基于分布外任务识别的图像分类类别增量学习

曹续生，卢浩日，刘夏雷，*Member, IEEE*，程明明，*Senior Member, IEEE*，

摘要—用于图像分类的类别增量学习（CIL）旨在通过允许模型在学习新类别时保留旧类别的知识，以应对现实世界场景。它比任务增量学习（TIL）更具挑战性，因为在测试时不会提供任务 ID。因此，从 CIL 转向 TIL 是处理图像分类中 CIL 问题的一种直观方法。目前，这种方法的主要挑战在于提高任务识别的准确性。为了解决这一问题，我们提出使用大规模图像-文本预训练模型（例如，CLIP）作为骨干网络，为不同任务训练并保存不同的分类器。每个分类器不仅包含当前任务的类别，还包含一个对应于所有先前任务中遇到的类别的分布外（OOD）类别。在测试时，我们从最后一个任务开始依次迭代分类器，以找到测试图像的正确任务 ID，并以 TIL 的方式进行分类。此外，为了应对由于模型对后续任务的偏好而导致迭代预测提前终止的问题，我们提出使用 CLIP 的零样本能力来辅助学习的 OOD 检测。实验表明，我们的方法在 CIFAR-100 和 ImageNet-Subset 数据集的传统多分类和更具挑战性的少分类设置中均取得了最先进的性能。

Index Terms—类别增量学习，图像分类，分布外检测，少样本学习，图像-文本预训练。

1 介绍

增量学习（IL）的目的是使深度模型适应现实世界的任务。在图像分类中遇到新类别时，深度模型应具备在学习新知识的同时保留先前所学知识的能力。图像分类中增量学习的主要挑战是应对灾难性遗忘 [1]，即模型在学习新任务时偏离其先前知识，导致在旧任务上的性能显著下降。

增量学习主要有三种场景 [2]：领域增量学习（DIL）、任务增量学习（TIL）和类别增量学习（CIL）。DIL 专注于学习具有不同领域分布的相同类别集合，例如相同类别的现实版本和仿真版本。模型可以学习跨领域的恒定信息，并在不同分布下进行预测。TIL 和 CIL 的区别在于测试时是否提供任务序号 (ID)。CIL 对应于测试时未提供任务 ID 的设置，这是一种更具挑战性的增量学习形式。因此，我们的研究主要解决这一问题。

现有的解决CIL的方法主要分为三类。基于重放的方法 [3]–[7] 会存储一部分先前的数据或生成一些合成样本用于重放，并将其与新数据一起用于模型训练，以防止遗忘。基于正则化的方法 [8]–[12] 在训练过程中对模型的参数空间或激活空间施加约束，以防止在学习新任务时发生灾难性遗忘。基于架构的方法 [13]–[18] 通过修改模型的架构来增加容量，例如添加神经元、模块或分支，以提高对新旧任务的性能。

解决CIL的另一种直观方法是通过首先识别任务 ID，将学习目标从类别增量转换为任务增量。这涉及到使用特定方法确定测试图像的任务 ID，然后以任务增量的方式进行测试。OOD检测可以作为识别任务的方法，这在最近的研究中有所尝试。Kim 等人 [19] 将持续学习问题分解为两个子问题，即任务内预测（WP）和任务 ID 预测（TP），并使用分布外（OOD）检测来解决 TP 问题。CLOM [20] 将每个任务训练为一个 OOD 检测模型，使模型不仅能对当前任务进行分类，还能检测不属于当前任务的 OOD 样本。MORE [21] 是一个基于多头变换器的模型，它将每个头训练为每个任务的独立分类器。ESN [22] 为增量学习提出了分阶段隔离的分类器，通过基于能量的置信度分数选择最佳分类器。 然而，尽管他们的方法完成了从类别增量到任务增量的转换，但 OOD 检测的性能仍然不尽如人意，导致整体持续学习任务的准确率较低。

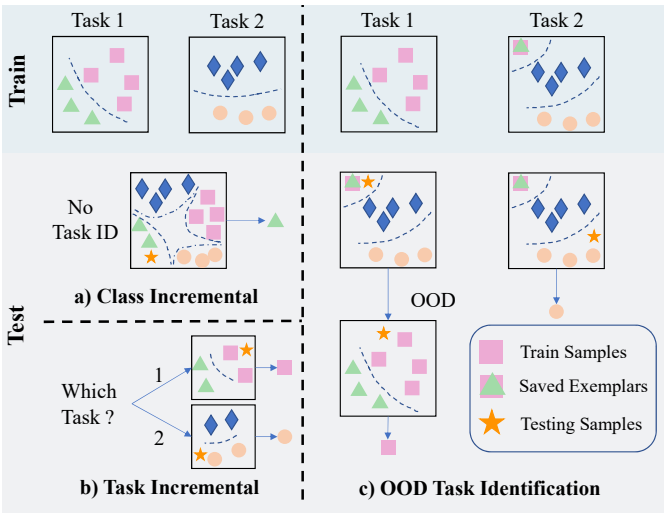


图 1: a) 和 b). 图像分类中训练和测试过程的 CIL 和 TIL 场景说明。c). 我们的方法。我们通过分布外检测的方式识别任务 ID，从而将 CIL 转化为 TIL。这要容易得多。

最近，图像-文本预训练模型受到了广泛的研究关注。由于在超过 4 亿对图像-文本对的大规模数据集上进行预训练，CLIP [23] 具有出色的泛化能力，其图像编码器有潜力预测大多数现实世界中的类别。此外，CLIP 在零样本评估方面表现出色，在许多下游任务上达到了高性能。这些预训练模型的特性使它们在持续学习和 OOD 检测任务上具有天然的优势。

The authors are with VCIP, CS, Nankai University, Tianjin 300350, X. Liu is the corresponding author(xialei@nankai.edu.cn).

Fort 等人 [24] 表明，预训练的 Transformer 在近似 OOD 基准测试中可以带来显著的改进，其中 OOD 样本与任务内分布（ID）样本具有相似的分布。Ming 等人 [25] 为 CLIP 编码器获得的相似性分数添加了单独的 softmax 缩放，进一步提高了 ID 和 OOD 数据之间的可分性。

因此，我们提出了一种基于 OOD 任务识别的CIL新方法，借助预训练的 CLIP 模型实现。如图 1所示，在训练过程中，我们对 CLIP 图像编码器上的分类器进行微调，使其具有 $(N+1)$ 个头部（第一个任务为 N 个头部），用于任务 t $(t>1)$，其中 N 表示当前任务的类别数量，而 1 代表之前遇到的所有类别（被视为 OOD 类别），并将该分类器添加到保存的列表中。在测试过程中，我们从后往前依次遍历保存的分类器列表，以找到为测试图像输出 ID 类别的分类器。为了进一步解决迭代测试过程中可能出现的提前终止问题，我们提出利用 CLIP 的零样本能力辅助 OOD 检测。

我们的贡献可以总结为：

- 我们将图像-文本预训练模型应用于CIL中的 OOD 任务识别，借助 CLIP 的泛化能力减少模型对当前任务的偏差，并缓解遗忘问题。
- 我们提出利用 CLIP 的零样本能力辅助 OOD 检测，减少在 OOD 预测迭代过程中出现的提前终止问题，并防止随着任务数量增加而导致的任务识别性能显著下降。
- 我们的模型在 CIFAR-100 和 ImageNet-Subset 数据集的传统多分类和更具挑战性的少分类设置中均取得了最先进的性能。

2 相关工作

2.1 类别增量学习（CIL）

CIL 是持续学习中一个具有挑战性的设置。方法可以分为三类：基于正则化的方法、基于重放的方法和基于架构的方法 [26]。

基于正则化的方法 基于正则化的方法通过在新旧任务之间添加额外的正则化项，以确保在学习新任务的同时保留旧知识。EWC [9] 根据参数的重要性对参数的变化施加惩罚。SI [12] 通过参数对总损失变化的贡献来近似其重要性。R-EWC [27] 改进了惩罚的实现方式，并对参数空间进行了因式分解的旋转。还可以利用知识蒸馏进行持续学习，例如 LWF [28]、LwM [29] 和 EBLL [30]。然而，随着任务数量的增加，通过正则化项缓解遗忘变得越来越困难。知识蒸馏也在其他应用中受到欢迎，例如图像分割 [31], [32] 和目标检测 [33], [34]。

基于重放的方法 基于重放的方法通过在记忆缓冲区中存储旧知识，以在训练新任务时估计和恢复旧数据分布 [35]。一些方法只是保留过去样本的一个有限子集，具有不同的选择策略 [6], [7]。

iCaRL [6] 和 EEIL [3] 在旧的和新的训练数据上执行知识蒸馏。PODNet [36] 引入了一种空间蒸馏损失，Co2L [4] 执行了一种自监督蒸馏损失。其他方法存储由生成模型产生的合成样本 [37]–[39]，或者存储中间特征表示 [10], [11]，而不是真实的样本。我们还使用了一个小的存储器来保存先前的类别，这些类别共同作为训练我们的 OOD 检测器的OOD类别。

基于架构的方法 基于架构的方法通过为每个任务训练掩码或在基础模型上添加新分支来控制哪些参数属于哪些任务。Piggyback [14]、HAT [16]、SupSup [17] 和 PackNet [40] 采用固定的网络架构，并优化一个二进制掩码以选择每个任务的特定参数。DyTox [41] 增量地训练任务特定的适配器层。DER [18] 为每个任务动态增加网络分支。Expert Gate [42] 为每个任务学习一个模块化的专家网络。PathNet [43] 和 RPSNet [44] 构建多个并行的逐层网络模块，并为不同任务选择不同的路径。在任务序列较长的场景中，这类方法会导致模型过大，参数数量显著增加，这使得其不切实际且缺乏实用性。

