Conditional Prompt Learning for Vision-Language Models

摘要

随着像 CLIP 这样强大的预训练视觉语言模型的出现，研究如何使这些模型适应下游数据集变得至关重要。最近提出的一种名为 Context Optimization(CoOp) 的方法将提示学习的概念引入视觉领域，以适应预先训练的视觉语言模型。具体来说，CoOp 将提示模板中的上下文单词转化为一组可学习的向量，并且只需要少量标记图像进行学习，就可以比需要高度调整的手工模板实现巨大的改进。但是 CoOp 容易过度拟合训练期间的基类，学习到的 context vectors 难以扩展到同一数据集的新类。为了解决这个问题，CoCoOp 引入了一个轻量级的神经网络，为每一个输入图像生成一个向量，并与 context vectors 结合，表现出了较强的域泛化性和可迁移能力。同 CoCoOp 类似，我们的方法也是动态地生成适应与每个输入图像实例的 prompt，不同的是我们的 prompt 生成器直接生成 prompt 的 context vectors，不再对给定初始模板进行调优。本文针对 CoCoOp 4.1 研究同一数据集中基类到新类泛化能力的研究，我们选取了其中 4 个数据集，复现了 CLIP，CoOp 和 CoCoOp 三种方法，同用在我们方法 CAPG 进行实验。实验结果表明，我们的方法与 CoCoOp 不相上下，比起 CoOp，在同一数据集上表现出了更强的可迁移性。

关键词：CLIP；CoOp；CoCoOp；Prompt Generator；跨模态；提示学习

1 引言

最近在大规模跨模态视觉语言预训练方面的研究，在零样本图像识别方面取得了惊人的表现[1]，证明了这种范式在学习开放的视觉概念方面的潜力。关键的设计在于如何对视觉概念进行建模。在传统的监督学习中，标签是离散的，每个类别都与一个随机初始化的权重向量相关联，通过学习使其与包含相同类别的图像的距离最小。这样的学习方法专注于封闭式的视觉概念，将模型限制在预先定义的类别列表中，当涉及到训练中未曾见过的新类别时，是无法扩展的。

相反，对于像 CLIP[2]和 ALIGN[1]这样的视觉语言模型，分类权重是由一个参数化的文本编码器（例如 Transformer[3]）通过 prompt[4]产生。例如，为了区分含有不同品种的狗和猫的宠物图片，我们可以采用一个提示模板，如”a photo of a class, a type of pet” 作为文本编码器的输入，结果，通过在”class” 中填写真实的类别名称，可以合成特定类别的分类权重。与离散的标签相比，视觉语言模型的监督来源来自于自然语言，这使得开放的视觉概念可以被广泛地探索，并被证明在学习可转移的表征方面是有效的[1]。

随着这种强大的视觉语言模型的兴起，学术界最近开始研究使这些模型适应下游数据集的方法[5-7]。例如 CLIP 使用的训练数据是 4 亿个图像文本对，模型的参数有数亿个。像深度学习研究中经常采用的对整个模型进行微调的方法，如今就行不通了，甚至可能会破坏学好的表示空间。

一种更安全的方法根据任务添加一些有意义的上下文词汇来调整提示，这种方法称之为提示工程[2]。如若数据集为 OxfordPets，那么在基础模板”a photo of a class 上添加”a type of pet”，研究表明这种方式有助于提高性能[2]。然而, 提示工程是基于试验和错误的，往往非常耗时又低效。最近，zhou 等人[7]首次将可学习的连续提示向量代替离散的提示模板，实现了提示工程的自动化。他们的方法，即语境优化（CoOp），将提示模板中的上下文单词转化为一组可学习的向量，并且只需要少量标记图像进行学习，就能在图像识别任务上比需要高度调整的手工模板取得更好的表现。

但是 CoOp 容易过度拟合训练期间的基类，学习到的 context vectors 难以扩展到同一数据集的新类。相反，CLIP 的 zero shot 方法所采用的人工设计的模板的在新类的可迁移性比 CoOp 强。为了解决这个问题，所复现的论文 CoCoOp[8]引入了一个轻量级的神经网络，为每一个输入图像生成一个向量，并与可学习的 context vectors 结合，表现出了较强的域泛化性和可迁移能力。同 CoCoOp 类似，我们的方法也是动态地生成适应与每个输入图像实例的 prompt，不同的是我们的 prompt 生成器直接生成 prompt 的 context vectors，不再对给定初始模板进行调优。我们的方法基于交叉注意力机制，我们称之为 Cross Attention Prompt Generator。

