# 用于非范例类增量学习的细粒度知识选择与恢复

**翟江天** [1] , **刘夏雷** [1,*], **余璐** [2], **程明明** [1]

[1] VCIP, CS, 南开大学

[2] 天津理工大学

Abstract

非范例类增量学习旨在在不访问任何过去训练数据的情况下同时学习新、旧任务。这一严格的限制使得减轻灾难性遗忘的难度加大，因为所有技术只能应用于当前任务的数据。针对这一挑战，我们提出了一种新颖的细粒度知识选择与恢复框架。传统的基于知识蒸馏的方法对网络参数和特征施加了过于严格的约束以防止遗忘，这限制了新任务的训练。为了放宽这一约束，我们提出了一种新颖的细粒度选择性分块级蒸馏方法，以自适应地平衡可塑性和稳定性。一些任务无关的图像分块可以用来保持旧任务的决策边界，而一些包含重要前景的分块则有利于学习新任务。此外，我们采用了一种与任务无关的机制，利用当前任务样本生成更真实的旧任务原型，以减少分类器偏差，从而实现细粒度知识恢复。在 CIFAR100、TinyImageNet 和 ImageNet-Subset 上的广泛实验验证了我们方法的有效性。代码可在 https://github.com/scok30/vit-cil 获取。

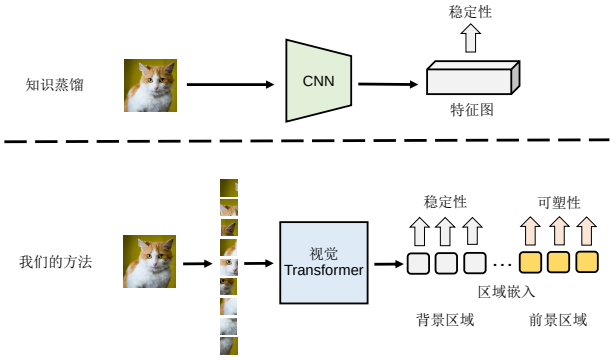


图 1: 传统的知识蒸馏（KD）与我们的基于视觉 Transformer 架构的分块级细粒度知识选择方法之间的比较。传统的 KD 将图像视为一个整体，而分块嵌入使我们能够在不同的局部区域之间实现更好的稳定性和可塑性平衡。此外，还提出了一种与任务无关的细粒度原型恢复方法，以更好地重现旧知识。

## 引言

深度神经网络在许多计算机视觉任务中表现出色。由于现实世界是开放且动态的，在应用这些网络期间，能够学习新知识是非常重要的。增量学习旨在在学习新任务的同时不忘记先前的任务，这也是深度神经网络在现实世界应用中一个至关重要的特性。例如，近年来人脸识别系统可能会遇到戴着口罩的人脸，适应这些新情况至关重要。然而，仅仅对深度神经网络进行新任务的微调会导致严重的灾难性遗忘 (Robins 1995)，因为网络几乎会完全将其参数调整到新任务上 (Goodfellow et al. 2013; McCloskey and Cohen 1989)。为了解决这个问题，许多近期的研究提出了缓解灾难性遗忘问题的方法。

在这篇论文中，我们考虑了一种具有挑战性的任务设置，称为非范例类别增量学习 (NECIL) (Gao et al. 2022; Zhu et al. 2021a, 2022)，该设置禁止模型在学习顺序任务时保留任何旧任务样本，而这些任务样本来自不相交的类别。与普通设置相比，非范例类别增量学习考虑了数据隐私问题和存储负担。为了解决这种任务中的灾难性遗忘问题，研究人员提出了各种方法，旨在不依赖过去的数据的情况下保留已获得的知识。

早期的工作，如 LwF (Li and Hoiem 2017)，引

*通讯作者 (xialei@nankai.edu.cn)。

入了旧模型和当前模型之间的基础知识蒸馏，但在这种具有挑战性的设置下表现不佳。一些基于重放的方法通常通过生成合成图像来缩小这一差距。然而，增量学习的效果可能会受到生成图像质量的影响。最近，DeepInversion (Yin et al. 2020) 提出从随机噪声中反向推断训练网络，以生成图像作为伪样本，与当前数据一起用于训练。R-DFCIL (Gao et al. 2022) 引入了关系引导的监督，以克服合成数据和真实数据之间严重的领域差距，从而生成更逼真的旧样本进行重放。与此相比，PASS (Zhu et al. 2021b) 关注分类器偏差的问题，并提出通过应用增强原型重放来维持旧任务的决策边界。SSRE (Zhu et al. 2022) 继承了这一设计，并引入了一种动态扩展的模型结构，以适应新任务知识，同时减少遗忘旧任务的负面影响。然而，这些方法中的大多数依赖于一个常见且基本的模块来减轻遗忘问题，即知识蒸馏 (Hinton, Vinyals, and Dean 2015)。

当知识蒸馏应用于 NECIL 时，它采用当前任务样本并减少旧模型与新模型之间的表示距离。其目的是提高模型在学习新任务时的稳定性，从而减少对旧知识的遗忘。然而，这一操作在 NECIL 设置下与学习当前任务部分矛盾，因为我们期望模型在新数据上具备可塑性。内在原因在于，之前的方法在样本层面处理这一蒸馏过程，其信息量较少，因为仅有图像中的部分区域，即前景，与当前分类任务高度相关。如果我们可以对任务相关区域和任务无关区域分别应用不同的策略，就能在可塑性和稳定性的权衡上实现更加细致的学习过程。换言之，模型被教会记住共同背景区域的表示，同时学习前景的判别性和任务相关知识。图 1 中展示了一个概念性图示。

