[CVPR2020] 基于元学习的少样本开集识别

Bo Liu
UC, San Diego
boliu@ucsd.edu

Hao Kang
Wormpex AI Research
haokheseri@gmail.com

Haoxiang Li
Wormpex AI Research
lhxustcer@gmail.com

Gang Hua
Wormpex AI Research
ganghua@gmail.com

Nuno V asconcelos
UC, San Diego
nuno@ece.ucsd.edu

摘要

研究了开集识别问题。虽然以前的方法只在大规模分类器训练的背景下考虑这个问题，但我们寻求一个统一的解决方案，用于此和 few-shot 分类设置。有人认为，经典的 softmax 分类器对于开集识别来说是一个糟糕的解决方案，因为它倾向于在训练类上过度拟合。然后提出随机化作为这个问题的解决方案。这建议使用元学习技术，通常用于 few-shot 分类，以解决开放集识别。然后介绍了一种新的开集元学习（oPen sEt mEta LEaRning, PEELER）算法。这结合了每 episode 一组新类的随机选择、使这些类示例的后验熵最大化的损失，以及基于马氏距离的新度量学习公式。实验结果表明，无论是 few-shot 识别还是大规模识别，PEELER 都达到了目前最先进的开放集识别性能。在 CIFAR 和 miniImageNet 上，对于给定的 SEED 类别分类精度，它在可见/不可见类别检测 AUROC 方面取得了巨大的进步。

1. 引言

深度卷积神经网络（CNN）的引入促进了计算机视觉的巨大进步。由于引入了大规模数据集（如 ImageNet[3]），包含许多类和每个类的许多示例，因此这些进步中的大多数可以追溯到对象识别方面的进步。许多现代计算机视觉体系结构完全或部分基于识别网络。在大规模环境下，通过交叉熵损失和小批量 SGD 训练的基于 CNN 的分类器具有优异的识别性能，在大多数识别基准上实现了最先进的结果。

然而，识别设置的变化可能导致性能大幅下降。众所周知的例子包括 few-shot 学习 [26,4,34,30]，其中每个类只有少数训练样本；域自适应 [33]，其中一个受训于源图像域的分类器 (例如合成图像) 必须部署到目标域 (例如自然图像)，其统计数据与源的统计数据不同；长尾识别 [19]，其中每个类的样本数量是高度不平衡的，或标签噪声问题 [21,31]。所有这些可供选择的识别设置都强调了大规模分类器的鲁棒性或概括能力。

Typical Recognition Paradigms	No. of Samples per Training Class	Supports unseen Class in Testing?
Closed-set [14]	Large	No
Few-shot [26, 4, 34, 30]	Small	No
Open-set [6, 1]	Large	Yes
Few-Shot Open-set	Small	Yes

Table 1. Comparison between different recognition tasks.

最近，人们有兴趣赋予 CNN 自我意识能力。本质上，自我意识意味着与人类一样，CNN 应该识别自己能做什么，拒绝自己不能做的。已经提出了该问题的几种变体，包括分布外检测 [10]，其中 CNN 拒绝训练分布以外的示例（例如，来自其他域的图像或对抗性攻击 [12]）；现实分类 [20]，其中拒绝其认为难以分类的示例；或开放集识别 [1]，要拒绝的例子来自在训练过程中看不到的新的类别。虽然已经提出了几种技术，但一种流行的方法是强制 CNN 在排斥区域产生高熵后验分布。然后，拒绝可以被认为是产生这种分布的例子。

在这项工作中，我们考虑开放集识别。这主要是在大规模环境中解决的，使用基于大规模分类器的解决方案。这些方法试图通过后处理后验类分布 [1] 来识别新类，定义一个“拒绝”类，该类通过人工生成 [6] 或从训练类中取样 [29] 或两者的组合来训练。我们试图将开放集识别推广到大规模环境之外，建议它应该针对一系列包含所有可选识别设置的任务。由于在许多环境下的训练是复杂的，我们认为这两个极端的连续体：large-scale 识别和 few-shot 识别。

这主要有三个原因。首先，在各种情况下，开放集识别都是一项挑战。一个在 few-shot 制度下训练的识别者，面对看不见的类别的可能性并不小。因此，一种支持 few-shot 设置的开放集识别技术比不支持 few-shot 设置的技术更有用。第二，由于标记数据的稀缺性，few-shot 开集识别比大规模开集识别更难解决。因此，few-shot 设置对开放式设置识别研究提出了更大的挑战。第三，像开集识别一样，few-shot 识别的主要挑战是在训练过程中对未见的数据做出准确的决策。由于这使得鲁棒性成为 few-shot 架构的主要特征，因此这些架构可能也