为了研究模型在同一个数据集从新类到基类的泛化效果，本文在 4 个数据集上，仿照所复现论文 CoCoOp 的 4.1 小节的实验，用 CLIP、CoOp、CoCoOp 和我们的方法 CAPG 进行了图像识别任务。具体来说，就是在同一个数据集中，首先使用基础类学习一个模型，然后在全新的类上进行测试。实验表明，与 zero-shot CLIP 和 CoOp[7]相比，我们的方法与 CoCoOp 一样，都取得了最佳的整体性能。重要的是，与 CoOp 相比，我们的方法 CAPG 以及 CoCoOp 在未见过的类中都获得了明显的改进，使人工 prompt 和可学习的连续型 prompt 之间的差距大大缩小。

2 相关工作

2.1 视觉语言模型

一种经典的视觉语言跨模态模型由三个关键元素组成：图像编码器，文本编码器和额外的损失函数。在早期，处理图像和文本的模型往往是独立设计，独立学习的，他们通过额外的损失函数模块连接起来，用于对齐。图像的表征通常使用手工制作的描述符[9-10]或神经网络[11]进行编码，而文本则使用例如预训练的词向量[11]或者基于频率的 TF-IDf[9]进行编码。在跨模态对齐方面，常见的方法包括度量学习[11]、多标签分类[12]和 n-gram 语言学习[13]。最近一项研究表明，用图像对应的文本描述来训练视觉编码器部分可以使视觉表征更容易迁移[14]。正如 Zhou 等人所说[7]的，最近在视觉语言模型方面的成功主要归功于以下三个方面的发展：i) Transformers[3]，ii) 对比学习[15] iii) 网络规模的训练数据集[1]。一个代表性的方法是 CLIP[2]，它使用对比损失来训练 2 个基于神经网络的编码器，从而来匹配成对的图像和文本。经过 4 亿对的图像文本对训练后，CLIP 模型表现出了显著的零样本图像识别能力。与 CoOp[7]，CoCoOp[8]类似，我们的方法旨在提供一个有效的解决方案，通过微调，使预训练好的跨模态视觉语言模型去适应下游应用。

2.2 提示学习

提示学习（Prompt Learning），即对输入的文本信息按照特定模板进行处理，把任务重构成一个更能够充分利用预训练语言模型处理的形式。具体而言，以情感分析为例。给定输入文本“今天天气很好。我很 [MASK]”，作为一个提示（prompt），让模型对这个语句进行完形填空。然后根据预测词 [MASK] 去做情感分类。提示学习的关键在于设计提示模板 prompt，而人工设计模板需要耗费大量的人工财力，因此提示学习的研究旨在借助标注好的数据自动生成 prompt。Jiang 等人[16]使用了离散的提示学习方法，使用文本挖掘和转述来生成一组候选提示，在其中选择最优的提示。而连续的提示学习方法[17]，其主要思想是将提示模板转化为一组连续的向量，根据目标损失函数进行端到端的优化。CoOp[7]最早将连续提示学习引入跨模态视觉语言模型中，根据目标函数对连续的提示模板向量进行

调优。而 CoCoOp 则是对可以生成提示模板向量的偏置的 Meta-Net 进行调优。我们的方法则是对能够生成提示模板向量的模板生成器进行优化。

3 本文方法

3.1 CLIP

Contrastive Language-Image Pre-training(CLIP)，是基于对比学习的视觉语言预训练模型。如图 1 所示，CLIP 由图像编码器和文本编码器组成。图像编码器可以是 ResNet[18]或 ViT[19]，它接受图像为输入，生成向量化表示。文本编码器是一个 Transformer[3], 可以将文本序列转换成特征向量。