2.2 少样本类别增量学习

与传统的CIL不同，少样本类别增量学习（FSCIL）致力于在标记数据非常有限的情况下获取新知识。通常情况下，初始任务拥有充足的数据，而在后续任务中，每个类别仅由 5-10 个样本表示。主要挑战在于防止模型对新数据过拟合。因此，大多数方法 [45], [46] 采用基于原型的方法，在初始任务中训练一个特征提取器，并在后续任务中保持其不变或仅更新其一小部分模块。尽管这种策略确保了旧知识的保留，但在新任务上的表现较差。其他方法采用元学习技术 [47]–[49]，希望模型能够从极少量的样本中学习如何识别模式。最近的方法 [50], [51] 精心设计了双层优化技术和合成特征生成方法，以提高后续任务中的性能。一般来说，为多样本场景设计的持续学习方法通常难以在少样本设置中取得良好效果。然而，我们的基于 OOD 的方法可以直接应用于这一更具挑战性的设置，并在没有任何专门修改的情况下实现了最先进的（SOTA）结果。

2.3 分布外（OOD）检测

OOD检测识别那些与训练分布不同的分布下的测试样本。这些方法可以分为三类：基于分类的方法、基于密度的方法和基于距离的方法 [52]。

基于分类的方法 基于分类的方法发现，OOD 样本通常显示出比 ID 样本更低的最大概率值，这一特性可以用于 OOD 检测 [53]。ODIN [54] 引入了温度缩放和输入扰动，以增强 ID 样本和 OOD 样本之间的可分性。G-ODIN [55] 通过采用一种名为 DeConf-C 的特定训练目标扩展了 ODIN。

基于密度的方法 对数据的概率密度进行建模，并将低密度的样本视为 OOD。[56] 估计特征的高斯分布，并计算马氏距离以检测 OOD。有时 OOD 数据在概率密度模型中也可能具

有高似然值 [57]。因此，一些研究提出计算其他指标来替代似然值，例如似然比 [58]、似然后悔值 [59] 和 SEM 分数 [60]。

基于距离的方法 通过计算测试样本与 ID 类别原型在特征空间中的距离来检测 OOD。 [56] 通过计算样本与所有 ID 类别分布之间的最小马氏距离来进行 OOD 检测。不假设特征服从高斯分布， [61] 和 [62] 计算特征与 ID 类别之间的余弦距离。 [63] 提出了一种基于深度最近邻距离的方法，直接检测 OOD 样本。

2.4 OOD 应用于增量学习

近期的研究开始将 OOD 检测应用于持续学习。这些方法主要使用 OOD 检测来判断测试样本是否属于当前任务或其他任务。Kim 等人 [19] 将 CIL 问题分解为任务内预测（WP）和任务 ID 预测（TP），采用传统的 OOD 检测方法来解决 TP 问题。由于传统 OOD 检测方法的局限性，这种方法受到 TP 准确率低的限制。CLOM [20] 将每个任务训练为一个 OOD 检测模型，而不是传统的分类模型。尽管这种方法可以将 TIL 方法适应到 CIL 并取得不错的结果，但与 DER [18] 等专门为 CIL 设计的方法相比，它未能达到最先进的性能。MORE [21] 构建了一个基于多头变换器的模型，n 个头对应 n 个任务，以及一个额外的 OOD 头来分类之前遇到的类别。我们认为，将 OOD 检测器分离并在测试时完全移除是多余的。

除了这些基于 OOD 的方法外，ESN [22] 还聚合了来自前一阶段或任务的分类器，以确保在学习新任务时保留旧知识。然而，与我们的方法不同——我们的方法使用 OOD 检测来判断样本属于当前任务还是之前任务，而基于能量的方法使用温度控制的能量度量来规范化分类器输出，确保跨任务的可比性。

3 动机

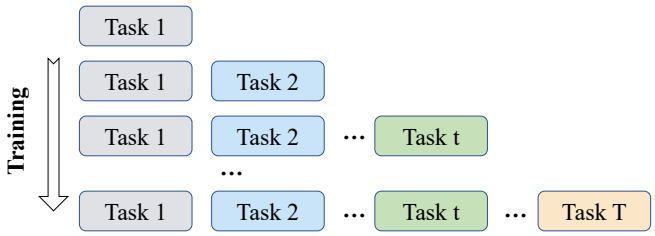
这里，我们首先简要介绍基于OOD和基于扩展分类器（EC）的持续学习方法。然后，我们将通过一个小示例说明为什么基于 OOD 的方法在持续学习中优于基于 EC 的方法。

如图 2（a）所示，基于 EC 的方法 [64], [65] 在新任务到来时会添加新的类别特定的分类头。这些新的分类头与现有的分类头一起训练，以识别迄今为止见过的所有类别。在测试过程中，使用整合了所有分类头的单一分类器进行分类。

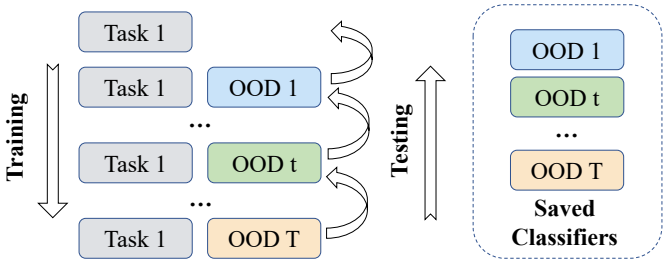
相比之下，我们的基于 OOD 的方法操作方式不同。如图 2（b）所示，当引入新任务时，仅添加一个 OOD 分类头，用于识别所有旧类别。在测试过程中，样本会被评估以确定它们属于当前任务还是之前的任务。如果它们属于之前的任务，则使用之前任务的分类器进行测试，直到识别出其任务 ID。

接下来，我们将通过 CIFAR100 的 B0-20 设置（将 100 个类别平均分配到 20 个任务中）的小示例来展示我们基于 OOD 的方法的优越性。

首先，在持续学习中，传统方法会随着任务的增加而不断扩展分类头，以适应不断增加的类别数量。线性分类器需要



(a) 基于 EC 的方法



(b) 基于 OOD 的方法

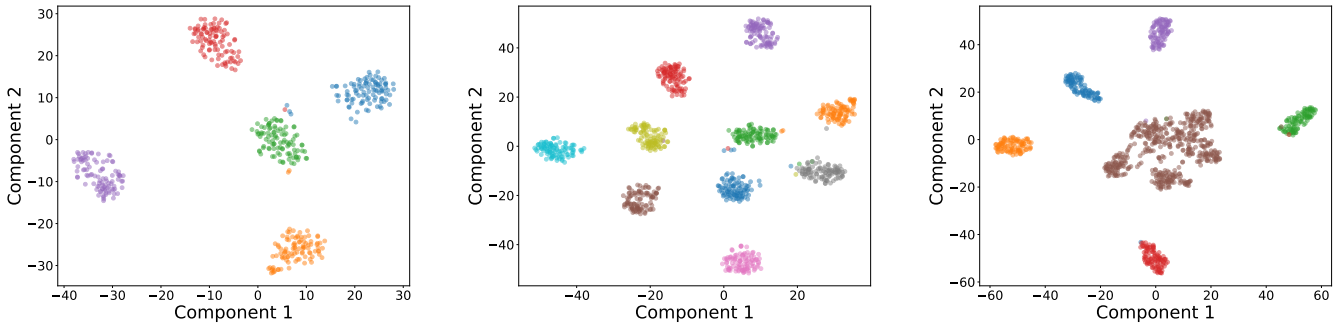
图 2: 基于 EC 和基于 OOD 的持续学习方法的比较。a) 基于 EC 的方法在每个新任务到来时添加新的分类头。b) 基于 OOD 的方法只为所有旧类别添加一个 OOD 头。基于 OOD 方法的详细训练和测试过程出现在后续章节。

在特征空间中为新类别划分新的特征空间，这不可避免地会影响已学习的特征分布，从而导致灾难性遗忘。相比之下，基于 OOD 的持续学习方法为每个任务保持恒定数量的分类头。模型专注于在新任务中学习新类别的特征分布，同时将所有旧类别视为一个类别。这显著简化了特征空间的变化，从而减少了遗忘。如图 3所示，基于 EC 的方法需要在后续任务中将五个新类别整合到之前五个类别的特征空间中，这使得误分类的可能性增加。另一方面，基于 OOD 的方法只需要扩展一个 OOD 分类头（用于所有旧类别）。通过充分学习新类别的特征分布，旧类别仍然可以被识别，从而实现了稳定性和可塑性之间的平衡。

此外，由于设备限制和隐私问题，在持续学习场景中存储（或仅保留一小部分）旧类别样本通常是不可行的。在基于 EC 的方法中，新类别的大量样本与旧类别的少量（或没有）样本之间的不平衡很容易导致分类器产生偏差，从而倾向于将样本分类为新类别，最终导致遗忘旧知识。相比之下，基于 OOD 的方法将所有旧类别归为一个类别（OOD 类别），显著减少了新旧类别之间的不平衡。这种方法对不断增加的任务数量（每个旧类别的样本数量减少）表现出强大的鲁棒性，同时缓解了灾难性遗忘的问题。

4 方法

在本节中，我们首先介绍CIL的问题定义。然后，我们介绍如何通过应用 OOD 检测进行任务识别来实现CIL，并进一步阐



(a) 基于 EC 的方法　(b) 基于 EC 的第二个任务　(c) 基于 OOD 的第二个任务

图 3: 基于 EC 和基于 OOD 的持续学习方法的差异可视化。我们以 CIFAR100 的 B0-5 设置（将 100 个类别分为 5 个任务）中的前两个任务作为小示例。(a) 是训练前 5 个类别的 t-SNE 可视化。(b) 和 (c) 分别是使用基于 EC 和基于 OOD 的方法对接下来的 5 个类别的分布进行可视化。

