arXiv:2301.03041v2 [cs.CV] 5 Jun 2023

自监督学习中相似性损失与聚类损失之间关系的研究

Jidong Ge、Yuxiang Liu、桂杰、IEEE高级会员、Lanting Fang、Ming Lin、James Tin-Yau Kwok、IEEE会士、LiGuo Huang、Bin Luo

摘要—**自监督学习使网络能够从海量数据本身学习判别特征。大多数最先进的方法基于对比学习，最大化同一图像的两种数据增强之间的相似性。利用两种数据增强的一致性，可以免去人工标注的负担。对比学习利用实例级信息来学习鲁棒特征。然而，学到的信息可能局限于同一实例的不同视图。本文尝试利用两幅不同图像之间的相似性，提升自监督学习中的表征。与实例级信息相比，两幅不同图像之间的相似性可能提供更多有用信息。此外，我们分析了相似性损失与特征级交叉熵损失之间的关系。这两种损失对大多数深度学习方法都至关重要。然而，这两种损失之间的关系尚不明确。相似性损失有助于获得实例级表征，而特征级交叉熵损失有助于挖掘两幅不同图像之间的相似性。我们通过理论分析和实验表明，恰当地结合这两种损失能够取得最先进的结果。代码见 https://github.com/guijiejie/ICCL。**

索引词—**自监督学习、图像表征、图像分类。**

I. 引言

RECENTLY，无监督/自监督表征学习已取得稳步进展。许多自监督方法 [1]–[12]在计算机视觉领域正逐步缩小与监督式预训练之间的性能差距。这些方法利用了数据本身的特性。大多数自监督方法试图基于实例判别 [13]–[16]任务，通过最大化同一图像两种数据增强之间的一致性，并分散不同实例，从而开展学习。自监督学习取得的令人鼓舞的成果有赖于强大的变换 [1], [8], [17] （例如图像裁剪和颜色失真）以及相似性损失。BYOL [18] 和 SimSiam [19]扩展了相似性损失，并摆脱了对负实例 [20]–[22]的依赖。这些方法隐式地实现了分散，并学习同一实例不同变换的鲁棒表征。本文中，基于对比学习的方法指的是 MoCo [23] 和 BYOL 等方法。这些方法的关键在于最小化数据增强之间的相似性。

J. Ge、Y. Liu 和 B. Luo 任职于南京大学软件学院新型软件技术国家重点实验室，中国南京 210093。（电子邮箱：gjd@nju.edu.cn、mg1932010@smail.nju.edu.cn、luobin@nju.edu.cn）。J. Gui 和 L. Fang 任职于东南大学网络空间安全学院及紫金山实验室，中国南京 210000。（电子邮箱：{guijie、lanting}@seu.edu.cn）。M. Lin 任职于亚马逊有限责任公司；本工作是在阿里巴巴集团完成的。（电子邮箱：minglamz@amazon.com）。James Tin-Yau Kwok 任职于香港科技大学计算机科学与工程学系，中国香港 999077。（电子邮箱：jamesk@cse.ust.hk）。L. Huang 任职于南卫理公会大学计算机科学系，美国德克萨斯州达拉斯 75205。（电子邮箱：lghuang@smu.edu）。

不同于通过对比学习来学习对变换的不变性的基于方法， [24], 一些工作尝试利用聚类 [25]–[32] 及伪标签。大多数基于实例级对比学习的方法可能会受到相似背景的误导 [33]。无论选择哪种变换，图像背景都可能无法被完全去除。背景像素为最小化相似性损失提供了一条捷径。相比之下，不同图像之间的相似性可能会提高模型对背景信息的鲁棒性。模型通过学习最大化不同背景下同一物体图像之间的相似性。这种学习方式至关重要，也更接近人类的学习方式。人们能够忽略背景，因为他们已经见过数十万张不同背景下的同一物体图像。传统的基于聚类的方法通过伪标签对图像进行分类。这些方法可能会关联同一类别的图像。然而，生成伪标签需要大量计算。一些在线聚类方法 [31] 通过 Sinkhorn-Knopp 算法 [34]为批次样本分配标签。Sinkhorn-Knopp 算法确保批次样本被原型均匀划分，从而避免所有图像都获得相同标签的平凡解。

