arXiv:2303.04751v2 [cs.CV] 8 Jan 2024

# 多模态参数高效少样本类增量学习

Marco D'Alessandro[1*] Alberto Alonso[1*]
EnriqueCalabrés[1,2] MikelGalar[2]

[1]Neuraptic AI [2]纳瓦拉公立大学
{marco.dalessandro, alberto.alonso, enrique.hernandez}@neuraptic.ai
mikel.galar@unavarra.es

## 摘要

少样本类增量学习（*FSCIL*）是一项具有挑战性的持续学习任务，在多个学习会话中可用的训练样本有限。要成功完成这项任务，必须避免少样本训练集中的偏置分布导致新类过拟合。解决这一问题的常见方法是，通过添加特殊模块来增强预定义骨干架构的表征能力，从而实现与旧类别的向后兼容性。然而，这种方法尚未解决这一难题：如何在降低较大训练集与较小训练集上的性能差距的同时，确保分类准确率随时间推移保持较高水平。本文提出一种替代方法，称为持续参数高效 *CLIP*（*CPE-CLIP*），以减少不同学习会话之间的信息损失。我们不通过调整额外模块来解决信息损失，而是利用 *CLIP* 在大规模预训练中获得的丰富知识及其对新概念的泛化能力。我们的方法兼具多模态与参数高效的特点，通过在语言和视觉编码器中使用可学习提示，实现跨会话迁移学习。我们还引入提示正则化，以提升性能并防止遗忘。实验结果表明，与当前最先进的方法相比，*CPE-CLIP* 显著提升了 *FSCIL* 性能，同时大幅减少了可学习参数数量和训练成本。

## 1. 引言

在动态环境中部署机器学习系统，需要考虑随时间到来的连续数据流。该环境可能会出现数据分布偏移，或新增类别。理想的学习系统必须能够学习新到来的类别，同时保持对先前已学习类别的区分能力，从而避免灾难性遗忘 [31]。这种持续学习问题形式化称为类增量学习（CIL），它需要应对稳定性与可塑性之间的两难问题 [32, 14], 即，也就是学习新类别与保留旧类别之间的权衡。本文聚焦于 CIL 的一种特殊情形，即少样本类增量学习（FSCIL, [44]），在每个学习会话中只有少量训练样本可用。此时，额外的挑战在于避免因少样本训练集中的偏置分布而对新到来的类别过拟合。这个问题在数据可用性有限的实际场景中尤为关键，例如制造场景 [59, 2]和医学影像 [18]。在制造业中，机器人被部署来执行各种任务，例如装配或抓取物体。为了完成这些任务，机器人可能需要适应新的物体或材料，而这些物体或材料可用的训练数据可能有限。在医学影像领域，由于数据收集成本高昂且涉及患者隐私，数据的可用性也可能有限，因此难以随着时间推移获取新的知识。

近期研究主要通过多种方法解决这些问题，例如元学习 [57, 34],正则化技术 [30], 或知识蒸馏 [38, 6, 62]。这些方法在数据有限的情况下，随着时间推移实现增量学习方面展现出良好前景。一般方法是在预定义的骨干架构上添加特殊模块，以增强其基础表征能力，从而在学习会话期间与旧类别保持向后兼容性。这些解决方案的计算成本较高，因为每个会话都需要大量迭代，才能使额外模块适应新类，同时保持向后兼容性。尽管计算成本高昂，它们仍无法有效缩小随着时间推移在较大训练集上获得的性能与在较小训练集上获得的性能之间的差距，

*贡献相同。

而这一难题至今仍未得到解决 [67, 25]。

在本研究中，我们提出持续参数高效 CLIP（CPE-CLIP），作为一种替代方案，以减少不同学习会话之间的信息损失。受零样本设置 [45], 下出色的持续学习表现启发，我们以 CLIP [36] 为起点，构建用于少样本类增量学习的持续学习系统。我们不再依赖适配额外模块来应对信息损失，而是提出通过轻量级可学习提示调整 CLIP 架构，以实现少样本图像分类。这样，我们便能充分利用 CLIP 在大规模预训练中获取的海量知识，以及其在泛化至新概念方面的固有优势。值得注意的是，这是一种多模态且参数高效的方法，因为它依靠可学习提示，而非对语言编码器和视觉编码器进行微调，从而实现跨会话的持续迁移学习。我们证明，与当前最先进的方法相比，我们的方法显著提升了少样本类增量学习的表现，同时大幅减少了可学习参数的数量和训练成本。我们还进行了广泛的超参数调优和消融研究，以了解模型多个组件的功能特性。我们的主要贡献总结如下:

- 我们提出一种基于提示的方法，通过减少遗忘并支持随时间进行知识迁移，将 CLIP 架构适配于少样本设置下的持续学习任务，同时仅需学习不到总参数量的 0.3%。
- 我们将两种不同的提示附加方法与提示正则化相结合，从而平稳地过渡到未来任务，同时保持性能稳定。
- 我们在三个广受欢迎的少样本类增量学习基准数据集上取得了最先进的性能，并大幅超越此前的最先进结果。

本文其余部分安排如下：第 2 节讨论与本文方法及小样本类增量学习相关的工作。第 3 节介绍问题形式化。第 4 节阐述所提出的方法。此外，第 5 节和第 6 节分别呈现实验结果和最终讨论。

## 2. 相关工作

我们针对少样本持续学习问题的方法与若干主题相关，因此我们将分别介绍这些主题。

### 2.1. 少样本图像分类

少样本图像分类旨在利用数量不足的训练样本拟合新的、未见过的类别[5, 50]。为此，人们提出了多种学习方法。例如，在基于度量的方法中，会构建不同的网络分支，通过计算测试集中的查询图像与少样本训练集中的训练图像之间的距离来对图像进行分类 [42, 48, 43]。与此不同，在元学习中，模型会在各种学习任务上进行训练，从而仅凭少量训练样本就能解决新的学习任务 [11, 10, 57, 34]。一种截然不同且较新的思路是依靠预训练的多模态视觉语言模型，根据带标签的描述文本，仅用少量训练样本对图像进行分类 [65, 64, 19]。在这种方法中，小样本学习依赖于查询图像与描述类别标签的文本说明之间的正确匹配。我们的方法可视为对后一种方法进行持续学习改造。

### 2.2. 增量学习

增量学习旨在解决从非平稳数据流中学习新信息的问题[46, 28, 35]。根据任务标识符（ID）是否随时间可用，问题形式化可分为任务增量学习或类增量学习。文献中提出了多种解决方案，试图在学习新任务传入信息的同时，尽可能减少对先前已获得知识的遗忘。在基于正则化的方法中，会对已学习任务的特定参数进行正则化，以保留从先前任务中获得的知识并避免灾难性遗忘[20, 24, 60]。基于架构的方法会为每个任务分配骨干网络中一个独立的部分，或额外分支中的一组独立参数 [40, 58, 29, 51, 9]。在基于回放的方法中，先前已学习任务的数据会存储在回放缓冲区中，并与当前训练集一同用于当前任务的学习 [3, 4, 33]。一种较新的、基于提示且无需回放的方法，将强大的预训练骨干网络与可学习提示相结合，在不修改主干骨干网络权重的情况下保留从不同任务中获得的知识，从而避免遗忘[52, 53, 41, 37]。我们的方法借鉴了后者提出的解决方案。