擅长于开集识别。最值得注意的是，它们可能会击败大规模分类器，而大规模分类器往往在鲁棒性方面得分不高。

大规模分类器 (用交叉熵进行训练) 的主要缺陷是容易在训练类别中过拟合。因为理想的分类嵌入将每个类的所有示例映射到特征空间中的一个唯一点，大规模分类器倾向于学习仅局部精确的嵌入。例如图像检索 [8]、人脸识别 [18]、姿势不变识别 [11]、人物再识别 [39] 或少量镜头学习 [34]。源自度量学习文献 [38] 或针对少数镜头识别等问题明确设计的嵌入 [37] 倾向于在整个特征空间更广泛地捕获数据的度量结构，并在这些泛化关键任务上获得更好的性能。虽然嵌入的度量结构由类代表（softmax 层的参数向量）邻域中的语义距离反映，但这些距离与后者无关。结果，大规模分类器嵌入在需要超过训练集泛化的任务上表现不佳，例如图像检索 [8]、人脸识别 [18]、姿态不变识别 [11]、人物重新识别 [39] 或 few-shot 学习 [34]。源自度量学习文献 [38] 或针对少数镜头识别等问题明确设计的嵌入 [37] 倾向于在整个特征空间更广泛地捕获数据的度量结构，并在这些泛化关键任务上取得更好的性能。因此，预计这些嵌入将在开集识别问题上优于大规模分类器。

在这项工作中，我们研究了一类流行的解决 few-shot 问题的解决方案的开放集性能，称为元学习（meta learning，ML）[34]。ML 方法用 episodic 训练取代了传统的小批量 CNN 训练。通过对类的一个子集和每个类一个子集示例进行随机抽样，生成一个支持集和查询集，对其应用分类损失。这将使分类任务随机化，在每个 ML 步骤中对嵌入进行优化，从而生成更健壮的嵌入，减少对任何特定类集的过度拟合。我们将这一思想推广到开集识别，通过在每个 episode 中随机选择一组新类，并为这些类的样例引入最大后验熵的损失。这迫使嵌入更好地解释看不见的类，在目标类区域之外产生高熵后验分布。

这个解决方案至少有两个好处。首先，由于它借鉴了最先进的方法来进行少样本学习，它立即将开放集识别扩展到了少样本问题。其次，由于 ML 嵌入是鲁棒的，而开集识别是泛化的，因此即使在大规模环境下，后者的性能也会提高。通过各种开放集识别基准的大量实验证明了这一点，其中我们展示了相对于现有技术的显著改进。我们还研究了特征空间中使用的度量对开集识别性能的作用，表明特定形式的马氏距离比常用的欧几里德距离具有显著的增益。

总的来说，这项工作的贡献可以总结如下：

- 一种新的基于 ML 的开集识别公式。这将开放集推广到 few-shot 识别环境。
- 一种新的 episodic 训练方法，结合交叉熵损失和一种新的开集损失，以提高开集在大规模和 few-shot 环境下的性能。
- 基于高斯嵌入的 ML 开集识别。

2. 相关工作

对不同结构中的泛化: 开放集识别处理分类设置，在该分类设置中，推理可以面对来自训练过程中看不见的类的样本。目标是赋予开放集分类器一种拒绝此类样本的机制。最早的深度学习方法之一是 Scheirer 等人 [1] 的工作，该工作为分类器生成的 Logit 提出了极值参数再分配方法。后来的工作考虑了判别或生成模型的问题。Schlachter 等人 [29] 提出了一种类内分割方法，其中使用封闭集分类器将数据分割为典型和非典型子集，将开放集识别重新表述为传统的分类问题。G-OpenMax[6] 利用一个经过训练的生成器，从一个表示所有未知类的额外类中合成示例。Neal 等人 [22] 介绍了反事实图像生成，其目的是生成无法分类为任何可见类别的样本，为分类器训练生成额外的类别。

所有这些方法都将一组看不见的类减少为一个额外的类。虽然开放样本可以从不同类别中提取，并且具有显著的视觉差异，但它们假设特征提取器可以将它们全部映射到单个特征空间簇中。虽然理论上是可能的，但实际很困难。相反，我们允许每个可见类有一个集群，并将不属于这些集群的示例标记为不可见。我们相信这是一种更自然的方法来检测看不见的类。

Out-of-Distribution: 与开集识别类似的一个问题是检测不符合分布 (OOD) 的例子。通常 [10] ，这是作为检测来自不同数据集的样本而制定的，即不同于用于训练模型的分布。而 [10] 通过直接使用 softmax 得分解决了这个问题，后来的工作 [17,35] 通过提高可靠性来改善结果。这个问题与开集识别的不同之处在于 OOD 样本不一定来自于不可见类。例如，它们可能是来自视觉类的样本的扰动版本，这在对抗性攻击文献中很常见。唯一的限制是它们不属于训练分布。通常情况下，这些样本比不可见类别的样本更容易被检测到。在文献中，它们往往是来自其他数据集的类，甚至是噪音的图像。这与开放式识别不同，在开放式识别中，不可见类往往来自相同的数据集。

few-shot 学习: 近年来出现了大量关于 few-shot 学习的研究 [26,4,34、30,32,27、7,23,16、2]。这些方法可以大致分为两个分支：优化和基于度量的方法。基于优化的方法通过展开反向传播过程来处理泛化问题。具体而言，Ravi 等人 [26] 提出了一个学习者模块，该模块经过培训以适应新任务。MAML[4] 及其变体 [5] 提出了一种训练程序，其中参数根据次级梯度计算的损失进行更新。基于度量的方法试图比较支持样本和查询样本之间的特征相似性。Vinyals 等人 [34] 介绍了 episode 训练的概念，其中训练程序旨在模拟测试阶段，该阶段与余弦距离一起用于训练循环网络。原型网络 [30] 通过结合度量学习和交叉熵损失，引入了由支持集特征构建的类原型。关系网络 [32] 利用神经网络隐式地探索了一对支持和查询特征之间的关系，而不是直接在特征空间上构建度量。