述如何利用 CLIP 零样本分类来提高 OOD 检测的准确性，从而提升CIL的性能。我们提出的方法的整体流程可以在图 4中找到。

4.1 预备知识

图像分类中的类别增量学习 CIL的目标是学习一系列任务 T，使得模型能够统一识别这些任务中包含的所有类别。给定一个任务序列 $D=\{D_t\}_{t=1}^{T}$，其中 $D_t=\{(\mathbf{x}_i^t,y_i^t)\}_{i=1}^{N_t}$ 是任务 t 的数据集，$\mathbf{x}_i^t\in\mathcal{X}_t$ 是输入样本，$y_i^t\in\mathcal{Y}_t$ 是其对应的标签，$|\mathcal{Y}_t|=N$ 是每个任务包含的类别数量。所有任务的类别空间是互不相交的（即，$\mathcal{Y}_t\cap\mathcal{Y}_{t'}=\emptyset,\forall t\neq t'$）。在任务 t 的训练过程中，当前任务的所有训练样本都是可用的，而所有先前任务的少量训练样本则存储在示例集 $\mathcal{V}$ 中以供使用。

从任务增量到类别增量的图像分类 除了CIL之外，TIL也得到了广泛研究。与 CIL 类似，TIL 在训练了一系列任务 D 之后，也需要识别来自特定任务 k 中某个类别的测试样本。然而，与 CIL 不同的是，CIL 在测试样本之前不提供任务 ID，而 TIL 为每个测试样本提供了真实的任务 ID。

受此启发，一些近期的研究工作 [19]–[21] 为每个任务引入了一组独立的分类器，并采用OOD检测机制来判断测试样本是否属于某个任务，而无需指定任务 ID。

为了实现高质量的 OOD 检测，他们的 OOD 检测模块要么使用现成的 OOD 方法，要么使用示例样本训练每个任务的分类头，并添加一个额外的 OOD 输出头。同时，为了确保每个任务的内部任务分类精度，他们引入了一些 TIL 的参数隔离方法来克服灾难性遗忘问题。尽管这些方法取得了改进的结果，但在从头开始增量训练中等数量的任务时，OOD 检测精度和内部任务分类精度仍然难以保证。

4.2 OOD 任务识别

用于持续学习的预训练图像-文本骨干网络 为了提高 OOD 检测和内部任务分类性能，引入一个大规模的图像-文本预训练模型是一个很好的选择。CLIP [23] 在许多下游任务上展示了强大的适应能力，这得益于其在多达 4 亿对图像-文本对上通过对比损失训练获得的泛化表示。

具体来说，CLIP 是一个双流网络，包含两个编码器 (E^I, E^T)，分别将图像和文本数据映射到多模态特征空间。对于一个测试图像，可以通过计算其视觉特征嵌入与所有候选类别的文本特征嵌入之间的余弦相似度，来执行零样本图像分类。

利用数据学习用于图像分类的 OOD 分类器 在本节中，我们详细说明如何通过使用预训练的 CLIP 的视觉编码器构建具有 OOD 检测的 CIL 模型。如上所述，大规模预训练的 CLIP 已经提供了足够好的视觉表示，因此我们使用冻结的 CLIP 视觉编码器作为 CIL 训练的骨干网络，以避免灾难性遗忘。为了实现基于 OOD 检测的 CIL，我们为每个任务学习一个独立的分类器，并在训练后依次将它们存储在分类器列表 $\mathcal{C}$ 中。给定一个任务序列 D，对于 CIL 的第一个任务，我们基于所有训练样本 $\mathcal{D}_1=\{(x_m,y_m):m\in[M]\}$，在骨干网络上微调一个线性分类层 $f(\cdot;\Phi_1)$，使用标准的交叉熵损失：

$$\mathcal{L}_s=\frac{1}{M}\sum_{m=1}^{M}\mathcal{H}\left(y_m,f\left(z_m;\Phi_1\right)\right) \tag{1}$$

其中，M 是 D_1 中的样本数量，$z_m\in\mathbb{R}^d$ 是通过冻结的 E^I 提取的 x_m 的特征嵌入，$\Phi_1\in\mathbb{R}^{d\times N}$ 是保存的分类器列表 $\mathcal{C}$ 中第一个线性分类层的参数。$d=512$ 表示由 CLIP 图像编码器获得的特征的维度，而 N 是一个任务中的类别数量（我们假设每个任务的类别数量相等，即 B0 设置）。$\mathcal{H}(\cdot,\cdot)$ 是交叉熵损失。

对于第 t^{th} $(t>1)$ 个任务，由于我们需要将当前任务之外的所有类别的样本视为 OOD 样本，我们将所有 OOD 样

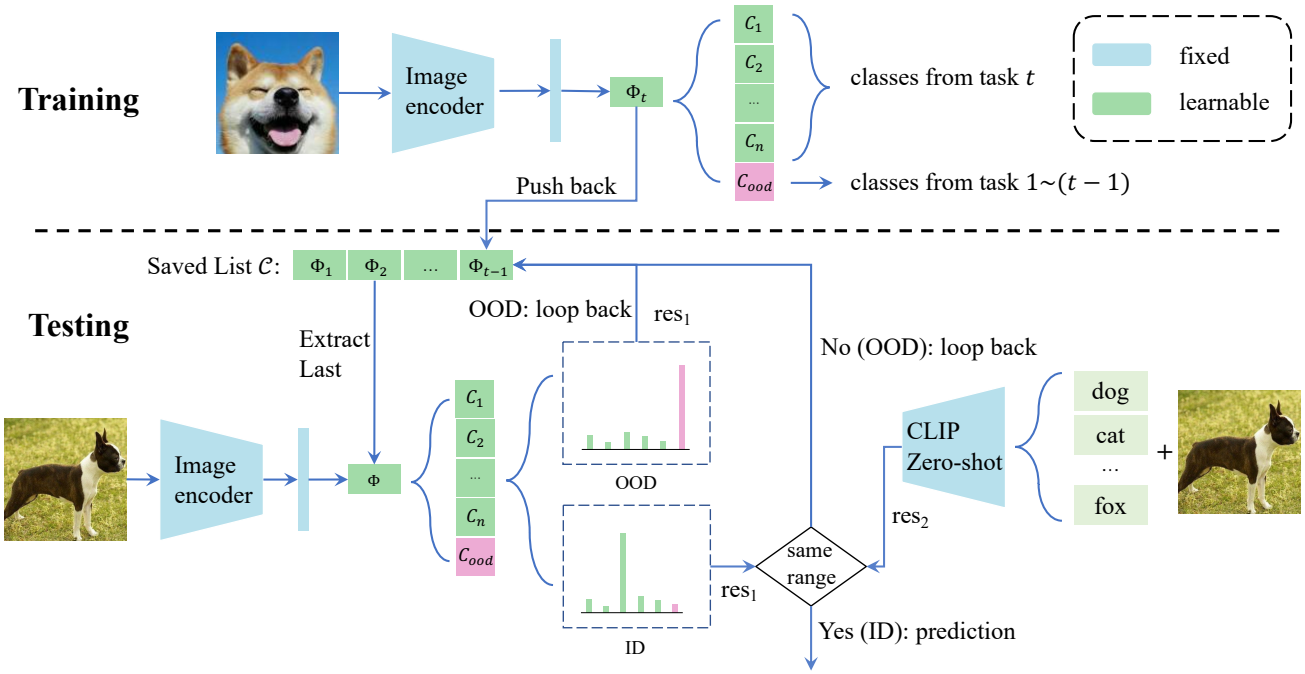


图 4: 在训练任务 t ($t > 1$) 时，我们微调一个具有 $(N+1)$ 个头部的分类器，其中 N 是当前任务中包含的类别数量，而 1 代表模型之前遇到的所有类别（被视为 OOD 类别），并将所有保存的示例标记为一个类别。训练完成后，将该分类器固定并添加到分类器列表中。在测试过程中，我们从保存的列表中提取最后一个分类器，并对测试图像进行推理。如果结果 res_1 显示它属于 OOD 类别（表明它不属于当前分类器对应的任务），我们丢弃当前分类器，并从保存的列表中提取下一个分类器继续推理，直到我们遇到一个输出 ID 类别的分类器。此时，我们将测试图像输入到 CLIP 零样本分类器中并获得结果 res_2。如果 res_1 和 res_2 属于同一任务，则我们获得测试图像的最终分类结果 res_1。否则，我们继续选择下一个分类器并重复上述过程。

本视为一个额外的 OOD 类别 y^{ood}，然后我们使用以下损失函数，结合所有训练样本 $\mathcal{D}_t = \{(x_m, y_m) : m \in [M]\}$ 和 $\mathcal{V} = \{(x_v, y_v) : v \in [V]\}$，微调带额外 OOD 类别的线性分类层 $f(;\Phi_t)$：

$$\mathcal{L}_s = \frac{1}{M+V}\Big(\sum_{m=1}^{M}\mathcal{H}\left(y_m, f\left(z_m;\Phi_t\right)\right) + \sum_{v=1}^{V}\mathcal{H}\left(y^{ood}, f\left(z_v;\Phi_t\right)\right)\Big) \tag{2}$$

其中，M 是 D_t 中的样本数量，V 是示例集 $\mathcal{V}$ 中的样本数量，$z_v \in \mathbb{R}^d$ 是通过冻结的 E^I 提取的 x_v 的特征嵌入，$\Phi_t \in \mathbb{R}^{d\times(N+1)}$ 是 $\mathcal{C}$ 中第 t^{th} 个线性分类层的参数。算法 1描述了对应于基于 OOD 的持续学习的特定任务的训练过程。