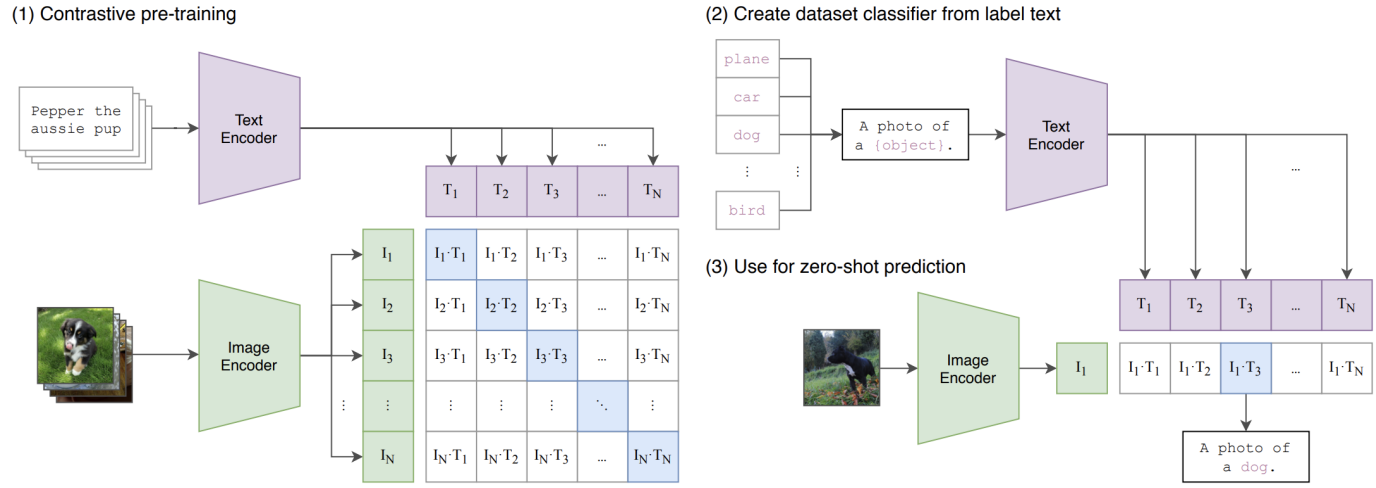

图 1: Contrastive Language-Image Pre-training(CLIP)[2]

预训练。为了学习到更容易转移到下游任务的各种视觉概念，CLIP 的团队从互联网收集了共由 4 亿个文本-图像对组成的大型数据集 WebImageText。在模型训练过程中，CLIP 采用对比损失函数来对齐图像和文本特征。给定一批图像-文本对，CLIP 最大化每个图像与匹配文本的余弦相似度，同时最小化与所有其他不匹配文本的余弦相似度。

零样本推理。经过训练，CLIP 可以无需任何标注的训练数据，进行零样本的图像识别。设 $\boldsymbol{x}$ 为, 图像编码器生成的图像特征向量，$\{\boldsymbol{w}_i\}_{i=1}^K$ 为文本编码器生成的权重向量集合，每个向量代表一个类别 (假设总共有 K 个类别)。当 prompt 是“a photo of a class”时，$\boldsymbol{w}_i$ 表示“a photo of a class”对应的权重向量，其中 class 是第 i 个 class 的对应的名称。预测为第 y 类的概率为

$$p(y \mid \boldsymbol{x}) = \frac{\exp\left(\text{sim}\left(\boldsymbol{x}, \boldsymbol{w}_y\right)/\tau\right)}{\sum_{i=1}^{K} \exp\left(\text{sim}\left(\boldsymbol{x}, \boldsymbol{w}_i\right)/\tau\right)} \tag{1}$$

其中，$sim(\cdot,\cdot)$ 表示余弦相似度，τ 表示可学习的温度参数。

3.2 CoOp

手工搭建模板，即“提示模板工程”，往往需要耗费大量人力，为了克服 CLIP“提示模板工程”中的低效率问题，使预训练的视觉语言模型更好地适应下游任务，Context Optimization (CoOp) 应运而生。如图 2所示，CoOp 不再使用“a photo of a”作为固定的 prompt，而是引入 M 个可学习的 context vector，$\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\}$，每个 context vector 都具有与 word embeddings 相同的维度。用 $\boldsymbol{t}_i = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M, \boldsymbol{c}_i\}$ 第 i 个类的 prompt，其中 $\boldsymbol{c}_i$ 是第 i 个类对应类名的 word embedding。context vectors$\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\}$ 是所有的类共享的。设 $g(\cdot)$ 为文本编码器，则预测为第 y 类的概率为

$$p(y \mid \boldsymbol{x}) = \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{x}, g(\boldsymbol{t}_y)\right)/\tau\right)}{\sum_{i=1}^{K} \exp\left(\operatorname{sim}\left(\boldsymbol{x}, g(\boldsymbol{t}_i)\right)/\tau\right)} \tag{2}$$