由于特征空间度量对于开放集识别也是有用的，我们主要关注基于度量的 few-shot 学习方法。虽然已有几种方法在 few-shot 任务中取得了很好的效果，但最终的分类器能否成功地剔除不可见样本仍不清楚。在这

项工作中，我们探讨了在大规模和少样本的情况下这个问题。

无遗忘学习：有几个工作 [7,24,25，28] 将传统的“few-shot”延伸到了无遗忘学习。该模型被训练用于处理额外的 few-shot 问题，并保持其在原始识别问题上的性能。我们的工作将集中在传统的几个问题上，这个方向超出了我们的范围，将来可以讨论。

3. 分类任务与元学习

在本节中，我们将讨论不同的分类设置（第 3.1）和元学习（第 3.2），它激励并为所提出的解决方案提供了基础，以在大规模和 few-shot 设置中实现开放集识别。

3.1. 分类设置

Softmax 分类. 用于对象识别和图像分类的最流行的深度学习体系结构是 softmax 分类器。这包括将图像 $\mathbf{x} \in \mathcal{X}$ 映射到特征向量 $f_\phi(\mathbf{x}) \in \mathcal{F}$（由多个神经网络层实现）的嵌入，以及使用线性映射估计类后验概率的 softmax 层，该线性映射使用下式估计类后验概率：

$$p(y = k|\mathbf{x}; \phi, \mathbf{w}_k) = \frac{exp(\mathbf{w}_k^T f_\phi(\mathbf{x}))}{\sum_{k'} exp(\mathbf{w}_{k'}^T f_\phi(\mathbf{x}))} \tag{1}$$

其中 ϕ 表示所有嵌入参数，$\mathbf{w}_k$ 是一组分类器权重向量。最近的工作 [30] 将度量学习与 softmax 分类相结合，使用如下距离函数实现 softmax 层：

$$p_\phi(y = k|\mathbf{x}) = \frac{exp(-d(f_\phi(\mathbf{x}), \mu_k))}{\sum_{k'} exp(-d(f_\phi(\mathbf{x}), \mu_{k'}))} \tag{2}$$

其中 $\mu_k = E[f_\phi(\mathbf{x})|y = k]$[30]。softmax 训练器通过训练集 $\mathbb{S} = (x_i^s, y_i^s)_{i=1}^{n^s}$ 学习，其中 $y_i^s \in \mathbb{C}^s, \forall i; \mathbb{C}^s$ 是一组训练图像类，n^S 是训练样例数。这包括找到使得交叉熵（如下所示）损失最小化的分类器和嵌入的参数。

$$\mathcal{L}_{CE} = \sum_{(x_i^s, y_i^s)} -logp(y_i^s|\mathbf{x}_i^s) \tag{3}$$

在测试集 $\mathbb{T} = (x_i^t, y_i^t)_{i=1}^{n^t}$ 上评估性能，其中 $y_i^t \in \mathbb{C}^t, \forall i; \mathbb{C}^t$ 是一组测试类，n^t 为测试样例数。

闭集分类 vs 开集分类. 在传统的分类定义下，训练类和测试类是相同的，即 $\mathbb{C}^s = \mathbb{C}^t$, 这被表示为闭集分类。最近，人们对另一种开集分类设置感兴趣，其中 $\mathbb{C}^t = \mathbb{C}^s \cup \mathbb{C}^u$。在这种情况下，$\mathbb{C}^s$ 中的类表示为可见类（在训练中出现的类），$\mathbb{C}^u$ 中的类表示为不可见类。

大规模识别 vs 小样本识别. 在大规模识别环境中，训练示例的数量 n^s 相当大，像 ImageNet 这样的数据集能达到数百万。相反，对于很少的 few-shot 识别，这个数字是相当小的，通常每类不到二十个例子。每类 K 个训练样本的 few-shot 问题通常被称为 K-shot 识别。请注意，测试示例的数量 n^t 在区分这两种设置方面没什么用。由于这些样例仅用于性能评估，测试集在这两种设置下应具有相同的基数。如表 1所示，培训数据的规模和覆盖范围的不同属性组合定义了不同的分类范式。few-shot 的开放式场景在文献和本文的研究重点中大多是未被探索的。

3.2. 元学习

元学习（ML）解决了“学会学习”的问题。在这种情况下，元学习器通过检查许多学习问题来学习学习算法。为此，元学习器依赖于元训练集 $\mathbb{MS} = (\mathbb{S}_i^s, \mathbb{T}_i^s)_{i=1}^{N^s}$ 其中 $(\mathbb{S}_i^s, \mathbb{T}_i^s)$ 是第 i 个学习问题的训练集和测试集，N^s 是用于训练的学习问题的数量；元测试集 $\mathbb{MT} = (\mathbb{S}_i^t, \mathbb{T}_i^t)_{i=1}^{N^t}$，其中 $(\mathbb{S}_i^t, \mathbb{T}_i^t)$ 是第 i 个测试问题的训练和测试集,N^t 是用于测试的学习问题的数量。给定 $\mathbb{MS}$，元学习器学习如何将一对 $(\mathbb{S}, \mathbb{T})$ 映射到一个利用 $\mathbb{S}$ 优化求解 $\mathbb{T}$ 的算法中。