### 2.3. 小样本类增量学习

少样本类增量学习（FSCIL）是近期提出的一个研究方向，旨在解决类增量场景中的少样本训练输入问题 [44]，其中评估时不提供任务 ID。其一般任务是先从若干基类中学习，然后持续更新模型，以表示不断到来的新类。这种设置的主要挑战是避免对新类的少样本样本过拟合。为解决这一问题，最早的尝试提出了神经气体（NG）网络

用于表示知识 [44], , 通过学习不同类别的特征空间拓扑结构，并通过扩展和调整网络拓扑来表示新类。在 [61], 中，作者将表示学习与分类器解耦，并通过图模型在分类器之间传播信息，使后者能够随时间更新。原型建模也被用于在嵌入空间中分配原型，以便为未来到来的新类预留空间 [62], ；或者使用新类嵌入表示的平均值作为类别原型来替代分类器 [63]。不同方法通过使用变分自编码器（VAE）将特征合成为增量类别的子空间混合体 [7], , 或调整通用深度学习架构，使每一批新到来的类别仅需更新少量参数 [30]，以解决这一问题。较新的方法则尝试结合有监督模型和自监督模型产生的特征，以增强分类器 [1],；或者通过检索旧类别的分布并估计新类别的分布来校准分布，从而避免遗忘 [26]。

### 2.4. 参数高效学习

将基础通用大规模预训练模型适配到下游任务，最常见的方式是微调模型的所有参数，但这会带来较高的计算成本和内存占用，而且需要为不同任务存储多个微调后的模型副本。参数高效学习领域提出了一种轻量级替代方案：仅更新少量额外参数，同时冻结骨干网络参数 [12, 22]。基于这一思路，人们提出了多种方法，以灵活地将预训练骨干网络适配到不同的下游任务。适配器调优 [15, 16] 通过在 Transformer 层之间插入带有跳跃连接的前馈瓶颈模块，在将层的输出传递到下一层之前对其进行调整。前缀调优 [23, 52, 17] 将可调的前缀向量作为可学习嵌入，添加到 Transformer[47]中多头注意力层的键和值之前。提示调优 [22, 53], 则在第一层的输入嵌入前添加一组可学习嵌入，随后由冻结的 Transformer 层按常规方式处理扩展后的输入。

## 3. 问题形式化

少样本类增量学习设置 [44]可定义如下。我们考虑一个带标签训练集流 $D_0$，$D_1$，……，$D_T$，其中 $D_t = \{(\boldsymbol{x}_{i,t},\ y_{i,\ t})\}_{i=1}^{N_t^D}$，$N_t^D$表示在会话 $t$中提供的训练样本数量，$T$表示最后一个增量会话。 $D_0$ 表示基类的大规模训练集， $D_t$ 表示新类的少样本训练集，适用于 $t > 0$。基类数据集 $D_0$应包含足够数量的训练样本。相反，新类仅提供了数量不足的训练集。考虑属于训练集 $D_t$ 的类别标签集合 $C_t$ 。少样本类增量学习具有以下要求：(1) 各会话之间的类别不重叠，$\forall t,\ \tau、\ t \neq \tau、\ C_t \cap C_\tau = \varnothing$；(2) 基类集合大于新类集合，其中 $|C_0| > |C_t|$，且 $N_0^D > N_t^D$ 对 $t > 0$ 成立；(3) 新类集合大小相同，即 $\forall t,\ \tau、\ t \neq \tau、\ |C_t| = |C_\tau|$和 $N_t^D = N_\tau^D$对 $t,\ \tau > 0$成立。对于评估阶段，唯一的要求是：计算各会话的性能时，应考虑截至当前会话 $t$ 所遇到的全部类别。考虑带标签评估集流 $E_0$，$E_1$，……，$E_T$，以及模型 $f$，则会话 $t$ 的评估准确率可计算如下：

$$A_t = \frac{\sum_{(\boldsymbol{x}_i, y_i) \in E_0 \cup E_1 \cup \ldots E_t} [f(\boldsymbol{x}_i) = y_i]}{N_0^E + N_1^E + \ldots + N_t^E} \tag{1}$$

其中， $N_\tau^E$表示会话$\tau$的评估样本数量， $[\cdot]$ 表示指示函数。

## 4. 方法

我们提出的方法 CPE-CLIP 总结于图1中。该方法使用 CLIP 基础视觉-语言模型 [36] 作为持续学习系统的主要构建模块。CLIP 是一种神经网络，经过训练能够对齐语言和视觉两种模态，并利用自然语言提供的丰富监督信息来推理视觉概念。在我们的工作中，我们利用 CLIP 的能力，将图像分类视为多模态任务，其中，文本提示（例如“一张<类别>的照片”）用作文本编码器的查询描述，而将测试图像与描述进行匹配则作为分类标准。我们的方法学习能够适配语言和视觉编码器 [19]的提示。通过这种方式，我们冻结 CLIP骨干网络，在保留预训练期间获得的知识的同时，让提示解决持续学习任务。为进一步提升性能并避免遗忘，我们引入了提示正则化。

**语言编码器**。 在这里，我们学习适用于给定下游任务中所有类别的共享语言上下文提示。上下文提示有两个作用：（1）在假设已知图像类别标签的情况下，避免手动选择低效的提示 [65, 64]来提供图像的文本表示；（2）作为稳定性参数，学习会话任务之间共享的通用任务不变属性。接下来，我们引入 $L$可学习标记 $\boldsymbol{g}$，称为 G-Prompt（遵循 Wang等人的记法 [52]），使得 $\boldsymbol{\Theta}_{\boldsymbol{g}} \in \mathbb{R}^{L \times d^{\text{NLP}}}$，其中 $d^{\text{NLP}}$是 CLIP 语言编码器的嵌入维度。此时，输入嵌入的形式为 $[g_1, g_2, \ldots, g_L, w]$ =

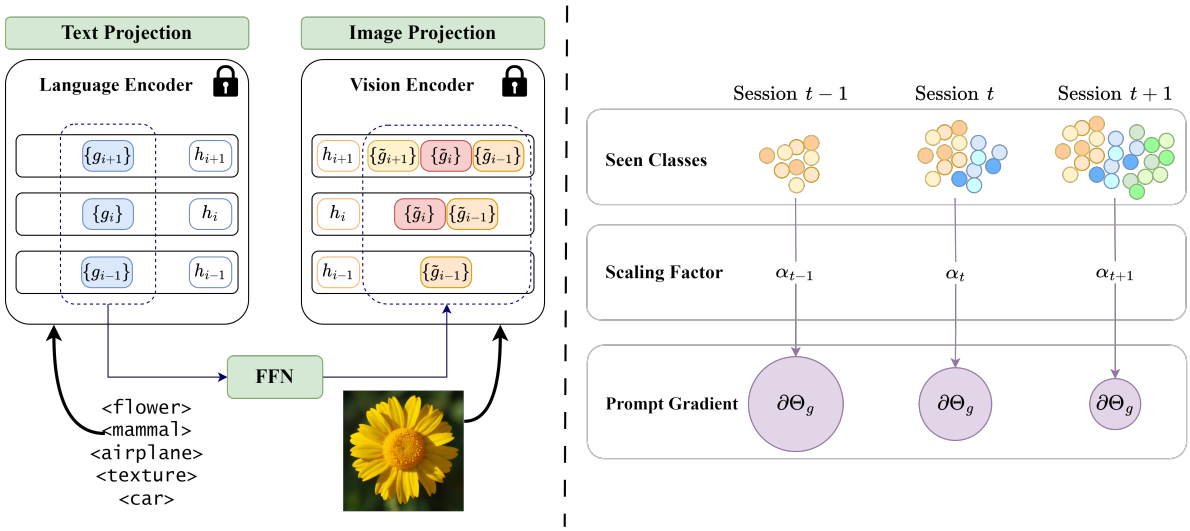


图 1. CPE-CLIP架构与训练过程概述。左图展示了 CPE-CLIP 的整体结构，其中 G-Prompt 有助于语言编码器泛化任务不变知识，随后将其投影到视觉编码器。视觉提示会在后续层中累积，而语言编码器则采用替换策略。右图展示了正则化过程：对于给定的会话 $t$，已见类别数量的增加会通过缩放因子 $\alpha_t$降低参数梯度。