为了使 CLIP 适应下游图像识别数据集，可以使用交叉熵损失函数作为学习目标。由于文本编码器 $g(\cdot)$ 是可微的，可以通过梯度的反向传播去更新 context vectors。注意在整个训练过程中，text encoder 和 image encoder 的参数是冻结的。

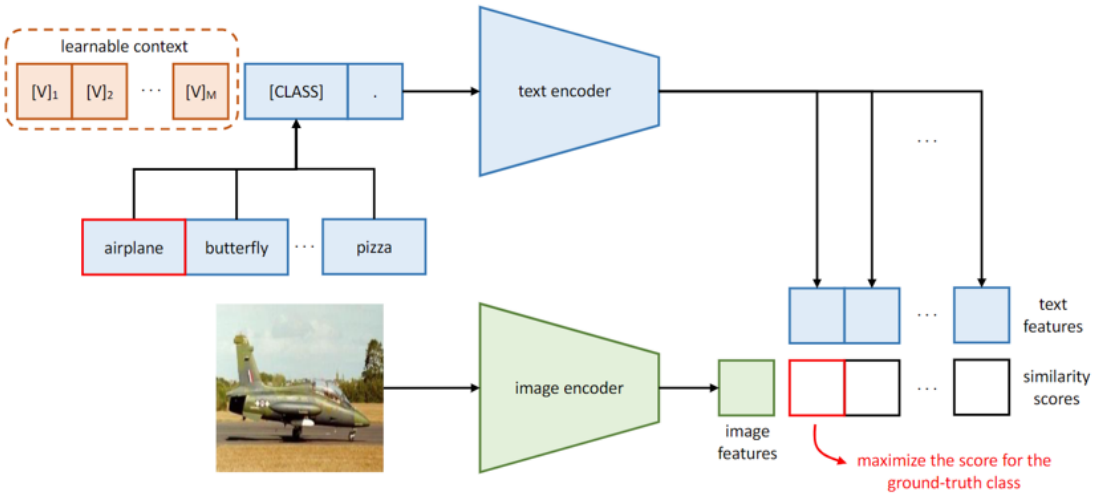

图 2: Context Optimization(CoOp)

3.3 CoCoOp

虽然 CoOp 可以利用少量的标注数据在下游任务对 context vectors 进行优化。但是 CoOp 的可迁移能力较差，在同一个任务中，通过训练得到的 context vectors，若用于与训练集样本类别不同的测试集中进行零样本推理，表现效果较差。即零样本泛化能力弱。为了解决这个问题，所复现的论文 CoCoOp（Conditional Context Optimization）[8]引入了一个称为轻量级神经网络 Meta-Net，可以接受每张输入图片的特征向量，生成对应的 conditional token，然后与 context vectors 结合。在这项工作中，Meta-Net 采用两层瓶颈结构 (Linear-ReLU-Linear)，隐含层将输入维数降低了 16 倍。Meta-Net 的输入仅仅是图像编码器产生的输出特征。CoCoOp 的架构如图 3所示。

设 $h_{\boldsymbol{\theta}}(\cdot)$ 表示参数为 $\boldsymbol{\theta}$ 的 Meta-Net，每个 context token 现在由 $\boldsymbol{v}_m(\boldsymbol{x}) = \boldsymbol{v}_m + \boldsymbol{\pi}$ 获得，其中 $\boldsymbol{\pi} = h_{\boldsymbol{\theta}}(\boldsymbol{x})$，$m \in \{1, 2, \ldots, M\}$。第 i 个类别的 prompt 为 $\boldsymbol{t}_i(\boldsymbol{x}) = \{\boldsymbol{v}_1(\boldsymbol{x}), \boldsymbol{v}_2(\boldsymbol{x}), \ldots, \boldsymbol{v}_M(\boldsymbol{x}), \boldsymbol{c}_i\}$。则预测为第 y 类的概率为

$$p(y \mid \boldsymbol{x}) = \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{x}, g(\boldsymbol{t}_y(\boldsymbol{x}))\right)/\tau\right)}{\sum_{i=1}^{K} \exp\left(\operatorname{sim}\left(\boldsymbol{x}, g(\boldsymbol{t}_i(\boldsymbol{x}))\right)/\tau\right)} \tag{3}$$