挖掘两幅不同图像之间的相似性，是提升判别能力的一种可行方式。大多数最先进的方法（例如，SwAV [31] 和 DINO [11]）直接利用特征级交叉熵损失来学习不同图像之间的相似性。一般而言，相似性损失和特征级交叉熵损失以不同的方式应用于自监督方法。在 SimSiam 中，作者发现交叉熵损失可能不适用于基于对比学习的方法。本文尝试分析这两种损失之间的关系。通过理论和实验分析，我们指出这两种损失可以互为补充。从梯度的角度出发，我们分析了相似性损失与交叉熵损失之间的差异。这些发现表明，恰当的组合或许能提升类别级和实例级表征。我们的贡献如下。

- 我们证明，监督学习能够捕捉同一类别的不同图像之间的关系。此外，自监督方法可以利用两幅不同图像之间的相似性。

- 我们通过理论和实验分析，解释了为什么以往基于对比学习的方法（例如 SimSiam）使用交叉熵损失时效果较差。这一点对于保持相似性损失和交叉熵损失的优势至关重要。
- 不同于基于聚类的方法，我们关注相似性损失与特征级交叉熵损失之间的关系。基于这一关系，我们提出了一种简单而有效的方法，利用实例级对比学习和类内对比学习（**ICCL**）。我们的方法试图以适当的方式挖掘两张不同图像之间的信息，从而减轻错误聚类的影响。与 SwAV 和 DINO 相比，我们的方法无需中心化（ DINO 使用）和 Sinkhorn-Knopp（SwAV 使用）。超参数设置对不同数据集具有稳健性。

II. 预备知识

本节旨在介绍本文中不同损失函数的一些符号。具体而言，我们为 SimSiam/BYOL 中的损失函数（称为相似性损失）以及 SwAV/DINO 中的损失函数（称为聚类/交叉熵损失）给出形式化定义。

A. 基于相似性损失的方法

许多方法使用相似性损失 [35] 来最大化同一图像的两个视图之间的一致性。一般来说，MoCo 中的损失函数是一种典型的相似性损失函数

$$L^{q}_{contrastive} = -\log \frac{\langle q, k_+\rangle/\tau}{\sum_{i=0}^{|B|}\langle q, k_i\rangle/\tau}. \tag{1}$$

这里， q 是图像的表征。$L^{q}_{contrastive}$ 表示表征 q的损失。由于大多数自监督学习损失函数都可以拆分为对应单个实例的损失之和，因此下文将省略损失函数的上标。 k_+ 表示批次中的正样本。通常，两个变换 T_1 和 T_2 会将图像 I 变换为同一图像的不同视图。这些视图的表征将被视为正样本。 B表示数据批次， $|B|$ 表示批量大小。相似性损失旨在增大正特征之间的相似性，并降低负特征之间的相似性。在 [36], 中，作者对相似性损失进行了详细的分解

$$\begin{aligned} L_{contrastive} &= -\log \frac{\langle q, k_+\rangle/\tau}{\sum_{i=0}^{|B|}\langle q, k_i\rangle/\tau} \\ &= \log \sum_{i=0}^{|B|} \exp \langle q, k_i\rangle/\tau - \frac{\langle q, k_+\rangle}{\tau}. \end{aligned} \tag{2}$$

第一项称为均匀性项。如果 q 和 $k+$ 归一化为单位向量，那么第二项中的 $\langle q,\ k_+\rangle$ 就是余弦相似度。因此，MoCo 的损失函数也可以视为带有均匀性项的余弦相似度损失。

对于相似性损失，我们分别将神经网络计算得到的输出特征定义为 z'_1 和 z'_2。 z'_1 和 z'_2 是同一图像的两个视图。在 BYOL [18] 和 Sim-Siam [19], 中，其中一个特征会经过额外的预测器（例如，预测器将 z'_1 编码为 q_1）。最终特征记为 q_1 和 z_2。需要注意的是， z_2不会将梯度传回网络（$z_2 = sg(z'_2)$和$sg()$表示停止梯度操作）。根据 $\tilde{q} \triangleq \frac{q}{||q||}$的定义，其中 $||.||$ 表示 l_2 范数，相似性损失可表示为

$$L_{similarity} = -\langle \tilde{q}_1, \tilde{z}_2\rangle. \tag{3}$$

此处 $\langle .,.\rangle$ 表示内积。均匀性项 [6], [8], [23]–[37]被忽略。该损失会最大化同一图像的两个视图之间的相似性，并学习能够抵抗强数据增强的有用表征。基于对比学习的方法（包括使用 $L_{contrastive}$ 和 $L_{similarity}$的方法）能够适应不同规模的数据集。

