中文CLIP：中文对比视觉-语言预训练

摘要：视觉-语言基础模型的巨大成功推动了计算机视觉和多模态表示学习的研究和应用。然而，有效地将这些基础模型转移到特定语言场景仍然具有一定难度。在本工作中，我们提出了中文CLIP，采用两阶段预训练方法，第一阶段对模型进行锁定图像调整，第二阶段对模型进行对比调整。具体而言，我们开发了5种不同规模的中文CLIP模型，参数从7700万到9580万不等，并在收集的大规模中文图像-文本对数据集上对它们进行了预训练。我们的综合实验表明，中文CLIP在零-shot学习和微调的设置下，在MUGE、Flickr30K-CN和COCO-CN上可以达到最先进的性能，并且在基于ELEVATER基准测试的零-shot图像分类评估中能够达到有竞争力的性能。我们已经发布了我们的代码、模型和演示。

1 引言 从NLP的预训练热潮开始，基础模型引起了多个研究社区的关注。从大规模无监督或弱监督数据中学习的基础模型成为下游模型的基础。多模态表示学习中的一个里程碑是CLIP（Radford等人，2021年），它是一种基于对比学习的模型，使用了从网络收集的约4亿个图像-文本对数据的大规模数据集进行预训练。尽管方法相对简单，但CLIP不仅在视觉-语言检索方面取得了出色的性能，更重要的是，它作为一个视觉基础模型，在一系列数据集上展示了零-shot图像分类的最先进性能。CLIP在视觉和语言之间建立了联系，已经在多模态表示学习和计算机视觉的研究中产生了深远影响。然而，将跨模态预训练模型有效地转移到另一种语言仍然存在一些困难。首先，学习建模语言本地视觉和语言数据的分布对于转移至关重要。虽然在大多数情况下，CLIP作为一个强大的基础模型表现出色，但我们发现，使用机器翻译的CLIP在中文本地跨模态检索基准上表现不佳。图1展示了原始CLIP和我们的中文CLIP在所有模型尺度上的显著性能差距。我们认为，让两个编码器都从语言本地的图像和文本中学习对于成功很重要。其次，中文多模态预训练的先前方法的性能受到了几个因素的限制。从头开始的预训练需要收集一个类似于OpenAI CLIP的Web Image Text（WIT）的大规模高质量语言特定图像文本对数据集（Fei等人，2021年；Xie等人，2022年）。尽管可以通过使用CLIP初始化和锁定图像调整（Zhai等人，2022年）来实现CLIP快速转移到中文数据，但视觉编码器仍然不能从语言特定领域的图像中学习信息（Gu等人，2022年）。因此，我们提出了中文CLIP，这是一个在公开可用的中文图像-文本对数据上预训练的语言特定的视觉-语言基础模型。此外，我们仍然使用与OpenAI CLIP相同的架构。为了实现将跨模态基础模型有效地转移到中文数据，我们开发了一个两阶段预训练方法，该方法也适用于其他视觉-语言基础模型，例如ALIGN、Florence等。在这项工作中，我们以CLIP为例。具体来说，我们首先使用预训练模型初始化两个编码器，即来自CLIP的视觉编码器和来自RoBERTa-wwm-Chinese的文本编码器（Cui等人，2020年）。在第一阶段，我们冻结图像编码器，只优化文本编码器与LiT，而在第二阶段，我们使用对比调整对两个编码器进行训练。通过这种方式，新模型可以通过初始化和LiT从基础模型中继承，通过对比调整有效地转移到语言特定数据。我们在包括MUGE2、Flickr30K-CN（Lan等人，2017年）和COCO-CN（Li等人，2019c年）在内的3个中文跨模态检索数据集上评估了中文CLIP。实验结果表明，无论是在零-shot学习还是微调的设置下，大尺寸和巨型尺寸的中文CLIP都在这3个数据集上达到了最先进的性能。此外，我们在ELEVATER基准测试（Li等人，2022b年）的“野外图像分类”赛道上评估了零-shot图像分类的能力。在分类数据集上，与最先进方法相比，中文CLIP表现出有竞争力的性能，并且优于中文基线方法。此外，我们提供了NVIDIA TensorRT和ONNX模型，用于部署，其推断速度比PyTorch模型快2到10倍。简言之，我们的贡献包括：• 我们提出了中文CLIP，这是一个简单的实现，使用我们收集的大规模中文图像-文本对数据进行了预训练，并提出了一个两阶段预训练方法，以实现高预训练效率和改进的下游性能。• 中文CLIP在零-shot学习和微调设置下，在跨模态检索方面达到了最先进的性能，并在零-shot图像分类方面具有竞争力的性能。 2 方法基于大规模弱监督数据的简单视觉-语言对比预训练的CLIP（Radford等人，2021年）是多模态表示学习中的一个重要基础模型。它可以直接转移到跨模态检索中，其图像编码器可以充当视觉骨干。在这项工作中，我们提出通过在大规模中文多模态数据上预训练视觉-语言模型来构建一个语言特定的CLIP模型。以下是我们中文CLIP方法设计和实现的详细细节。