在训练过程中，使用交叉熵损失函数作为学习目标，text encoder 和 image encoder 的参数是冻结的，仅更新 context vectors$\{\boldsymbol{v}_m\}_{m=1}^{M}$ 和 Meta-Net 的参数 $\boldsymbol{\theta}$。

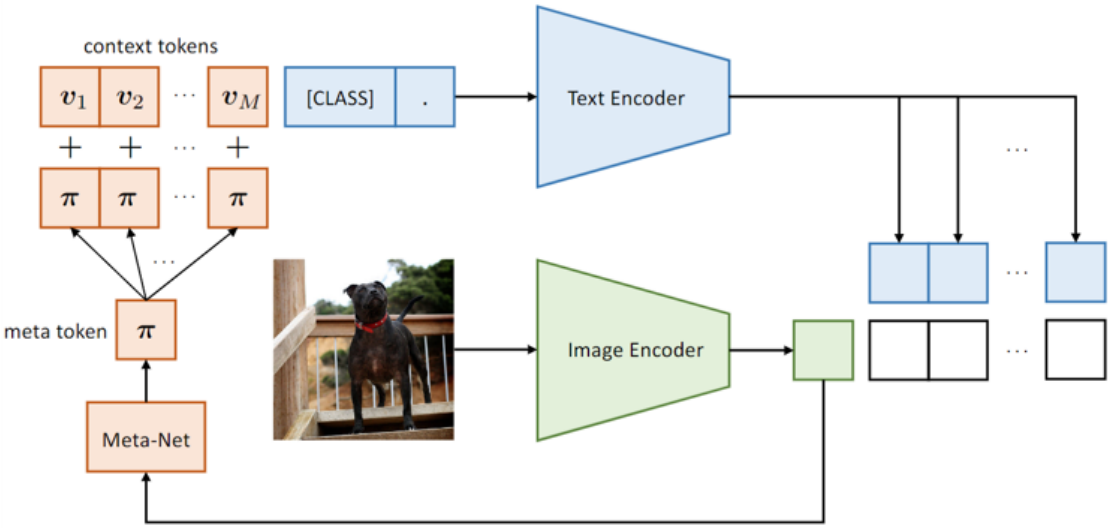

图 3: Conditional Context Optimization(CoCoOp)

3.4 我的方法

本文提出了一种综合了提示工程和适应实例的动态提示学习的模型，即基于交叉注意力机制的 prompt 生成器（CAPG）。同 CoCoOp 类似，我们的方法也是动态地生成适应与每个输入图像实例的 prompt，不同的是我们的 prompt 生成器直接生成 prompt 的 context vectors，不再对给定初始模板进行调优。根据图像特征所隐含的视觉概念对词库的 word embeddings 进行加权，通过训练，让模型生成最适配的连续型的上下文向量。CAPG 的输入有两个，一个是根据任务提前构造好的词库的预训练好的 word embeddings，二是输入图像的特征向量。CAPG 的架构如图 4所示。

设 $f_{\phi}(\cdot)$ 表示参数为 $\boldsymbol{\phi}$ 的 CAPG，此时 $f_{\phi}(\boldsymbol{x})$ 表示 PG 生成的 context vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\}$。因此，当输入图片属于第 i 个类别时，text encoder 的输入为 $\boldsymbol{t}_i(\boldsymbol{x}) = \{f_{\phi}(\boldsymbol{x}), \boldsymbol{c}_i\}$。则预测为第 y 类的概率为

$$p(y \mid \boldsymbol{x}) = \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{x}, f(\boldsymbol{t}_y(\boldsymbol{x}))\right)/\tau\right)}{\sum_{i=1}^{K} \exp\left(\operatorname{sim}\left(\boldsymbol{x}, f(\boldsymbol{t}_i(\boldsymbol{x}))\right)/\tau\right)} \tag{4}$$

在训练过程中，使用交叉熵损失函数作为学习目标，仅更新 Prompt Generator 的参数 $\boldsymbol{\phi}$。对于 CAPG，使用 CLIP 提示工程中采用的各种模板的词汇，构造 CAPG 的词库。为了更好将预训练过程中得学到得视觉概念适应下游任务，我们将预训练好的模型 CLIP 的文本编码器的第一层 Attention 参数来初始化 CAPG 的权重参数。同样，在整个训练过程中，text encoder 和 image encoder 的参数是冻结的。