$[\boldsymbol{g}, w]$，而 $w$ 是输入图像的类别名称嵌入。我们将 $f_i^{\text{NLP}}$定义为语言编码器的第 $i^{th}$ 个 Transformer 层，其中 $i = 1, 2, \ldots\ldots, K$，且 $K$ 为层总数。我们考虑在每个语言编码器层中进一步引入新的可学习标记的情况，即 $f_i^{\text{NLP}}$，直到某个深度 $D$。这样，不同层可以独立使用不同的提示，以适应不同的抽象层次。前向传播可描述如下：

$$[_, h_1] = f_1^{\text{NLP}}([\boldsymbol{g}_1, w]) \tag{2}$$

对于第一层，以及

$$[_, h_i] = f_i^{\text{NLP}}([\boldsymbol{g}_{i-1}, h_{i-1}]) \qquad i = 2, 3, 4, \ldots, D \tag{3}$$

对于后续层，直至第 $D$ 层。这里，$[\cdot, \cdot]$ 表示垂直拼接，$h_i$ 是与类别词嵌入相关联的第 $i$ 层输出的潜在状态，而$\boldsymbol{g}_i$ 表示第 $i$ 层的可学习提示集合。在此阶段，输入提示 $\boldsymbol{g}_i$ 的输出嵌入会被丢弃，不传递到下一层。值得注意的是，引入特定于层的提示时，我们有 $\boldsymbol{\Theta_g} \in \mathbb{R}^{D \times L \times d^{\text{NLP}}}$。经过 $D^{th}$ 个Transformer层后，后续层会以标准方式处理前一层的输出，直到得到最终的文本表示：

$$[\boldsymbol{g}_i, h_i] = f_i^{\text{NLP}}([\boldsymbol{g}_{i-1}, h_{i-1}]) \qquad i = D+1, \ldots, K. \tag{4}$$

为简便起见，我们将 [EOS] 标记对应的最后隐藏状态记为 $h^{\text{NLP}}$；在 CLIP 语言编码器中，该状态用于表示整个句子。随后将该隐藏状态投影到较低维空间：

$$h_*^{NLP} = p^{\text{NLP}}(h^{\text{NLP}}), \tag{5}$$

其中 $p^{\text{NLP}}$ 是一个线性投影层，$h_*^{\text{NLP}}$是文本表示的最终低维向量。

**视觉编码器**。与语言编码器类似，我们为视觉分支设计了 G-Prompt。即便如此，我们仍会在层级结构的多个层中，将提示与图像块嵌入拼接起来，以便与较低层和较高层的图像特征处理过程交互。我们引入 $L$ 个标记 $\tilde{\boldsymbol{g}}$，使得 $\tilde{\boldsymbol{g}} \in \mathbb{R}^{L \times d^{\text{CV}}}$，其中 $d^{\text{CV}}$是 CLIP视觉编码器的嵌入维度，且$d^{\text{CV}} > d^{\text{NLP}}$。此时，输入嵌入的形式为 $[c_1, c_2, \ldots, c_J, \tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_L] = [\boldsymbol{c}, \tilde{\boldsymbol{g}}]$，其中 $\boldsymbol{c}$是输入图像的嵌入图块集合加上额外的[CLS] 标记，而 $J$ 是嵌入总数。现在将 $f_i^{\text{CV}}$定义为视觉编码器的第 $i^{th}$ 个 Transformer层，其中 $i = 1, 2, \ldots\ldots, K$。与 Khattak 等人 [19], 类似，我们通过明确将视觉提示表示为语言提示的函数，弥合语言提示与视觉提示之间的差距。我们引入一个可学习的线性投影 $f^{\text{PROJ}}$，$\boldsymbol{\Theta}_{f^{\text{PROJ}}} \in \mathbb{R}^{d^{\text{NLP}} \times d^{\text{CV}}}$，并约束视觉任务不变提示以语言 G-Prompt 为条件，即$\tilde{\boldsymbol{g}}_i = f^{\text{PROJ}}(\boldsymbol{g}_i)$，其中 $\tilde{\boldsymbol{g}}_i$ 是视觉编码器各层的提示集合

$i$。尽管视觉分支中的提示源自语言上下文提示，但我们发现，采用不同策略在层级结构中传播提示效果更好。这里，我们提出一种累积方法，以替代语言分支中使用的替换方法。在这种方法中，不同层的提示不再彼此独立，因为它们可以与前几层提示经过处理后的输出嵌入进行交互。通常，提示累积会持续至深度 $D$。形式化地，我们将视觉编码器中的前向传播描述如下：

$$[h_1, \tilde{\boldsymbol{g}}_1] = f_1^{\text{CV}}([\boldsymbol{c}, \tilde{\boldsymbol{g}}_1]) \tag{6}$$

对于第一层，以及

$$\begin{aligned} \tilde{\boldsymbol{g}}_i &= [\tilde{\boldsymbol{g}}_i, \tilde{\boldsymbol{g}}_{i-1}] \\ [h_i, \tilde{\boldsymbol{g}}_i] &= f_i^{\text{CV}}([\boldsymbol{c}, \tilde{\boldsymbol{g}}_i]) \end{aligned} \qquad i = 2, 3, 4, \ldots, D \tag{7}$$

对于后续层，直到第 $D$ 层。在经过 $D^{th}$ Transformer层后，不再累积提示，后续层将以标准方式处理前一层的输出，直至得到最终图像表示：

$$[\tilde{\boldsymbol{g}}_i, h_i] = f_i^{\text{CV}}([\tilde{\boldsymbol{g}}_{i-1}, h_{i-1}]) \qquad i = D+1, \ldots, K \tag{8}$$

其中， $\tilde{\boldsymbol{g}}_i$ 现在是最终池化提示，使得 $\tilde{\boldsymbol{g}}_i \in \mathbb{R}^{LD \times d^{\text{CV}}}$。我们将与[CLS] 标记相关的最后隐藏状态称为 $h^{\text{CV}}$，在 CLIP视觉编码器中，该状态用于表示整幅图像。随后将该隐藏状态投影到一个更低维的空间：

$$h_*^{\text{CV}} = p^{\text{CV}}(h^{\text{CV}}) \tag{9}$$

其中， $p^{\text{CV}}$ 是一个线性投影层， $h_*^{\text{CV}}$是图像表征的最终低维向量。

**多模态分类**。对于给定的每个输入图像 $x$，其属于类别 $z$、 $z = 1$，2，.....， $Z$ 的预测概率计算如下：

$$p(y = z|x) = \frac{\exp[\rho(h_*^{\text{CV}}, h_{*z}^{\text{NLP}})]}{\sum_{j=1}^{Z} \exp[\rho(h_*^{\text{CV}}, h_{*j}^{\text{NLP}})]} \tag{10}$$

其中， $\rho$ 是余弦相似度， $h_*^{\text{CV}}$是图像 $x$的投影表示， $h_{*z}^{\text{NLP}}$是训练集中类别名称为 $z^{th}$ 的类别对应句子的投影表示。

