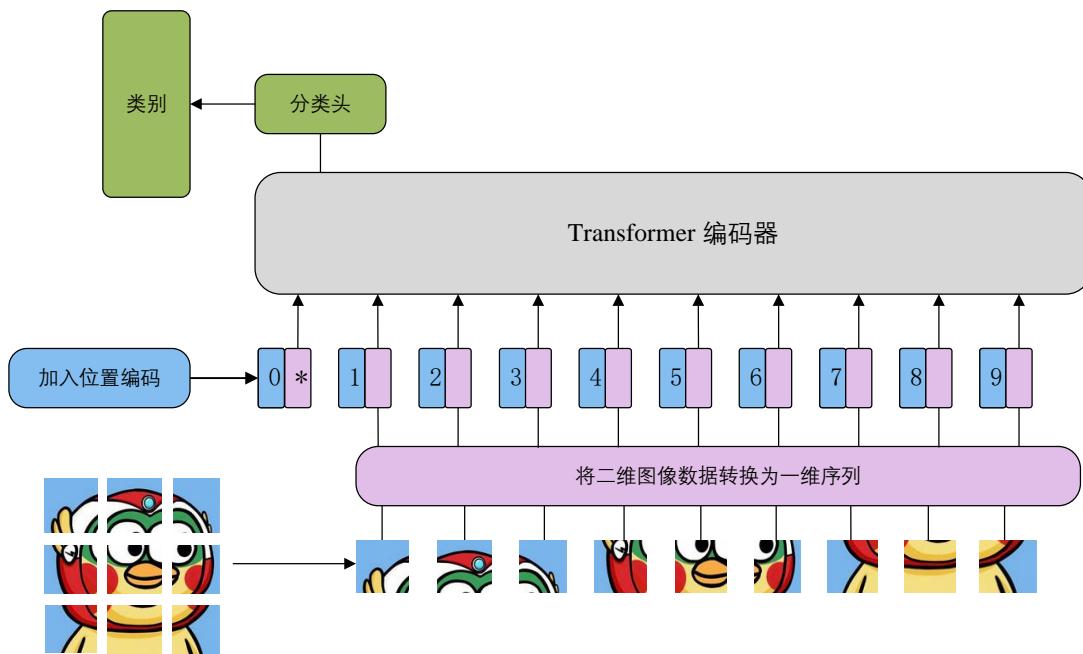


图 13: ViT (Vision Transformer) 模型结构示意图。

后续 encoder 的操作和原始 Transformer 中完全相同。但是因为对图片分类，因此在输入序列中加入一个特殊的 token，该 token 对应的输出即为最后的类别预测。这种序列化的方式可以使得 VT 模型更好地处理全局视觉信息，同时也可以避免卷积神经网络中存在的参数共享限制。ViT 的训练过程可以使用预训练和微调两个阶段。在预训练阶段，VT 会使用大量未标记的图像数据来学习视觉特征，而在微调阶段，则会使用少量标记的图像数据来对模型进行微调和分类任务的训练。ViT 已经在例如图像分类、目标检测、语义分割等多个视觉任务中取得了很好的表现。作者根据模

型大小（包括“Base”、“Large”和“Huge”）和图像块大小提供了一系列变体，例如 ViT-B/16 表示输入的图像块分辨率是 16×16 ，型号是“Base”其中，Transformer 的序列长度与图像块大小的平方成反比，因此图像块大小越小的模型计算成本越高。Transformer 编码器内部结构如图14所示。

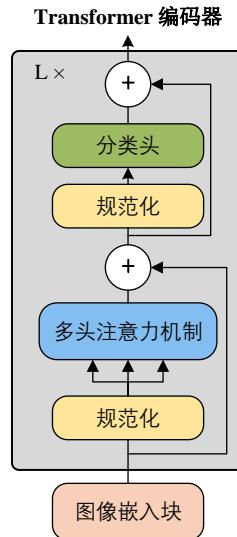


图 14: Transformer Encoder 结构示意图

具体原理如下 计算查询 (Query)、键 (Key)、值 (Value): 输入序列首先通过三个不同的线性变换生成查询 (Q)、键 (K)、值 (V)。

$$Q = \mathbf{X} \mathbf{W}^Q, \quad K = \mathbf{X} \mathbf{W}^K, \quad V = \mathbf{X} \mathbf{W}^V \quad (9)$$

计算注意力分数：

对于序列中的每个元素（以查询 i 为例），计算它对序列中所有元素（以键 j 为例）的注意力分数。注意力分数的计算公式为：

$$\text{Score}(i, j) = \frac{Q_i K_j^T}{\sqrt{d_k}} \quad (10)$$

计算注意力权重：

使用 softmax 函数将注意力分数转换为权重：

$$\alpha_{ij} = \frac{\exp(\text{Score}(i, j))}{\sum_k \exp(\text{Score}(i, k))} \quad (11)$$

计算加权的值：

$$O_i = \sum_j \alpha_{ij} V_j \quad (12)$$

输出自注意力层的结果：

$$O = \text{Concat}(O_1, O_2, \dots, O_N) \quad (13)$$

多头自注意力 (Multi-Head Self-Attention):

Transformer 模型进一步扩展了自注意力机制，通过“多头”自注意力来并行处理信息。在多

头自注意力中，输入序列被分割成多个“头”，每个头独立地计算自注意力，最后将所有头的输出拼接起来，再通过一个线性层进行处理。公式如下：

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O \quad (14)$$

其中每个头的计算为：

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (15)$$

(5) 词向量模型

词向量模型是将自然语言中的单词转换成固定长度的向量，以便计算机能够更好地理解自然语言的含义。目前，大多数跨模态匹配领域的文本描述都采用单词到向量（Word2vec）和单词表示的全局向量（Glove）两个词向量模型生成单词嵌入。

单词到向量（Word2Vec）是一种通过对输入语料中的单词进行学习，学习其相互关系，从而生成向量表示的模型。其基本思想是通过学习单词在上下文中的分布来捕捉其语义信息。有两种策略可以实现 Word2Vec 模型：一个是连续词袋模型（Continuous Bag of Words, CBOW），CBOW 模型是从周围的单词预测中心单词的概率。

另一个实现单词到向量的策略是采用跳字模型（Skip-gram），这个模型是从中心单词预测周围单词的概率，可以通过学习大规模文本数据的词汇共现关系，将每个单词表示成一个固定长度的向量。单词表示的全局向量（Global Vectors for Word Representation, Glove）旨在通过学习单词的全局共现信息来捕捉词语之间的语义关系。

与 Word2Vec 中的模型不同，GloVe 是在全局语料库中对所有单词对的共现信息进行建模，而不是基于局部的上下文信息来学习词向量。其优点在于能够同时利用全局的词频信息和局部的上下文信息，因此得到的词向量具有较好的语义表示能力。

(6) BERT

BERT 来自 Google 的论文 Pre-training of Deep Bidirectional Transformers for Language Understanding，BERT 是“Bidirectional Encoder Representations from Transformers”的首字母缩写，整体是一个自编码语言模型（Autoencoder LM），并且其设计了两个任务来预训练该模型。

第一个任务是采用 MaskLM 的方式来训练语言模型，通俗地说就是在输入一句话的时候，随机地选一些要预测的词，然后用一个特殊的符号 [MASK] 来代替它们，之后让模型根据所给的标签去学习这些地方该填的词。

第二个任务在双向语言模型的基础上额外增加了一个句子级别的连续性预测任务，即预测输入 BERT 的两段文本是否为连续的文本，引入这个任务可以更好地让模型学到连续的文本片段之间的关系。

BERT 模型通过对 Masked LM 任务和 Next Sentence Prediction 任务进行联合训练，使模型输出的每个字 / 词的向量表示都能尽可能全面、准确地刻画输入文本（单句或语句对）的整体信息，为后续的微调任务提供更好的模型参数初始值。最后的实验表明 BERT 模型的有效性，并在 11 项