NECIL 的另一个挑战是跨任务的分类器偏差，这一点在 PASS (Zhu et al. 2021b) 中有所描述。原型是输入图像的一个一维嵌入，用于分类，通常由网络计算得出（例如，通过对 CNN 特征图进行平均池化得到的结果，或通过视觉 Transformer 的 [CLS] 嵌入）。PASS 和 SSRE 中的原型重放通过增强类别中心（对该类所有样本原型取平均值）来合成旧任务的样本级原型，并使用这些原型来维持旧的决策边界。然而，如 (Ma et al. 2023) 中所提到的，样本原型不一定是正态分布的。由于深度神经网络（DNN）倾向于对新任务的训练样本过拟合，这些用于分类器重放的有偏差的合成原型可能导致分类器记住错误的决策边界，而不是保持最初的旧边界，从而导致更严重的遗忘。

为了解决上述两个问题，我们提出了一种新颖的 NECIL 框架，利用视觉 Transformers 的天然优势：其分块表示。我们的方法包括分块级知识选择和原型恢复。一方面，视觉 Transformer 为每个输入图像计算分块级表示。根据每个分块与 [CLS] 标记嵌入的相似度，模型感知每个分块与任务的相关性，并应用加权知识蒸馏。对于前景分块，模型倾向于减少正则化的强度，并鼓励在这些分块上保持可塑性。而背景分块可能与任务的相关性较小。因此，模型可以利用这些背景分块来在不同历史版本的模型之间保持一致的表示。

另一方面，考虑到 PASS 中的假设，即原型具有类似的数据分布（正态分布），我们将其扩展为两个步骤。首先，我们明确计算每个样本到其类别中心（即原型）的偏移距离，并对其进行正则化，以维持任务无关的数据分布。然后，我们利用这一特性，通过当前任务原型和旧类别原型来恢复旧任务的原型。这些操作减轻了由于在 PASS 中采用高斯模型而导致的不准确恢复旧原型的问题，获得更真实的原型用于分类器重放，从而缓解分类器偏差和遗忘。

我们的主要贡献可以概括如下：

(i) 我们提出了一种新颖的视觉 Transformer 框架用于 NECIL 任务，其分块级知识选择可以自然地应用，以实现网络可塑性和稳定性的更好权衡。

(ii) 我们采用了一种新颖的原型恢复策略，以生成更真实的合成原型，从而减轻分类器中的遗忘问题。

(iii) 在 CIFAR-100、TinyImageNet 和 ImageNet-Subset 上进行的广泛实验展示了我们框架的效果与优越性。我们方法的每个部分都可以轻松应用于其他相关工作，并显著提高性能。

# 相关工作

**增量学习**。增量学习涉及学习具有不同知识的序列任务，这些年来引起了广泛关注，包括多种方法 (Belouadah, Popescu, and Kanellos 2021; Delange et al. 2021; Liu et al. 2018, 2022; Zhou et al. 2023)。为了克服由于无法充分访问旧任务数据而导致的灾难性遗忘，iCaRL (Rebuffi et al. 2017) 在当前模型和旧模型的类别 logits 之间使用知识蒸馏。PODNet (Douillard et al. 2020) 进一步在主干网络中的每个中间特征图上应用蒸馏。

最近，视觉 Transformer 在图像分类及其衍生任务如类增量学习（CIL）中变得越来越流行。DyTox (Douillard et al. 2022) 将主干网络从 CNN 更换为视觉 Transformer，并引入任务相关的嵌入，以使模型适应增量任务。L2P 和 DualPrompt (Wang et al. 2022b,a) 通过动态提示来引导预训练的 Transformer 模型，使其能够在不同任务之间顺序学习。此外，(Zhai et al. 2023) 引入了掩码自编码器，以扩展用于类增量学习的重放缓冲区。在本文中，我们重新思考了视觉 Transformer 的特点，并将其扩展到 NECIL，以提供一种减轻灾难性遗忘的新见解：细粒度的分块级知识选择。

**非范例类增量学习**。非范例类增量学习（NECIL）在训练数据敏感且无法长期存储的情况下更为适用。DAFL 采用 GAN 生成过去任务的合成样本，避免了存储实际数据的需求 (Chen et al. 2019)。DeepInversion 是另一种 NECIL 方法，通过反转训练网络使用随机噪声生成图像 (Yin et al. 2020)。SDC 通过估计和利用原型漂移来解决在旧样本上训练新任务时的语义漂移问题 (Yu et al. 2020)。像 PASS 和 IL2A (Zhu et al. 2021b,a) 这样的方法通过生成旧类的原型而无需保留原始图像，从而提供高效的 NECIL。SSRE 引入了一种重新参数化方法平衡旧知识和新知识，自训练则利用外部数据作为 NECIL 的替代方案 (Zhu et al. 2022; Hou, Han, and Cheng 2021; Yu, Liu, and Van de Weijer 2022)。

在我们的方法中，NECIL 中分别的原型重放将原型中心分为任务相关原型中心和任务无关原型偏移，通过首先监督模型生成任务无关的偏移，然后利用这些偏移恢复旧类原型，从而提高了重放质量。这相较于 PASS (Zhu et al. 2021b) 中的标准方法，改进了原型生成。

# 方法

## 基础知识

**问题定义与分析**。类增量学习顺序地学习不同的任务。这些任务中的每一个都与先前的任务没有重叠的类。令 $t \in \{1, 2, ..T\}$ 表示增量学习任务，其中 $T$ 是所有任务的数量。训练数据 $D_t$ 包含 $C_t$ 个类别，具有 $N_t$ 个训练样本 $\{(x_t^i, y_t^i)\}_{i=1}^{N_t}$。$x_t^i$ 表示图像，$y_t^i \in C_t$ 是其类别标签。

大多数类增量学习的深度网络可以分为两个部分：特征提取器 $F_\theta$ 和分类器 $G_\phi$，其中分类器会随着每个新任务 $t+1$ 的增加而扩展，以包括类别 $C_{t+1}$。输入 $x$ 通过特征提取器 $F_\theta$ 转换为深度特征向量 $z = F_\theta(x) \in \mathbb{R}^d$，然后使用统一的分类器 $G_\phi(z) \in \mathbb{R}^{|C_t|}$ 以学习一个关于类别 $C_t$ 的概率分布，以预测 $x$ 的标签。