B. 基于聚类损失的方法

另一种自监督学习形式基于聚类 [11], [29]–[31]。这些方法生成伪标签，并利用伪标签最大化交叉熵损失：

$$\begin{aligned} L_{ce} &= -\sum_{i=1}^{D} f(i, z_2)\log p(i|q_1, \tau), \\ where \quad p(i|x,\tau) &= \frac{\exp(\frac{x^{(i)}}{\tau})}{\sum_{j=1}\exp(\frac{x^{(j)}}{\tau})}. \end{aligned} \tag{4}$$

此处 f 是生成伪标签的函数（例如， f是 SwAV 中的 Sinkhorn-Knopp 算法，而 $p(i|z_2)$是在 DINO 中采用中心化机制的函数）。 C 是向量 q_1 和 z_2的维度。 W 表示聚类原型，它是一个 $C\times D$ 矩阵。 τ 是温度 [38] ，用于调节概率分布的尖锐程度，默认值为 1。这些基于聚类的方法试图利用不同实例之间的关系来学习鲁棒的表征。然而，这些方法可能会受到错误聚类的影响。学到的表征基于伪标签。超参数可能难以推广到大量数据集。

III. 方法

本节首先从召回率指标的角度说明监督学习与自监督学习之间的差异。这种差异表明，自监督方法的类内数据点分布较为分散。然而，监督学习可能会聚合类内特征，并扩大不同类别之间的距离。这一点启发了我们接下来的研究。随后，我们分析 L_{ce} 为何能够捕捉两张不同图像之间的相似性。接着，我们说明如何建立相似性损失与交叉熵损失之间的关系。最后，我们介绍所提方法的细节，以及它与其他基于聚类的方法之间的差异。

表I 监督方法与自监督方法的比较。默认网络骨干为ResNet-50。默认训练轮数为100。

方法	ImageNet	
	$Precision@k = 5$	Top-1准确率
监督学习	**52.1**	**76.5**
SimSiam	27.1	67.3
BarlowTwins	28.2	67.4
BYOL	28.1	66.0
BYOL-300轮	32.4	72.2
MoCo	27.5	67.4
MoCo-ViT	37.0	69.1

A. 自监督学习的类内距离

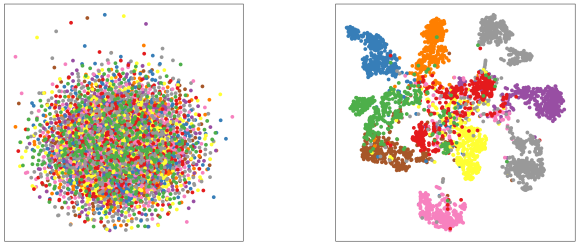
尽管自监督学习在许多任务和数据集上取得了令人鼓舞的结果，但它仍存在一些问题。例如，大多数自监督学习方法在图像分类任务中都需要采用线性评估协议。可训练的全连接层用于区分不同类别的特征。如表 I 所示，这个全连接层至关重要。我们使用 $Precision@k = num(true)/k$ 表示不同自监督方法的召回率指标。$num(true)$ 表示返回的 k张图像中，查询图像的正样本最近邻数量。正样本定义为属于同一类别的图像。该指标表示检索同一类别图像的能力。换言之，如果同一类别的特征分布较为紧凑，且不同类别之间的质心距离相对较大，那么 $Precision@k$ 可能会表现良好。骨干网络的输出用作检索特征。对于 Top-1准确率，我们会额外训练一个全连接层来进行分类。

如表 I 所示，自监督学习方法难以从不同实例的视图中学习紧凑的表征。尽管 Top-1准确率的差距不大，但我们发现，自监督学习方法的召回率仍远低于监督学习。这表明，大多数自监督学习方法可能只能学到粗略的表征。

图1展示了网络在不同阶段的结果。训练初期，由于不同类别的质心彼此接近，自监督学习难以捕捉类内关系。例如，SwAV 会在训练初期尝试进行聚类。然而，从大量无关数据中学习有用的聚类信息并非易事。相比之下，如果能在图1b中进行聚类，聚类信息或许有助于网络压缩同一类别的特征。一种方法是在训练初期避免使用类内信息，并在经过若干轮次后利用类内信息。因此，关键在于从梯度的角度找到一种损失函数，能够直接替代实例级损失函数。