程序如下。在元迭代 i 中，用前一个元迭代生成的模型初始化元模型 h。然后执行两个步骤。首先，元学习算法通过映射

$$h' = \mathcal{M}(h, \mathbb{S}_i^s) \tag{4}$$

得到训练集 $\mathbb{S}_i^s$ 的最优模型估计 h'。然后，使用适当的优化程序（例如反向传播），使用测试集 $\mathbb{T}_i^s$ 来寻找适当损失函数 L（例如交叉熵）的模型

$$h^* = \arg\min_h \sum_{(x_k, y_k) \in \mathbb{T}_i^s} L[y_k, h'(x_k)] \tag{5}$$

最后返回 h^* 作为元迭代 i 的最佳元模型。在测试过程中，元学习器用元测试集 $\mathbb{MT}$ 中的最终元模型 h^* 和训练集 $\mathbb{S}_i^t$ 生成一个新模型

$$h'' = \mathcal{M}(h^*, \mathbb{S}_i^t) \tag{6}$$

其性能通过 $\mathbb{T}_i^t$ 进行评估。

用于 few-shot 识别的元学习. 虽然已经提出了将 ML 应用于 few-shot 识别的不同方法，但在这项工作中，我们采用了流行的原型网络体系结构 [30] 引入的程序，这是各种其他方法的基础。在这种情况下，ML 主要是一种随机化过程。成对的 $(\mathbb{S}_i, \mathbb{T}_i)$ 表示为片段，训练集表示为支持集，测试集表示为查询集。元训练集 MS 是通过抽样训练和测试类生成的。特别地，第 i 个 episode 的训练集 $\mathbb{S}_i^s$ 是通过从 low-shot 问题的类集合中抽样 N 个类和每个类的 K 个示例获得的。这定义了一组 K-shot 学习问题，称为 N-way k-shot 问题。模型 h 是(2)的 softmax 分类器，(4)的元学习映射实现了高斯平均最大似然估计量

$$\mu_k = \frac{1}{|\mathcal{P}_{i,k}|} \sum_{\mathbf{x}_j \in \mathcal{P}_{i,k}} f_\phi(\mathbf{x}_j) \tag{7}$$

其中 $\mathcal{P}_{i,k} = \{\mathbf{x}_j \in \mathbb{S}_i^s | y_j = k\}$ 是 k 类的支持样本集，在(5)中使用反向传播更新嵌入 f_ϕ。

ML 对开放集分类的好处.ML 程序与训练 softmax 分类器的标准小批量程序非常相似。除了使用 episode 训练而不是更流行的小批量训练外，还有两个主要区别。首先，通过 $\mathbb{T}_i^s$ 而不是 $\mathbb{S}_i^s$ 中示例的反向传播来更新 f_ϕ。由于与元测试的一致性，这对于 few-shot 学习是有利的，而在大规模测试中则不是这样。第二，来自所有类的小批量随机示例被来自 $\mathbb{S}_i^s$ 和 $\mathbb{T}_i^s$ 中包含的 N 个类子集的示例替换。将每个 episode 学习的分类任务随机化迫使嵌入 f_ϕ 更好地推广到看不见的数据。这个特性使得 ML 成为一个很好的解决 few-shot 学习的方法，因为在这些 few-shot 学习中，训练数据缺乏对类内变化的良好覆盖。在这种情况下，来自已知类的测试样本的大子集在训练期间是看不见的。我们建议，同样的属性使 ML 成为开集分类的一个很好的候选者，根据定义，分类器必须处理来自不可见类的不可见样本。这个观察激励我们设计一个统一的基于 ML 的 open-set 分类解决方案，它支持大规模和少样本分类。

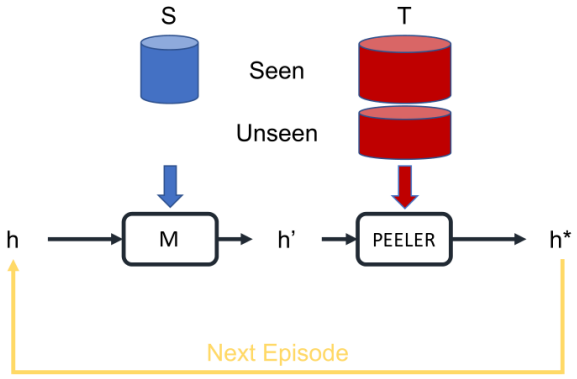

Figure 1. 图 1. 提出的开放集元学习的一般框架：对元训练集 $\{\mathbb{S},\mathbb{T}\}$ 进行抽样，其中 $\mathbb{T}$ 包含不在 $\mathbb{S}$ 中的类作为“看不见”的类，并最小化(8)中的损失以获得 h*。