类增量学习要求模型在任何训练任务中都能对来自先前任务的所有已学习样本进行分类。换言之，模型在执行任务 $t$ 时应该保留对属于任务 $t' < t$ 的类别样本进行分类的能力。考虑到这些要求，非范例类增量学习（NECIL）施加了一个额外的约束，即模型必须在不使用任何来自先前任务样本的情况下学习每个新任务。大多数相关方法都以一个基本目标进行监督，该目标是最小化定义在当前训练数据 $D_t$ 上的损失函数 $\mathcal{L}_t^{\text{CIL}}$：

$$\mathcal{L}_t^{\text{CIL}}(x, y) = \mathcal{L}_t(G_{\phi_t}(F_{\theta_t}(x)), y). \tag{1}$$

**视觉 Transformer 架构**。DyTox (Douillard et al. 2022) 已经证明，视觉 Transformer 在 CIL 中是有效的，因为其动态任务相关的标记可以轻松地适应不同的任务。在本文中，我们发现了视觉 Transformer 的一个重要特性，它可以促进 CIL 并自适应地减轻新任务中的遗忘：即图像的分块级表示。以下是对视觉 Transformer 过程的回顾：

视觉 Transformer 首先将输入图像 $x$ 裁剪成 $K \times K$ 个不重叠的分块，我们用 $N$ 表示完整图

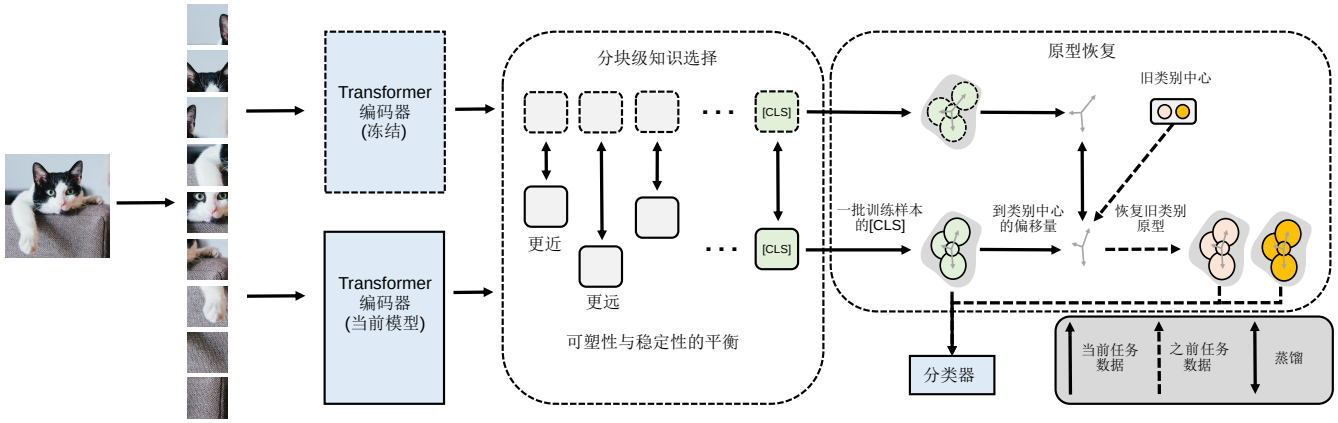


图 2: 我们方法的细粒度知识选择和恢复框架的示意图。分块嵌入通过不同的权重进行正则化。我们首先训练当前网络以保持与旧网络相似的原型分布，并用旧类中心和当前任务原型恢复旧原型。这两种原型被送入分类器，以减少任务偏差。

像 $x$ 中的分块数量。在此操作之后，这些分块通过一个 MLP 层映射到维度为 $d$ 的视觉嵌入。将这些嵌入与形状为 $\mathbb{R}^d$ 的类别标记 [CLS] 连接起来，得到一个大小为 $\mathbb{R}^{(N+1)\times d}$ 的张量。在对原始分块位置进行位置编码后，输入被传递到视觉 Transformer 的编码器。每个编码器 Transformer 块有两个顺序部分：自注意力层和前馈层。在每个部分之前都会应用层归一化。这些操作保持相同的嵌入尺寸，即 $\mathbb{R}^{(N+1)\times d}$，并为每个区域生成分块级表示。从处理结果中得到的类别标记嵌入可以用于分类，通过交叉熵损失 $\mathcal{L}_t^{\text{CIL}}$ 进行训练。我们采用线性分类器和 softmax 操作来预测每个学习类别的概率。

## 分块级知识选择 (PKS)

在学习任务 $t$ 时，仅有 $D_t$ 对模型可用。先前的 NECIL 方法中的基础知识蒸馏，如 PASS 和 SSRE，直接使用当前数据来维持任务间的稳定性。这种操作没有考虑当前任务样本与旧任务样本之间的语义差距，从而导致在减轻对旧任务的遗忘效果不佳。为了解决这个问题，我们重新思考了在分块化图像中使用知识蒸馏的方法。一个自然的想法是，在应用知识蒸馏时为每个分块分配不同的权重，因为它们对分类任务的重要性不同：前景区域的分块通常包含更多与任务相关的上下文，而背景中的分块则主要具有任务无关的像素和随机信息。

考虑到在 $D_t$ 上平衡模型的可塑性和稳定性的双边策略，我们将其分为两步实施：(a) 定义一个度量标准，用于评估每个分块与当前任务 $t$ 的相关性，(b) 对每个分块应用分块级知识选择，使用这些分块特定的权重来进行当前模型和旧模型之间的知识蒸馏。

为了简化问题而不失一般性，我们采用 [CLS] 标记的嵌入 $P_{t,cls}$ 和每个图像分块 $P_{t,i}$ 之间的 L2 距离来计算它们对任务 $t$ 的重要性：

$$W_i = \frac{1}{||P_{t,cls} - P_{t,i}||_2 + \epsilon}. \tag{2}$$

我们为接近 [CLS] 标记的分块分配更大的 $W_i$，并将每个 $W_i$ 除以它们的最大值进行归一化，得到 $w_i$。$\epsilon$ 设置为 $1e-8$ 以避免分母中出现零值。分块级知识选择 $\mathcal{L}_t^{pks}$ 被定义为：

$$\mathcal{L}_t^{pks} = \sum_{i=1}^{N} w_i ||P_{t,i} - P_{t-1,i}||_2 + ||P_{t,cls} - P_{t-1,cls}||_2, \tag{3}$$