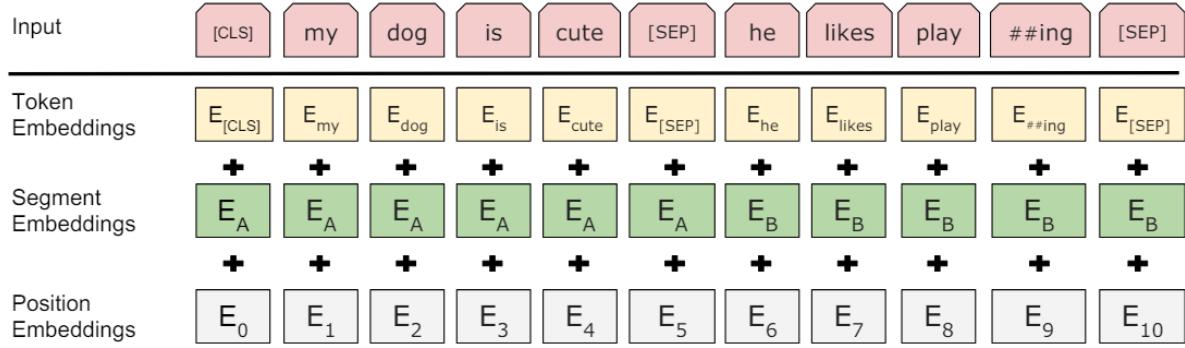


图 15: BERT (Bidirectional Encoder Representations from Transformers) 模型输入示意图。

NLP 任务中夺得 SOTA 结果。

BERT 相较于原来的 RNN、LSTM 可以做到并发执行，同时提取词在句子中的关系特征，并且能在多个不同层次提取关系特征，进而更全面反映句子语义。相较于 word2vec，其又能根据句子上下文获取词义，从而避免歧义出现。同时缺点也是显而易见的，模型参数太多，而且模型太大，少量数据训练时，容易过拟合。

从准确率上来看，BERT 利用更深的模型，以及海量的语料，得到的 embedding 表示，来做下游任务时的准确率显著高于 word2vec。从方法的意义角度来说，BERT 的重要意义在于给大量的 NLP 任务提供了一个泛化能力很强的预训练模型，而仅仅使用 word2vec 产生的词向量表示，不仅能够完成的任务比 BERT 少了很多，而且很多时候直接利用 word2vec 产生的词向量表示给下游任务提供信息，下游任务的表现不一定会很好，甚至会比较差。

3. 模型结构

CLIP 模型的核心思想是通过学习图像和文本之间的匹配关系来提高模型的性能。具体来说，CLIP 模型包含两个主要组成部分：一个用于处理图像的 CNN 模型或 ViT 模型，和一个用于处理文本的 BERT 模型。这两个组件都被训练成能够将输入的信息映射到相同的嵌入空间中，并使得相似的图像和文本在嵌入空间中的距离更近。图 16 演示了 CLIP 模型的结构。

在实现上，为了兼顾模型精度与效率，我们实现的 CLIP 模型采用 **ViT-L/14** 模型作为图像编码器，以及 **RoBERTa** 模型作为文本编码器。同时，这些编码器均经过充分预训练，显著加快了后续训练的收敛速度。

下面，我们分别介绍使用的预训练图像编码器与文本编码器。

(1) 图像编码器

ViT [3] 是 Google 团队提出的将 Transformer [4] 应用在图像分类的模型，因为其模型“简单”且效果好，可扩展性强，于是成为了 Transformer 在 CV 领域应用的里程碑著作，也引爆了后续相关研究。

ViT 最核心的结论是，当拥有足够多的数据进行预训练的时候，ViT 的表现就会超过 CNN，突破 Transformer 缺少归纳偏置的限制，可以在下游任务中获得较好的迁移效果。基于该结论，我们

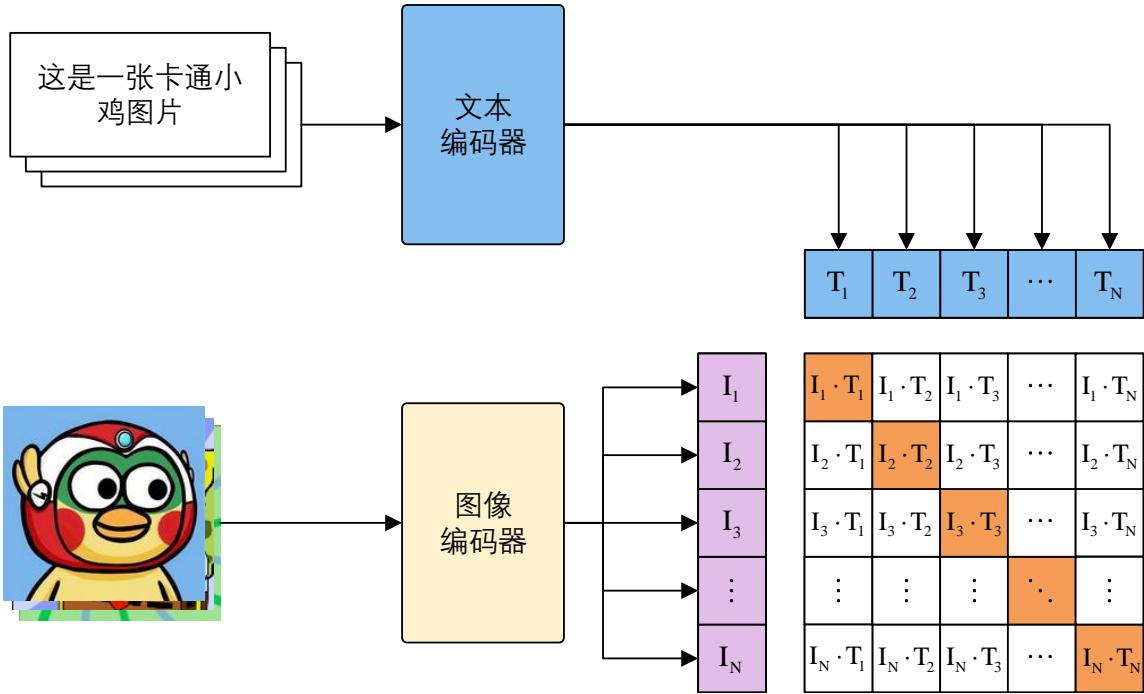


图 16: CLIP (Contrastive Language-Image Pre-training) 模型结构示意图。

将充分预训练的 ViT 模型作为 CLIP 模型的图像编码器，并选择参数量适中的 ViT-L/14 变种，以平衡精度与计算量的冲突。

具体而言，我们基于 HuggingFace 平台，获取了在 ImageNet 数据集中充分预训练的 ViT-L/14 网络的权重，用来初始化 CLIP 模型的图像编码器，以获得更佳的迁移学习性能。

(2) 文本编码器

RoBERTa [5] 是在论文 *RoBERTa: A Robustly Optimized BERT Pretraining Approach* 中被提出的。此方法属于 BERT 的强化版本，也是 BERT 模型更为精细的调优版本。RoBERTa 主要在三方面对之前提出的 BERT 做了该进，其一是模型的具体细节层面，改进了优化函数；其二是训练策略层面，改用了动态掩码的方式训练模型，证明了 NSP (Next Sentence Prediction) 训练策略的不足，采用了更大的 Batch Size；其三是数据层面，一方面使用了更大的数据集，另一方面是使用字节级别的 BPE (Bytes-level BEP) 来处理文本数据。