4. 基于元学习的开放集识别

在本节中，我们将介绍所提出的开集识别 ML 方法。我们首先介绍一般的过程，然后讨论在我们的 PEELER 实现中使用的具体嵌入度量。

4.1. 开集元学习

如图1所示，开放集元学习（PEELER）依赖于元训练集 $\mathbb{MS} = \{(\mathbb{S}_i^s, \mathbb{T}_i^s)\}_{i=1}^{N^s}$ 和元测试集 $\mathbb{MT} = \{(\mathbb{S}_i^t, \mathbb{T}_i^t)\}_{i=1}^{N^t}$。与标准 ML 相比，唯一的区别是，episodes$(\mathbb{S},\mathbb{T})$ 是开放集。虽然训练集 $\mathbb{S}$ 与标准 ML 中使用的训练集相同，但测试集 $\mathbb{T}$ 增加了看不见的类。因此，PEELER 可使用类似于第3.2节 ML 程序的解决方法。当 ML 步骤保持如(4)中所示时，(5)的优化步骤变为：

$$h^* = \arg\min_h \left\{ \sum_{(x_k,y_k)\in\mathbb{T}_i^s | y_k\in\mathbb{C}_i^s} L_c[y_k, h'(x_k)] + \lambda \sum_{(x_k,y_k)\in\mathbb{T}_i^s | y_k\in\mathbb{C}_i^u} L_o[h'(x_k)] \right\} \tag{8}$$

其中 $\mathbb{C}_i^s(\mathbb{C}_i^u)$ 是 $\mathbb{T}_i^s$ 的可见（不可见）类的集合，$L_c[.,.]$ 是应用于可见类的分类损失（通常为交叉熵损失），L_o 是应用于 $\mathbb{S}_i^s$ 中未见类的开放集损失。

few-shot 开集识别.few-shot 设置要求对组成支持集和查询集的类进行采样。与闭集 few-shot 识别相似，第 i 个 episode 的支持集 $\mathbb{S}_i^s$ 是通过采样 N 个类和每个类采样 K 个样例来获得的。这定义了可见类 $\mathbb{C}_i^s$。但是，查询集 $\mathbb{T}_i^s$ 由这些类与 M 个额外的未见类 $\mathbb{C}_i^u$ 组合而成。这些支持和查询集在(8)中使用。

大规模开集识别. 在大规模环境中，可见类有大量的示例，并且经过良好的训练，无需重新采样。然而，对不可见类进行重采样仍然是有利的，因为它可以使嵌入更好地推广到这些类。在每一个 episode 中，从类标签空间中随机抽取 M 个类，形成一组不可见的类 $\mathbb{C}_i^u$，其余的类用作可见类 $\mathbb{C}_i^s$。如果没有支持集 $\mathbb{S}_i^s$，则不再需要(4)中的元学习步骤。相反，我们依靠一组可见类来调整模型，使其仅将样本分类到这些类中，即仍然应用损失函数为(8)的映射：

$$h' = \mathcal{M}(h, \mathbb{C}_i^s) \tag{9}$$

开集损失. 在推理过程中，当面对来自不可见类的样本时，模型不应该将很大的概率分配给任何类。在这种情况下，如果所见类别中的最大类别概率 $\max_k p_\phi(y = k|\mathbf{x})$ 很小，则可以拒绝样本。为了实现这一点，学习算法应该最小化 $\mathbb{C}_i^u$ 样本在可见类上的概率。这可以通过最大化可见类概率的熵来实现，即，使用负熵

$$L_o[\mathbf{x}] = \sum_{k\in\mathbb{C}_i^s} p(y = k|\mathbf{x}) \log p(y = k|\mathbf{x}) \tag{10}$$

作为损失函数。

4.2. 高斯嵌入

虽然 PEELER 是一个通用框架，但在这项工作中，我们提出了一个基于原型网络架构的实现 [30]。首先定义一组类原型，并将与最近的类原型有较大距离的样本分配给该组不可见类。对于 low-shot 分类，类原型是(7)的类平均值。对于大规模分类，我们假设固定的原型嵌入到网络中，并通过反向传播学习。

虽然在文献 [30] 中讨论了几个距离，但是原型网络通常是用欧几里得度量来实现的，即

$$d(f_\phi(\mathbf{x}), \mu_k) = (f_\phi(\mathbf{x}) - \mu_k)^T (f_\phi(\mathbf{x}) - \mu_k) \tag{11}$$

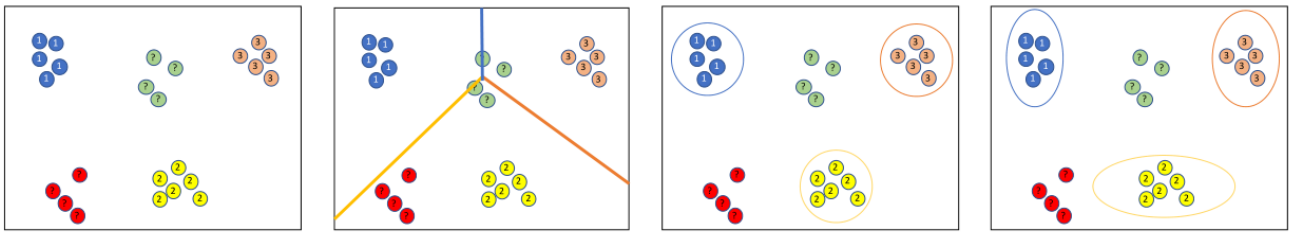

Figure 2. （a）具有 3 个可见类和 2 个不可见类的开集识别问题的特征空间；（b）闭集分类的最佳边界；（c）每个类定义一个等半径的欧氏距离。虽然对于闭集是最优的，但对于开集识别来说，这是次优的；（d）通过允许沿每个维度的类相关方差的不同高斯聚类，可以学习一组更好的开集距离。