$P_{t,i}$ 表示任务 $t$ 的模型计算得到的第 $i$ 个分块的嵌入表示，我们计算当前模型 $F_{\theta_t}$ 和旧模型 $F_{\theta_{t-1}}$ 之间的嵌入表示的 L2 距离。

## 原型恢复 (PR)

我们首先描述原型偏移和类别中心的定义。特征提取器 $F_\theta$ 从任务 $t$ 的输入图像 $X_t$ 中计算出表示

Algorithm 1: 训练过程的伪代码

**输入**: 任务数量 $T$，第 $t$ 个任务的训练样本 $D_t = (x_i, y_i)_{i=1}^{N}$，任务 $t$ 中类别 $k$ 的类别原型 $\mu_{t,k}$（在训练过程中维护），初始参数 $\Theta_0 = \{\theta_0, \phi_0\}$ 包含视觉 Transformer 的特征提取器 $F_{\theta_t}$ 和分类器 $G_{\phi_t}$ 的参数。CE 代表交叉熵损失。

**输出**: 模型 $\Theta_T$

for $t \in \{1, 2, ..., T\}$ do
  $\Theta_t \leftarrow \Theta_{t-1}$
  while 未收敛 do
    从 $D_t$ 中采样 $(x, y)$
    $P_{t,i}, P_{t-1,i} \leftarrow F_{\theta_t}(x), F_{\theta_{t-1}}(x)$
    $\mathcal{L}_t^{pks} \leftarrow$ 计算 $(P_{t,i}, P_{t-1,i})$ 依照 Eq. (3)
    $O_t, O_{t-1} \leftarrow P_{t,y} - \mu_{t,y}, P_{t-1,y} - \mu_{t,y}$
    $\mathcal{L}_{t,pr} \leftarrow \mathcal{L}_{mse}(O_t, O_{t-1})$ 依照 Eq. (4)
    $F_{t_{old},y_{old}} \leftarrow O_t + \mu_{t_{old},y_{old}}$ 依照 Eq. (5)
    $\mathcal{L}_t^{CIL} \leftarrow \mathcal{L}_t^{CE}\ (G_{\phi_t}(F_{t,y}, F_{t_{old},y_{old}}), y, y_{old})$
    通过最小化 Eq. (7) 中的 $\mathcal{L}_t^{\text{all}}$ 更新 $\Theta_t$
  end while
end for

$z \in \mathbb{R}^d$，该表示用于通过分类器 $G_\phi$ 预测类别标签。对于视觉 Transformer，我们采用 [CLS] 标记作为图像分类的表示。设 $N_{t,k}$、$X_{t,k}$ 和 $\mu_{t,k}$ 分别表示任务 $t$ 中类别 $k$ 的样本数量、图像集以及类别中心。而 $\mu_{t,k} = \frac{1}{N_{t,k}} \sum_{i=1}^{N_{t,k}} F_\theta(X_{t,k}^i)$，即对该类别的所有样本取平均值。

为了引入更真实且无遗忘的样本级原型，以减轻分类器的偏差，我们通过使用类别中心和当前样本来恢复旧任务的原型。我们将原型重放分为两个步骤：1) 引入监督机制，使这些原型偏移与任务无关；2) 利用这一特性恢复旧的样本级原型。

**任务无关原型偏移的监督**。首先，我们考虑当前任务和上一个任务的模型，即 $F_{\theta_t}$ 和 $F_{\theta_{t-1}}$，并应用偏移正则化。令 $bs$ 表示批大小，我们考虑批次中的训练样本 $(x_i^t, y_i^t), i = 1, 2, ..., bs$，并将它们随机划分为两个大小相同的子集 $S_1$ 和 $S_2$，每个子集的大小为 $\lfloor \frac{bs}{2} \rfloor$。然后，我们计算两个子集中样本的原型偏移：对于 $S_1$，计算 $\{O_{t,i} = F_{\theta_t}(x_i^t) - \mu_{t,y_i^t} | i \in S_1\}$；对于 $S_2$，计算 $\{O_{t-1,i} = F_{\theta_{t-1}}(x_i^t) - \mu_{t,y_i^t} | i \in S_2\}$。我们采用旧模型 $F_{\theta_{t-1}}$ 来计算子集 $S_2$ 的原型偏移，这样可以利用旧模型中包含的偏移分布的先前知识。我们从中随机采样 $\lfloor \frac{bs}{2} \rfloor$ 对原型偏移，并最小化它们之间的均方误差。

$$\mathcal{L}_t^{pr} = \frac{1}{sz} \sum_{(i_k, j_k) \in Idx} \mathcal{L}_{mse}(O_{t,i_k}, O_{t-1,j_k}), \tag{4}$$

其中 $sz = \lfloor \frac{bs}{2} \rfloor$，$Idx = (i_1, j_1), \ldots, (i_{sz}, j_{sz})$ $(i_k \in S_1, j_k \in S_2)$，$O_{t,i_k}$ 表示来自 $S_1$ 的第 $i_k$ 个原型偏移，而 $O_{t-1,j_k}$ 具有类似的含义。

**旧任务原型的恢复**。我们使用当前样本 $x_i^t$ 的原型偏移量以从旧任务中恢复原型:

$$F_{t_{old},k_{old}} = \mu_{t_{old},k_{old}} + (F_{\theta_t}(x_i^t) - \mu_{t,y_i^t}). \tag{5}$$