RoBERTa 在训练方面与原始 BERT 模型保持一致，使用类似“完形填空”的代理任务，让模型学习填补缺失词。输入句子中部分词被随机遮掩，替换为 “[PAD]”，并在最前方添加一个特殊词汇 “[CLS]”。对于缺失值填补任务，要求 “[PAD]” 的输出能还原原始词汇；对于分类任务，使用 “[CLS]” 的输出作为分类器的特征。

与图像编码器类似，我们使用基于中文数据预训练的 RoBERTa 模型 [6] 作为 CLIP 的文本编码器。该策略极大地提高了 CLIP 模型预训练与微调的效率，使得我们能在有限的训练回合中，获得更大的精度收益。

4. 训练目标

CLIP 使用图像文本对作为训练标签。这里举例一个包含 N 个图像文本对的训练 Batch，对提取的文本特征和图像特征进行训练的过程：

1. 输入图像 → 图像编码器 → 图像特征向量；输入文字 → 文字编码器 → 文字特征向量；并进行线性投射，得到相同维度；
2. 将 N 个图像特征和 N 个文本特征两两组合，形成一个形状为 $N \times N$ 的矩阵 s ；CLIP 模型会预测计算出这 N^2 个图像文本对的相似度（即余弦相似度）；
3. 对角线上的 N 个元素因为图像-标签对应正确被作为训练的正样本，剩下的 $N(N - 1)$ 个元素作为负样本；
4. CLIP 的训练目标是基于对比损失，最大化 N 个正样本的相似度，同时最小化 $N(N - 1)$ 个负样本的相似度。

具体而言，对于任意一个图像文本对，CLIP 的对比损失如公式 (16) 所示。最后考虑所有可能存在的图像文本对，需要最小化的目标函数如公式 (17) 所示。其中， $s_{i,j}$ 表示相似度矩阵第 i 行的第 j 列的元素，其数值含义为第 i 幅图像与第 j 段文本的相似度，由公式 (7) 计算。

$$l(i, j) = -\log \frac{\exp(s_{i,j})}{\sum_{k=1}^N \mathbf{1}_{[k \neq i]} \exp(s_{i,k})} \quad (16)$$

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [l(k-1, k) + l(k, k-1)] \quad (17)$$

5. 模型推理

基于上述损失函数，我们在大规模数据集上对 CLIP 模型训练后，得到的权重可以用于推理。这里，我们以问题一的 I2T 任务为例，演示 CLIP 进行文本检索推理的过程。

(1) 数据预处理

设我们进行 I2T 任务的目标图像为 $Image_1$ ，潜在文本集合为 $\{Text_i \mid 1 \leq i \leq N\}$ 。模型首先对目标图像进行预处理，通过裁剪、缩放和插值操作，得到结构化的图像 $X \in \mathbb{R}^{1 \times H \times W \times C}$ 。对于原始类别标签，我们将其拼接至模板“这是一张 {XX} 图像”中，使其更符合中文语法习惯，如图 17 所示。接下来，对拼接后的文本进行处理，将其分词后映射至词汇表，通过填充与截断操作控制序列长度，并转换为训练得到的词向量矩阵，得到结构化的文本 $Y \in \mathbb{R}^{N \times L \times D_{emb}}$ 。

(2) 图像文本编码

对于结构化的图像与文本数据，使用 CLIP 的图像编码器与文本编码器，将其映射至同维度的编码向量。为了计算方便，我们对所有编码向量进行归一化，得到归一化图像编码 $I \in \mathbb{R}^{1 \times D_{model}}$ 以及归一化文本编码 $T \in \mathbb{R}^{N \times D_{model}}$ 。对于任意向像 $Image_i$ 和文本 $Text_j$ ，其归一化编码的计算过程分别如公式 (18) 和公式 (19) 所示。通过进行归一化，我们可以使用向量的点乘操作来替代此前

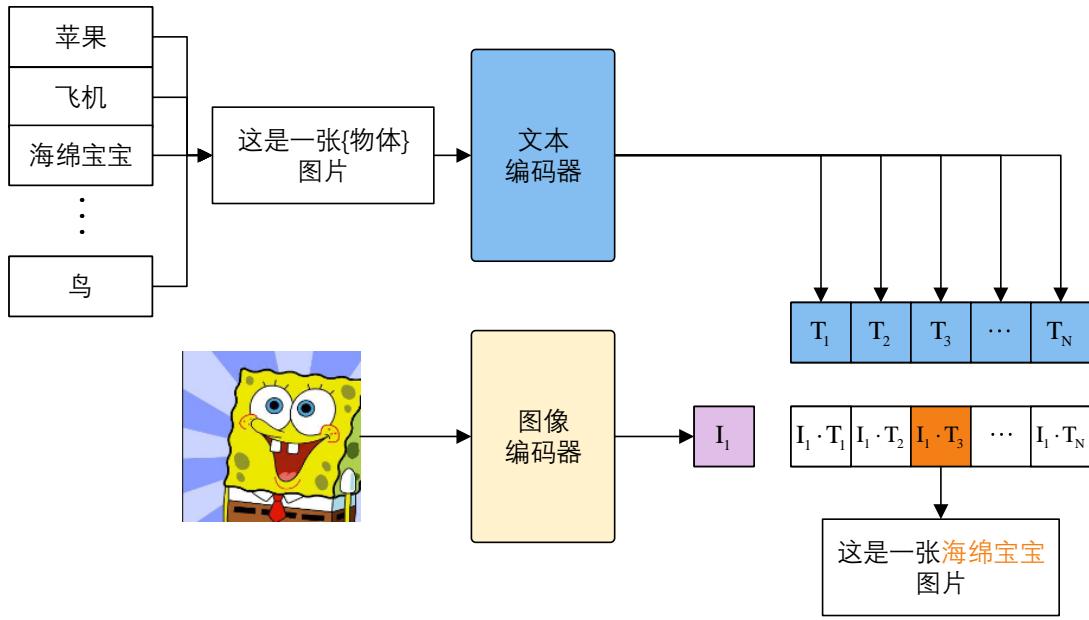


图 17: CLIP 模型推理——文本检索

的余弦相似度计算。

$$I_i = \frac{r_i}{\|r_i\|} \quad \text{其中, } r_i = \text{ImageEncoder}(Image_i) \quad (18)$$

$$T_j = \frac{h_j}{\|h_j\|} \quad \text{其中, } h_j = \text{TextEncoder}(Text_j) \quad (19)$$

(3) 相似度计算

接下来，对目标图像编码 I_1 与所有文本编码 T 进行点乘，得到目标图像与各段文本的相似度数组 $Sims = \{I_1 \cdot T_i \mid 1 \leq i \leq N\}$ 。最后，取相似度数组 $Sims$ 的最大值，与之对应的文本即为最佳匹配文本。类似的，若取最大的 K 个值，则可得到最佳的 K 段匹配文本。图 17 演示了 CLIP 模型进行文本检索推理的完整流程。

6. 改进策略

(1) 梯度累积 (Gradient Accumulation)

在深度学习训练的时候，数据的 batch size 大小受到 GPU 内存限制，batch size 大小会影响模型最终的准确性和训练过程的性能。在 GPU 内存不变的情况下，模型越来越大，那么这就意味着数据的 batch size 只能缩小。然而，有研究表示主流的对比学习模型通常 batch size 越大，训练过程越平稳，最终的性能表现也越好。

通常，当一块 GPU 的内存不足时，可以采用多 GPU 进行分布式训练。将一个大的 Batch 均摊到多块 GPU 上，计算完梯度后再汇总，进行梯度下降，如图 18 所示。然而，分布式训练会带来额外的通讯与同步开销，且成本高昂。