在所有任务的增量训练之后，我们可以使用类似 TIL 的测试形式进行 CIL 测试，任务 ID 由 OOD 检测模块自动确定，而不是手动指定：给定一个测试样本 $\boldsymbol{x}_{\text{test}}$，我们从分类器列表 $\mathcal{C}$ 中按从后到前的顺序提取每个任务的分类器，并对 $\boldsymbol{x}_{\text{test}}$ 进行分类。如果 $\boldsymbol{x}_{\text{test}}$ 在某个分类器中被识别为 OOD，则认为该样本不属于此任务，我们使用下一个分类器进行分类，直到样本被识别为该任务中的某个类别（ID），并使用这个类别作为最终结果。

Algorithm 1 训练算法

Input: X^t ▷ 当前任务的训练样本
Input: $x_v^1, \ldots, x_v^{t-1}$ ▷ 前一个任务的代表性样本

require: Φ_r ▷ 随机初始化新的分类器
require: $\mathcal{C}_{t-1} = (\Phi_1, \ldots, \Phi_{t-1})$ ▷ 保存的分类器列表

$X^{OOD} \leftarrow \text{RELABEL}(x_v^1, \ldots, x_v^{t-1})$
$\Phi_t \leftarrow \text{UPDATE}(X^{OOD}, X^t, \Phi_r)$ ▷ 根据公式（2）更新
$x_v^t \leftarrow \text{HERDING}(X^t)$ ▷ 选择代表性样本
$\mathcal{C}_t = (\Phi_1, \ldots, \Phi_{t-1}, \Phi_t)$ ▷ 将新的分类器添加到列表中
return Φ

利用图像-文本预训练模型的零样本 OOD 检测器 尽管为每个分类头添加一个额外的通道以检测不属于当前任务的 OOD 样本是一种简单且高效的做法，但 OOD 检测的准确性仍然不尽如人意，最终导致分类性能次优。为了缓解这一问题，我们增加了一个额外的 OOD 模块，利用 CLIP 强大的零样本识别能力。

具体来说，我们设计了一个基于微调视觉分类器和 CLIP 图像-文本零样本分类器的两级 OOD 样本检测机制。在测试阶段，测试样本 $\boldsymbol{x}_{\text{test}}$ 首先使用从 $\mathcal{C}$ 中按从后到前顺序弹出的

Algorithm 2 测试过程

Input: x_{test} ▷ 待测试的图像
require: $\Phi = (\Phi_1, \ldots, \Phi_T)$ ▷ 保存的分类器
require: E^I, E^T ▷ CLIP 编码器
require: $\mathcal{W} = \{W_t\}_{t=1}^T$ ▷ 类别名称

1: **for** $t = T...2, 1$ **do** ▷ 从后向前迭代
2: $\text{res}_1 \leftarrow f(E^I(x_{\text{test}}); \Phi_t)$ ▷ 分类器的结果
3: **if** $\text{res}_1 = \text{OOD}$ **then**
4: **continune** ▷ OOD: 循环回退
5: **end if**
6: $\text{res}_2 \leftarrow \text{ZERO-SHOT}(x_{\text{test}}, \mathcal{W}, E^I, E^T)$
7: **if same-task**$(\text{res}_1, \text{res}_2)$ **then**
8: **return** res_1 ▷ ID: 返回预测结果
9: **end if**
10: **end for**

视觉分类器进行分类/ OOD 检测。如果 $\boldsymbol{x}_{\text{test}}$ 被一个视觉分类器识别为 OOD，我们将放弃当前分类器，并继续迭代选择下一个分类器，直到我们遇到一个将当前测试图像识别为 ID 类别的分类器。一旦 $\boldsymbol{x}_{\text{test}}$ 被识别为 ID 类别，我们将进一步执行 CLIP 零样本分类：给定每个类别在 $D = \{D_t\}_{t=1}^T$ 中的文本标签 $\mathcal{W} = \{W_t\}_{t=1}^T$，由于我们的 CIL 测试按从后到前的顺序从分类器列表中获取视觉分类器，对于当前任务 t，我们只为模型在训练此任务时见过的所有类别提供文本标签（即，$\mathcal{W} = \{W_t\}_{t=1}^t$）。这个文本标签通过一个提示词来生成文本编码器 E^T 的输入，提示词通常具有 “一张 [CLS] 的照片” 的形式。E^T 的最终输出可以定义为 $\{\boldsymbol{w}_i\}_{i=1}^{t\times N}$。最终的零样本预测概率可以计算为：

$$p(y = i \mid \boldsymbol{x_{test}}) = \frac{\exp\left(\cos\left(\boldsymbol{w_i}, \boldsymbol{z_{test}}\right)/\tau\right)}{\sum_{j=1}^{t\times N} \exp\left(\cos\left(\boldsymbol{w_j}, \boldsymbol{z_{test}}\right)/\tau\right)} \quad (3)$$

其中，$z_{\text{test}} \in \mathbb{R}^d$ 是通过冻结的 E^I 提取的 x_{test} 的特征嵌入，τ 是一个温度缩放参数，用于控制输出的锐度。

如果零样本分类的结果也表明测试图像属于当前分类器对应的任务，我们将使用该视觉分类器获得的分类结果作为最终预测结果。否则，我们继续迭代并选择下一个分类器，重复上述操作。

算法 2 给出了我们提出模型的测试过程的详细描述。在第 2 行，我们获得了当前分类器的结果 res_1。只有当我们遇到一个将当前样本预测为 ID 类别的分类器时，我们才会进入第 3 行；否则，我们返回第 1 行并继续提取下一个分类器。在第 5 行，我们使用与视觉分类器预测的任务 ID 对应的标签名称进行 CLIP 零样本预测 res_2。在第 6 行，我们比较从两个模块获得的两个结果，如果这两个结果都表明测试样本属于当前任务，我们返回最终预测结果 res_1。否则，我们返回第 1-2 行并选择上一个分类器，再次重复整个过程。

5 实验

在本节中，我们首先描述实现细节和基线，然后将我们的方法与其他竞争方法进行比较。接下来通过消融研究从不同角度理解我们的方法。所有结果均在三种不同类别顺序下取平均值，详细结果可在补充材料中找到。

5.1 实验设置

实现细节 我们在配备 4 个 V100 GPU 的 Linux 集群上进行了所有实验。我们在 CIFAR-100、ImageNet-Subset [66] 和 5-Datasets [72] 上测试了模型。5-Datasets 是一个为持续学习的跨数据集评估而设计的特殊基准，包含五个数据集，分别是 CIFAR-10 [73]、MNIST [74]、Fashion-MNIST [75]、SVHN [76] 和 notMNIST，每个数据集包含十个类别，被视为一个任务，总共五个任务。

B50 设置表示模型首先在一个包含 50 个类别的较大数据集上进行训练，随后每个任务增加 5 个或 10 个新类别。另一种设置 B0 保持每个任务的类别数量不变，将 100 个类别平均分配到 5 个、10 个或 20 个任务中。我们使用预训练的 CLIP 图像编码器，并遵循 CLIP 原始文本提示，使用 “一张糟糕的 [CLS] 照片”。分类器由一个仅包含一个隐藏层的 MLP 组成。我们使用 Adam [77] 作为优化器，并为每个任务训练 10 个周期，学习率为 0.01，权重衰减为 0.0002。对于所有设置，我们将示例集的大小固定为 2000，并将温度缩放参数 τ 设置为 2。在 CLIP 零样本 OOD 检测模块中，我们遵循 Continual CLIP [69] 的方法。随着我们深入分类器列表，零样本需要区分的类别数量减少，零样本的准确率相应提高，这有助于我们更好地识别任务 ID。

对比方法 我们将我们的方法与 10 种传统基线方法进行了比较，包括 iCaRL [6]、LUCIR(LUCIR-DDE) [66]、PODNet [36]、RM [71] 和 DER [18]。基线 Linear Probe 和我们的方法基于相同的预训练模型，Linear Probe 的分类器头随着每个任务扩展，并直接使用分类器头进行分类。此外，我们还与基于提示的方法 L2P [64]、DualPrompt [65] 和 CODA-Prompt [70] 进行了比较。这些方法都使用在 ImageNet-21K 上预训练的骨干网络，并已在 CIFAR-100 和 ImageNet-Subset 上进行了测试。如文献 [78] 所述，仅使用 NMC 而不进行任何额外训练时，ImageNet-21k 预训练的 ViT 模型在所有五个数据集上的表现均优于 L2P。因此，为了公平比较，我们将这些方法的预训练模型替换为 CLIP 图像编码器。

我们将基线分为两部分：参数量增长和参数量不变。DER [18] 在整个学习过程中持续复制和扩展骨干网络，而 DualPrompt [65] 为每个新任务分配一个专家提示，我们将这些方法归类为参数量增长，并在表格中用斜体加粗文本表示。仅扩展分类器的方法被视为参数量不变。

表 1: 我们的方法与其他最先进方法在不同设置下的 CIFAR-100 上的结果。“# tasks” 表示任务数量。* 表示我们用 CLIP 图像编码器替换了 ImageNet-21K 预训练模型，以确保公平比较。在方法列中，斜体加粗文本表示参数量增长，而其他则表示参数量不变。