这意味着每个类的特征遵循一个具有平均 c_k 和对角线方差 $\sigma^2 I$ 的正态分布，其中 σ 由所有类共享。虽然对于闭集 few-shot 学习是明智的，因为嵌入是学习产生这样的特征分布，但当引入开放集样本时，这可能是非常次优的。图 2 说明了具有三个可见类和两个不可见类的设置的问题。即使如图2（a）所示，嵌入是这样的，可见类呈球形分布并具有相同的协方差，但在训练期间看不到的开放集样本仍然以随机位置嵌入到特征空间中。因此，尽管封闭集分类器的最佳边界（如图2（b）所示）与图2（c）所示的距离(11)的轮廓相匹配，但它们不是开放集识别的最佳边界。事实上，如图2（d）所示，可见类和不可见类之间的最佳边界形状甚至可能因类而异。

为了说明这一点，我们假设 k 类的平均 μ_k 和协方差 $\sum_k$ 的高斯分布。因此，(11)的欧几里德距离被马氏距离代替：

$$d(f_\phi(\mathbf{x}), \mu_k) = [f_\phi(\mathbf{x}) - \mu_k]^T \sum_k^{-1} [f_\phi(\mathbf{x}) - \mu_k] \quad (12)$$

为了保持参数的数量可控，我们假设所有协方差矩阵都是对角的，即 $\sum_k = diag(\sigma_{k1}, \sigma_{k2}, ..., \sigma_{kM})$，精度矩阵 $A_k = \sum_k^{-1}$ 用于简化计算。与类原型类似，精度矩阵的学习取决于识别设置。对于大规模的开集识别，A_k 是通过反向传播直接学习的网络参数。在 few-shot 设置中，在支持样本可用的情况下，我们引入了一个新的拥有可学习参数 φ 的嵌入函数 g_φ，并定义了

$$A_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} g_\varphi(\mathbf{x}_i). \quad (13)$$

5. 实验

将该方法与 SOTA 的开放集识别方法进行了比较。根据 Neal 等人 [22]，我们评估了分类精度和开放集检测性能。为了澄清术语，闭集类是在训练期间看到的类别，而开放集类是仅用于测试的新类别。所有的训练都是基于来自封闭集合类的训练样本。对于测试，我们使用来自训练和开放集类的测试样本。分类精度用于衡量模型对封闭集样本（即来自封闭集类的测试样本）的分类程度。AUROC（ROC 曲线下面积）度量用于测量模型在所有测试样本中检测开放集样本（即来自开放集类的测试样本）的程度。为了简化写作，我们定义了以下首字母缩略词：our basic 表示具有欧几里德距离的原型网络，用于开集检测；GaussianE 表示 Sec.4.2 中引入的高斯嵌入；OpLoss 表示提出的开集损失。

5.1. 大规模开集识别

以前大多数关于开集识别的工作都是为大规模环境而设计的。这里，我们在 CIFAR10[14] 和扩展 miniImageNet[7] 上评估 PEELER。CIFAR10 由 10 个类的 60000 个图像组成。在 [22] 之后，首先随机选择 6 个类作为闭集类，其他 4 个类作为开集类。结果在闭集/开集类的 5 个随机分区上求平均值。扩展的 miniImageNet 是为 few-shot 学习而设计的。我们使用 64 个训练类别和每个类别 600 个图像作为闭集训练数据，而每个类别的 300 个额外图像用于闭集测试。来自 20 个测试类别的图像用于开放集测试。

训练. 在 CIFAR10 上，我们从每个 episode 的 6 个封闭集类中随机抽取 2 个类，以应用开放集损失，其余 4 个类用于训练分类。在 miniImageNet 上，闭合/开放集分区是固定的。我们使用 Adam[13]，初始学习率为 0.001，λ = 0.5(8)和 10000 次训练。6000 episodes 和 8000 episodes 之后，学习率衰减了 0.1 倍。

结果. 我们将所提出的方法与几种开放集识别 SOTA 方法进行比较，包括 OpenMax[1]、G-OpenMax[6] 和 Counterfactual[22]，以及分布外 SOTA 方法 Confidence[15]。所有模型都使用相同的 CNN backbone 进行公平比较。我们测试了由 [22] 提出的 CNN（表示为 ConvNet）和 ResNet18[9]。