这个时候，梯度累积 (Gradient Accumulation) 可以作为一种简单的解决方案来解决这个问题。

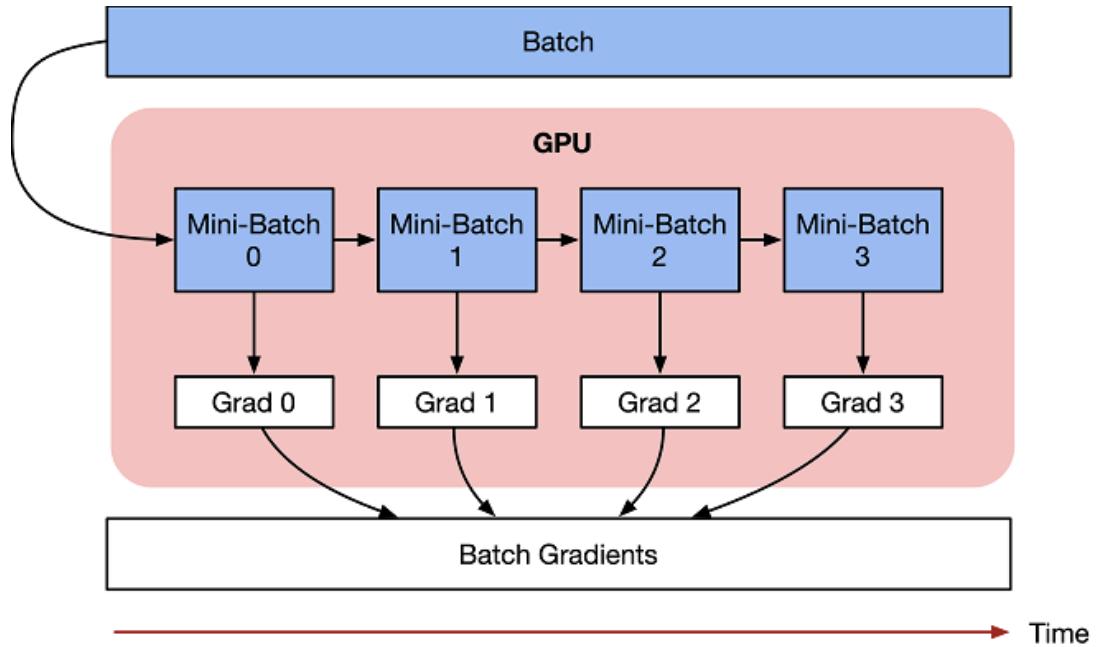


图 18: 分布式训练示意图。

梯度累积是一种不需要额外硬件资源就可以增加批量样本数量（Batch Size）的训练技巧。这是一个通过时间换空间的优化措施，它将多个 Batch 训练数据的梯度进行累积，在达到指定累积次数后，使用累积梯度统一更新一次模型参数，以达到一个较大 Batch Size 的模型训练效果。累积梯度等于多个 Batch 训练数据的梯度的平均值。

所谓梯度累积过程，其实很简单，我们梯度下降所用的梯度，实际上是多个样本算出来的梯度的平均值，以 `batch_size=1024` 为例，你可以一次性算出 1024 个样本的梯度然后平均，我也可以每次算 64 个样本的平均梯度，然后缓存累加起来，算够了 16 次之后，然后把总梯度除以 16，然后才执行参数更新。当然，必须累积到了 16 次之后，用 16 次的平均梯度才去更新参数，不能每算 64 个就去更新一次，不然就是 `batch_size=64` 了。

上述整个过程可以由以下步骤描述：

1. 正向传播，出入数据，得到预测结果；
2. 根据预测结果与 label，计算损失值；
3. 利用损失进行反向传播，计算参数梯度；
4. 重复上面步骤，不清空梯度，将梯度累加；
5. 梯度累加达到固定次数之后，更新参数，然后将梯度清零。

总结来讲，梯度累积就是每计算一个 batch 的梯度，不进行清零，而是做梯度的累加，当累加到一定的次数（`accumulation_steps`）之后，再更新网络参数，然后将梯度清零。通过这种参数延迟更新的手段，可以实现与采用大 batch size 相近的效果。在平时的实验过程中，我一般会采用梯度累加技术，大多数情况下，采用梯度累加训练的模型效果，要比采用小 batch size 训练的模型效果要好很多。

(2) 掩码特征学习

我们借鉴了 FLIP 模型的图像掩码策略，对 CLIP 模型进行改进。我们首先将图像划分为不重叠的 patch，随机屏蔽掉大部分（例如 50% 或 75%）的 patch，图像编码器仅应用于可见 patch，如图 19 所示。

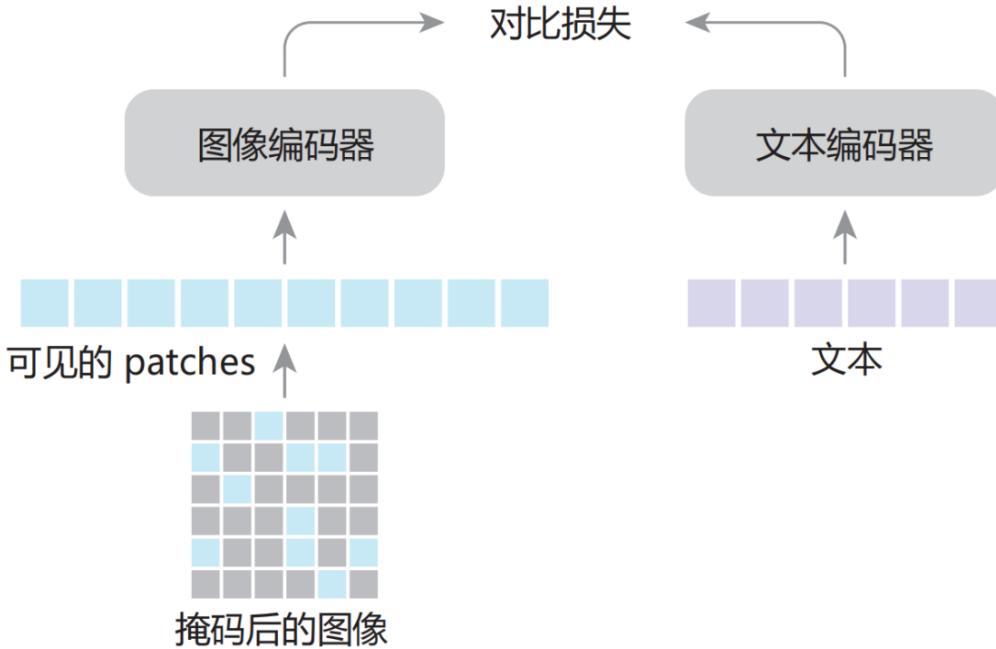


图 19: 掩码特征学习示意图。

与原始的 CLIP 模型相比，改进后的模型在同样的训练时间内可以学习更多的图像-文本对，并且 ImageEncoder 的显存使用也下降（mask 掉 50%，显存消耗就下降 50%），这样在一定的硬件资源下就可以实现更大的 batch size，而对比学习往往需要较大的 batch size。同时，由于图像往往具有较强的冗余性，即使 mask 大部分图像，特征网络的学习性能并不会变差，反而由于更大的 batch size 而可以获得更好的性能。

综上，FLIP 使用类似 MAE 模型的掩码输入策略，减少了编码序列的长度，极大地节省了显存的消耗量，使得我们可以使用更大的 batch size 进行训练，以获得更佳的泛化性能。

(3) FlashAttention

在标准注意力实现中，注意力的性能主要受限于内存带宽，是内存受限的。频繁地从 HBM 中读写 $\mathbb{R}^{N \times N}$ 的矩阵是影响性能的主要瓶颈。稀疏近似和低秩近似等近似注意力方法虽然减少了计算量 FLOPs，但对于内存受限的操作，运行时间的瓶颈是从 HBM 中读写数据的耗时，减少计算量并不能有效地减少运行时间（wall-clock time）。针对内存受限的标准注意力，Flash Attention 是 IO 感知的，目标是避免频繁地从 HBM 中读写数据。图 20 以一个金字塔的形式，形象地说明了 GPU 中各存储器的 IO 效率差异。