B. 交叉熵损失的作用

为简单起见，我们首先分析监督学习的情况。给定一个包含 $X = \{x_1, ..., x_N\}$ 张图像及其标签 $Y = \{y_1, ..., y_N\}$ 的批次，其中批量大小为 N，相应的



(a) (b)

图1. t-SNE [39] 可视化了网络在不同阶段的结果。左图是在训练开始时生成的。右图是在自监督训练过程进行到一半时生成的。

图像编码器（网络）的输出为 $Z = \{z_1, ..., z_N\}$。所有图像被分为 C 个类别。因此，z_i 是一个 1×C 向量。损失函数为

$$L = -\frac{1}{|B|}\sum_{i=1}^{|B|}\log p(y_i|z_i, \tau), \tag{5}$$

其中，B 表示批量数据，$|B|$ 表示批次大小。τ 的默认值为 1。

命题1. 假设网络具备区分实例的基本能力。对于批次 B 中属于同一类别 y 的图像 x_i 和 x_j，最小化损失函数等价于最大化两个示例的概率分布之间相似性的下界。

证明. 交叉熵损失可以表示为

$$\begin{aligned} L = &-\frac{1}{|B|}\sum_{i'=1, i'\notin\{i,j\}}^{|B|} log\ p(y_{i'}|z_{i'}) \\ &-\frac{1}{|B|}(log\ p(y|z_i) + log\ p(y|z_j)), \end{aligned} \tag{6}$$

其中，x_i 和 x_j 属于同一类别 y。将$\mathbf{P}(z_i) \in \mathbb{R}^C$ 定义为不同类别上的 $p(c|z_i)$ 的堆叠，它表示图像 x_i的概率分布，最小化交叉熵损失可以最大化

$$\begin{aligned} p(y|z_i)\cdot p(y|z_j) &\le \sum_c p(c|z_i)\cdot p(c|z_j) \\ &\le S(\mathbf{P}(z_i), \mathbf{P}(z_j)), \\ where \quad S(p,q) &\triangleq \frac{\langle p, q\rangle}{\|p\|\cdot\|q\|}. \end{aligned} \tag{7}$$

如果在批次中找不到另一个属于同一类别的实例（例如，找不到 x_j），独热向量 $p(y|z_j) = 1$和 $log\ p(y|z_j) = 0$ 也能满足我们的命题，并且不会影响损失。证明完毕。□

根据命题 1，如果网络具备区分实例的基本能力，那么监督学习在最小化交叉熵时会学习两幅不同图像之间的相似性。对于那些基于聚类的方法，当伪标签的概率分布

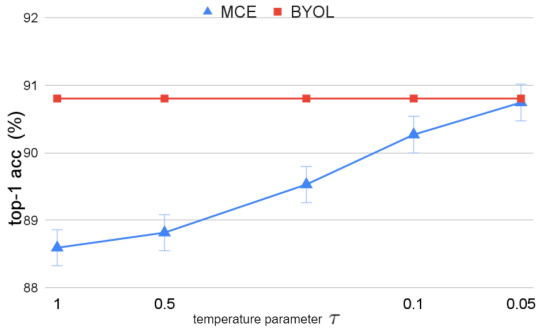


图 2。**不同 τ 在 L_{mce}的实验**。在 Imagenette [40] 上通过线性评估获得 Top-1 准确率。超参数与 BYOL 保持一致。

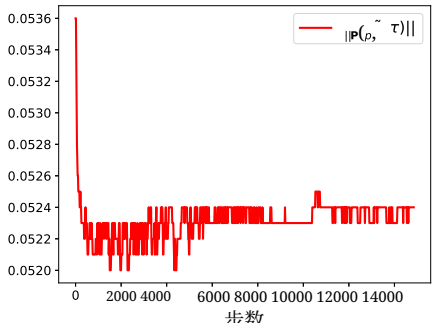


图 3. **$\|\mathbf{P}(z_2)\|$ 在训练过程中**。图 2 中 $\|\mathbf{P}(z_2)\|$ 的值接近 τ 的最佳设置。在式 (14) 中，$\|\mathbf{P}(z_2)\|^2/\tau^2$ 直接影响 L_{ce} 的梯度幅值。测量 $\|\mathbf{P}(z_2)\|$ 有助于选择 τ 的超参数。