公式 5 中的第二项是当前样本的计算偏移。样本 $(x_i^t, y_i^t)$ 在当前批次中随机选择。它可以融入公式 6 中如下，

$$\mathcal{L}_t^{\text{CIL}} = \mathcal{L}_t^{CE}(G_{\phi_t}(F_{t,y}, F_{t_{old},y_{old}}), y, y_{old}), \tag{6}$$

其中 $\mathcal{L}_t^{CE}$ 是交叉熵（CE）损失。整体算法在 Alg. 1 中进行了说明。

## 学习目标

整体学习目标结合了分类损失、样本原型一致性损失和分块级知识选择:

$$\mathcal{L}_t^{\text{all}} = \mathcal{L}_t^{\text{CIL}} + \lambda_{pks}\mathcal{L}_t^{\text{pks}} + \lambda_{pr}\mathcal{L}_t^{\text{pr}}. \tag{7}$$

# 实验

**数据集**。我们在三个数据集上进行实验：CIFAR100、TinyImageNet 和 ImageNet-Subset，这些数据集在先前的研究中广泛使用。对于每个实验，我们首先从数据集中选择部分类别作为基础任务，然后将剩余的类别均匀分配到每个顺序任务中。这个过程可以表示为 $F + C \times T$，其中 $F$、$C$、$T$ 分别表示基础任务中的类别数、每个任务中的类别数以及任务数。对于 CIFAR100 和 ImageNet-Subset，我们采用三种配置：$50 + 5 \times 10$、$50 + 10 \times 5$、$40 + 20 \times 3$。对于 TinyImageNet，设置为：$100 + 5 \times 20$、$100 + 10 \times 10$ 和 $100 + 20 \times 5$。

| 数据集 | | | CIFAR100 | | | | | | TinyImageNet | | | | | | ImageNet-Sub | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 设置 | | | 5 任务 | | 10 任务 | | 20 任务 | | 5 任务 | | 10 任务 | | 20 任务 | | 10 任务 | |
| | 方法 | Para.(M) | Avg↑ | Last↑ | Avg↑ | Last↑ | Avg↑ | Last↑ | Avg↑ | Last↑ | Avg↑ | Last↑ | Avg↑ | Last↑ | Avg↑ | Last↑ |
| E=20 | iCaRL-CNN† | 11.2 | 51.07 | 40.12 | 48.66 | 39.65 | 44.43 | 35.47 | 34.64 | 22.31 | 31.15 | 21.10 | 27.90 | 20.46 | 50.53 | 41.08 |
| | iCaRL-NCM† | 11.2 | 58.56 | 49.74 | 54.19 | 45.13 | 50.51 | 40.68 | 45.86 | 33.45 | 43.29 | 33.75 | 38.04 | 28.89 | 60.79 | 51.90 |
| | LUCIR† | 11.2 | 63.78 | 55.06 | 62.39 | 50.14 | 59.07 | 48.78 | 49.15 | 37.09 | 48.52 | 36.80 | 42.83 | 32.55 | 66.16 | 56.21 |
| | EEIL† | 11.2 | 60.37 | 52.35 | 56.05 | 47.67 | 52.34 | 41.59 | 47.12 | 34.24 | 45.01 | 34.26 | 40.50 | 30.14 | 63.34 | 54.19 |
| | RRR† | 11.2 | 66.43 | 57.22 | 65.78 | 55.74 | 62.43 | 51.35 | 51.20 | 42.23 | 49.54 | 40.12 | 47.46 | 35.54 | 67.05 | 58.22 |
| E=0 | LwF_MC | 14.5 | 45.93 | 36.17 | 27.43 | 50.47 | 20.07 | 15.88 | 29.12 | 17.12 | 23.10 | 12.33 | 17.43 | 8.75 | 31.18 | 20.01 |
| | EWC | 14.5 | 16.04 | 9.32 | 14.70 | 8.47 | 14.12 | 8.23 | 18.80 | 12.71 | 15.77 | 10.12 | 12.39 | 8.42 | - | - |
| | MUC | 14.5 | 49.42 | 38.45 | 30.19 | 19.57 | 21.27 | 15.65 | 32.58 | 17.98 | 26.61 | 14.54 | 21.95 | 12.70 | 35.07 | 22.65 |
| | IL2A | 14.5 | 63.22 | 55.13 | 57.65 | 45.32 | 54.90 | 45.24 | 48.17 | 36.14 | 42.10 | 35.23 | 36.79 | 28.74 | - | - |
| | PASS | 14.5 | 63.47 | 55.67 | 61.84 | 49.03 | 58.09 | 48.48 | 49.55 | 41.58 | 47.29 | 39.28 | 42.07 | 32.78 | 61.80 | 50.44 |
| | SSRE | 19.4 | 65.88 | 56.33 | 65.04 | 55.01 | 61.70 | 50.47 | 50.39 | 41.67 | 48.93 | 39.89 | 48.17 | 39.76 | 67.69 | 57.51 |
| | Ours | 9.3 | 68.17 | 59.02 | 70.13 | 57.90 | 66.86 | 54.25 | 54.88 | 44.97 | 52.72 | 43.35 | 51.68 | 41.94 | 70.18 | 61.42 |

表 1: 在不同任务数量下，CIFAR100、TinyImageNet 和 ImageNet-Subset 上的平均准确率和最终准确率。基于重放的方法存储每个先前类别的 20 个样本，用符号 † 标记。最佳总体结果以粗体显示。

**对比方法**。我们将我们的方法与其他非范例类增量学习方法进行比较：SSRE (Zhu et al. 2022)、PASS (Zhu et al. 2021b)、IL2A (Zhu et al. 2021a)、EWC (Kirkpatrick et al. 2017)、LwF-MC (Rebuffi et al. 2017) 和 MUC (Liu et al. 2020)。我们还与几种基于范例的方法进行比较，如 iCaRL（最近均值和 CNN）(Rebuffi et al. 2017)、EEIL(Castro et al. 2018) 和 LUCIR (Hou et al. 2019)。