Method	Backbone	No. Para (M)	CIFAR-100 B0						CIFAR-100 B50			
			# tasks						# tasks			
			5		10		20		5		10	
			avg	last	avg	last	avg	last	avg	last	avg	last
iCaRL [6]	ResNet	11.2	71.1	59.6	65.3	50.9	61.2	44.9	65.1	56.0	58.6	49.5
LUCIR [66]			62.8	46.9	58.7	42.9	58.2	41.1	64.3	52.7	59.9	48.2
BiC [67]			73.1	61.5	68.8	53.6	66.5	47.8	66.6	55.1	60.3	48.7
WA [68]			72.8	60.3	69.5	53.7	67.3	48.2	64.0	52.8	57.9	48.1
PODNet [36]			66.7	51.5	58.0	40.7	54.0	35.8	67.3	56.0	64.0	51.7
RPSNet [44]			70.5	60.8	68.6	56.2	-	-	-	-	-	-
DER [18]			76.8	67.3	75.4	64.4	74.1	62.6	73.2	66.0	72.8	65.6
Dytox [41]	conViT	11.0	-	-	72.9	60.0	72.2	57.0	-	-	-	-
Continual-CLIP [69]	ViT-B/16	85.9	74.0	66.7	75.1	66.7	75.9	66.7	69.7	66.7	69.5	66.7
Linear Probe			78.8	72.9	80.2	72.4	78.9	69.4	78.9	71.2	77.6	72.3
L2P* [64]			78.1	65.7	79.2	67.5	79.2	68.4	77.2	68.3	76.5	69.2
DualPrompt* [65]			78.6	72.6	80.3	69.4	80.6	70.2	79.6	56.3	62.5	32.8
CODA-Prompt* [70]			68.3	33.8	71.1	42.9	69.3	39.6	76.6	51.3	64.3	32.5
Ours	ViT-B/16	86.1	81.1	75.5	84.3	74.2	84.0	73.9	82.8	76.3	**83.0**	75.6
Ours (order 1)			81.3	**77.2**	83.6	**77.6**	83.0	72.9	**84.2**	76.0	82.1	**75.7**
Ours (order 2)			**82.4**	75.1	**85.0**	73.4	82.3	75.0	82.5	75.6	81.8	73.5
Ours (order 3)			82.1	74.4	83.0	74.3	**84.1**	**75.8**	82.8	**78.5**	83.4	74.4

5.2 与最先进方法的比较

在 CIFAR-100 上的评估 如表 1所示，我们的结果在每个设置中均一致优于基线方法。在 B0-5 任务设置中，我们在最后一个任务上比 DER 提高了约 8.2%。请注意，使用 CLIP 图像编码器的基于提示的方法表现远低于预期。我们认为这些方法的大部分性能提升可能来自于 ImageNet 预训练的骨干网络，而不是提示池的设计。

此外，我们还可以看到，在任务数量较多的设置（B0-20、B50-10）中，我们的模型相对于最先进方法的改进幅度略有减少。这是因为任务数量越多，我们的 OOD 测试链就越长，出错的可能性也就越大。然而，由于我们的零样本 OOD 检测器不仅协助识别任务 ID，还保证了我们模型的下限性能，即使在处理大量任务时，我们也能保持显著的高性能。

在 ImageNet-Subset 上的评估 在 ImageNet-Subset 上的实验（表 2）也证明了我们模型的有效性。在每个设置中，我们的结果都显著高于最先进水平。值得注意的是，我们的模型在所有五个设置中分别保持了大约 85 和 77 的平均值和最终结果，这是其他模型无法匹敌的。这表明，尽管更多的任务和更长的测试链可能会引入更多的错误，但这些错误对性能的影响远小于随着任务数量增加而引入到分类器中的偏差。这证明了我们基于 OOD 的方法的优越性。

多次运行的评估 表 1和表 2的最后三行展示了我们的方法在不同类别顺序下的结果。尽管在不同顺序下的性能略有变化，但我们的方法始终显著优于比较方法。这表明我们的基于 OOD 的方法不依赖于固定的顺序。

在 5-Datasets 上的评估 由于 5-Datasets 基准中的每个任务代表一个不同的数据集，因此为了与其他基线进行公平比较，有必要遵循统一的训练顺序。我们遵循了 DualPrompt [65] 的设置，使用 SVHN、MNIST、CIFAR-10、NotMNIST 和 FashionMNIST 的顺序。以最后一个任务的准确率作为评估指标。如表 3所示，我们的模型显著优于其他方法。这主要是由于我们的 OOD 检测方法更适合涉及不同数据集的场景，从而显著提高了任务识别的准确性，几乎没有遗忘的现象出现。

在少样本设置上的实验 我们还在 CIFAR100 上进行了少样本设置的实验（表 4），其中第一个任务包括所有 60 个类别的所有数据，而随后的 8 个任务每个包含 5 个类别，每个类别只有 5 个样本。竞争方法包括基于 EC 或原型的方法，如 iCaRL [6]、EEIL [3]、LUCIR [66]、TOPIC [79]、CEC [46]、F2M [45]、MetaFSCIL [50]、Entropy-seg [80]，以及基于提示的方法 L2P [64]、DualPrompt [65] 和 CODA-Prompt [70]。可以看出，我们的方法最初可能不如某些基于 EC 的方法表现好，但随着已见类别数量的增加（特征空间更加拥挤），我们的优势变得更加明显。在最后一个任务中，我们实现了 71.04 的性能，至少比基于 EC 的方法高出 10 个百分点。

表 2: 我们的方法与其他最先进方法在不同设置下的 ImageNet-Subset 上的结果。

Method	Backbone	No. Para (M)	ImageNet-Subset B0						ImageNet-Subset B50			
			# tasks						#tasks			
			5		10		20		5		10	
			avg	last	avg	last	avg	last	avg	last	avg	last
iCaRL [6]	ResNet	11.2	78.1	65.2	74.1	58.5	69.0	50.9	60.7	44.7	57.3	44.4
End2End [3]			75.5	64.0	70.1	53.0	68.3	48.9	61.0	52.1	58.5	52.2
UCIR [66]			76.0	64.0	70.5	55.3	64.7	47.8	77.2	68.2	66.9	56.8
PODNet [36]			78.2	66.2	72.3	57.2	66.7	48.9	80.3	73.5	79.0	70.8
UCIR-DDE [66]			77.2	65.8	71.7	56.8	66.2	40.0	78.8	68.1	68.4	57.9
RM [71]			75.5	62.2	70.4	53.2	65.4	45.7	56.9	41.8	57.7	37.3
DER(w/o P) [18]			-	-	77.2	66.7	-	-	-	-	78.2	74.9
DyTox+ [41]	conViT	11.0	-	-	77.2	67.7	-	-	-	-	-	-
Continual-CLIP [69]	ViT-B/16	85.9	84.8	75.3	85.0	75.3	86.6	75.3	79.2	75.3	79.3	75.3
Linear Probe			79.4	64.5	81.8	67.4	84.0	72.0	81.4	67.2	83.1	72.0
L2P* [64]			81.7	76.5	80.3	75.2	80.1	75.7	74.9	74.2	72.3	72.6
DualPrompt* [65]			75.4	61.1	80.7	67.4	83.9	74.2	74.2	49.8	62.1	22.4
CODA-Prompt* [70]			51.6	24.9	64.1	34.8	69.8	44.0	65.1	28.8	57.3	20.0
Ours	ViT-B/16	86.1	84.9	77.6	85.3	77.4	85.6	77.2	85.2	**78.5**	**85.0**	78.5
Ours (order 1)			83.5	77.1	84.4	76.2	87.1	**79.0**	85.4	77.3	83.2	78.4
Ours (order 2)			85.7	76.9	85.1	75.6	86.6	75.8	84.4	77.5	84.3	77.5
Ours (order 3)			**85.8**	**78.5**	**86.6**	**78.5**	**87.8**	77.3	**86.6**	77.0	84.7	**79.3**

表 3: 我们的方法与其他最先进方法在 5-Datasets 上的结果。

Method	Backbone	Para	Acc
ER	ResNet	11.0	80.32
BiC		11.0	78.74
L2P	ViT-B/16	85.9	81.84
DualPrompt		85.9	77.91
CODA-Prompt		85.9	64.18
Ours	ViT-B/16	86.2	**87.31**

5.3 运行时间分析

通过将 CLIP 图像编码器保持为固定组件，图像编码器的一次传递足以确定后续的分类器。此外，后续的分类器仅由一个隐藏层组成，参数极少，因此计算开销和内存消耗的增加可以忽略不计。在表 5中，我们对 B0-10 ImageNet-Subset 设置中的模型参数、推理延迟和准确性进行了全面比较。这包括各种方法的初始参数数量、训练后的最终参数数量以及可训练参数数量。结果明确表明，与最近的方法相比，我们的方法仅增加了极小的额外延迟。值得注意的是，我们的方法仅通过适度增加参数，就实现了显著的性能提升。

5.4 成功和失败的案例

这里我们讨论 OOD 分类中的常见失败案例以及如何通过整合 CLIP 来提高整体准确性。以 CIFAR100 B0-20 设置为例，模型在学习了三个任务（15 个类别）后，在测试中遇到了“水族馆鱼”这一类别。

在没有 CLIP 的情况下（图 6的左侧），ID 样本和 OOD 样本之间数量的不平衡会导致误分类，模型将图像分配至错误的任务。这发生是因为第一个分类器中的有限示例记忆导致了“早期停止”。有了 CLIP 的帮助（右侧），CLIP 和分类器必须在样本是否属于 ID 类别上达成一致，才能进行最终分类。如果不是这样，样本将移至前一个分类器。CLIP 的零样本分类使用类别名称来改善任务识别。尽管 CLIP 帮助了任务分配，但细粒度分类仍由分类器处理，从而导致更准确的结果。这种组合显著提高了整体模型的准确性。

在图 7中，我们展示了 5-Datasets 和 CIFAR100 B0-10 设置下的成功和失败案例。在 5-Datasets 基准测试中，由于 SVHN 的最终准确率仅为 72%，许多错误表现为同一任务内的误分类。可以观察到，在错误预测中，图像内容通常与误分类结果相似。而在正确预测中，任务之间的分布差异较大，这表明我们的方法能够准确预测任务。