较为尖锐时，交叉熵损失也可能捕捉到这一信息。然而，如图1所示，在自监督学习的训练初期，很难利用相关图像之间的相似性。在 SwAV 和 DINO 等自监督学习方法中，训练初期的交叉熵损失可能会使彼此不相关的图像变得接近（图1a）。经过一半的自监督训练后，同一类别的图像可能会彼此接近（图1b）。在这一阶段，交叉熵损失可能会学习相关图像之间的相似性。因此，问题在于如何在训练过程中自然地将相似性损失转化为交叉熵损失。

C. L_{ce} 与 $L_{similarity}$之间的关系

交叉熵损失对于捕捉两幅不同图像之间的相似性至关重要。然而，大多数使用 L_{ce} 的基于聚类的方法可能会受到伪标签质量不佳的影响。数据集规模也可能影响这些方法。输出特征维度等超参数可能比较敏感。相比之下，基于对比学习的方法受此问题的影响可能较小。例如，在 CIFAR-10 上，当输出维度为 2048 时，SimSiam 表现良好。然而，在 CIFAR-10 上，当原型数量为 2048 时，SwAV 的表现较差。在 SimSiam 中，作者注意到，直接将 $L_{similarity}$ 替换为

$$L_{ce} = -\sum_i p(i|z_2,\tau)\log p(i|q_1,\tau) \tag{8}$$

可能会降低性能。在 SimSiam 中，他们不使用 τ。因此此处的 $\tau = 1$。

为了探究 $L_{similarity}$ 与 L_{ce} 之间的关系，我们首先分析式 (3) 中 $L_{similarity}$ 的梯度。对于来自同一图像的两个向量 q_1 和 $z_2 \in \mathbb{R}^C$，q_1 的梯度为

$$\frac{\partial L}{\partial q_1} = \frac{1}{\|q_1\|}\left(\frac{\partial L}{\partial \tilde{q}_1} - \tilde{q}_1 \cdot \langle \tilde{q}_1, \frac{\partial L}{\partial \tilde{q}_1}\rangle\right), \tag{9}$$

$$\left\|\frac{\partial L}{\partial q_1}\right\|^2 = \frac{1}{\|q_1\|^2}(1 - \langle \tilde{q}_1, \tilde{z}_2\rangle^2) \le \frac{1}{\|q_1\|^2}, \tag{10}$$

当 L 是 $L_{similarity}$ 的简称时。若在训练开始时使用合适的初始化器 [41]，则 $\|q_1\|^2 \approx C$。q_1 的梯度将受到限制。

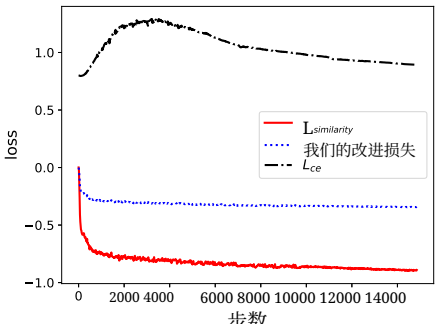


图 4. **与我们的损失之间的相关性**。训练过程仅最小化 $L_{similarity}$。其他项（蓝色曲线和黑色曲线）仅作记录，以展示它们与 $L_{similarity}$ 的相关性。我们的损失减去常数 $\log C$，以便清晰地显示曲线。

接下来，我们分析 SimSiam 中使用的交叉熵损失。对于式（8），q_1 的梯度为

$$\frac{\partial L_{ce}}{\partial q_1} = \frac{1}{\tau}(\mathbf{P}(q_1) - \mathbf{P}(z_2)), and\ \|\frac{\partial L_{ce}}{\partial q_1}\|^2 \le \frac{2}{\tau^2}. \tag{11}$$

显然，L_{ce} 中 q_1 的梯度仅受概率分布 $\mathbf{P}(q_1)$ 和 $\mathbf{P}(z_2)$ 之间距离的影响。因此，更新可能不受控制。尽管 L_{ce} 和 $L_{similarity}$ 似乎都能提高两个视图之间的一致性，但如图 4 所示，这两种损失无法关联起来。