2.1 数据

CLIP成功的关键之一应该是用于预训练的大规模数据集。基于对CLIP的重新实现的实验（Ilharco等人，2021年），扩大数据规模和延长训练过程可以持续改善零-shot学习的模型性能。今年，最新的多模态预训练模型Wukong（Gu等人，2022年）和R2D2（Xie等人，2022年）在一个公开数据集（1亿图像-文本对）和一个内部数据集（2.5亿样本）上进行了预训练，其中只发布了2300万样本的子集。为了便于重新实现，我们的目标是在尽可能多的公开可用数据上进行中文CLIP的预训练，因此我们专注于收集高质量的公开数据集。我们从最新的LAION-5B（Schuhmann等人，2021年）中提取了中文数据（带有“zh”标记），并从Wukong数据集中收集数据。然而，由于链接不可用的问题，我们只能从LAION-5B和Wukong中分别收集约1.08亿样本和7200万样本。我们还从经典的英文多模态数据集，包括Visual Genome（Krishna等人，2017年）和MSCOCO（Chen等人，2015年），中添加了翻译后的数据，其中移除了测试集。最终，我们构建了一个包含约2亿图像-文本对的中文多模态预训练数据集。以下说明了数据预处理的过程。对于来自LAION-5B的数据部分，我们删除了mCLIP计算得出的CLIP分数低于0.26的样本（Carlsson等人，2022年）。此外，我们还删除了包含内部黑名单中的单词的样本。黑名单包含与广告、图像文件名等相关的单词。我们删除了太短（少于5个字符）或太长（超过50个字符）的样本。对于图像，我们将它们调整大小为224×224的分辨率，大多数情况下为336×336的分辨率，对于ViT-L/14@336px为336×336的分辨率。

2.2 预训练方法

预训练中文CLIP模型有多种设计选择。其中一种最简单的方法是从头开始预训练，图像和文本编码器都是随机初始化的。然而，我们认为其性能将受限于当前预训练数据的数量和质量。为了利用现有预训练模型的优势，我们使用来自官方发布的CLIP 4预训练检查点的权重初始化模型，其中图像编码器使用了来自OpenAI CLIP的权重，文本编码器使用了RoBERTa-wwm-ext和RBT3的权重。为了使模型适应引入的预训练数据，可以使用“对比调整”对其进行预训练，类似于将CLIP转移到下游检索数据的方式。与对比调整相比，锁定图像调整（LiT）（Zhai等人，2022年）在下游转移中表现出改进的性能。在这项工作中，我们提出了一个两阶段预训练方法，如图2所示。核心思想是首先利用LiT使文本编码器能够从OpenAI CLIP的基础视觉模型中读取高质量的表示，然后将整个模型转移到引入的预训练数据域。仅仅使用LiT来预训练中文CLIP是不够的，因为图像编码器需要学习来自中文数据集的图像信息，并对这些数据进行建模。在两阶段预训练之前，我们首先使用预训练模型初始化两个编码器。在第一阶段，我们通过在预训练过程中冻结其参数来“锁定”图像编码器。我们只对文本编码器进行视觉-语言对齐的预训练，基于这样一个假设：使用预训练权重的视觉骨干已经是一个强大的视觉基础模型（Zhai等人，2022年；Gu等人，2022年）。我们将其预训练，直到在下游任务中没有显著的性能改善，即使我们延长了预训练进程。然后我们切换到第二阶段，在第二阶段中，我们通过启用图像编码器的优化来“解锁”它。在第二阶段，我们继续预训练，没有任何参数被冻结，以便图像编码器可以学习对中国网站上的图像数据进行建模。在割离研究中，我们讨论了预训练检查点初始化和预训练方法对下游性能的影响。实验结果表明，两阶段预训练方法可以优于从头开始预训练或直接从预训练模型进行微调。