**实现细节**。关于视觉 Transformer 的结构，我们在编码器中使用了 5 个 Transformer 块，在解码器中使用了 1 个，比 Vit-Base 的原始版本要轻量得多。所有 Transformer 块的嵌入维度为 384，具有 12 个自注意力头。我们为每个任务训练了 400 个轮次。在任务 $t$ 之后，我们为每个类别保存一个平均原型（类别中心）。在实验中，我们将 $\lambda_{pks}$ 和 $\lambda_{pr}$ 设置为 10。我们报告 CIL 任务的三种常见指标：在学习最后一个任务后，所有已学习任务的平均和最终 top-1 准确率，以及迄今为止学习的所有类别的平均遗忘率。我们用 $Acc_i$ 表示在任务 $i$ 之后所有已学习类别的准确率。然后，平均准确率定义为 $Avg_{acc} = \frac{\sum_{i=1}^{T} Acc_i}{T}$，最终准确率为 $Acc_T$。设 $a_{m,n}$ 表示在学习任务 $m$ 后任务 $n$ 的准确率。任务 $i$ 在学习任务 $k$ 后的遗忘度量 $f_k^i$ 被计算为 $f_k^i = \max_{t\in 1,2,\ldots,k-1}(a_{t,i} - a_{k,i})$。平均遗忘率 $F_k$ 定义为 $F_k = \frac{1}{k-1}\sum_{i=1}^{k-1} f_k^i$。所有实验均运行三次，并报告平均性能。

| 数据集 | TinyImageNet | | | ImageNet-Subset |
|---|---|---|---|---|
| 方法 | 5 任务 | 10 任务 | 20 任务 | 10 任务 |
| LwF_MC | 54.26 | 54.37 | 63.54 | 56.07 |
| EWC | 67.55 | 70.23 | 75.54 | 71.97 |
| MUC | 51.46 | 50.21 | 58.00 | 53.85 |
| IL2A | 25.43 | 28.32 | 35.46 | 32.43 |
| PASS | 18.04 | 23.11 | 30.55 | 26.73 |
| SSRE | 9.17 | 14.06 | 14.20 | 23.22 |
| Ours | 11.45 | 12.21 | 12.82 | 18.39 |

表 2: 与其他方法的平均遗忘率比较。实验在 CIFAR100、TinyImageNet 和 ImageNet-Subset 上进行，任务数量为 5、10 和 20。

## 与最先进方法的对比

在表 1 中，我们将我们的方法与几种非范例和基于范例的方法进行了比较。在非范例设置中，我们

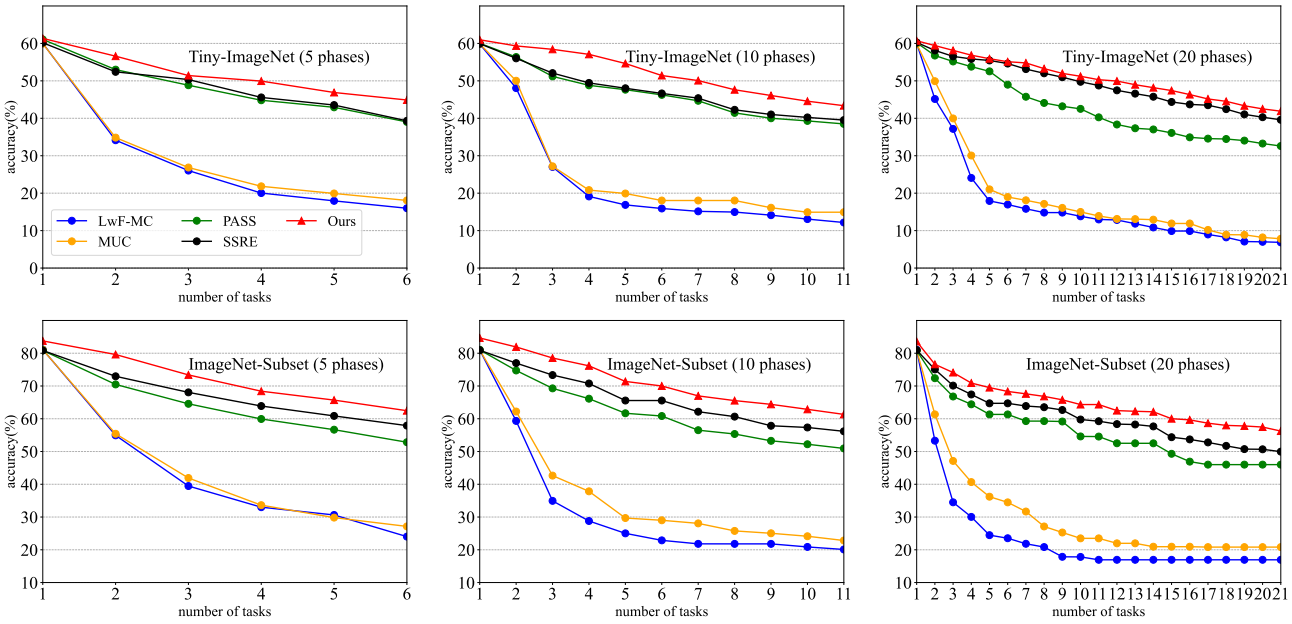


图 3: 在 TinyImageNet 和 ImageNet-Subset 上不同任务数量的结果表明，我们的方法优于其他方法。

| 方法 | PKS | PR | 5 任务 | 10 任务 | 20 任务 |
|---|---|---|---|---|---|
| PASS | - | - | 55.67 | 49.03 | 48.48 |
| Baseline (ViT) | | | 56.44 | 51.90 | 51.20 |
| | ✓ | | 58.27 | 56.82 | 52.87 |
| | | ✓ | 57.78 | 54.61 | 51.51 |
| | ✓ | ✓ | 59.02 | 57.90 | 54.25 |

表 3: 对我们提出的方法各个组成部分的消融实验。实验在 CIFAR-100 上进行，我们报告 top-1 准确率（以百分比表示）。我们用 PKS 和 PR 分别表示分块级知识选择和原型恢复。