从 GPU 显存效率分级来看，SRAM 的读写速度比 HBM 高一个数量级，但内存大小要小很多。通过 kernel 融合的方式，将多个操作融合为一个操作，利用高速的 SRAM 进行计算，可以减少读

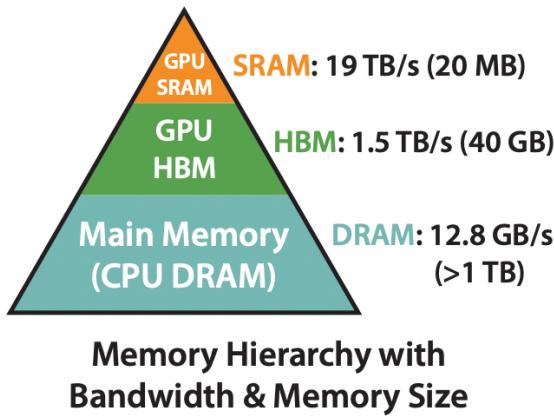


图 20: GPU 中的 IO 速度金字塔。

写 HBM 的次数，从而有效减少内存受限操作的运行时间。但 SRAM 的内存大小有限，不可能一次性计算完整的注意力，因此必须进行分块计算，使得分块计算需要的内存不超过 SRAM 的大小。

Flash Attention 的作者将多头注意力的计算抽象为以下操作：

$$S = \tau Q K^\top \in \mathbb{R}^{N \times N} \quad (20)$$

$$S^{\text{masked}} = \text{MASK}(S) \in \mathbb{R}^{N \times N} \quad (21)$$

$$P = \text{softmax}(S^{\text{masked}}) \in \mathbb{R}^{N \times N} \quad (22)$$

$$P^{\text{dropped}} = \text{dropout}(P, p_{\text{drop}}) \in \mathbb{R}^{N \times N} \quad (23)$$

$$O = P^{\text{dropped}} V \in \mathbb{R}^{N \times d} \quad (24)$$

为了提高运算效率，在 Flash Attention 算法中，并没有将 S 、 P 写入 HBM 中去，而是通过分块写入到 HBM 中去，存储前向传递的 softmax 归一化因子，在后向传播中快速重新计算片上注意力，这比从 HBM 中读取中间注意力矩阵的标准方法更快。即使由于重新计算导致 FLOPS 增加，但其运行速度更快并且使用更少的内存（序列长度线性），主要是因为大大减少了 HBM 访问量。

简而言之，Flash Attention 实现了不使用中间注意力矩阵，通过存储归一化因子来减少 HBM 内存的消耗，同时减少了 HBM 访问量，使得 Attention 计算过程中在 IO 方面的开销大大减少，从而提高了前向运算的速度，并降低了显存的开销。

五、实验与结果分析

1. 模型预训练

(1) 数据集构建

在预训练阶段，我们首先根据 MUGE Retrieval、Flickr30K-CN、COCO-CN 这三个中文检索方向常用数据集构建了约 800 万样本的数据集。为了确保数据集的质量，我们对数据集进行了彻底的清洗，移除了重复项、低分辨率图像以及包含错误或不完整文本的样本。此外，我们还执行了

数据增强，图像包括随机裁剪、翻转、旋转，文本包括同义词替换、翻译互转等操作，以增加数据的多样性并提高模型对不同图像变换的鲁棒性。在数据清洗过程中，需要保持数据的多样性和代表性，确保模型能够学习到丰富的视觉和语言特征，同时使数据分布尽量平衡，防止模型模型出现偏向性。数据增强策略的实施，旨在模拟真实世界条件下图像可能出现的各种情况，以增强模型的泛化性。通过这种方式，能够提升模型对于图像几何变换的适应能力。

(2) 模型预训练

预训练环境 我们使用 Ubuntu20.04 操作系统，基于 PyTorch1.12 深度学习框架实现所提出的改进 CLIP 模型，并在 4 块 NVIDIA RTX4090D 24G 显卡上对该模型进行分布式训练。

预训练参数设置 预训练阶段，我们使用 AdamW 优化器，初始学习率设置为 5×10^{-5} ，权重衰减系数设置为 0.001，掩码特征学习的 Mask 比例为 50%。此外，每张显卡上的 batch_size 设置为 32，累积梯度系数设置为 16，因此训练时的 等价batch_size = $32 \times 16 \times 4 = 2048$ (来自公式：逻辑batch_size = 单卡batch_size × 累计梯度系数 × GPU 数量)。

预训练效率 共计预训练 10 个 Epoch，耗时 11.8 小时，平均每个 Epoch 耗时 71 分钟。此外，每张 GPU 的最大显存使用量为 20.5GB/24GB。

(3) “零样本”测试

在仅基于公共数据集预训练，不使用任何比赛数据集相关信息的情况下，对模型进行性能评估，称为“零样本”测试。具体而言，我们按照章节 5. 中描述的方法，使用基于公共数据集预训练的模型，在比赛数据集的 **T2I 测试集** 和 **I2T 测试集** 上进行测试。

为了进行更全面的性能评估，我们除了计算题目所要求的 R@5 指标，还分别计算了 R@1、R@10 指标，以及各个指标的均值 MR。

图像检索 按照章节 5. 中描述的方法，我们使用基于公共数据集预训练的模型，在比赛数据集的 T2I 验证集上进行测试。最终，得到的评估指标如表 6 所示。同时，图 28 直观地展示了各个 epoch 之间的精度差异。

表 6: “零样本”测试结果——图像检索

回合 \ 评估指标	R@1	R@5	R@10	MR
2 epochs	0.2814	0.3800	0.4562	0.3725
4 epochs	0.4098	0.5576	0.6230	0.5301
6 epochs	0.4810	0.6538	0.7184	0.6177
8 epochs	0.5072	0.6924	0.7628	0.6541
10 epochs	0.5156	0.7094	0.7712	0.6654

文本检索 与图像检索类似，我们按照章节 5. 中描述的方法，使用基于公共数据集预训练的模型，在比赛数据集的 T2I 验证集上进行测试。最终，得到的评估指标如表 7 所示。同时，图 29 直观地展示了各个 epoch 之间的精度差异。

表 7：“零样本”测试结果——文本检索

回合 \ 评估指标	R@1	R@5	R@10	MR
2 epochs	0.2566	0.3662	0.4426	0.3551
4 epochs	0.3912	0.5396	0.6078	0.5129
6 epochs	0.4642	0.6386	0.7024	0.6017
8 epochs	0.4934	0.6924	0.7538	0.6465
10 epochs	0.5012	0.6972	0.7604	0.6529

我们根据表 6 和表 7 的数据绘制了模型在预训练阶段的精度迭代曲线，如图 21 所示。不难发现，在预训练初期，模型在两项任务中的表现一般；但随着预训练回合的增加，模型的精度先快速增长，再趋于平衡。最终，模型在两个任务上的平均召回率（MR）收敛于一个较高的值。