3 评估

为了全面探究中文CLIP的效果，我们遵循常规做法，首先评估其跨模态检索的基本能力，即文本到图像的检索和图像到文本的检索，在不同领域，包括电子商务和通用领域。此外，由于对比学习为基础的预训练构建了与自然语言语义连接的基础视觉模型，我们遵循Radford等人（2021年）的做法，评估其零-shot分类能力。具体而言，我们在ELEVATER基准测试的分类数据集上验证中文CLIP，该基准测试被称为“野外图像分类（ICinW）”。

3.1 跨模态检索

3.1.1 数据集和指标

我们在3个跨模态检索数据集上验证中文CLIP，即MUGE-Retrieval、Flickr30K-CN（Lan等人，2017年）和COCO-CN（Li等人，2019c年）。MUGE-Retrieval是一个图像-文本检索数据集，数据从中国电子商务网站中提取。Flickr30K-CN和COCO-CN是从经典数据集Flickr30K和MSCOCO-1K构建的，其文本被翻译成了中文。我们的评估包括零-shot学习和微调的设置。对于零-shot学习，我们使用中文CLIP模型计算图像和文本之间的相似性分数，并返回前K个最相似的候选项。对于微调，我们使用对比调整对中文CLIP模型进行跨模态检索的微调。评估与零-shot学习中的一致。评估指标为Recall@K，其中K = {1, 5, 10}，以及均值召回（MR，即Recall@K的平均值）。为了比较，我们选择了基准线的基准大小和大尺寸Wukong和R2D2，它们是中文多模态表示学习中的先前SOTA模型。遵循这些基准线，我们在MUGE上报告验证性能，在Flickr30K-CN和COCO-CN上报告测试性能。请注意，在微调设置中，R2D26本质上是一个检索和排名的端到端模型。

3.1.2 结果

表1报告了MUGE-Retrieval上的模型性能。对于基准大小的模型，CN-CLIPViT-B/16在所有指标上均优于基线，并在零-shot学习和微调的两种设置中都如此。具体而言，对于基准大小的模型，CN-CLIPViT-B/16在零-shot学习中超过WukongViT-B/32约17.0 MR，在微调中超过R2D2ViT-B约8.7 MR。此外，小型模型CN-CLIPRN50在零-shot学习中可以在几个指标上超过基准大小的WukongViT-B/32，超过了8.9 MR，在微调中超过8.0 MR。对于大型模型，CN-CLIPViT-L/14可以在所有指标上超过两个基线，而在图像分辨率较大的图像上预训练的CN-CLIPViT-L/14@336px可以实现最先进的性能。CN-CLIPViT-L/14@336px在零-shot学习中超过R2D2ViT-L/14约6.6 MR，在微调中超过3.8 MR。当扩展到CN-CLIPViT-H/14时，性能进一步提高。与最佳大型模型CN-CLIPViT-L/14@336px相比，CN-CLIPViT-H/14在零-shot学习中超过了2.7 MR，在微调中超过了2.3 MR。表2和表3报告了在Flickr30K-CN和COCO-CN上的模型性能。我们专注于R@1的评估。在这两个数据集中，CN-CLIP的性能均优于基线。对于基准大小的模型，在Flickr30K-CN的零-shot学习设置中，CN-CLIPViT-B/16在文本到图像检索中超过了WukongViT-B/32约17.0 R@1，在图像到文本检索中超过了8.4 R@1，在微调设置中，CN-CLIPViT-B/16在图像检索中超过了R2D2ViT-B约0.8 R@1，文本检索中超过了0.9 R@1。类似地，在COCO-CN的微调设置中，CN-CLIPViT-B/16超过R2D2ViT-B约1.9 R@1，在图像检索中超过文本检索约1.3 R@1。对于小型CN-CLIPRN50，它在Flickr30K-CN和COCO-CN的一些指标中再次实现或超越了WukongViT-B/32的性能。具体而言，在Flickr30K-CN图像检索的零-shot学习中，CN-CLIPRN50在R@1方面超过WukongViT-B/32约3.1，在COCO-CN文本检索的微调中超过2.6 R@1。对于大型模型，在Flickr30K-CN的零-shot学习设置中，CN-CLIPViT-L/14在文本到图像检索中超过了WukongViT-L/14约16.3 R@1，在图像到文本检索中超过了4.1 R@1。CN-CLIPViT-L/14@336px在图像检索中超过1.0 R@1，在文