在 CIFAR100 中，错误通常发生在文本和视觉含义重叠的情况下，例如“水獭”和“海狸”，导致 CLIP 零样本和 OOD 增量分类器同时失败。

5.5 基于 OOD 方法的优势

如动机部分所述，我们的基于 OOD 的方法不需要随着任务数量的增加而持续扩展分类头，因此它不需要将新类别的特征插入到已从旧类别中很好地学习到的特征空间中。因此，1）模型不需要太多调整，因此微调过程很快。2）它对示例样本的数量不敏感，因为我们把所有示例都视为属于同一个类别，极大地减少了传统基于示例方法的类别不平衡问题。我们现在将通过两个实验来证明这些观点。

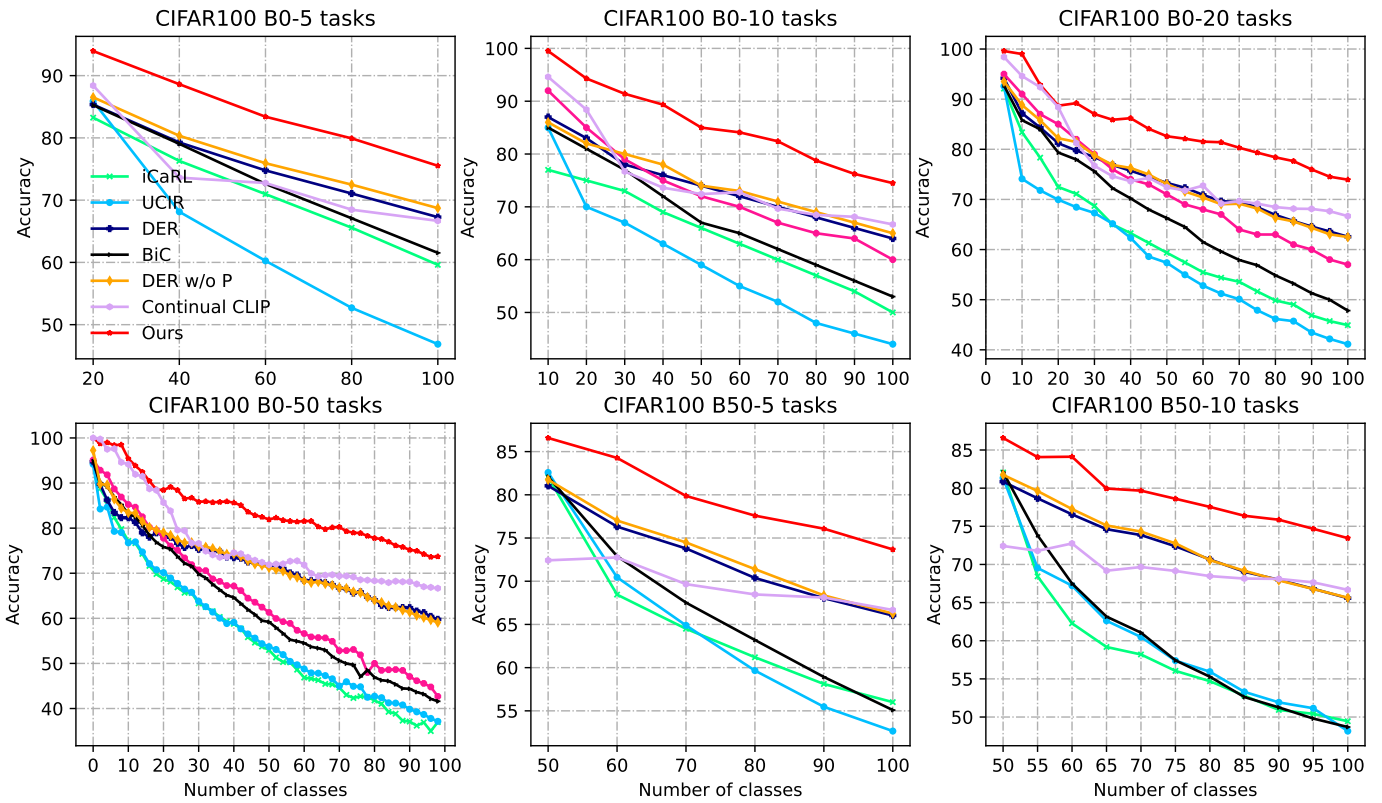


图 5: 我们的方法与 CIFAR-100 上的最先进方法的详细比较结果。

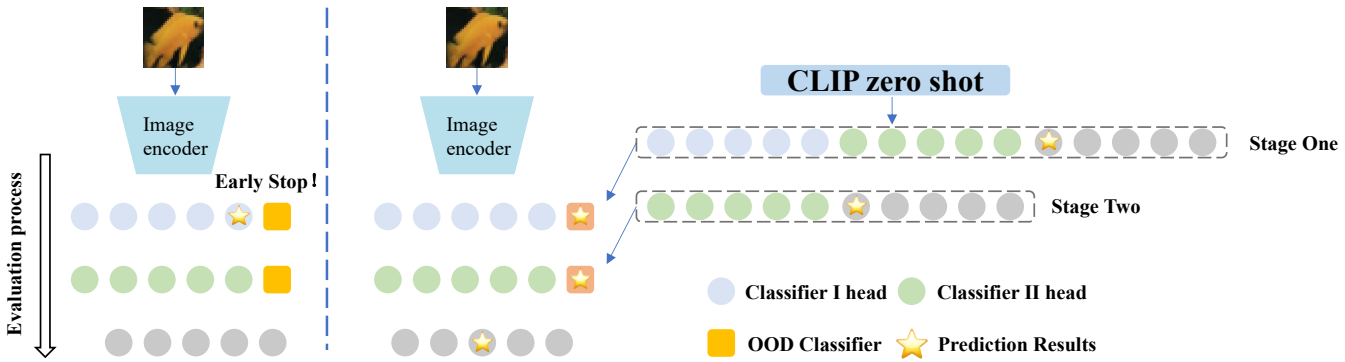


图 6: OOD 增量分类器的失败案例以及我们基于 CLIP 零样本引导方法的成功案例。

我们在 CIFAR100 上进行了实验，采用 B0-10 设置，将我们的基于 OOD 的方法与两种基于扩展分类器（EC）的方法：线性探测和 L2P 进行比较。首先，如图 8（a）所示，即使只训练一个周期，我们的方法也能实现相当的性能，并且随着周期数的增加，其性能逐渐提高。另一方面，基于 EC 的方法在周期数较少时表现不佳（欠拟合），并且随着周期数的增加，它们倾向于由于类别不平衡而过拟合到新任务的类别，导致灾难性遗忘和次优性能。

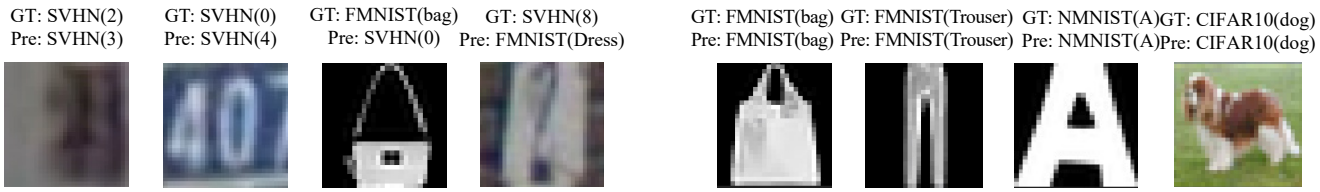
此外，在图 8（b）中，可以观察到，当示例样本数量较少（500、1000）时，基于 EC 的方法的性能显著下降，表明它们的分类器严重依赖于复习记忆。相比之下，我们的方法对示例样本数量的变化表现出很强的鲁棒性，表明我们的性能来自于所提出的基于 OOD 的分类方法，而不是示例样本。

5.6 消融研究

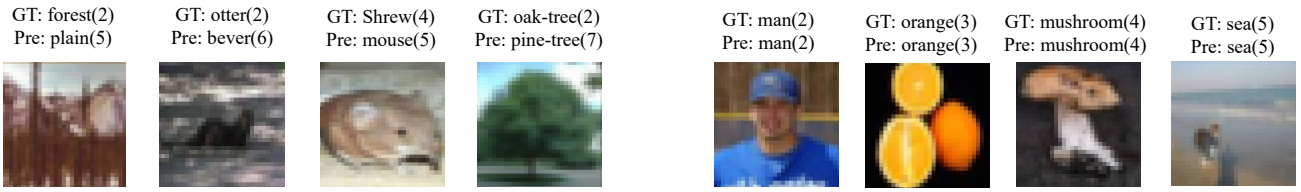
对两个 OOD 任务识别模块的消融研究 在本节中，我们讨论了我们提出的每个模块的有效性。首先，我们测试了不使用 CLIP 作为辅助工具的简单基于分类器的 OOD 检测（Only Classifier）。此外，我们仅使用 CLIP 进行 OOD 检测。具体来说，我们为十个任务训练了十个完全独立的分类器，并在测试时，首先使用零样本结果来确定其任务 ID，然后使用该任务保存的分类器进行分类（Only CLIP）。如图 10所示，仅使用分类器或仅使用 CLIP 进行 OOD 检测的效果还不够理