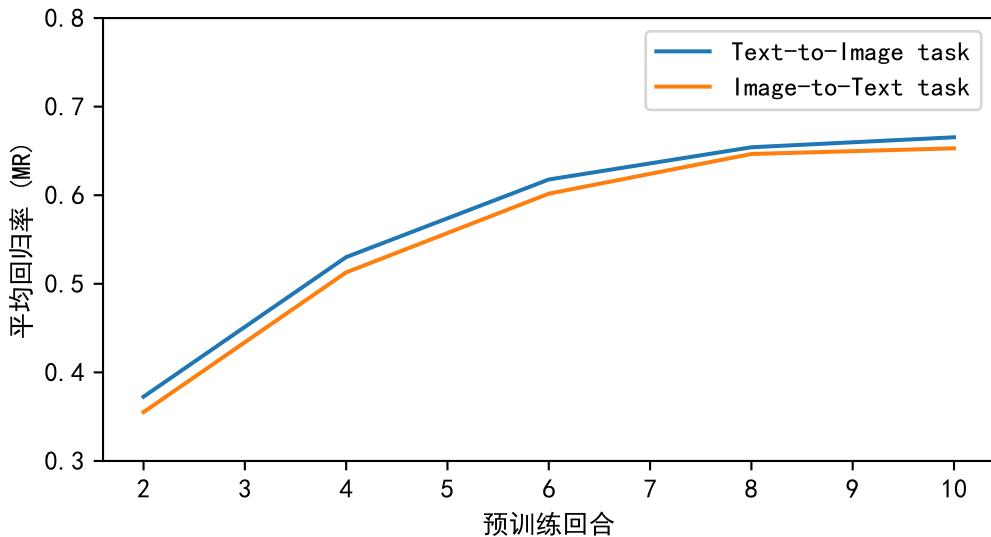


图 21：预训练精度迭代曲线

2. 模型微调

(1) 数据集构建

我们使用比赛附件 1 中所给的数据构建微调模型所使用的数据集，对数据集进行增强，如对图像进行随机裁剪、翻转、旋转，对文本进行同义词替换、翻译互转，人为地增加了数据的多样性，有助于模型学习到更加鲁棒的特征，从而在面对未见过的样本时能够做出准确的预测。在数据集准备好之后，我们将其划分为训练集、验证集和测试集，以便在训练过程中监控模型的性能，并最终评估其泛化能力。同时我们采用交叉验证的方法，以确保模型在不同的数据子集上都能保持一致的性

能。

(2) 模型微调

微调环境 微调环境与预训练阶段保持一致，使用 Ubuntu20.04 操作系统，基于 PyTorch1.12 深度学习框架实现所提出的改进 CLIP 模型，并在 4 块 NVIDIA RTX4090D 24G 显卡上对该模型进行分布式训练。

微调参数设置 微调阶段，我们去除动量机制，将优化器切换为随机梯度下降（SGD），同时将初始学习率减小十倍，设置为 5×10^{-6} ，并不再使用掩码特征学习策略。除此之外的设置与预训练阶段保持一致。

微调效率 微调时采用十折交叉验证策略，共计微调 5 个 Epoch，共计耗时 6.2 小时，平均每个 Epoch 耗时 7.4 分钟。同时，每张 GPU 的最大显存使用量与预训练阶段一致。

(3) 微调结果测试

图像检索 首先，基于十折交叉验证策略，重复了十次微调实验，并将结果平均值汇总，得到微调模型在验证集上的性能评估指标，如表 8 所示。

表 8: 微调测试结果——图像检索

回合 \ 评估指标	R@1	R@5	R@10	MR
0 epoch	0.5156	0.7094	0.7712	0.6654
1 epoch	0.5776	0.7630	0.8194	0.7200
2 epochs	0.5802	0.7684	0.8230	0.7239
3 epochs	0.5856	0.7676	0.8238	0.7257
4 epochs	0.5840	0.7682	0.8240	0.7254
5 epochs	0.5856	0.7694	0.8240	0.7263

为了进一步分析各个微调回合的精度收益情况，我们根据表 8 的数据绘制了验证集精度随微调回合数的变化曲线，如图 22 所示。

由表 8 和图 22 可以看出，基于预训练结果微调时，第一个回合由于引入了与比赛验证集分布一致的比赛训练集，其在验证集上的性能表现迎来了巨大的提升。然而，当微调继续进行时，模型的精度提升已经非常细微，到第三个回合后，模型的精度已经开始震荡。

文本检索 对于问题二，我们同样基于十折交叉验证策略，重复了十次微调实验，并将结果平均值汇总，得到微调模型在验证集上的性能评估指标，如表 9 所示。

同样的，我们根据表 9 的数据绘制了验证集精度随微调回合数的变化曲线，如图 23 所示。

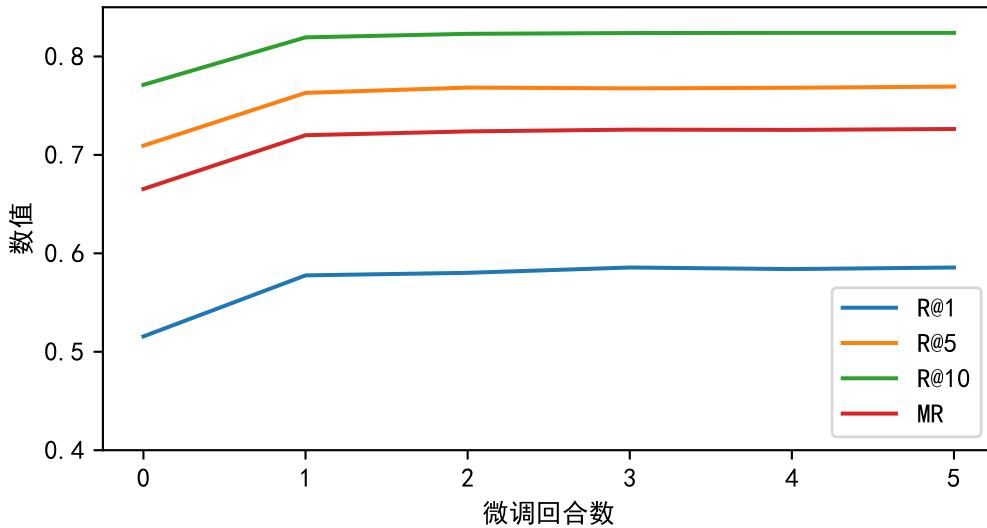


图 22: 问题一验证集精度随微调回合数的变化曲线

表 9: 微调测试结果——文本检索

回合 \ 评估指标	R@1	R@5	R@10	MR
0 epoch	0.5012	0.6972	0.7604	0.6529
1 epoch	0.5636	0.7584	0.8096	0.7105
2 epochs	0.5690	0.7568	0.8110	0.7123
3 epochs	0.5732	0.7590	0.8140	0.7154
4 epochs	0.5724	0.7626	0.8118	0.7156
5 epochs	0.5734	0.7624	0.8140	0.7166

由表 9 和图 23 可以看出，基于预训练结果微调时，第一个回合由于引入了与比赛验证集分布一致的比赛训练集，其在验证集上的性能表现迎来了巨大的提升。然而，当微调继续进行时，模型的精度提升已经非常细微，到第三个回合后，模型的精度已经开始震荡。

测试集推理 接下来，我们将微调得到的模型应用于问题一的测试集中，得到了最终的结果。

图 24 和图 25 分别展示了问题一结果的两个示例。在示例一中，我们以文本“开学第一课 军训 你成长了吗？”对所有图像进行检索，最终得到五幅最相似的图像。观察图 24 中的图像，发现它们都与军训这一主题高度相关，所以该结果比较符合人的知觉，效果较好。

在示例二中，我们以文本“华为用这一招让美国始料未及：手机和 5G 基站都拒绝美国”对所有图像进行检索，同样得到五幅符合要求的图像。其中，第一幅图像内容是 5G 基站；第二、三幅是描述美国对通讯企业制裁的插画；第四幅图绘制了一个巨大的“5G”标语以及通讯设备；最后一幅是印着中国国企的芯片。这些图像都与“5G”、“美国制裁”、“中国科技发展”相关。