本检索中超过3.1 R@1。在微调设置中，CN-CLIPViT-L/14在文本检索中超过R2D2ViT-L/14约0.5 R@1。CN-CLIPViT-L/14@336px在图像检索中与R2D2ViT-L/14的性能相当，在文本检索方面超过1.0 R@1。同样，在COCO-CN的微调设置中，CN-CLIPViT-L/14超过R2D2ViT-L/14约0.9 R@1。CN-CLIPViT-L/14@336px在图像检索中进一步超过R2D2ViT-L/14约1.0，在文本检索中超过1.9 R@1。在Flickr30K-CN和COCO-CN上，从CN-CLIPViT-L/14到CN-CLIPViT-H/14的扩展提高了几乎所有指标的性能。具体而言，在Flickr30K-CN的零-shot设置中，CN-CLIPViT-H/14在图像检索方面超过CN-CLIPViT-L/14约3.2 R@1，在文本检索方面超过1.4 R@1。此外，在COCO-CN的微调设置中，CN-CLIPViT-H/14甚至超过了具有更大图像分辨率的CN-CLIPViT-L/14@336px，图像检索方面超过1.4 R@1，在文本检索方面超过2.3 R@1。我们还将我们的CN-CLIPViT-H/14与一个大型的CLIP类似模型T-Bletchley7进行了比较。该模型具有25亿个参数，经过数十亿个多语言图像-标题对的预训练。在COCO-CN的微调设置中，尽管模型参数和预训练数据集的规模较小，CN-CLIPViT-H/14仍然超过T-Bletchley约3.9 MR。

3.1.3 割离研究

在这里，我们对我们提出的两阶段训练方法进行了割离研究。为了验证其重要性和有效性，我们设计了几个割离研究的设置。我们的实验是在CN-CLIPViT-B/16上进行的。为了评估初始化的影响，我们从头开始预训练一个模型；为了检验LiT的影响，我们预训练一个没有冻结图像编码器的模型。为了更好地展示，我们报告了零-shot检索的模型性能曲线，以不同的预训练进度为指示，即已处理的样本数量。图3展示了不同任务上的性能，即MUGE文本到图像检索，Flickr30K-CN和COCO-CN上的文本到图像和图像到文本检索。与使用预训练模型初始化的预训练相比，从头开始的预训练效果要差得多，尽管在预训练进展方面表现出一致的性能改善。关于LiT的重要性，我们观察到在不同数据集上存在不同的现象。在MUGE数据集上，这是一个最初从中国网站收集的样本集，我们发现没有LiT的预训练可能是最好的解决方案，尽管它与两阶段预训练的性能差距非常小。然而，在其他数据集上，即Flickr30K-CN和COCO-CN，其样本是从英文数据集翻译而来的，我们发现我们的两阶段预训练比没有LiT的预训练性能显著更好。此外，我们观察到一个共同现象，即在两阶段预训练中，从阶段1切换到阶段2可以有效地提升模型性能到更高水平。这反映了将预训练模型适应于中国多模态数据的数据分布的重要性，尤其是那些涉及视觉信息的数据。

3.2 零-shot 图像分类

3.2.1 开放域图像分类基准测试中的中文对照训练

在图像-文本对上进行的对比性预训练在视觉和自然语言之间建立了联系。自然语言监督而不是众包标注赋予了模型零-shot图像分类的能力，通过计算给定图像与候选集中标签的文本描述之间的相似性分数。这个领域的最新进展是ELEVATER基准测试（Li等人，2022b年）。ICinW跟踪用于开放域图像分类，包括一系列图像分类数据集，如ImageNet（Deng等人，2009年）、CIFAR（Krizhevsky等人，2009年）、MNIST（Deng，2012年）等。为了评估中文CLIP在这些数据集上的表现，我们首先通过将标签和提示翻译为中文来为中文模型转化数据集。

3.2.2 实验结果

表4报告了英文模型和中文模型的性能。在英文数据上预训练的基准线包括DeCLIP（Li等人，2021b年）、ALIGN（贾等人，2021年）、CLIP（Radford等人，2021年）和OpenCLIP（Ilharco等人，2021年），在中文数据上预训练的基准线包括BriVL（费等人，2021年）和悟空（顾等人，2022年）。我们报告具有最佳下游性能的变体模型的结果。我们首先专注于与中文基准线的比较。具体而言，对于包括ImageNet分类在内的所有数据集，中文CLIP在很大程度上超过了两个基准线，其中某些数据集的相对成就超过了100%。此外，我们还将中文CLIP与基于英文数据预训练的基础模型（如CLIP和ALIGN）进行比较。可以发现，在CIFAR-10、CIFAR-100、FER-2013、KITTI-Distance、MNIST、PatchCamelyon和Pascal-VOC-2007等数据集上，中文CLIP的性能超过了CLIP或ALIGN。此外，在既有西方文化又有东方文化的常见概念或对象的分类数据集上，中文CLIP始终表现出更好的性能。这表明中文CLIP能够将图像分类为一般的原型。然而，对于与专有名词相关的分类，例如FGVC-Aircraft，所有模型都很难实现高精度。我们推测，预训练数据集中相关的图像和文本并不常见，模型在没有微调的情况下也很难理解飞机的名称。具体来说，对于中文模型，翻译甚至音译都会显著影响中文CLIP的性能。这鼓励建立一个“面向中文模型的野外图像分类基准测试”。