表 4: 在 CIFAR100 上的少样本类别增量设置的实验结果。

Method	0	1	2	3	4	5	6	7	8	PD↓
iCaRL [6]	64.10	53.28	41.69	34.12	27.93	25.06	20.41	15.48	13.73	50.37
EEIL [3]	64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	48.25
LUCIR [66]	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	50.56
TOPIC [79]	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	34.73
CEC [46]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	23.93
F2M [45]	71.45	68.10	64.43	60.80	57.76	55.26	53.53	51.57	49.35	22.06
MetaFSCIL [50]	74.50	70.10	66.84	62.72	59.48	56.52	54.36	52.56	49.97	24.53
Entropy-reg [80]	74.40	70.20	66.54	62.51	59.71	56.58	54.52	52.39	50.14	24.26
L2P* [64]	91.22	88.35	82.80	70.34	68.66	64.34	60.78	58.32	54.89	36.33
DualPrompt* [65]	91.08	87.96	84.55	71.31	68.45	64.52	61.20	59.31	54.67	36.41
CODA-Prompt* [70]	**93.55**	**89.91**	**86.54**	76.65	71.91	67.12	64.52	62.89	59.32	34.23
Ours	88.93	85.86	83.14	**80.57**	**78.04**	**76.41**	**74.02**	**71.96**	**71.04**	**17.89**



(a) 5-Datasets



(b) CIFAR100 B0-10 tasks

图 7: 5-Datasets 和 CIFAR100 基准测试下成功和错误案例的示例：(a) 括号内表示真实标签，(b) 括号内显示对应的任务 ID，总共 10 个任务。

表 5: 参数数量和推理延迟的比较。我们在一台 3090 Ubuntu 机器上进行了所有实验，使用了 DER、DyTox 和 L2P 的官方代码。参数以百万为单位进行测量，延迟以秒为单位。延迟是针对尺寸为 (3, 224, 224) 的 128 张图像批次进行报告的。

Method	Initial para	Final para	Trainable para	Latency	Accuracy
DER	11.22	112.27	11.22	1.35	77.2
DyTox	11.01	11.02	11.02	1.83	77.2
L2P	85.9	85.9	0.12	0.63	80.3
Ours	86.19	88.89	0.27	1.86	85.3

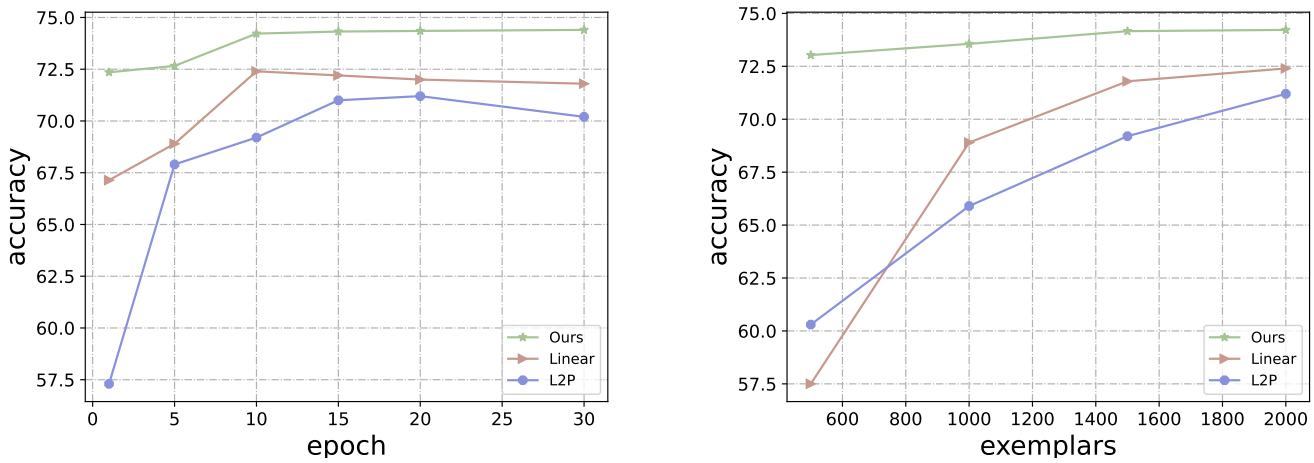
想。然而，将这两个组件结合起来，至少在最后一个任务和平均准确率上都提高了 2%。

我们认为 CLIP 中的预训练图像编码器也是我们模型实现高性能的基本组成部分。正如文献 [24] 所验证的那样，CLIP 已经证明了其自身检测 OOD 样本的能力。

在这里，我们测试了从头开始训练一个 ResNet32 网络用于基于 OOD 检测的持续学习，但发现其性能远不如微调预训练模型。如图 9和图 10所示，没有预训练的方法在后期任务中的表现非常差，在最后一个任务上仅达到 47%。

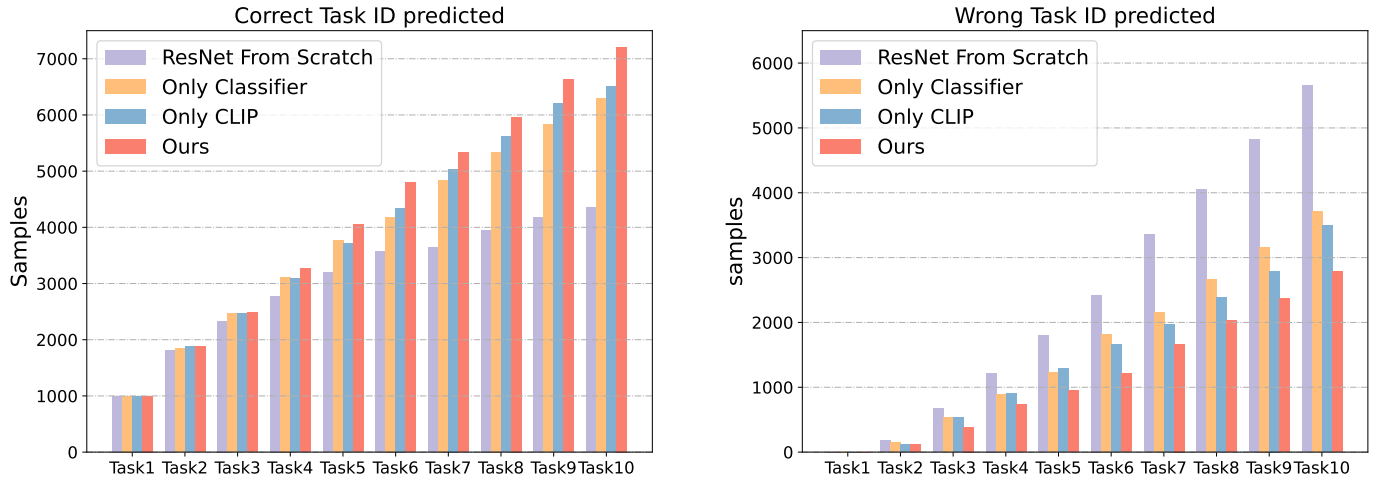
与其他 OOD 实现的比较 我们还探索了其他基于 OOD 检测的持续学习方法 [25]。我们为每个任务训练了一个特定于任务的分类器和一个任务识别器，将每个任务中的所有类别视为同一类别。在测试过程中，我们首先使用这个识别器来确定测试图像属于哪个任务，然后使用相应的特定于任务的分类器得出最终分类结果。

我们在表 6中报告了两个 OOD 模块的准确率。可以观察到，直接在任务 ID 上训练的分类器随着任务数量的增加，性能迅速下降。在最后一个任务中，OOD 的准确率仅为 51.1%，



(a) 不同数量的训练轮次　　(b) 不同数量的示例样本

图 8: (a) 展示了模型性能随训练周期数的变化，而 (b) 展示了模型性能随示例样本数量的变化。可以观察到，基于 EC 的方法更依赖于微调和大量的示例样本，而我们的方法对训练周期数和示例样本数量都表现出一定的鲁棒性。



(a) 正确任务的 ID 预测　　(b) 错误任务的 ID 预测

图 9: 我们的方法与其他消融设置在 CIFAR-100 B0-10 上的任务预测性能。“Only CLIP” 表示我们仅使用 CLIP 进行 OOD 检测。“Only Classifier” 表示我们仅使用视觉分类器进行 OOD 检测，而不使用 CLIP 作为辅助工具。我们还从头开始训练了一个 ResNet32 网络用于 OOD 检测，结果在 “Resnet From Scratch” 中展示。

这意味着模型本身的准确率更低。相比之下，我们的方法利用了 CLIP 零样本的高性能和 CLIP 图像编码器的出色泛化能力。

6 结论

在本工作中，我们提出了一种基于 OOD 检测的方法，将 CIL 转化为 TIL，并利用 CLIP 预训练模型进一步提高任务识别的准确性。我们发现，在持续学习过程中，预训练的骨干网络可以防止模型偏向当前任务，而零样本 OOD 检测不仅能够提高 ID 识别的性能，还能在任务数量较多的设置中确保整体模型准确率的下限。我们相信，基于预训练模型的持续学习具有重要的研究价值，而基于 OOD 的方法能够充分发挥预训练模型的特性。在未来的研究中，值得探索如何利用现有的 OOD 方法设计更合理的持续学习训练和测试方法。

参考文献

[1] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, “Overcoming catastrophic forgetting in neural networks,” *PNAS*, 2017.

表 6: 在 CIFAR-100 B0-10 设置下，训练独立的任务识别器 [25] 与我们方法的性能比较。

task	2	3	4	5	6	7	8	9	10
Additional OOD	94.9	83.4	73.5	70.9	67.9	63.3	59.7	58.3	51.1
Ours	88.0	84.0	80.9	78.3	77.0	75.1	74.3	71.7	70.4

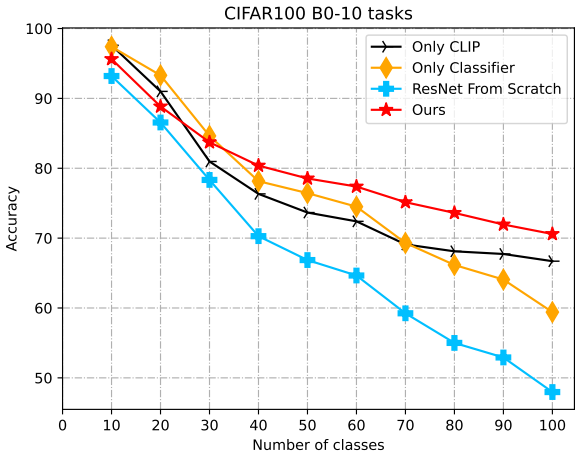


图 10: 我们的方法与其他消融设置在 CIFAR-100 B0-10 上的性能表现。

[2] G. M. van de Ven and A. S. Tolias, "Three scenarios for continual learning," *NIPS Workshops*, 2019.