**提示正则化**。在少样本类增量学习中，基类训练对于初步调整网络至关重要，有助于提升模型对新类别的泛化能力。在我们的方案中，G-Prompt会引入一组用于实现这一目标的词元。这些词元能够提供高效的文本表示，可与图像进行匹配，从而将图像正确分类到相应的（语义）类别/标签中。然而，与会话相关类别集合相比，基类集合提供了更大的泛化机会，因为它包含更多类别和训练样本。为此，我们提出一种机制，根据不同会话中遇到的类别数量，按比例保留知识。我们为给定会话 $t$、 $t = 1$，2、.....、$T$定义一个缩放因子 $\alpha_t$，该因子会影响在会话 $t$ 上训练时G-Prompt参数的更新率：

$$\alpha_t = \frac{|C_t|}{\sum_{\tau=0}^{t} |C_\tau|}. \tag{11}$$

随后，我们按如下方式进行正则化：

$$\frac{\partial \mathcal{L}_t}{\partial \boldsymbol{\Theta}_{\boldsymbol{g}}} \leftarrow \alpha_t \frac{\partial \mathcal{L}_t}{\partial \boldsymbol{\Theta}_{\boldsymbol{g}}} \tag{12}$$

and

$$\frac{\partial \mathcal{L}_t}{\partial \boldsymbol{\Theta}_{f^{\text{PROJ}}}} \leftarrow \alpha_t \frac{\partial \mathcal{L}_t}{\partial \boldsymbol{\Theta}_{f^{\text{PROJ}}}} \tag{13}$$

其中， $\mathcal{L}_t$ 是会话 $t$中的分类任务损失函数。这种正则化使得随着已见类别总数的增加，G-Prompt 以及语言-视觉提示投影的更新频率降低。

## 5. 实验

我们在三个基准数据集 [44]上评估 CPE-CLIP，这些数据集是少样本类增量学习文献中模型比较的主要基线。基准数据集包括 CIFAR100 [21],迷你ImageNet [39],和 CUB200-2011 [49]。

### 5.1. 评估基准数据集

对于所有基准数据集，我们均采用 Tao 等人提出的划分方式。 [44] 这是因为该划分方式为文献中的所有方案提供了标准，并确保模型比较公平。基准数据集介绍如下：

- **CIFAR100** 该数据集包含来自 100 个类别的 60.000 $32 \times 32$ RGB 图像。我们将其中 60 个类别用作基类集合。其余 40 个类别被划分为 8 个会话，每个会话包含 5 个新类，且少样本训练集中的每个类别包含 5 个样本（5-way 5-shot 增量任务）。
- 迷你**ImageNet** 该数据集包含 60.000 $84 \times 84$ 张 RGB 图像。我们将 100 个类别用作基类集合。其余 40 个类别被划分为 8 个会话，每个会话包含 5 个少样本训练示例（5类5样本增量任务）。

**•加州理工学院-加州大学圣迭戈分校鸟类-200-2011 (CUB200)** 该数据集包含来自 200 个鸟类物种类别的 11.788 张细粒度 $224 \times 224$ RGB 图像。我们将 100 个类别用作基类集合。其余 100 个类别被划分为 10 个会话（或时间戳），每个会话包含 10 个新类；少样本训练集中的每个类别包含 5 个样本（10类5样本增量任务）。

### 5.2. 实现细节

我们使用 HuggingFace 的 Transformers库中的 $16 \times 16$ patches OpenAI CLIP [36] 版本作为初始骨干网络。借助 Avalanche库 [27]，我们在 PyTorch 中构建模型和流水线。对于所有基准数据集，我们使用带动量的 SGD优化器，学习率设为 0.00325，权重衰减设为 $1e^{-5}$，并采用带预热的余弦退火。基类训练中，CIFAR100 和 *mini*ImageNet 的批量大小设为 32，训练轮数设为 3；CUB200 的批量大小设为 4，训练轮数设为 6。对于新类别会话的训练集，所有基准数据集的批量大小均设为 4，训练轮数设为 5。所有实验均在一台 GeForce RTX 2080 Ti 上进行。为了比较模型，我们报告基类以及每个会话的 Top-1评估准确率，这是 FS-CIL 中的标准做法。我们还报告准确率下降率（PD）指标，该指标衡量最后一个会话的准确率相较于基类会话准确率的下降幅度，以此作为遗忘程度的度量；同时报告跨会话平均准确率，以衡量整体性能。

### 5.3. 超参数调优

我们通过调整影响 CPE-CLIP 在少样本类增量学习中整体表现的两个超参数，进行超参数调优，以选出最佳候选模型。我们采用网格搜索，在穷举 $2 \times 5$ 搜索中探索以下取值范围： $L = [2, 4]$,$D = [1, 3, 6, 9, 12]$。对于每种配置，我们使用随机参数初始化进行 5 次运行，并以各会话准确率的平均值作为模型选择指标。鉴于穷举超参数搜索的计算负担，我们仅聚焦于 CUB200 基准，因为与鸟类物种相关的细粒度图像带有专业且不常见的文本标签，对我们基于 CLIP 的方法构成了特殊挑战。结果见表1

| | $D$ | | | | |
|---|---|---|---|---|---|
| $L$ | 1 | 3 | 6 | 9 | 12 |
| 2 | 67.48 (0.25) | 68.63 (0.38) | 68.54 (0.38) | 69.49 (0.14) | **70.23** (0.31) |
| 4 | 68.03 (0.26) | 68.35 (0.07) | 69.61 (0.45) | 69.11 (0.36) | 69.28 (0.28) |

表1。 $L$ 和 $D$的超参数调优。报告了每种配置的平均跨会话准确率和标准误（括号内）。

最佳模型对应的超参数集为 $L = 2$,$D = 12$。因此，我们采用此配置进行模型比较。

### 5.4. 与最先进模型的比较

在本节中，我们展示了在 CIFAR100、*mini*ImageNet 和 CUB200 基准数据集上的主要结果，分别如表2、表3和表4所示，其中 CPE-CLIP 与最新的最先进FSCIL方法 [66, 1, 61, 55,56, 67, 62, 63]进行了比较。对于最新的最先进方法而言性能大幅落后的模型，未纳入模型比较研究。

根据这些结果，我们的模型大幅优于当前最先进的模型。如 PD 指标所示，CPE-CLIP 在基类会话中取得了最高的分类准确率，并能更有效地减少遗忘，同时长期保持稳定且较高的分类性能。从在较大数据集（如基类集合）上训练，转向在学习会话中使用较小数据集时，CPE-CLIP 在减少信息损失方面展现出更强的能力。值得一提的是，当前模型对比中纳入的其他方法主要依赖 ResNet [13] 和 ViT [8] 作为主要骨干网络，且这些网络仅针对 CUB200 基准进行了预训练。此外，与 ResNet 和 ViT 不同，CPE-CLIP 使用的 CLIP 并未直接在 ImageNet [39],等常见基础数据集上进行预训练。

我们还比较了研究中各种模型的训练时间和可学习参数数量。该比较旨在了解完成整个学习会话序列所需的计算成本。我们仅纳入了能够保证可复现性且提供超参数的模型。该比较结果见表 5。总体而言，我们的研究结果表明，CPE-CLIP 在不牺牲性能的情况下显著降低了计算成本。