3.2.3 分析对手工提示的敏感性

尽管ELEVATER基准测试为每个数据集提供了具体的提示，但我们发现与我们的基线，即由OpenAI CLIP提供的提示翻译相比，这并不总是最佳选择。大约90个提示的基线平均表现最佳。然而，对于某些数据集，基于人类知识设计的特定提示可以显著提升性能。典型案例是飞机的分类。我们测试了具有与飞机知识相关的特定提示的CN-CLIPViT-L/14，例如“标签，一张飞机的照片”、“标签，一张战斗机的放大图像”等，以及OpenAI提示的翻译。实验结果显示，使用特定提示时，模型可以达到16.0的准确率，而使用OpenAI提示时仅为13.8。无法理解否定性语句之前的研究（Khandelwal和Sawant，2019；Hosseini等人，2021）表明，即使是强大的NLP预训练模型在否定问题上也经常出错。我们通过在KITTI-Distance（Fritsch等人，2013年）和PatchCamelyon（Veeling等人，2018年）上进行实验，探究了CLIP理解否定的能力。KITTI-Distance为模型提供4个选项进行判断，包括“靠近汽车”、“在汽车附近”、“与汽车距离较远”和“没有汽车”。最后一个涉及否定。我们比较了使用文本“没有汽车”和“其他”作为最后一个标签的模型性能。我们观察到，模型很难理解否定。通过将标签从“其他”更改为“没有汽车”，准确率下降了48.1%（49.9 vs. 25.9）。类似地，在PatchCamelyon的实验中，将标签从“主要为红色”和“中间有绿色方块”更改为“中间没有绿色方块”和“中间有绿色方块”时，性能从63.5下降到50.2。这显示了CLIP在学习否定方面的局限性。预训练数据集中的文本主要是图像的描述，表明它们的对象或特征，但通常不表明物体的缺失。

3.3 部署

为了部署，我们基于我们的Pytorch预训

练的中文CLIP模型开发了基于ONNX和TensorRT的模型。如预期所料，我们观察到推理效率显著提高，几乎没有性能牺牲。具体而言，基于TensorRT的模型的推理效率比基于Pytorch的模型快2到10倍。更多统计数据在附录A.6中列出。

4 相关工作

以往的视觉-语言预训练模型主要是BERT/T5风格的（Devlin等人，2019年；Raffel等人，2020年），涉及跨模态融合（Chen等人，2020年；Li等人，2019a,b；Lu等人，2019年；Lin等人，2020年；Li等人，2020年；Huang等人，2020年；Xu等人，2021年；Zhang等人，2021年；Shen等人，2021年；Wang等人，2021b，2022a；Li等人，2021a，2022c,a；Wang等人，2021a，2022b）。CLIP（Radford等人，2021年），则是基于对比学习的双塔模型，可以作为视觉基础模型。在CLIP之后，出现了一系列类似的基于对比学习的跨模态预训练模型，并在跨模态检索和零-shot分类方面达到了新的SOTA（Jia等人，2021年；姚等人，2021年；袁等人，2021年）。此外，CLIP还可以适应其他模型。一个典型案例是CLIP对许多图像生成模型的重要性，例如DALL-E（Ramesh等人，2021年）、DALL-E 2（Ramesh等人，2022年）、Stable Diffusion（Rombach等人，2022年）等。多模态预训练的成功鼓励了将现有方法转移到中文预训练中，包括生成性预训练模型（林等人，2021a；费等人，2021年；杨等人，2021年；林等人，2021b；王等人，2022a）和对比预训练模型（费等人，2021年；顾等人，2022年；谢等人，2022年；陈等人，2022b年）。