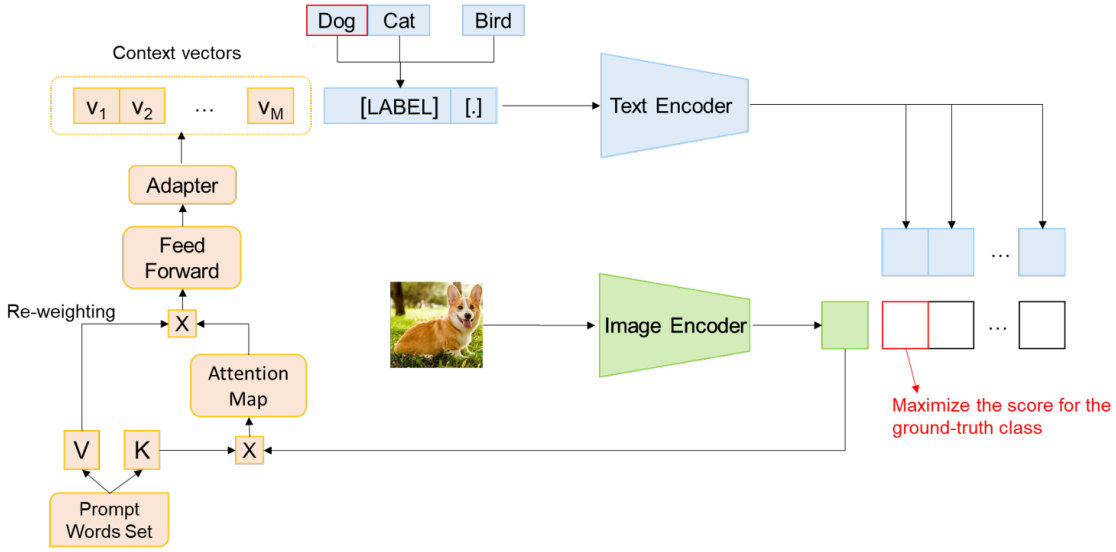

图 4: Cross Attention Prompt Generator(CAPG)

4 复现细节

4.1 实验细节

本文复现论文 CoCoOp[8]4.1 小节的从基类到新类的泛化性实验，实验细节如下。

4.1.1 实验环境配置

本实验在 Ubuntu 64 位的操作系统上，程序语言选择 python。相关的软硬件配置如表 1 所示。

表 1: 软硬件配置

名称	参数
操作系统	Ubuntu 18.04.3 LTS (x86_64)
显卡	NVIDIA A100 80GB PCIe
python	3.8
pytorch	1.8.1+cu111
transformers	4.26.0
dassl	0.6.3

4.1.2 实验设置

在从基类到新类的泛化性实验中,我们选取 Caltech101[20],OxfordPets[21],StanfordCars[22],Food101[23]了 4 个数据集。在 4 个数据集上，我们将类平均分为两组，一组作为基类，另一组作为新类。CLIP 无需训练，直接在基类和新类的测试集上进行评估。基于学习的模型，即 CoOp、CoCoOp 和 CAPG，只使用基类进行训练，而分别对基类和新类进行评估，以测试泛化性。同时，当使用基类进行训练时，随机抽取一个少量的训练集，我们只评估 16-shot，即每类样本只抽取 16 张图片。对于基于可学习的提示向量的模型，即 CoOp、CoCoOp 以及 CAPG，评估结果使随机种子 seed 分别设置为 1，2，3 三次运行的准确度的平均值。对于 CAPG，使用 CLIP 提示工程中采用的各种模板的词汇，构造 CAPG 的词库。为了更好将预训练过程中得学到得视觉概念适应下游任务，我们将预训练好的模型 CLIP 的文本编码器的第一层 Attention 参数来初始化 CAPG 的权重参数。

预训练好的模型选择 CLIP 中图像编码器为 ViT-B/16 的 CLIP 模型，这是 CLIP 中最好用的。Zhou

等人[7]认为较短的上下文长度和良好的初始化可以带来更好的性能和更强的对域迁移的鲁棒性。因此，我们将上下文长度固定为 4，并使用针对 CoOp 和 CoCoOp 的“a photo of a”预训练的词嵌入初始化上下文向量。由于 CAPG 不对上下文向量进行调优，因此 CAPG 仅将上下文长度固定为 4，无需设置“a photo of a”预训练的词嵌入去初始化上下文向量。由于实例条件设计，我们的方法训练速度较慢，并且比 CoOp 消耗更多的 GPU 内存。因此，为了保证模型能适合 GPU，同时减少训练时间，训练时我们将 CoCoOp 和 CAPG 的 batchsize 设置为 1，epoch 设置为 10。