### 5.5. 消融实验

这里我们分析 CPE-CLIP 中相关组件的重要性。为简洁起见，我们仅采用 CUB200 基准。具体而言，我们重点考察了三种消融模

| 方法 | 各会话准确率（%） | | | | | | | | | 平均值 ↑ | 性能下降 ↓ | Δ 平均值 | Δ PD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | | | |
| CEC † [61] | 73.07 | 68.88 | 65.26 | 61.19 | 58.09 | 55.57 | 53.22 | 51.34 | 49.14 | 59.53 | 23.93 | +**23.89** | +**16.62** |
| SPPR ‡ [66] | 76.33 | 72.33 | 67.33 | 63.33 | 59.00 | 55.33 | 53.00 | 50.33 | 47.33 | 60.47 | 29.00 | +**22.95** | +**21.69** |
| CLOM † [67] | 74.2 | 69.83 | 66.17 | 62.39 | 59.26 | 56.48 | 54.36 | 52.16 | 50.25 | 60.57 | 23.95 | +**22.85** | +**16.64** |
| FeSSSS † [1] | 75.35 | 70.81 | 66.70 | 62.73 | 59.62 | 56.45 | 54.33 | 52.10 | 50.23 | 60.92 | 25.12 | +**22.50** | +**17.81** |
| MFS3 † [55] | 73.42 | 69.85 | 66.44 | 62.81 | 59.78 | 56.94 | 55.04 | 53.00 | 51.07 | 60.93 | 22.35 | +**22.49** | +**15.04** |
| PC † [56] | 76.30 | 71.89 | 67.70 | 63.40 | 60.21 | 57.31 | 55.01 | 52.79 | 50.74 | 61.71 | 25.56 | +**21.71** | +**18.25** |
| LIMIT † [63] | 73.81 | 72.09 | 67.87 | 63.89 | 60.70 | 57.77 | 55.67 | 53.52 | 51.23 | 61.84 | 22.58 | +**21.58** | +**15.27** |
| FACT † [62] | 74.60 | 72.09 | 67.56 | 63.52 | 61.38 | 58.36 | 56.28 | 54.24 | 52.10 | 62.24 | 22.50 | +**21.18** | +**15.19** |
| CLIP 零样本 | 74.45 | 72.83 | 72.11 | 70.25 | 69.71 | 69.55 | 69.52 | 68.78 | 68.04 | 70.58 | 6.40 | +12.84 | −0.91 |
| CPE-CLIP | **87.83** | **85.86** | **84.93** | **82.85** | **82.64** | **82.42** | **82.27** | **81.44** | **80.52** | **83.42** | **7.31** | | |

表2。CIFAR100基准测试。 ΔPD表示与其他模型相比性能下降方面的改进。 Δ 平均值表示与其他模型相比跨会话平均准确率的提升。 † 表示取自相应论文的结果， ‡ 表示由于相应论文未提供表格结果，故根据论文图表估算的结果。

| 方法 | 各会话准确率（%） | | | | | | | | | 平均值 ↑ | 性能下降 ↓ | Δ 平均值 | Δ PD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | | | |
| CEC † [61] | 72.00 | 66.83 | 62.97 | 59.43 | 56.70 | 53.73 | 51.19 | 49.24 | 47.63 | 57.74 | 24.37 | +**28.39** | +**16.91** |
| CLOM † [67] | 73.08 | 68.09 | 64.16 | 60.41 | 57.41 | 54.29 | 51.54 | 49.37 | 48.00 | 58.52 | 25.08 | +**27.61** | +**17.62** |
| LIMIT † [63] | 72.32 | 68.47 | 64.30 | 60.78 | 57.95 | 55.07 | 52.70 | 50.72 | 49.19 | 59.05 | 23.13 | +**27.08** | +**15.67** |
| PC † [56] | 73.20 | 68.35 | 64.06 | 60.85 | 58.00 | 54.98 | 52.82 | 51.17 | 50.16 | 59.28 | 23.04 | +**26.85** | +**15.58** |
| MFS3 † [55] | 73.65 | 68.91 | 64.60 | 61.48 | 58.68 | 55.55 | 53.33 | 51.69 | 50.26 | 59.79 | 23.39 | +**26.34** | +**15.93** |
| FACT † [62] | 72.56 | 69.63 | 66.38 | 62.77 | 60.60 | 57.33 | 54.34 | 52.16 | 50.49 | 60.69 | 22.07 | +**25.44** | +**14.61** |
| SPPR ‡ [66] | 80.00 | 74.00 | 68.66 | 64.33 | 61.00 | 57.33 | 54.66 | 51.66 | 49.00 | 62.29 | 31.00 | +**23.84** | +**23.54** |
| FeSSSS † [1] | 81.50 | 77.04 | 72.92 | 69.56 | 67.27 | 64.34 | 62.07 | 60.55 | 58.87 | 68.24 | 22.63 | +**17.89** | +**15.17** |
| CLIP 零样本 | 77.13 | 76.49 | 75.31 | 77.30 | 75.35 | 75.28 | 73.92 | 74.18 | 73.17 | 75.35 | 3.96 | +10.78 | −3.50 |
| CPE-CLIP | **90.23** | **89.56** | **87.42** | **86.80** | **86.51** | **85.08** | **83.43** | **83.38** | **82.77** | **86.13** | **7.46** | | |

表3. 迷你ImageNet 基准。 Δ PD 表示相较于其他模型在 PD 方面的提升。 Δ 平均值表示相较于其他模型在跨会话平均准确率方面的提升。 † 表示结果取自各自论文， ‡ 表示由于论文中未提供表格结果，因此根据相应论文中的图近似得出的结果。

型。首先，我们考虑视觉编码器不采用提示传播累积策略的情况。在这种情况下，与语言分支一样，采用标准替换策略。此外，我们通过完全移除视觉提示，考察将 G-Prompt 投影到视觉分支的作用。最后，我们考虑不进行正则化的情况，此时 G-Prompt 会在各个会话中持续更新。结果如图2所示。

主模型与其消融版本之间的比较揭示了一些值得注意的发现。具体而言，提示正则化似乎是确保性能随时间保持稳定的关键因素，因为它能够抵消问题会话中少样本训练示例的偏置分布所带来的影响。相反，移除视觉提示系统似乎并未产生显著对模型在各会话期间易发生遗忘和信息损失的影响。然而，与完整模型相比，该方法在各个会话中的表现通常较差，初始会话期间的下降尤为明显，之后在后续会话中逐渐趋于稳定。值得注意的是，将提示从视觉分支中完全移除，与采用提示传播替代累积策略相比，性能差异极小。总体而言，结果证实了我们的提示系统在持续学习设置中能够取得最佳性能。总之，提示正则化通过确保训练随时间推移保持稳定，减少了信息损失和遗忘；视觉编码器中的提示传播累积策略则能提供更好的图像表征，从而确保更好的文本-图像匹配，并在特定会话中取得更高的准确率。