5 结论

在这项工作中，我们提出了中文CLIP，这是一种特定于中文的视觉-语言基础模型。具体而言，我们构建了一个约2亿样本的预训练数据集，并使用提出的两阶段预训练方法预训练了一系列中文CLIP模型，该方法改善了预训练的效率和效果。我们的综合评估显示，中文CLIP在多个跨模态检索数据集中可以达到最先进的性能，在零-shot学习和微调中也表现出色。此外，我们证明了中文CLIP模型在10个数据集上也可以实现有竞争力的零-shot图像分类性能。局限性一些问题反映了这项工作的局限性，但也指出了未来研究的一些方向。在本节中，我们通常讨论了关于数据和模型规模的一些局限性。数据CLIP预训练的核心是极大规模数据上简单但有效的对比预训练。尽管我们已经利用了约2亿样本，但与近期研究（袁等人，2021年；陈等人，2022a年）相比，我们的预训练数据规模相对较小。因此，我们下一步的研究之一是增加预训练数据的数量，以评估随着数据规模的增加而带来的性能改进。此外，我们仍然很难确定CLIP高质量数据集是什么。在先前的研究中（贾等人，2021年；李等人，2021b年），预处理方法大多是简单的，以避免数据丢失。然而，仍然有许多图像和文本不匹配的样本，这可能会向预训练提供负面信息。在未来的研究中，我们计划使用预训练的中文CLIP模型为较大数据集中的每个图像-文本对计算一个分数，过滤掉那些分数低于指定阈值的对，并使用新数据对新模型进行预训练。这是探索数据质量与预训练效果之间关系的一种可能解决方案。此外，这种迭代可能会在下游任务中带来持续的性能提升。模型最近，我们已经看到在许多领域，模型规模的扩大可以带来持续的性能改进（Gordon等人，2021年；魏等人，2022年），在这项工作中，我们还发现中文CLIP的模型规模扩大可以在不同的下游任务中实现稳定的性能改进，包括检索和分类。最近的研究已经将ViT和CLIP类似的模型扩展到比我们最大的CN-CLIPViT-H/14大得多的规模，例如3B Swin-v2（刘等人，2022年），4B ViT-e（陈等人，2022a年）等。在未来，我们将继续探索与数据扩展相匹配的模型扩展，以建立一个更

有效的中文CLIP。与实际应用相关的模型扩展的另一个问题是如何构建有效的小模型。实验结果表明，我们最小的中文CLIP CN-CLIPRN50的性能要比ViT变体差得多。然而，在实际应用中，可以用于部署的有效小模型通常更受欢迎。因此，有必要探索蒸馏技术，以便将大型模型的能力转移到小型模型中以进行应用。伦理声明所提出的模型是基于对比学习的视觉-语言基础模型，可以为图像和文本生成特征。这些特征可以是视觉和语言信息的表征，并且可以支持诸如搜索引擎、推荐系统等应用。此外，该模型还可以作为支持最新图像生成模型的基础模型，例如扩散模型（Ramesh等人，2022年）。这可能会带来风险，因为AI生成的内容可能反映出有害信息，例如仇恨、偏见、色情等。在大多数情况下，应该将这些情况归因于图像生成模型的训练。尽管如此，我们不能避免CLIP表示对生成产生的负面影响。在未来，我们将研究如何过滤预训练数据，以避免潜在风险。

附录

A.1 模型架构细节

我们开发了5个不同规模的中文CLIP模型，参数数量从大约7700万到9.58亿不等。我们包括1个ResNet-50模型CN-CLIPRN50和4个ViT模型，即CN-CLIPViT-B/16、CN-CLIPViT-L/14、CN-CLIPViT-L/14@336px和CN-CLIPViT-H/14，其中模型在未指定分辨率的情况下对224×224像素的图像进行预训练。表5呈现了模型架构的详细信息。最小的模型CN-CLIPRN50由ResNet-50图像编码器和RBT3文本编码器组成。基础大小的模型CN-CLIPViT-B/16由ViT-B/16@224px图像编码器和RoBERTa-wwm-Base文本编码器组成。大型模型CN-CLIPViT-L/14由ViT-L/14@224px图像编码器和RoBERTa-wwm-Base文本编码器组成，而CN-CLIPViT-L/14@336px由ViT-L/14@336px和RoBERTa-wwm-Base组成。具体而言，我们通过在预训练的CN-CLIPViT-L/14上继续预训练来预训练CN-CLIPViT-L/14@336px。为了适应更大的分辨率，我们通过对CN-CLIPViT-L/14的位置嵌入应用插值来初始化图像位置嵌入，遵循Ilharco等人（2021年）的方法。巨大大小的模型CN-CLIPViT-H/14由ViT-H/14图像编码器和RoBERTa-wwm-Large文本编码器组成。在附录A中还提供了更多的实现细节。我们在表6和表7中提供了它们的模型架构的更多细节。我们保持ResNet-50、ViT-B/16和ViT-L/14的骨干架构与OpenAI CLIP一致，ViT-H/14的架构与LAION CLIP8相同。这使得我们可以使用它们的权重初始化中文CLIP图像编码器。文本编码器是中文RoBERTa模型（崔等人，2020）。具体而言，我们最轻量级的小型中文CLIP使用3层RBT3模型的架构。基础大小和大型大小的中文CLIP模型使用RoBERTa-wwm-Base的12层架构。对于巨大大小的CN-CLIP，我们使用RoBERTa-wwm-Large的24层架构。文本分词器的词汇表大小为21,128。