4.2 与已有开源代码对比

本文所参考的源代码来自于论文 CoCoOp 发布的代码 https://github.com/KaiyangZhou/CoOp。本文的主要工作有两个：

一是使用源代码复现了 4.1 小节的从基类到新类泛化性（Generalization From Base to New Classes）在其中四个数据集的实验。其中基于 CoCoOp 的方法有相关的配置文件脚本，可以根据论文稍微调整跑通，但是源码中缺少了基于 CLIP 和 CoCoOp 方法的部分，因此本文根据复现的论文的配置自行编写了模型的配置文件和脚本。

第二个工作则是提出了另一种适应输入实例,为每个输入图像动态生成连续的提示向量的方法,我们称之为基于交叉注意力机制的模板生成器（CAPG）。代码基于 CoCoOp 的源码实现。已知 CoCoOp 是在 CLIP 两个编码器 Text Encoder 和 Image Eocoder 的基础上，加入了 PromptLearner 模块，利用 PromptLearner 生成 Image Encoder 的输入。其 Prompt Learner 模块中设置了 Meta-Net，CoCoOp 在反向传播时更新 Meta-Net 和初始 context vectors。而我们的方法主要是修改 PromptLearner 模块，将 Mete-Net 换成 CAPG，并适应性的修改其他相关的部分代码。最后模仿 CoCoOp 设计 CAPG 的脚本和配置文件，同样在 4 个数据集上完成了从基类到新类泛化性实验。

5 实验结果分析

表 2: 在 base-to-new 泛化实验中 CLIP、CoOp、CoCoOp 和 CAPG 的比较

(a) Caltech101

	Base	New
CLIP	97.40	94.00
CoOp	98.03	89.00
CoOpOp	98.00	92.90
CAPG	97.80	94.70

(b) OxfordPets

	Base	New
CLIP	91.30	97.20
CoOp	93.57	94.70
CoOpOp	95.40	97.43
CAPG	95.40	94.73

(c) StanfordCars

	Base	New
CLIP	63.90	75.00
CoOp	78.67	63.00
CoOpOp	70.83	74.63
CAPG	70.13	74.10

(d) Food101

	Base	New
CLIP	90.00	91.20
CoOp	88.10	84.43
CoOpOp	90.70	91.63
CAPG	90.67	91.67

1) **CoOp 从基类到新类表现出了较弱的泛化性**。由表 2可见，CoOp 在新类上的准确度有 3/4 的都比基类准确度低得多，平均相差 9%。CoOp 与 CLIP 相比，尽管在基类上的准确度有 3/4 都要比 CLIP 的高，但是在新类上的准确度全都比 CLIP 的低得多。这是因为 CoOp 在下游任务的训练阶段生成的提示向量过度拟合，导致在新类的泛化性比较差。这突出了提高基于可学习的连续型提示向量泛化性

的必要性。

2）**CAPG 和 CoCoOp 泛化能力比 CoOp 强**。与 CoOp 相比，CoCoOp 在泛化方面的收益远远大于基类精度的损失，CAPG 和 CoCoOp 在新类上地准确度在 4 个数据集上全都高于 CoCoOp 的，而在基类上的准确度不相上下。因为 CoOp 专门针对基类进行优化，而 CAPG 和 CoCoOp 针对每个实例进行优化，以便在整个任务中获得更多的泛化。

3）**CAPG 和 CoCoOp 在整体性能上比 CLIP 强**。与 CLIP 相比，CAPG 和 CoCoOp 在新类上的准确度相差不大，而且在基类上的准确度都要比 CLIP 高。

6 总结与展望

大规模预训练模型在各种下游任务的表现能力十分出色。如何使让这种预训练模型更好地适应下游任务，在 NLP 和图像领域收到了越来越多的关注。然而，在数据规模和计算资源方面，这种大规模预训练模型训练成本极高。因此倘若以传统的方式，即根据下游任务的数据集对已预训练好的模型的整个或部分进行重新训练过微调，可能会波坏学好的表征空间。这表现出研究预训练模型的在下游任务的必要性。我们的方法 CAPG 以及 CoCoOp 为 CoOp 静态的 prompt 泛化问题提供了及时的见解。并通过实验证明，我们的模型 CAPG 与 CoCoOp 类似，即适应输入实例的动态调整 prompt 的方法，在同一数据基类到新类的泛化性中有着优良的表现。