发现我们的方法在所有三个数据集的不同数据划分设置（5/10/20 任务）下，均优于所有先前的相关方法。

以 20 个任务的结果为例，我们在 CIFAR100 的 20 任务设置下（最终平均准确率）超越了最佳非范例方法 SSRE 3.31%。此外，我们的方法甚至取得了比使用存储样本来减轻遗忘的所有基于范例的方法更高的准确率。这一现象在分辨率更大的数据集（如 TinyImageNet 和 ImageNet-Subset）中仍然保持不变。对于表 2 中报告的平均遗忘率，我们的方法优于大多数非基于范例的方法。在 ImageNet-Subset 数据集上，差距（高达 4.17%）更加明显。这从另一个角度证明了我们的方法在增量训练过程中的优越性能。我们还在图 3 中展示了动态准确率曲线，结果显示，所提方法（红色部分）在所有训练阶段的下降速度更慢。

## 消融实验

**各组成部分**。我们的方法由两个组成部分构成：分块级知识选择和原型恢复。我们在表 3 中分析了每个方面的影响。一种基线是使用基础知识蒸馏和原型增强在 PASS 中进行训练的。我们将视觉 Transformer（ViT）训练视为另一种基线。我们发现：(a) 分块级知识选择显著提高了性能，提升幅度为 3.71%。(b) 原型恢复也带来了一定的提升，提供了更真实的原型重放。(c) 这两个因素可以相互协作，实现更高的性能。这验证了在非范例类增量学习设置中，我们的分块级知识选择和原型恢复两者的重要性。

我们在 PASS 中引入了两个模块来进行该主干网络的实验：原型增强和自监督，如表 3 第二行所列。例如，与 PASS 中的原始网络（即 ResNet18）相比，我们的框架在性能上相似或略高于前两行的结果，表明视觉 Transformer 可以作为进一步研究

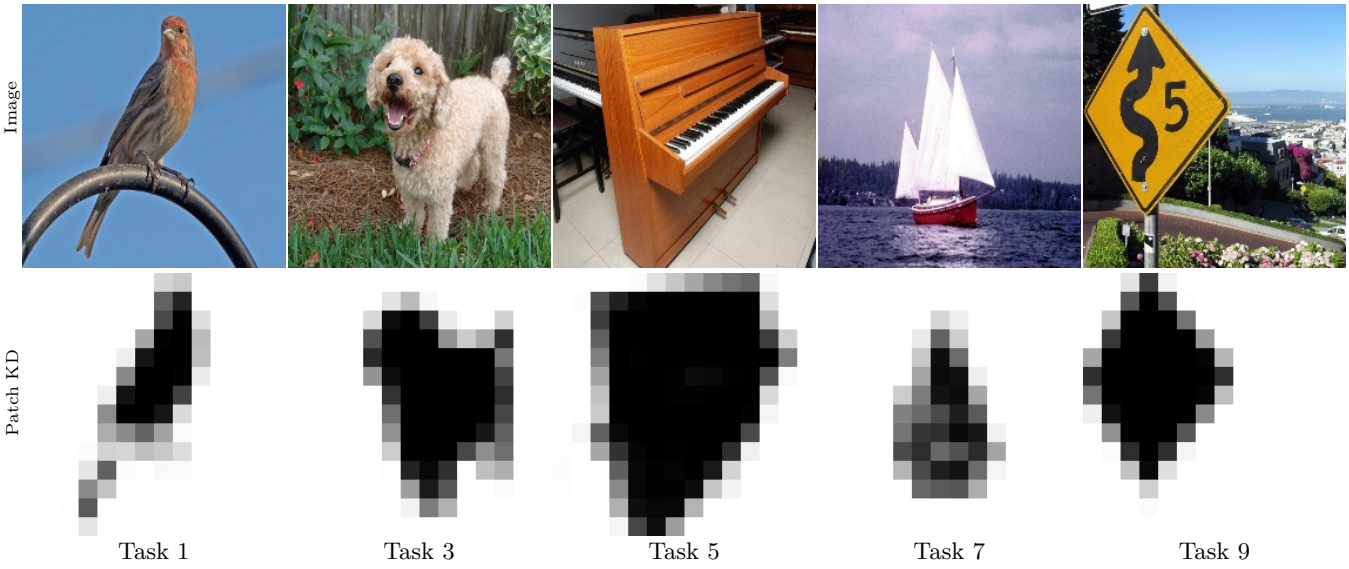


图 4: 我们的分块级知识选择的可视化及其与基础知识蒸馏的比较。白色分块表示在保持稳定性方面具有更大蒸馏权重。

| 设置 | $N$=10 | | | $N$=20 | | |
|---|---|---|---|---|---|---|
| | Avg↑ | Last↑ | $F\downarrow$ | Avg↑ | Last↑ | $F\downarrow$ |
| DyTox | 75.47 | 62.10 | 15.43 | 75.10 | 59.41 | 21.60 |
| Ours | 78.35 | 66.47 | 13.12 | 77.63 | 63.90 | 15.76 |

表 4: 在 CIFAR-100 上应用 Dytox 的结果，包括平均准确率（%）、最终准确率（%）和遗忘率 $F$（%），针对 10 任务和 20 任务的场景。

| $W_i$ | 5 任务 | 10 任务 | 20 任务 |
|---|---|---|---|
| 1 | 56.18 | 51.99 | 50.53 |
| $\|\|P_{t,cls}-P_{t,i}\|\|_2$ | 53.81 | 49.78 | 48.31 |
| Eq. 2 (Ours) | 59.02 | 57.90 | 54.25 |

表 5: 实验在 CIFAR-100 上进行，我们报告 top-1 准确率（以百分比表示）。第一行将所有 $W_i$ 设置为 1，第二行则使用与分块嵌入和 [CLS] 嵌入的距离成比例的 $W_i$。

的新基线。这也展示了我们提出的两个模块对视觉 Transformer 的影响，而非更强的基线。由于 SSRE 中的动态结构重组（DSR）是针对卷积层设计的，因此我们没有进行相关实验。