表2显示，对于两个 backbones，PEELER 的性能都比以前的所有方法有很大的提高。对于相似的已知类分

Model	Accuracy(%)	AUROC(%)
ConvNet on CIFAR10		
Softmax	80.1	67.7
OpenMax	80.1	69.5
G-OpenMax	81.6	67.9
Counter	82.1	69.9
Confidence	82.4	73.19
Our basic	82.4	74.62
Our basic + GaussianE	82.3	75.65
Our basic + GaussianE + OpLoss	82.3	**77.22**
ResNet18 on CIFAR10		
Softmax	94.2	78.90
OpenMax	94.2 7 79.02	
Confidence	94.0	80.90
Our basic	94.7	82.94
Our basic + GaussianE	94.3	83.12
Our basic + GaussianE + OpLoss	94.4	**83.99**
ResNet18 on miniImageNet		
Softmax	76.1	76.65
OpenMax	76.1	77.80
Confidence	76.5	80.67
Our basic	76.4	80.59
Our basic + GaussianE	76.1	81.06
Our basic + GaussianE + OpLoss	76.3	**82.12**

Table 2. 与 SOTA 在大规模开放集识别上的比较：PEELER 在开放集样本检测 AUROC 方面优于所有其他方法，具有相当的分类精度

Model	Accuracy(%)	AUROC(%)
5-way 1-shot		
GaussianE + OpenMax	57.89±0.59	58.92±0.59
GaussianE + Counterfactual	57.89±0.59	52.20±0.61
Our basic	56.31±0.57	58.94±0.60
Our basic + OpLoss	56.34±0.57	60.94±0.61
Our basic + GaussianE	57.89±0.59	58.66±0.60
Our basic + GaussianE + OpLoss	**58.31±0.58**	**61.66±0.62**
5-way 5-shot		
GaussianE + OpenMax	75.31±0.76	67.54±0.67
GaussianE + Counterfactual	75.31±0.76	53.25±0.59
Our basic	74.19±0.75	66.00±0.67
Our basic + OpLoss	74.14±0.74	67.92±0.68
Our basic + GaussianE	75.31±0.76	66.50±0.67
Our basic + GaussianE + OpLoss	**75.08±0.72**	**69.85±0.70**

Table 3. few-shot 开放集识别结果。与几种基线和先前的开放集方法进行了比较。

Model	Accuracy(%)	AUROC(%)
Our basic	39.61±0.40	71.32±0.70
Proto+Entropy	39.41±0.40	72.23±0.72
Our basic + GaussianE	40.18±0.40	71.31±0.70
Our basic + GaussianE + OpLoss	**41.90±0.39**	**74.97±0.74**

Table 4. 10-way 1-shot 开放集识别结果。AUROC 值高于 5-way 的结果。

类精度，它实现了更高的 AUROCs 来检测未知类。提出的高斯嵌入和开集损失都提高了开集检测性能。

5.2. few-shot 开集识别

数据集. 在 mini-Imagenet[34] 上，使用 [26] 的分割来评估 few-shot 开放式设置的性能。64 个类用于训练，16 个类用于验证，另外 20 个类用于测试

训练. 开放集问题遵循 5-way few-shot 识别设置。在训练期间，每个 episode 从训练集中随机选择 10 个类，5 个类作为封闭类，另外 5 个类作为不可见类。所有支持集样本都来自封闭集类。查询集包含来自封闭集类（封闭查询集）的样本和来自开放集类（开放查询集）的样本。使用相同的策略从测试集中抽取评估集。评估重复 600 次，以尽量减少不确定性。训练总次数为 30,000 次。在 10,000 和 20,000 次发作后，学习率下降了 0.1 倍。

结果. 由于并非所有先前的开放集方法都支持 few-shot 设置，因此需要进行一些修改。例如，生成方法 [6,22] 不支持 few-shot 样本。相反，我们在预训练集上训练模型，并在支持集上对其进行微调。闭集分类器如上所述，并且进一步训练两类分类器来检测开集样本。OpenMax[1] 在 few-shot 设置下更容易应用。我们将 OpenMax 应用于预 softmax 层的激活，即高斯设置距离的负值。所有方法均采用 ResNet18[9] 进行公平比较。

如表3所示，OpenMax 和 Counterfact 的性能都不如提出的方法。请注意，50% 的 AUROC 对应于 chance performance。提出的开集损失、高斯嵌入以及它们的组合都提供了增益。

5.3. 消融研究

训练中的抽样策略. 我们研究了训练抽样策略对开放集检测结果的影响。在确定开集样本总数的情况下，通过训练集产生开集样本所用的不可见类的数目是变化的。相应的开放集检测结果如图3（a）所示。与基线方法的平均增益相比，性能变化较小。我们假设这种鲁棒性是由于任务的随机性。当训练中 episodes 总数较大时，无论一个训练 episode 包含多少个 open-set 类，该模型都能很好地收敛。

影响开放集测试的因素. 对于大规模开集识别，开集样本的数量取决于开集类的数量。开放集问题的难度取决于后者。我们试图找出决定 few-shot 开放集问题难度的因素。第一个因素是开放查询集中的类数。我们将这个数字从 1 变为 10，同时保持训练过程和开放查询集中的样本总数不变。结果如图3（b）所示。第二个因素是每个类的样本数，当类数保持为 5 时，样本数会发生变化。结果如图3（c）所示。图中显示，变化会导致 AUROC 性能的微小变化。这意味着这些因素对问题的难度影响不大。