然而，我们的方法存在一些不足之处。一方面笔者提出的 CAPG 模型中，词库的构建仅采用了 CLIP 模板工程中 ImageNet 的常用模板，未来可考虑针对每个数据搭建配套的词库，扩建词库容量，可能更有效地提高模型在未见类的泛化性。另一方面，CAPG 的表现与 CoCoOp 不相上下，几乎没有什么提高。因此在未来可进一步研究更有效的动态适应实例的提示学习方法，进一步增强模型的泛化能力。

参考文献

[1] JIA C, YANG Y, XIA Y, et al. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision[J]. arXiv: Computer Vision and Pattern Recognition, 2021.

[2] RADFORD A, KIM J W, HALLACY C, et al. Learning Transferable Visual Models From Natural Language Supervision[J]. arXiv: Computer Vision and Pattern Recognition, 2021.

[3] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All you Need[J]. Neural Information Processing Systems, 2017.

[4] LIU P, YUAN W, FU J, et al. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.[J]. ACM Computing Surveys, 2021.

[5] GAO P, GENG S, ZHANG R, et al. CLIP-Adapter: Better Vision-Language Models with Feature Adapters[C]//. 2023.

[6] YAO Y, ZHANG A, ZHANG Z, et al. CPT: Colorful Prompt Tuning for Pre-trained Vision-Language Models[C]//. 2023.

[7] ZHOU K, YANG J, LOY C C, et al. Learning to Prompt for Vision-Language Models[J]. International Journal of Computer Vision, 2021.

[8] ZHOU K, YANG J, LOY C, et al. Conditional Prompt Learning for Vision-Language Models[C]//.

[9] ELHOSEINY M, SALEH B, ELGAMMAL A. Write a Classifier: Zero-Shot Learning Using Purely Textual Descriptions[J]. International Conference on Computer Vision, 2013.

[10] SOCHER R, GANJOO M, MANNING C D, et al. Zero-Shot Learning Through Cross-Modal Transfer [J]. Neural Information Processing Systems, 2013.

[11] FROME A, CORRADO G S, SHLENS J, et al. DeViSE: A Deep Visual-Semantic Embedding Model [J]. Neural Information Processing Systems, 2013.

[12] GOMEZ L, PATEL Y, RUSIÑOL M, et al. Self-supervised learning of visual features through embedding images into text topic spaces[J]. Cornell University - arXiv, 2017.

[13] LI A, JABRI A, JOULIN A, et al. Learning Visual N-Grams from Web Data[J]. Cornell University - arXiv, 2016.

[14] DESAI K, JOHNSON J. VirTex: Learning Visual Representations from Textual Annotations[J]. Computer Vision and Pattern Recognition, 2021.

[15] CHEN T, KORNBLITH S, NOROUZI M, et al. A Simple Framework for Contrastive Learning of Visual Representations[J]. arXiv: Learning, 2020.

[16] JIANG Z, XU F F, ARAKI J, et al. How Can We Know What Language Models Know[J]. arXiv: Computation and Language, 2019.

[17] LESTER B, AL-RFOU R, CONSTANT N. The Power of Scale for Parameter-Efficient Prompt Tuning [C]//. 2021.

[18] HE K, ZHANG X, REN S, et al. Deep Residual Learning for Image Recognition[J]. Cornell University - arXiv, 2015.

[19] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[J]. arXiv: Computer Vision and Pattern Recognition, 2020.

[20] FEI-FEI L, FERGUS R, PERONA P. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories[J]. Computer Vision and Pattern Recognition, 2004.

[21] PARKHI O M, VEDALDI A, ZISSERMAN A, et al. Cats and dogs[J]. Computer Vision and Pattern Recognition, 2012.

[22] KRAUSE J, STARK M, DENG J, et al. 3D Object Representations for Fine-Grained Categorization[J]. International Conference on Computer Vision, 2013.

[23] BOSSARD L, GUILLAUMIN M, GOOL L V. Food-101 – Mining Discriminative Components with Random Forests[J]. Lecture Notes in Computer Science, 2014.