A.2 预训练细节

初始化

如第2.2节所述，我们使用OpenAI CLIP的权重初始化CN-CLIPRN50、CN-CLIPViT-B/16和CN-CLIPViT-L/14的图像编码器。CN-CLIPViT-H/14的图像编码器使用LAION CLIP进行初始化。除了ResNet或ViT参数外，温度和视觉输出投影参数也使用预训练的CLIP权重进行初始化。对于文本编码器，我们使用相应模型规模的已发布中文RoBERTa权重进行参数初始化，其中舍弃了它们的池化层权重。文本输出投影权重使用正态分布进行随机初始化。

阶段1

阶段1的预训练超参数如表8所示，适用于CN-CLIPRN50、CN-CLIPViT-B/16、CN-CLIPViT-L/14和CN-CLIPViT-H/14。超参数的值通常与OpenAI CLIP（Radford等人，2021年）中的值类似。关于数据增强，我们在输入图像上使用随机大小裁剪和AutoAugment（Cubuk等人，2019年）。我们通过GPU工作进程之间的全聚合通信来计算全局批次上的对比损失。在这个阶段，这4个模型分别进行了大约20、44、64和26个时期的预训练，图像编码器被冻结。CN-CLIPRN50的批归一化层的运行方差和均值在这个阶段不会更新。通过在训练过程中测量3个下游零-shot检索任务的平均召回来确定预训练的最佳时期。激活混合精度训练。在这个阶段，我们使用64个NVIDIA V100 GPU预训练CN-CLIPRN50，使用128个NVIDIA V100 GPU预训练CN-CLIPViT-B/16，使用128个NVIDIA V100 GPU预训练CN-CLIPViT-L/14，使用184个NVIDIA A100 GPU预训练CN-CLIPViT-H/14，分别花费了1.6天、4.5天、11.5天和3.8天。

阶段2

在阶段2中，我们解冻图像编码器并更新所有模型参数。除了峰值学习率、批大小和训练时期之外，阶段1中提到的所有其他超参数保持不变。我们将学习率减小到2e -5，以进行更细致的优化。对于CN-CLIPRN50、CN-CLIPViT-B/16和CN-CLIPViT-L/14，由于GPU内存的限制，批大小分别缩小到16、384、16、384和4,608。当扩展到CN-CLIPViT-H/14时，我们实现了

梯度检查点，从而使批大小增加到32,768。这4个模型在阶段2分别进行了大约44、15、7和7个时期的预训练。在这个阶段，我们使用64个NVIDIA V100 GPU预训练CN-CLIPRN50，使用128个NVIDIA V100 GPU预训练CN-CLIPViT-B/16，使用128个NVIDIA V100 GPU预训练CN-CLIPViT-L/14，使用184个NVIDIA A100 GPU预训练CN-CLIPViT-H/14，分别花费了5.8天、3.0天、8.0天和2.2天。为了预训练更大分辨率的模型，我们对CN-CLIPViT-L/14的图像位置嵌入实现了插值，以适应更大的分辨率，并继续使用336×336像素的图像进行预训练。我们从CN-CLIPViT-L/14开始，连续进行2个时期的预训练。这个预训练仅使用了128个NVIDIA A100 GPU，耗时0.7天。

A.3 微调细节

如表1、表2和表3所报告的，我们主要在3个跨模态检索数据集上对CN-CLIP进行微调：MUGE、Flickr30K-CN和COCO-CN。大多数微调实验都在32个NVIDIA A100 GPU上进行。微调策略和损失与预训练过程一致。为了提高时间效率和充分利用计算资源，我们将批大小设置得尽可能大。我们在CN-CLIPViT-L/14@336px和CN-CLIPViT-H/14的微调过程中实现了梯度检查点，以实现更大的批大小。表9显示了微调过程中批大小、最大学习率、最大时期和预热迭代的具体设置。我们将其他超参数默认设置为与预训练中相同。我们在每个时期结束时保存模型参数。对于MUGE，我们报告验证集上的最佳结果。对于Flickr30K-CN和COCO-CN，我们选择在验证集上性能最好的检查点，并报告测试集上的结果。