注意，L_{ce} 可以表示为

$$L_{ce} = (\log\sum_j \exp\frac{q_1^{(j)}}{\tau}) - (\sum_i p(i|z_2,\tau)\frac{q_1^{(i)}}{\tau}). \tag{12}$$

该公式与 [36] 中的均匀性项和对齐项相似。如果忽略第一项，该损失与相似性损失类似。为了模拟相似性损失，改进交叉熵（MCE）损失可表示为

$$L_{mce} = -\sum_i p(i|z_2,\tau)\frac{\tilde{q}_1^{(i)}}{\tau}. \tag{13}$$

表II 不同损失函数的比较。我们对所分析的方法进行了详细比较。L_{align} 表示对齐项的梯度。为方便起见，我们给出了对齐项的梯度幅值。

损失函数	对齐项	均匀性项	梯度幅值（$\|\frac{\partial L^{align}}{\partial q_1}\|^2$）	梯度幅值上界
$L_{similarity}$	$-\langle\tilde{q}_1, \tilde{z}_2\rangle$	-	$(1-\langle\tilde{q}_1, \tilde{z}_2\rangle^2)/\|q_1\|^2$	$1/\|q_1\|^2$
$L_{contrastive}$	$-\langle\tilde{q}_1, \tilde{z}_2\rangle/\tau$	对数 $\sum_{i=0}^{\|B\|}$指数 $\langle q, k_i\rangle/\tau$	$(1-\langle\tilde{q}_1, \tilde{z}_2\rangle^2)/\tau^2\|q_1\|^2$	$1/\tau^2\|q_1\|^2$
L_{ce}	$-\langle q_1, \mathbf{P}(z_2)\rangle/\tau$	$\log\sum_j \exp q_1^{(j)}/\tau$	$\|\mathbf{P}(z_2)\|^2/\tau^2$	$1/\tau^2$
L_{mce}	$-\langle\tilde{q}_1, \mathbf{P}(z_2)\rangle/\tau$	-	$(1-\langle\tilde{q}_1, \frac{\mathbf{P}(z_2)}{\|\mathbf{P}(z_2)\|}\rangle^2)\cdot\|\mathbf{P}(z_2)\|^2/\tau^2\|q_1\|^2$	$\|\mathbf{P}(z_2)\|^2/\tau^2\|q_1\|^2$
我们的方法 (L_{iccl})	$-\langle\tilde{q}_1, \mathbf{P}(z_2)\rangle/\tau_1$	$\sum_i q(i)log\frac{q(i)}{p(i\|\theta)}$	$(1-\langle\tilde{q}_1, \frac{\mathbf{P}(z_2)}{\|\mathbf{P}(z_2)\|}\rangle^2)\cdot\|\mathbf{P}(z_2)\|^2/\tau_1^2\|q_1\|^2$	$\|\mathbf{P}(z_2)\|^2/\tau_1^2\|q_1\|^2 \approx 1/\|q_1\|^2$

我们不将 q_1 用作 $softmax$ 的输入，而是将 $\tilde{q}_1$ 用作输入。这一简单的修改可以带来

$$\|\frac{\partial L_{mce}}{\partial q_1}\|^2 = \frac{1}{\|q_1\|^2}(\|\frac{\partial L_{mce}}{\partial \tilde{q}_1}\|^2 - \langle\tilde{q}_1, \frac{\partial L_{mce}}{\partial \tilde{q}_1}\rangle^2)$$
$$= \frac{\|\mathbf{P}(z_2)\|^2}{\tau^2\|q_1\|^2}(1-\langle\tilde{q}_1, \frac{\mathbf{P}(z_2)}{\|\mathbf{P}(z_2)\|}\rangle^2). \quad (14)$$

与相似性损失的梯度不同，$\|\nabla q_1 L_{mce}\| \leq \|\mathbf{P}(z_2)\|/\tau$ 可能会导致较小的梯度，前提是 $\|\mathbf{P}(z_2)\|$ 较小（例如，均匀分布）。因此，τ 被证明对梯度的幅值至关重要。图 2 展示了不同 τ下的结果。实验结果使我们确信，通过恰当的 τ，可以建立 $L_{similarity}$ 与 L_{mce} 之间的联系。根据图 3、图 2 和公式（14），我们可以发现，$\|\mathbf{P}(z_2)\|$ 的范围接近 L_{mce}中τ 的最佳设置。从梯度的角度来看，可以解释 DINO 中中心化机制的 τ增加。增加$\mathbf{P}(z_2)$中的τ 会降低 $\|\mathbf{P}(z_2)\|$，从而使参数更好地收敛。