| 方法 | 各会话准确率（%） | | | | | | | | | | | 平均值↑ | 性能下降↓ | Δ平均值 | ΔPD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | | | |
| SPPR † [66] | 68.68 | 61.85 | 57.43 | 52.68 | 50.19 | 46.88 | 44.65 | 43.07 | 40.17 | 39.63 | 37.33 | 49.32 | 31.35 | +**21.47** | +**14.37** |
| PC † [56] | 74.06 | 70.89 | 68.13 | 63.98 | 61.54 | 58.85 | 57.56 | 55.96 | 54.28 | 53.73 | 52.40 | 61.03 | 21.66 | +**9.76** | +**4.68** |
| CEC † [61] | 75.85 | 71.94 | 68.50 | 63.50 | 62.43 | 58.27 | 57.73 | 55.81 | 54.83 | 53.52 | 52.28 | 61.34 | 23.57 | +**9.45** | +**6.59** |
| MFS3 † [55] | 75.63 | 72.51 | 69.65 | 65.29 | 63.13 | 60.38 | 58.99 | 57.41 | 55.55 | 54.95 | 53.47 | 62.45 | 22.16 | +**8.34** | +**5.18** |
| FeSSSS † [1] | 79.60 | 73.46 | 70.32 | 66.38 | 63.97 | 59.63 | 58.19 | 57.56 | 55.01 | 54.31 | 52.98 | 62.85 | 26.62 | +**7.94** | +**9.64** |
| FACT † [62] | 75.90 | 73.23 | 70.84 | 66.13 | 65.56 | 62.15 | 61.74 | 59.83 | 58.41 | 57.89 | 56.94 | 64.42 | 18.96 | +**6.37** | +**1.98** |
| LIMIT † [63] | 75.89 | 73.55 | 71.99 | 68.14 | 67.42 | 63.61 | 62.40 | 61.35 | 59.91 | 58.66 | 57.41 | 65.50 | 18.48 | +**5.29** | +**1.50** |
| CLOM † [67] | 79.57 | 76.07 | 72.94 | 69.82 | 67.80 | 65.56 | 63.94 | 62.59 | 60.62 | 60.34 | 59.58 | 67.17 | 19.99 | +**3.62** | +**3.01** |
| CLIP 零样本 | 65.46 | 63.37 | 62.15 | 58.58 | 58.66 | 58.57 | 56.95 | 55.97 | 54.57 | 54.64 | 55.31 | 58.56 | 10.15 | +12.23 | −6.83 |
| CPE-CLIP | **81.58** | **78.52** | **76.68** | **71.86** | **71.52** | **70.23** | **67.66** | **66.52** | **65.09** | **64.47** | **64.60** | **70.79** | **16.98** | | |

表 4. CUB200 基准。 Δ PD 表示与其他模型相比的性能下降改善幅度。 Δ Avg. 表示与其他模型相比，跨会话平均准确率的提升幅度。 †表示结果取自各自对应的论文。

| | CIFAR100 | | 迷你ImageNet | | CUB200 | |
|---|---|---|---|---|---|---|
| 模型 | #参数 | 训练时间 | #参数 | 训练时间 | #参数 | 训练时间 |
| CEC [61] | 295K | 0.32 | 12.2M | 1.41 | 12.3M | 0.96 |
| FACT [62] | 280K | 1.48 | 11.2M | 7.30 | 11.3M | 1.41 |
| LIMIT [63] | 295K | 1.02 | 12.2M | 2.00 | 12.3M | 0.99 |
| CLOM [67] | 350K | 0.24 | 14M | 1.56 | 18.9M | 0.35 |
| CPE-CLIP | 400K | 0.69 | 400K | 0.65 | 400K | 0.27 |

表5. 训练时间（训练时间），以小时为单位，以及可学习参数的数量（参数）。结果通过模拟模型的开源训练方案获得。所有模拟均在同一台 GeForce RTX 2080 Ti 上进行。

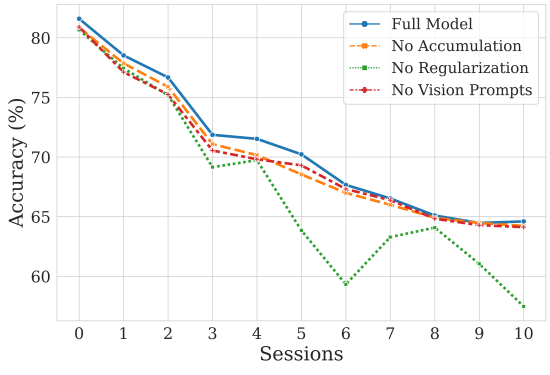


图2. 消融实验展示了主模型（完整模型）及三个消融版本在5次模拟中的 Top-1准确率：其中一个版本移除了视觉分支中的累积机制（无累积），一个版本不由视觉分支处理任何提示（无视觉提示），还有一个版本未应用正则化（无正则化）。会话0指基类。

## 6. 结论

受少样本图像分类和参数高效学习领域进展的启发，我们提出了一种新颖方案，用于解决具有挑战性的少样本类增量学习任务，其中每个会话仅有数量有限的标注数据。我们提出的 CPE-CLIP 有效结合了多种技术与现代理念，构建出一种能够长期保持高性能的多模态小样本持续学习器。我们证明，所提出的方法依靠少量参数和较低的总体计算成本，能够超越专为少样本类增量学习设计的其他方法。CPE-CLIP 引入了提示传播累积策略，似乎有助于增强图像表征，从而确保最佳分类准确率。提示正则化则通过减少随时间推移的信息损失，确保学习过程稳定。

**局限性**。 CPE-CLIP架构以CLIP框架为基础骨干网络。CLIP利用文本监督对视觉概念进行推理，这为少样本类增量学习带来了主要优势。然而，在处理缺少图像类别标签、无法直接由CLIP词汇表处理或本身存在歧义的任务时，这也会带来挑战，

从而导致图文匹配不可靠。此外，尚未探索正则化对更多会话的影响。随着类别数量增加，降低G-Prompt参数的更新率似乎对避免过拟合至关重要，但随着会话数增加，梯度的缩放因子会趋近于零。在这种情况下，若未能适当更新G-Prompt参数，当出现意外的分布偏移时，可能会损害模型对新类别的泛化能力。

## 参考文献


[1] Touqeer Ahmad、Akshay Raj Dhamija、Steve Cruz、Ryan Rabinowitz、Chunchun Li、Mohsen Jafarzadeh 和 Terrance E Boult。利用自监督特征进行小样本类增量学习。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第3900–3910页，2022年。[2] Ali Ayub 和 Alan R Wagner。告诉我这是什么：机器人通过少样本增量目标学习进行学习。载于*2020*年*IEEE/RSJ*智能机器人与系统国际会议（*IROS*），第8344–8350页。IEEE，2020年。[3] Pietro Buzzega、Matteo Boschini、Angelo Porrello、Davide Abati 和 Simone Calderara。用于通用持续学习的暗经验：一种强大而简单的基线。神经信息处理系统进展，33：15920–15930，2020年。[4] Hyuntak Cha、Jaeho Lee 和 Jinwoo Shin。Co2l：对比持续学习。载于*IEEE/CVF*计算机视觉国际会议论文集，第9516–9525页，2021年。[5] Wei-Yu Chen、Yen-Cheng Liu、Zsolt Kira、Yu-Chiang Frank Wang 和 Jia-Bin Huang。深入探究小样本分类。*arXiv*预印本 *arXiv:1904.04232*，2019年。[6] Ali Cheraghian、Shafin Rahman、Pengfei Fang、Soumava Kumar Roy、Lars Petersson 和 Mehrtash Harandi。面向小样本类增量学习的语义感知知识蒸馏。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第2534–2543页，2021年。[7] Ali Cheraghian、Shafin Rahman、Sameera Ramasinghe、Pengfei Fang、Christian Simon、Lars Petersson 和 Mehrtash Harandi。基于合成特征的子空间混合小样本类增量学习。载于*IEEE/CVF*计算机视觉国际会议论文集，第8661–8670页，2021年。[8] Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、Xiaohua Zhai、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Sylvain Gelly 等。一张图像胜过16×16个词：用于大规模图像识别的Transformer。*arXiv*预印本 *arXiv:2010.11929*，2020年。[9] Sayna Ebrahimi、Franziska Meier、Roberto Calandra、Trevor Darrell 和 Marcus Rohrbach。对抗持续学习。载于计算机视觉—*ECCV 2020*：第*16*届欧洲会议，英国格拉斯哥，*2020*年*8*月*23*日至*28*日，论文集，第*XI*卷（第*16*部分），第386–402页。施普林格，2020年。