**分块级知识选择的进一步研究**。由于我们的分块级知识选择是基于视觉 Transformer 的基础知识蒸馏的简单但有效的扩展，我们将其应用于 DyTox 和基于范例的任务，以验证其在更多视觉 Transformer 方法和问题设置中的通用性。在表 4 的每个实验中，我们按照 DyTox 的要求存储每个学习类别的 20 个范例，以便进行公平比较。基础知识蒸馏被我们的分块级知识选择所替代。我们观察到，所提出的方法在平均/最终平均准确率和遗忘率方面显著优于原始 DyTox。这进一步证明了我们在细粒度分块蒸馏方面的有效性。与基础知识蒸馏相比，我们的方法通过对不同分块使用自适应权重来保留知识，为模型在可塑性和稳定性的权衡之间提供了更大的灵活性。

此外，考虑到我们的分块级知识选择方法在蒸馏过程中对每个分块嵌入使用不同的权重，我们还进行了比较实验，以评估该策略在两种不同设置中的有效性。第一个实验将所有蒸馏权重 $W_i$ 设置为 1，第二个实验则通过 $W_i = ||P_{t,cls} - P_{t,i}||_2$ 来计算权重，以替代 Eq. 2。根据表 5 中第一个设置的结果，我们发现对所有分块的蒸馏施加相同的权重导致性能比我们的第三行结果更差。我们认为这种严格的限制虽然保留了更多关于先前任务的信息，但却损害了

当前任务的学习能力。此外，对距离 [CLS] 嵌入更远的嵌入施加更大的蒸馏权重（与我们的方法相反）导致的结果也比基线更差。这两个设置的结果证明了在与任务相关的分块上保持可塑性以及在与任务无关的分块上保持稳定性的重要性。

**分块级知识选择的可视化**。我们可视化了一些示例，并展示了在不同任务中我们的分块级知识选择所应用的实际权重，如图 4 所示。这些图像选自 ImageNet-Subset 的 10 任务设置。与基础知识蒸馏对每个分块使用相同权重不同，我们的方法可以自适应地选择一些背景补丁以保持稳定性，同时在前景分块上提供更多可塑性，以学习与任务相关的知识。

## 总结

本文介绍了一种新颖的框架，利用视觉 Transformer 进行非范例类增量学习（NECIL），以减少灾难性遗忘和分类器偏差。该框架利用分块嵌入在不同区域实现稳定性与可塑性之间的平衡，并采用独特策略处理与任务相关和与任务无关的区域。此外，还引入了一种新的原型恢复模块，以保留旧任务的决策边界而不引入分类器偏差。该框架为未来的研究提供了一种潜在的基线。

## 鸣谢

该项目得到国家自然科学基金（NO. 62225604, 62206135, 62202331）以及中央高校基本科研业务费专项资金资助（南开大学，070-63233085）。算力由南开大学超算中心提供。

## 参考文献

Belouadah, E.; Popescu, A.; and Kanellos, I. 2021. A comprehensive study of class incremental learning algorithms for visual tasks. Neural Networks, 135: 38–54.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In ECCV.

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-free learning of student networks. In ICCV.

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

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In ECCV.

Douillard, A.; Ramé, A.; Couairon, G.; and Cord, M. 2022. Dytox: Transformers for continual learning with dynamic token expansion. In CVPR.

Gao, Q.; Zhao, C.; Ghanem, B.; and Zhang, J. 2022. R-DFCIL: Relation-Guided Representation Learning for Data-Free Class Incremental Learning. ECCV.

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6211.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Hou, Q.; Han, L.; and Cheng, M.-M. 2021. Autonomous Learning of Semantic Segmentation from Internet Images (in Chinese). Sci Sin Inform, 51(7): 1084–1099.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In CVPR.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. IEEE TPAMI.

Liu, X.; Hu, Y.-S.; Cao, X.-S.; Bagdanov, A. D.; Li, K.; and Cheng, M.-M. 2022. Long-tailed class incremental learning. In European Conference on Computer Vision, 495–512. Springer.

Liu, X.; Masana, M.; Herranz, L.; Van de Weijer, J.; Lopez, A. M.; and Bagdanov, A. D. 2018. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In 2018 24th International Conference on Pattern Recognition (ICPR), 2262–2268. IEEE.

Liu, Y.; Parisot, S.; Slabaugh, G.; Jia, X.; Leonardis, A.; and Tuytelaars, T. 2020. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In ECCV.

Ma, C.; Ji, Z.; Huang, Z.; Shen, Y.; Gao, M.; and Xu, J. 2023. Progressive Voronoi Diagram Subdivision Enables Accurate Data-free Class-Incremental Learning. In ICLR.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation, volume 24, 109–165. Elsevier.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In CVPR.

Robins, A. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. Connection Science.

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

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. Learning to prompt for continual learning. In CVPR.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In CVPR.

Yu, L.; Liu, X.; and Van de Weijer, J. 2022. Self-training for class-incremental semantic segmentation. IEEE Transactions on Neural Networks and Learning Systems.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic drift compensation for class-incremental learning. In CVPR.

Zhai, J.-T.; Liu, X.; Bagdanov, A. D.; Li, K.; and Cheng, M.-M. 2023. Masked Autoencoders are Efficient Class Incremental Learners. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 19104–19113.

Zhou, D.-W.; Wang, Q.-W.; Qi, Z.-H.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2023. Deep class-incremental learning: A survey. arXiv preprint arXiv:2302.03648.

Zhu, F.; Cheng, Z.; Zhang, X.-y.; and Liu, C.-l. 2021a. Class-Incremental Learning via Dual Augmentation. NIPS.

Zhu, F.; Zhang, X.-Y.; Wang, C.; Yin, F.; and Liu, C.-L. 2021b. Prototype augmentation and self-supervision for incremental learning. In CVPR.

Zhu, K.; Zhai, W.; Cao, Y.; Luo, J.; and Zha, Z.-J. 2022. Self-Sustaining Representation Expansion for Non-Exemplar Class-Incremental Learning. In CVPR.