[3] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *ECCV*, 2018.

[4] H. Cha, J. Lee, and J. Shin, "Co2l: Contrastive continual learning," in *ICCV*, 2021.

[5] W. Li, B.-B. Gao, B. Xia, J. Wang, J. Liu, Y. Liu, C. Wang, and F. Zheng, "Cross-modal alternating learning with task-aware representations for continual learning," *IEEE Transactions on Multimedia*, pp. 1–14, 2023.

[6] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *CVPR*, 2017.

[7] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," *ICLR*, 2019.

[8] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," *NIPS*, 2019.

[9] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *PNAS*, 2017.

[10] M. Toldo and M. Ozay, "Bring evanescent representations to life in lifelong class incremental learning," in *CVPR*, 2022.

[11] K. Wang, J. van de Weijer, and L. Herranz, "Acae-remind for online continual learning with compressed feature replay," *Pattern Recognition Letters*, 2021.

[12] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *ICML*, 2017.

[13] S. Thuseethan, S. Rajasegarar, and J. Yearwood, "Deep continual learning for emerging emotion recognition," *IEEE Transactions on Multimedia*, vol. 24, pp. 4367–4380, 2022.

[14] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *ECCV*, 2018.

[15] K. Du, F. Lyu, L. Li, F. Hu, W. Feng, F. Xu, X. Xi, and H. Cheng, "Multi-label continual learning using augmented graph convolutional network," *IEEE Transactions on Multimedia*, vol. 26, pp. 2978–2992, 2024.

[16] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *ICML*, 2018.

[17] M. Wortsman, V. Ramanujan, R. Liu, A. Kembhavi, M. Rastegari, J. Yosinski, and A. Farhadi, "Supermasks in superposition," *NIPS*, 2020.

[18] S. Yan, J. Xie, and X. He, "Der: Dynamically expandable representation for class incremental learning," in *CVPR*, 2021.

[19] G. Kim, C. Xiao, T. Konishi, Z. Ke, and B. Liu, "A theoretical study on solving continual learning," *NIPS*, 2022.

[20] G. Kim, S. Esmaeilpour, C. Xiao, and B. Liu, "Continual learning based on ood detection and task masking," in *CVPR*, 2022.

[21] G. Kim, B. Liu, and Z. Ke, "A multi-head model for continual learning via out-of-distribution replay," in *Conference on Lifelong Learning Agents*, 2022.

[22] Y. Wang, Z. Ma, Z. Huang, Y. Wang, Z. Su, and X. Hong, "Isolation and impartial aggregation: A paradigm of incremental learning without interference," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 10 209–10 217.

[23] 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 *ICML*, 2021.

[24] S. Fort, J. Ren, and B. Lakshminarayanan, "Exploring the limits of out-of-distribution detection," *NIPS*, 2021.

[25] Y. Ming, Z. Cai, J. Gu, Y. Sun, W. Li, and Y. Li, "Delving into out-of-distribution detection with vision-language representations," *NIPS*, 2022.

[26] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *TPAMI*, 2021.

[27] X. Liu, M. Masana, L. Herranz, J. Van de Weijer, A. M. Lopez, and A. D. Bagdanov, "Rotate your networks: Better weight consolidation and less catastrophic forgetting," in *ICPR*, 2018.

[28] Z. Li and D. Hoiem, "Learning without forgetting," *TPAMI*, 2017.

[29] P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, and R. Chellappa, "Learning without memorizing," in *CVPR*, 2019.

[30] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, "Encoder based lifelong learning," in *ICCV*, 2017.

[31] G. Yang, E. Fini, D. Xu, P. Rota, M. Ding, T. Hao, X. Alameda-Pineda, and E. Ricci, "Continual attentive fusion for incremen-

tal learning in semantic segmentation," *IEEE Transactions on Multimedia*, 2022.

[32] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, "Plop: Learning without forgetting for continual semantic segmentation," in *CVPR*, 2021.

[33] T. Feng, M. Wang, and H. Yuan, "Overcoming catastrophic forgetting in incremental object detection via elastic response distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9427–9436.

[34] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3400–3409.

[35] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *TPAMI*, 2023.

[36] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *ECCV*, 2020.

[37] Y. Cong, M. Zhao, J. Li, S. Wang, and L. Carin, "Gan memory with no forgetting," *NIPS*, 2020.

[38] X. Liu, C. Wu, M. Menta, L. Herranz, B. Raducanu, A. D. Bagdanov, S. Jui, and J. v. de Weijer, "Generative feature replay for class-incremental learning," in *CVPR Workshops*, 2020.

[39] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *NIPS*, 2017.

[40] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *CVPR*, 2018.

[41] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "Dytox: Transformers for continual learning with dynamic token expansion," in *CVPR*, 2022.

[42] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *CVPR*, 2017.

[43] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," 2017.

[44] J. Rajasegaran, M. Hayat, S. H. Khan, F. S. Khan, and L. Shao, "Random path selection for continual learning," *NIPS*, 2019.

[45] G. Shi, J. Chen, W. Zhang, L.-M. Zhan, and X.-M. Wu, "Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima," *Advances in neural information processing systems*, vol. 34, pp. 6747–6761, 2021.

[46] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, "Few-shot incremental learning with continually evolved classifiers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 455–12 464.

[47] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," *arXiv preprint arXiv:1805.08136*, 2018.

[48] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," *arXiv preprint arXiv:1807.05960*, 2018.

[49] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol *et al.*, "Meta-dataset: A dataset of datasets for learning to learn from few examples," *arXiv preprint arXiv:1903.03096*, 2019.

[50] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "Metafscil: A meta-learning approach for few-shot class incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14 166–14 175.

[51] K. Zhu, Y. Cao, W. Zhai, J. Cheng, and Z.-J. Zha, "Self-promoted prototype refinement for few-shot class-incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6801–6810.

[52] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," 2022.

[53] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *ICLR*, 2017.

[54] S. Liang and Y. Li, "Enhancing the reliability of out-of-distribution image detection in neural networks," *ICLR*, 2018.

[55] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *CVPR*, 2020.

[56] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *NIPS*, 2018.

[57] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" *ICLR*, 2019.

[58] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," *NIPS*, 2019.

[59] Z. Xiao, Q. Yan, and Y. Amit, "Likelihood regret: An out-of-distribution detection score for variational auto-encoder," *NIPS*, 2020.

[60] J. Yang, K. Zhou, and Z. Liu, "Full-spectrum out-of-distribution detection," *arXiv preprint arXiv:2204.05306*, 2022.

[61] E. Techapanurak, M. Suganuma, and T. Okatani, "Hyperparameter-free out-of-distribution detection using cosine similarity," in *ACCV*, 2020.

[62] X. Chen, X. Lan, F. Sun, and N. Zheng, "A boundary based out-of-distribution classifier for generalized zero-shot learning," in *ECCV*, 2020.

[63] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *ICML*, 2022.

[64] Z. Wang, Z. Zhang, C. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. G. Dy, and T. Pfister, "Learning to prompt for continual learning," *CVPR*, 2022.

[65] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy *et al.*, "Dualprompt: Complementary prompting for rehearsal-free continual learning," *ECCV*, 2022.

[66] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *CVPR*, 2019.

[67] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *CVPR*, 2019.

[68] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *CVPR*, 2020.

[69] V. Thengane, S. Khan, M. Hayat, and F. Khan, "Clip model is an efficient continual learner," 2022.

[70] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. Kira, "Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 909–11 919.

[71] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, "Rainbow

memory: Continual learning with a memory of diverse samples," in *CVPR*, 2021.

[72] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach, "Adversarial continual learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 386–402.

[73] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Toronto, ON, Canada*, 2009.

[74] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[75] H. Xiao, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[76] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng *et al.*, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 2. Granada, 2011, p. 4.

[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[78] P. Janson, W. Zhang, R. Aljundi, and M. Elhoseiny, "A simple baseline that questions the use of pretrained-models in continual learning," 2023.

[79] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 183–12 192.

[80] H. Liu, L. Gu, Z. Chi, Y. Wang, Y. Yu, J. Chen, and J. Tang, "Few-shot class-incremental learning via entropy-regularized data-free replay," in *European Conference on Computer Vision*. Springer, 2022, pp. 146–162.

Xusheng Cao is currently a Ph.D. candidate in the College of Computer Science, Nankai University, under the supervision of Assoc. Prof. Xialei Liu. His research interests include continual learning and few-shot learning.

Haori Lu is currently a master's student at the College of Computer Science, Nankai University, under the supervision of Assoc. Prof. Xialei Liu. His research interest is continual learning.

Xialei Liu is currently an associate professor at Nankai University. Before that, He was a postdoctoral researcher at the University of Edinburgh. He received his Ph.D. degrees from the Autonomous University of Barcelona in 2019, supervised by Prof. Joost van de Weijer and Prof. Andrew D. Bagdanov. He works in the field of computer vision and machine learning. His research interests include continual learning, self-supervised learning, and few-shot learning.

Mingming Cheng received his Ph.D. degree from Tsinghua University in 2012. Then, he did two years research fellow with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer vision, and image processing. He received research awards, including ACM China Rising Star Award, IBM Global SUR Award, and CCF-Intel Young Faculty Researcher Program. He is on the editorial boards of IEEE TPAMI/TIP.