[10] Thomas Elsken、Benedikt Staffler、Jan Hendrik Metzen 和 Frank Hutter。面向小样本学习的神经架构元学习。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第12365–12375页，2020 – 。[11] Chelsea Finn、Pieter Abbeel 和 Sergey Levine。用于深度网络快速适应的模型无关元学习。载于国际机器学习会议，第1126–1135页。PMLR，2017。[12] Junxian He、Chunting Zhou、Xuezhe Ma、Taylor Berg-Kirkpatrick 和 Graham Neubig。迈向参数高效迁移学习的统一视角。*arXiv*预印本 *arXiv:2110.04366*，2021。[13] 何恺明、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集，第770–778页，2016年。[14] Donald Olding Hebb。《行为的组织：一种神经心理学理论》。Psychology Press，2005年。[15] Neil Houlsby、Andrei Giurgiu、Stanislaw Jastrzebski、Bruna Morrone、Quentin De Laroussilhe、Andrea Gesmundo、Mona Attariyan 和 Sylvain Gelly。面向自然语言处理的参数高效迁移学习。载于国际机器学习会议，第2790–2799页。PMLR，2019年。[16] Neil Houlsby、Andreea Giurgiu、Stanislaw Jastrzebski、Bruna Morrone、Quentin De Laroussilhe、Andrea Gesmundo、Mona Attariyan 和 Sylvain Gelly。K-adapter：通过适配器将知识注入预训练模型。载于*2020*年自然语言处理实证方法会议：系统演示论文集，第56–61页，2020年。[17] Menglin Jia、Luming Tang、Bor-Chun Chen、Claire Cardie、Serge Belongie、Bharath Hariharan 和 Ser-Nam Lim。视觉提示调优，2022年7月。arXiv:2203.12119 [cs]。[18] Yifan Jiang、Han Chen、Hanseok Ko 和 David K Han。基于计算机断层扫描的小样本COVID-19诊断。载于*ICASSP 2021-2021 IEEE*国际声学、语音与信号处理会议（*ICASSP*），第1045–1049页。IEEE，2021年。[19] Muhammad Uzair Khattak、Hanoona Rasheed、Muhammad Maaz、Salman Khan 和 Fahad Shahbaz Khan。Maple：多模态提示学习。*arXiv*预印本 *arXiv:2210 03117*，2022。。[20] James Kirkpatrick、Razvan Pascanu、Neil Rabinowitz、Joel Veness、Guillaume Desjardins、Andrei A Rusu、Kieran Milan、John Quan、Tiago Ramalho、Agnieszka Grabska-Barwinska 等。克服神经网络中的灾难性遗忘。美国国家科学院院刊，114(13):3521–3526，2017年。[21] Alex Krizhevsky、Geoffrey Hinton 等。学习微小图像中的多层特征。2009年。[22] Brian Lester、Rami Al-Rfou 和 Noah Constant。规模对参数高效提示调优的作用。*arXiv*预印本 *arXiv:2104.08691*，2021年。[23] Xiang Li、Liunian Harold Li、Chi Li、Xiaodan Liang 和 Li Dong。前缀调优：优化用于生成的连续提示。载于*2021*年自然语言处理实证方法会议（*EMNLP*）论文集，第5016–5027页，2021年。

[24] Zhizhong Li 和 Derek Hoiem。无遗忘学习。*IEEE*模式分析与机器智能汇刊，40(12):2935–2947，2017年。[25] Bin Liu、Yue Cao、Yutong Lin、Qi Li、Zheng Zhang、Ming-sheng Long 和 Han Hu。负边际至关重要：理解小样本分类中的边际。载于计算机视觉—*ECCV 2020*：第*16*届欧洲会议，格拉斯哥，英国，*2020*年*8*月*23*日至*28*日，论文集，第四部分*16*，第438–455页。施普林格，2020年。[26] Binghao Liu、Boyu Yang、Lingxi Xie、Ren Wang、Qi Tian 和 Qixiang Ye。面向小样本类增量学习的可学习分布校准。*arXiv*预印本 *arXiv:2210 00232*，2022。。[27] Vincenzo Lomonaco、Lorenzo Pellegrini、Andrea Cossu、Antonio Carta、Gabriele Graffieti、Tyler L. Hayes、Matthias De Lange、Marc Masana、Jary Pomponi、Gido van de Ven、Martin Mundt、Qi She、Keiland Cooper、Jeremy Forest、Eden Belouadah、Simone Calderara、German I. Parisi、Fabio Cuzzolin、Andreas Tolias、Simone Scardapane、Luca Antiga、Subutai Amhad、Adrian Popescu、Christopher Kanan、Joost van de Weijer、Tinne Tuytelaars、Davide Bacciu 和 Davide Maltoni。Avalanche：用于持续学习的端到端库。载于*IEEE*计算机视觉与模式识别会议论文集，第2届计算机视觉持续学习研讨会，2021年。[28] Zheda Mai、Ruiwen Li、Jihwan Jeong、David Quispe、Hyun-woo Kim 和 Scott Sanner。图像分类中的在线持续学习：实证综述。神经计算，469:28–51，2022年。[29] Arun Mallya 和 Svetlana Lazebnik。PackNet：通过迭代剪枝将多个任务添加到单个网络中。载于*IEEE*计算机视觉与模式识别会议论文集，第7765–7773页，2018年。[30] Pratik Mazumder、Pravendra Singh 和 Piyush Rai。小样本终身学习。载于*AAAI*人工智能会议论文集，第35卷，第2337–2345页，2021年。[31] Michael McCloskey 和 Neal J Cohen。联结主义网络中的灾难性干扰：序列学习问题。载于学习与动机心理学，第24卷，第109–165页。爱思唯尔，1989年。[32] Martial Mermillod、Aurélia Bugaiska 和 Patrick Bonin。稳定性-可塑性困境：探究从灾难性遗忘到年龄限制性学习效应的连续谱，2013年。[33] Quang Pham、Chenghao Liu 和 Steven Hoi。DualNet：快与慢的持续学习。神经信息处理系统进展，34:16131–16144，2021年。[34] Marcin Przewi˛ezlikowski、Przemysław Przybysz、Jacek Tabor、M Zi˛eba 和 Przemysław Spurek。HyperMAML：借助超网络实现深度模型的小样本自适应。*arXiv*预印本 *arXiv:2205.15745*，2022年。[35] Haoxuan Qu、Hossein Rahmani、Li Xu、Bryan Williams 和 Jun Liu。计算机视觉持续学习的最新进展：综述。*arXiv*预印本 *arXiv:2109.11369*，2021年。[36] Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark 等。学习基于自然语言监督的可迁移视觉模型。在国际机器学习会议，第8748–8763页。PMLR，2021年。[37] Anastasia Razdaibiedina、Yuning Mao、Rui Hou、Madian Khabsa、Mike Lewis和Amjad Almahairi。渐进式提示：语言模型的持续学习。*arXiv*预印本 *arXiv:2301.12314*，2023年。[38] Sylvestre-Alvise Rebuffi、Alexander Kolesnikov、Georg Sperl和Christoph H Lampert。iCaRL：增量分类器与表征学习。在*IEEE*计算机视觉与模式识别会议论文集，第2001–2010页，2017年。[39] Olga Russakovsky、Jia Deng、Hao Su、Jonathan Krause、Sanjeev Satheesh、Sean Ma、Zhiheng Huang、Andrej Karpathy、Aditya Khosla、Michael Bernstein等。ImageNet大规模视觉识别挑战赛。国际计算机视觉杂志，115：211–252，2015年。[40] Andrei A Rusu、Neil C Rabinowitz、Guillaume Desjardins、Hubert Soyer、James Kirkpatrick、Koray Kavukcuoglu、Razvan Pascanu和Raia Hadsell。渐进式神经网络。*arXiv*预印本 *arXiv:1606.04671*，2016年。[41] James Seale Smith、Leonid Karlinsky、Vyshnavi Gutta、Paola Cascante-Bonilla、Donghyun Kim、Assaf Arbelle、Rameswar Panda、Rogerio Feris和Zsolt Kira。CODA-Prompt：用于无回放持续学习的持续分解式注意力提示。*arXiv*预印本 *arXiv:2211.13218*，2022。[42] Jake Snell、Kevin Swersky和Richard Zemel。面向小样本学习的原型网络。神经信息处理系统进展，30，2017年。[43] Flood Sung、Yongxin Yang、Li Zhang、Tao Xiang、Philip HS Torr和Timothy M Hospedales。学习比较：用于小样本学习的关系网络。在*IEEE*计算机视觉与模式识别会议论文集，第1199–1208页，2018年。[44] Xiaoyu Tao、Xiaopeng Hong、Xinyuan Chang、Songlin Dong、Xing Wei和Yihong Gong。小样本类增量学习。在*IEEE/CVF*计算机视觉与模式识别会议论文集，第12183–12192页，2020年。[45] Vishal Thengane、Salman Khan、Munawar Hayat和Fahad Khan。CLIP模型是一种高效的持续学习器。*arXiv*预印本 *arXiv:2210.03114*，2022。[46] Gido M van de Ven、Tinne Tuytelaars和Andreas S Tolias。三种增量学习类型。《自然·机器智能》，第1–13页，2022。[47] Ashish Vaswani、Noam Shazeer、Niki Parmar、J akob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser和Illia Polosukhin。Attention Is All You Need。神经信息处理系统进展，30，2017年。[48] Oriol Vinyals、Charles Blundell、Timothy Lillicrap、Daan Wierstra等。用于单次学习的匹配网络。神经信息处理系统进展，29，2016年。[49] Catherine Wah、Steve Branson、Peter Welinder、Pietro Perona和Serge Belongie。加州理工学院-加州大学圣迭戈分校鸟类200-2011数据集。2011年。[50] Yaqing Wang、Quanming Yao、James T Kwok和Lionel M Ni。从少量示例中泛化：小样本学习综述。*ACM*计算调查（*CSUR*），53(3)：1–34，2020年。