与问题一类似，我们将微调得到的模型应用于问题二的测试集中，得到了最终的结果。图 26 和图 27 分别展示了问题二结果的两个示例。

在示例一中，我们以一幅布偶猫的照片对所有文本进行检索。画面中，一只猫趴在桌子一边，

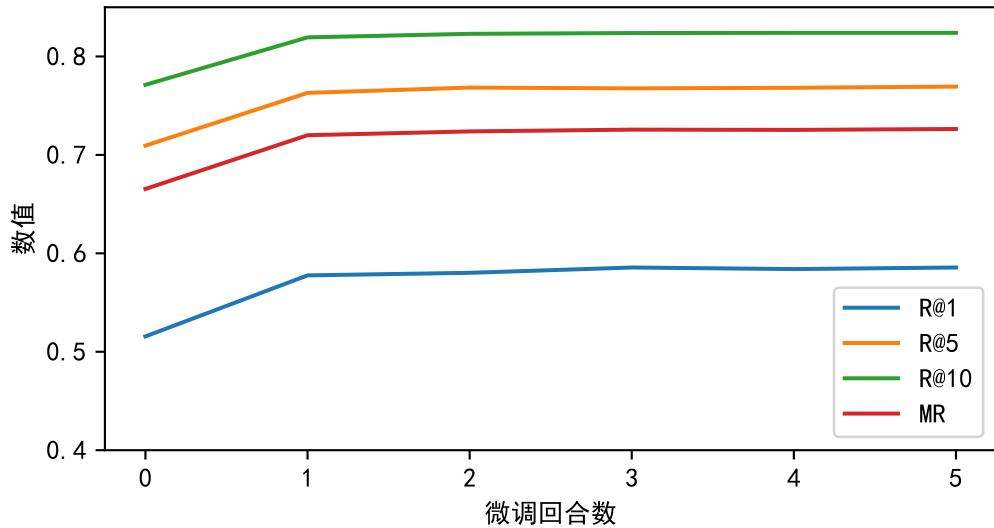


图 23: 问题二验证集精度随微调回合数的变化曲线

文本	(Word-1000000495) 开学第一课 军训 你成长了吗?				
相似度最高的前五张图像	Image14001010-8447.jpg 	Image14001009-9737.jpg 	Image14001013-1696.jpg 	Image14001010-8849.jpg 	Image14001008-1337.jpg

图 24: 问题一结果示例 1

文本	(Word-10000001302) 华为用这一招让美国始料未及:手机和5G基站都拒绝美国				
相似度最高的前五张图像	Image14001010-8072.jpg 	Image14001010-4371.jpg 	Image14001012-4569.jpg 	Image14001009-6880.jpg 	Image14001008-1408.jpg

图 25: 问题一结果示例 2

另一边则是写有“全价猫粮”的包装袋，这很可能是一封猫粮的广告。最终，我们检索得到五段最相关的文本，如图 26 所示。第一段文本显然符合图片的意思，是猫粮的广告词；后面四段文本虽然没有直接涉及猫粮，但都与猫、宠物等主题高度相关，较为符合题意。

而在示例二中，我们则以一幅仓鼠的照片对所有文本进行检索。画面中，一只仓鼠卡在一间宠物小屋的窗户里向外探头。图 27 展示了检索得到五段最相关的文本。前三段文本比较符合图片的意思，都是关于仓鼠和宠物的话题；第五段文本以第一人称描述困意，与图中仓鼠的神态也较为符合。

图像	相似度最高的前五条文本
Image14105001-0347.jpg 	<p>(Word-1000065271) 软便不消化猫咪玻璃胃试试三只小宠无谷肠胃养护猫粮</p> <p>(Word-1000079059) 因为有了猫老板, 你的七夕比别人多一丝甜蜜</p> <p>(Word-1000067738) 来源 咖门作者 政雨5年, 1006家店.</p> <p>(Word-1000089843) 招收铲屎官</p> <p>(Word-1000096655) 夏季养猫攻略必看防暑喵招</p>

图 26: 问题二结果示例 1

图像	相似度最高的前五条文本
Image14105001-3018.jpg 	<p>(Word-1000093238) 怎样让妈妈喜欢上仓鼠</p> <p>(Word-1000098641) (流量慎)随便发点我存过的可爱动物图片(不含猫狗)</p> <p>(Word-1000084821) 快救救我们家的老鼠!</p> <p>(Word-1000069215) 不让趴花盆, 蓝宝就喜欢趴在电风扇下面的小框子里打滚</p> <p>(Word-1000064447) 啊~不行啦, 好困好困</p>

图 27: 问题二结果示例 2

3. 结果分析

(1) 预训练收益

为了直观分析由预训练带来的验证集精度提升，我们分别针对图像检索任务和文本检索任务，绘制了各个预训练回合下的评估指标情况，如图 28 和图 29 所示。

观察图 28 和图 29，可以发现，在最初几个预训练回合中，各项评估指都有明显的增长；而随着预训练回合数的增加，精度变化趋于平缓，最终收敛到一个较高的值。这表示，从预训练中获取的精度提升是有限度的，为了继续提高在比赛数据集的表现，我们必须在比赛训练集上进行微调。

(2) 微调有效性

为了验证“预训练—微调”策略的有效性，我们对比了三种训练模式得到的模型——仅预训练、直接训练，以及预训练 + 微调。这三种模式下，图像检索任务和文本检索任务在比赛验证集上的性能表现分别如图 30 和图 30 所示。