影响 few-shot 测试的因素. 对于训练集和测试集的固定开集成分，我们将 5-way 分类与 10-way 分类进行了比

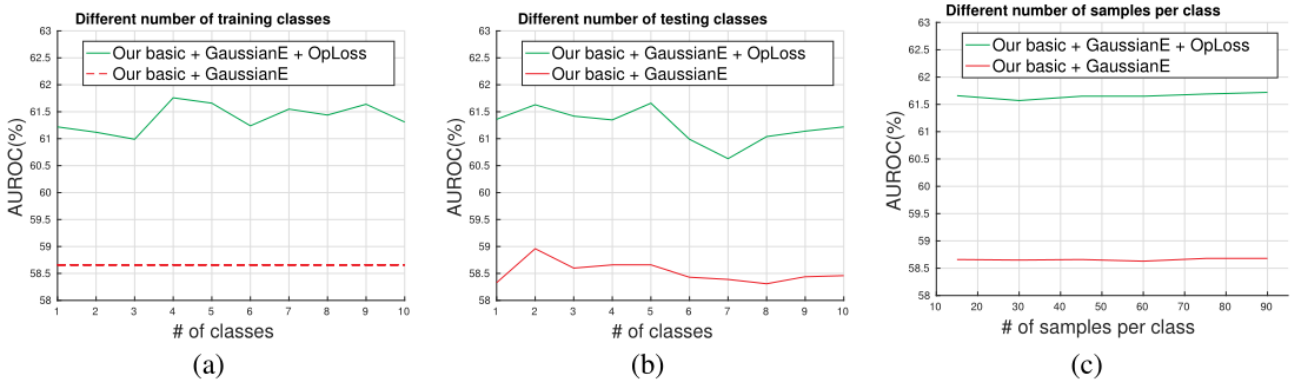

Figure 3. 消融研究: 提出的开集损失在不同的 (a) 训练班数、(b) 测试班数或 (c) 每个班级的测试样本数下产生一致的增益。

Category	VODC[36]			Ours		
	I	II	III	I	II	III
airplane	20	10	**0**	6	6	**0**
balloon	13	4	3	11	7	**2**
bear	3	3	**2**	**2**	**2**	**2**
cat	4	5	5	5	**3**	**3**
eagle	23	12	8	5	2	**0**
ferrari	11	7	6	11	**3**	**3**
figure skating	**0**	**0**	**0**	**0**	**0**	**0**
horse	5	**1**	**1**	5	4	3
parachute	14	10	2	10	2	**0**
single diving	18	13	5	4	2	**1**
Avg.	11.1	6.5	3.2	5.9	3.1	**0**

Table 5. 通过改变带注释的帧数 (第二行中的 I、II、III)，错误分类的帧数 (越低越好)。

较。结果如表4所示。尽管如预期的那样，10-way 分类不如 5-way 分类，但开集性能显著提高。这表明具有更多类别多样性的 few-shot 分类任务对开集样本的鲁棒性更强。

5.4. 弱监督对象检测

最后，我们研究了一个 few-shot 开放集识别的应用。[36] 中考虑了弱监督对象发现问题。给出了一个对象的视频，但有些帧与对象的存在无关。任务是在带注释的帧 (呈现或不呈现对象) 的数量有限的情况下找到这些不相关的帧。对于开集问题，只给出相关的帧标签，即包含对象的帧。不相关的帧被检测为开放集样本。

采用 XJTU-Stevens 数据集 [36] 进行评估。它有来自 10 个类别的 101 个不同实例的视频，这些视频中的一些帧没有对象。在训练过程中，按照5.2中的说明训练 5-way 1-shot few-shot 开放集模型。我们对每个帧执行实例级分类，而不是类别级分类。在测试期间，使用标注为与支持集相关的帧，只考虑一个视频。这意味着只提供一个高斯类中心。未标记的帧根据其到中心的马氏距离被标记为可见或不可见。距离大于阈值的帧被检测为无关。带注释的相关帧的数量从 1 到 3 不等。结果显示为错误分类的帧的数量。文中列出了 [36] 中最好的 VODC 方法，以供比较。请注意，VODC 需要相同数量的带注释的无关帧，而 PEELER 不需要。所提出的方法在很大程度上优于 VODC，在所有三种设置下都将误分类帧的数量减半。这说明它的特征嵌入是一种较好的开集检测解决方案。

6. Conclusions

在这项工作中，我们重新探讨了在 few-shot 学习的情况下开放集识别的问题。我们提出了元学习的扩展，包括一个开放集损失和一个更好的度量学习设计。所得到的分类器为 mini-Imagenet 上的 few-shot 开放集识别提供了一种新的 SOTA。实验结果表明，该方法在不作任何修改的情况下，也可以应用于大规模识别，其性能优于开放集识别的最新方法。最后，使用 XJTU-Stevens Dataset 验证了所提出模型在弱监督对象发现任务中的有效性。

致谢. 获得国家自然科学基金 61A001 和国家自然科学基金 61A0400 的部分资助。刘波和 Nuno Vasconcelos 获得了 NSF 奖项 IIS-1637941、IIS-1924937 和 NVIDIA GPU 捐款的部分支持。