[51] Zifeng Wang、Tong Jian、Kaushik Chowdhury、Yanzhi Wang、Jennifer Dy和Stratis Ioannidis。面向终身学习的学习-剪枝-共享。在*2020*年 *IEEE*国际数据挖掘会议（*ICDM*），第641–650页。IEEE，2020年。[52] Zifeng Wang、Zizhao Zhang、Sayna Ebrahimi、Ruoxi Sun、Han Zhang、Chen-Yu Lee、Xiaoqi Ren、Guolong Su、Vincent Perot、Jennifer Dy等。DualPrompt：用于无回放持续学习的互补提示。在计算机视觉—*ECCV 2022*：第*17*届欧洲会议，特拉维夫，以色列，*2022*年*10*月*23*日至*27*日，会议论文集，第二十六卷，第631–648页。施普林格，2022。[53] Zifeng Wang、Zizhao Zhang、Chen-Yu Lee、Han Zhang、Ruoxi Sun、Xiaoqi Ren、Guolong Su、Vincent Perot、Jennifer Dy和Tomas Pfister。学习为持续学习设计提示。在*IEEE/CVF*计算机视觉与模式识别会议论文集，第139–149页，2022。[54] Thomas Wolf、Lysandre Debut、Victor Sanh、Julien Chaumond、Clement Delangue、Anthony Moi、Pierric Cistac、Tim Rault、Rémi Louf、Morgan Funtowicz、Joe Davison、Sam Shleifer、Patrick von Platen、Clara Ma、Yacine Jernite、Julien Plu、Canwen Xu、Teven Le Scao、Sylvain Gugger、Mariama Drame、Quentin Lhoest和Alexander M. Rush。Transformer：自然语言处理领域的最先进技术。在*2020*年自然语言处理实证方法会议：系统演示论文集，第38–45页，线上，2020年10月。计算语言学协会。[55] Xinlei Xu、Saisai Niu、Zhe Wang、Wei Guo、Lihong Jing和Hai Yang。面向小样本类增量学习的多特征空间相似性补充。知识型系统，第110394页，2023年。[56] Xinlei Xu、Zhe Wang、Zhiling Fu、Wei Guo、Ziqiu Chi和Dongdong Li。基于原型容器的灵活小样本类增量学习。神经计算与应用，第1–15页，2023年。[57] Li Yin、Juan M Perez-Rua和Kevin J Liang。Sylph：用于增量式少样本目标检测的超网络框架。在*IEEE/CVF*计算机视觉与模式识别会议论文集，第9035–9045页，2022年。[58] Jaehong Yoon、Eunho Yang、Jeongtae Lee和Sung Ju Hwang。利用动态可扩展网络进行终身学习。*arXiv*预印本 *arXiv:1708.01547*，2017年。[59] Tianhe Yu、Chelsea Finn、Annie Xie、Sudeep Dasari、Tianhao Zhang、Pieter Abbeel和Sergey Levine。通过领域自适应元学习，从观察人类行为中进行单次示范模仿。*arXiv*预印本 *arXiv:1802.01557*，2018年。[60] Friedemann Zenke、Ben Poole和Surya Ganguli。通过突触智能实现持续学习。在国际机器学习会议，第3987–3995页。PMLR，2017年。[61] Chi Zhang、Nan Song、Guosheng Lin、Yun Zheng、Pan Pan和Yinghui Xu。利用持续演化的分类器进行小样本增量学习。在*IEEE/CVF*计算机视觉与模式识别会议论文集，第12455–12464页，2021年。

[62] Da-Wei Zhou、Fu-Yun Wang、Han-Jia Ye、Liang Ma、Shil- iang Pu 和 De-Chuan Zhan。前向兼容的少样本类增量学习。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 9046–9056 页，2022。[63] Da-Wei Zhou、Han-Jia Ye、Liang Ma、Di Xie、Shiliang Pu 和 De-Chuan Zhan。通过多阶段任务采样实现少样本类增量学习。*IEEE*模式分析与机器智能汇刊，2022。[64] Kaiyang Zhou、Jingkang Yang、Chen Change Loy 和 Ziwei Liu。面向视觉语言模型的条件提示学习。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 16816–16825 页，2022。[65] Kaiyang Zhou、Jingkang Yang、Chen Change Loy 和 Ziwei Liu。学习为视觉语言模型设计提示。国际计算机视觉期刊，130 2348，2022 9 2337( ): – 。[66] Kai Zhu、Yang Cao、Wei Zhai、Jie Cheng 和 Zheng-Jun Zha。用于小样本类增量学习的自提升原型细化。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 6801–6810 页，2021。[67] Yixiong Zou、Shanghang Zhang、Yuhua Li 和 Ruixuan Li。采用类别级过拟合缓解的基于边际的少样本类增量学习。*arXiv*预印本 *arXiv:2210 04524*，2022。