A.4 长文本的跨模态检索

在第3.1.2节报告的结果表明了中文CLIP出色的跨模态检索能力。需要注意的是，MUGE、Flickr30K-CN和COCO-CN的平均文本长度分别为7.4、19.7和16.8。我们还在具有平均文本长度45.3的ICR（Xie等人，2022年）数据集上进行了微调实验。表10中展示了实验结果。由于ICR数据集中的文本较长，我们将微调的最大文本长度设置为128。结果显示，中文CLIP在具有更长文本的跨模态检索任务中取得了最先进的性能。

A.5 零-shot图像分类实验细节

我们在表11中提供了ELEVATER基准测试中ICinW轨迹的20个图像分类数据集的数据统计和指标。为了将中文CLIP适应英语原生基准测试，我们采用了一系列预处理策略。具体而言，我们将标签的文本描述和手动提示的模板翻译成中文。例如，CIFAR-10中的标签包括“car, dog, ...”，我们将这些词手动翻译成中文。还有特殊情况，例如FGVC-Aircraft（Maji等人，2013年）中的标签很难翻译或音译。我们在Google上搜索名称，找出每个标签的最佳中文名称。尽管如此，我们无法保证我们拥有最佳的中文翻译，更重要的是，中文预训练模型仍然很难理解一些概念，这可能导致相关下游任务的性能不佳。至于模板，对于一些数据集，我们使用ELEVATER工具包提供的模板的我们的翻译，对于其他数据集，我们使用OpenAI CLIP中的模板的翻译。在表12中展示了所有中文CLIP模型在零-shot图像分类上的实验结果。可以发现，模型规模的扩大可以始终提升模型的性能。中文CLIP的规模可预测的改进表明我们可以在未来的工作中进一步扩大模型以获得更好的性能。然而，遗憾的是，微小的CN-CLIPRN50的性能明显比显著更大的ViT变体要差得多。这表明小型模型仍然有很大的改进空间，而将CLIP的知识从大型模型传递到小型模型应该是多模态表示学习中的一个重要研究主题。

A.6 部署

中文CLIP支持部署为基于ONNX和TensorRT的模型，可以实现更快的文本和图像表示生成（尤其是用于在线推理）。在本节中，我们提供有关模型转换以及性能改进的更多细节。具体而言，我们使用PyTorch中的ONNX模块和ONNXMLTOOLS12软件包将中文CLIP PyTorch模型转换为FP16精度的基于ONNX的模型。借助ONNXRUNTIME-GPU13软件包的支持，基于ONNX的模型能够在NVIDIA GPU上进行推断。TENSORRT软件包使从基于ONNX的模型获得的TensorRT-based模型成为可能，并提供GPU推断运行时。我们的TensorRT-based模型也采用FP16精度。我们使用单个NVIDIA T4 GPU的服务器对PyTorch实现的中文CLIP模型进行基于ONNX和TensorRT的模型的转换进行基准测试。该服务器包含16个英特尔至强（Skylake）Platinum 8163 CPU核心和64GB内存。对于每个模型，我们推断100个批次的视觉和文本表示，并计算平均时间。模拟在线部署场景，我们使用批大小为1。所有模型均使用FP16精度进行推断。表13显示了推断时间成本的比较。对于几乎所有模型规模，基于ONNX和TensorRT的模型在原生PyTorch实现的中文CLIP模型上具有优化后的推断速度，特别是在较小的模型尺寸上。对于视觉表示推断，TensorRT-based模型的速度约为Pytorch-based模型的1.3倍（CN-CLIPViT-H/14）到9.5倍（CN-CLIPRN50）。对于文本表示推断，TensorRT-based模型的速度约为PyTorch对应模型的6.2倍（CN-CLIPViT-H/14）到8.2倍（CN-CLIPViT-L/14）。我们还通过在MUGE检索数据集上测量零-shot模型表示的质量来评估基于ONNX和TensorRT的模型表示的质量。表14提供了实验零-shot结果，显示转换的基于ONNX或TensorRT的模型在保持视觉和文本表示质量方面表现良好，检索性能的退化不超过0.1 MR。