根据图 30 中的结果，我们可以清晰地得到结论，对于图文检索任务，预训练 + 微调训练的效果优于直接训练或者仅预训练，且性能差距非常显著。

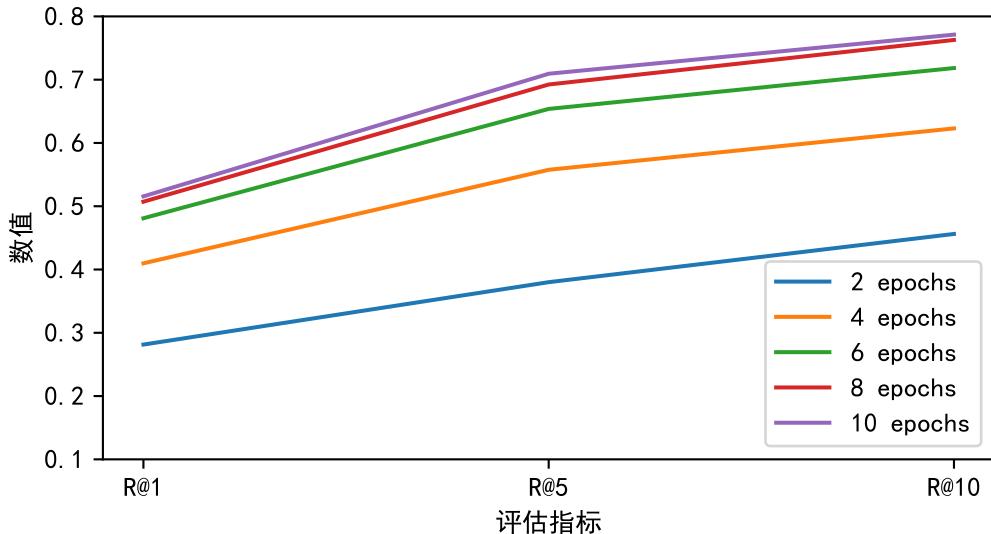


图 28: “零样本”测试结果收敛情况——图像检索

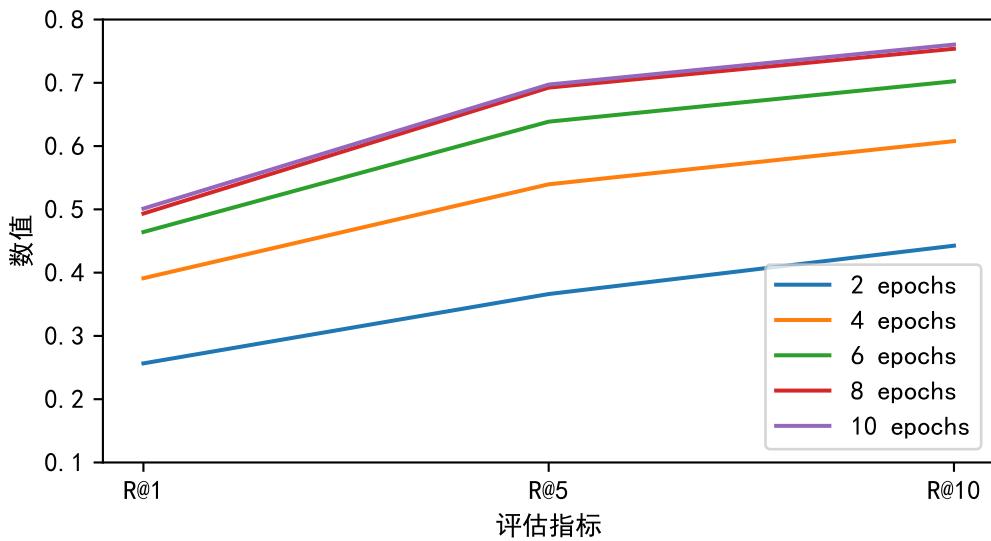


图 29: “零样本”测试结果收敛情况——文本检索

4. 模型总结

本文提出了一种基于大规模预训练 CLIP 模型微调的图文检索方法。下面是该模型的优点和缺点的总结：

模型优点：

1. **统一建模：**该模型通过将图像检索和文本检索问题转化为图像与文本的相似度衡量问题，实现了统一的建模框架。这种统一建模使得模型能够处理复杂的多模态数据，提高了检索的效果和准确性。
2. **多模态特征融合：**该模型利用 CLIP 模型的对比学习方法，将图像和文本映射到同一嵌入空间，实现了图像与文本的多模态特征融合。这种融合使得模型能够计算不同模态数据之间的余

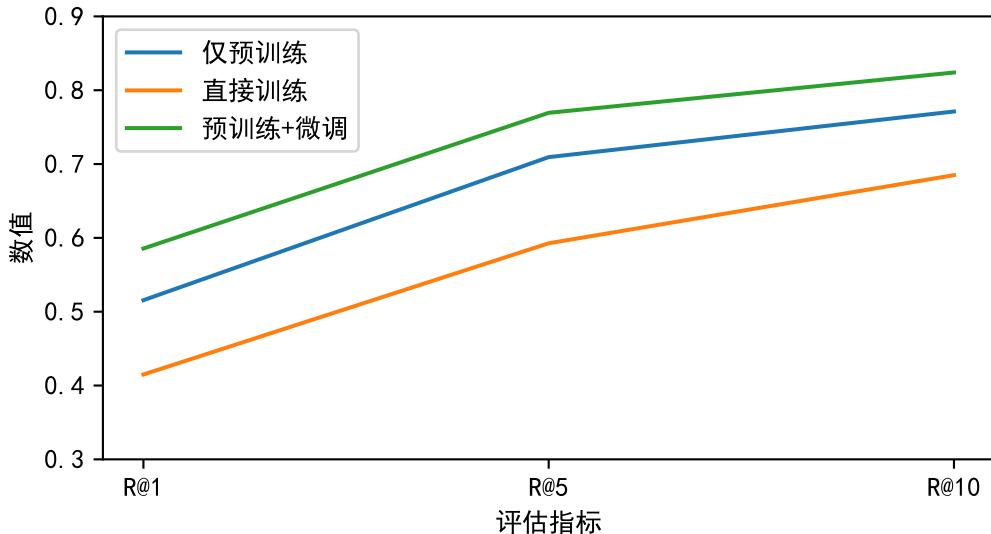


图 30: 三种训练模式下的图像检索任务性能比较

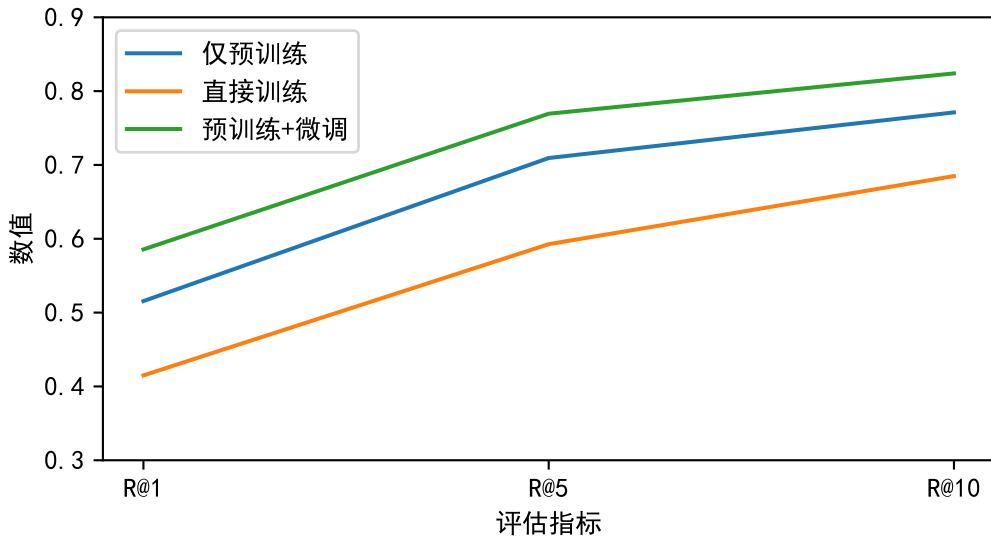


图 31: 三种训练模式下的文本检索任务性能比较

弦相似度，从而实现高效的图文互检索功能。

3. 预训练与微调：该模型通过在大量公共数据集上进行预训练，得到了一个泛化性能强大的基线模型。同时，通过针对比赛数据集的数据增强和微调，进一步提升了模型的性能表现。预训练和微调策略的应用使得模型能够适应特定任务并取得显著的性能提升。
4. 效率优化：所提出的方法对原始 CLIP 进行了一系列效率优化，包括使用累积梯度模拟大的训练 Batch，得到更加的收敛性能；利用掩码特征学习减少训练时的显存占用，同时学习更加鲁棒的特征；使用 Flash Attention，从内存分布层面提高运算效率等。
5. 高性能表现：基于所提出的模型，在图像检索和文本检索任务中，取得了较高的 R@5 精度，即在前 5 个检索结果中的准确率。这表明该模型在处理图文检索问题上具有较好的性能和效果。

模型不足：

1. **数据集依赖性：**该模型的性能和效果受到所使用的数据集的影响。虽然作者使用了大规模的预训练数据集和增强数据集，但模型的泛化能力仍然可能受到数据集的局限性。
2. **可解释性有限：**尽管该模型在图文检索任务上表现出色，但由于其基于深度学习模型的黑盒特性，其内部的决策过程和解释性有限。这使得模型的可解释性和可理解性不如传统的基于规则或特征工程的方法。

总体而言，该论文提出的基于大规模预训练 CLIP 模型微调的图文检索方法在处理多模态数据和提高检索精度方面具有优势。然而，它也面临数据集依赖性和可解释性有限等挑战。

□

参考文献

- [1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.
- [2] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, “Self-supervised learning: Generative or contrastive,” *IEEE transactions on knowledge and data engineering*, vol. 35, no. 1, pp. 857–876, 2021.
- [3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. Int. Conf. Learn. Represent.*, 2020.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [6] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, “Pre-training with whole word masking for chinese bert,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3504–3514, 2021.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 248–255, Ieee, 2009.
- [9] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, 2016.
- [10] P. K. Diederik, “Adam: A method for stochastic optimization,” (*No Title*), 2014.
- [11] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [12] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 1989.

- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016.
- [16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.