D. 详细方法

L_{ce} 可能捕获类别级信息，而 $L_{similarity}$可能捕获实例级信息。基于上述分析，我们提出一种简单的方法来利用 L_{ce} 与 $L_{similarity}$之间的关系。

(8) 中有两个 τ。我们将 q_1 的 τ 记为 τ_1，将z_2 的 τ 记为 τ_2。τ_1 和 τ_2 对梯度的作用截然不同。本质上，(14) 中的 τ 是 τ_1，它直接影响$\|\frac{\partial L_{mce}}{\partial q_1}\|^2$的幅值。$\tau_2$ 会调整 $\|\mathbf{P}(z_2)\|$的幅值。τ_1 和 $\tilde{q}_1$ 可能会影响 q_1的梯度，这对于构建 L_{ce} 与 $L_{similarity}$之间的关系至关重要。上述分析解释了为什么我们可以为 τ_1和 τ_2设置不同的值。如 (14) 所示，使 τ_1 具有自适应性的一个基本设置是 $\tau_1 = \|\mathbf{P}(z_2)\|$。我们还在附录中对自适应 τ_1 的设置进行了详细分析。此外，(14) 与 (11) 之间的差异表明，p的 l_2范数对于获得合适的梯度至关重要。因此，损失函数为

$$L_{iccl} = -\sum_i p(i|z_2,\tau_2)\log p(i|\frac{q_1}{\|q_1\|},\tau_1)$$
$$= (\log\sum_j \exp\frac{\tilde{q}_1^{(j)}}{\tau_1}) - (\sum_i p(i|z_2,\tau_2)\frac{\tilde{q}_1^{(i)}}{\tau_1}). \quad (15)$$

该损失函数与 DINO 和 SwAV 中的损失相似。不过，DINO 和 SwAV 中损失的输入并未进行 l_2归一化。此外，在该公式中，使用 τ_1 来控制 $\tilde{q}_1$ 的幅值。对于 SwAV 和 DINO，τ_1 可用于生成基本概率分布。如图 4 所示，我们的损失函数可以与 $L_{similarity}$ 相关联。这一特性或许能防止我们的损失导致不平衡聚类，并使训练更加合理。例如，在训练开始时，我们可以使用 $L_{similarity}$；当网络基本能够提取实例级特征时，则使用 L_{iccl}。

L_{ce} 与 $L_{similarity}$ 之间的关系有助于网络同时利用实例级信息和类别级信息。与某些基于聚类的方法不同，所推导出的方法受超参数和平衡机制（例如 Sinkhorn-Knopp 算法）的影响较小。我们的方法可以简单地理解为关注较大值所在的维度，以提供跨实例监督。

E. 概率的均匀性

在 SwAV 和 DINO 中，监督标签 $\mathbf{P}(z_2)$ 通过平衡机制生成，例如 Sinkhorn-Knopp 算法和移动平均中心化，以防止不平衡聚类。传统的交叉熵损失可能无法避免平凡解。基于 $L_{similarity}$与 L_{iccl}之间的关系，我们的方法受不平衡聚类的影响较小。在我们的方法中，只需对损失函数添加一些正则化。我们假设批量数据B的输出是 $P = \{q_1, ..., p_N\}$。均匀性假设表明，每个维度上的输出都应大致接近，可表示为

$$p(i|\theta) = \frac{1}{|B|}\sum_{j=1}^{|B|} p(i|p_j,\tau,\theta),$$
$$\min_\theta D_{KL}(P||Q) = \sum_i q(i)log\frac{q(i)}{p(i|\theta)}, \quad (16)$$

其中，$q(i) = \frac{1}{C}$ 表示均匀分布。θ 表示参数。P 和 Q 分别表示估计概率分布和期望概率分布。我们在（16）中使用 P和 Q 之间的 KL 散度。表征的偏差将用于更新网络参数。λ_r 用于调整均匀性正则化的强度。最终损失为

$$L_{final} = L_{iccl} + \lambda_r \times (\sum_i q(i)log\frac{q(i)}{p(i|\theta)}). \quad (17)$$

默认情况下，一半的训练使用 $L_{similarity}$ 来保留实例级信息，另一半训练使用 L_{final}来保留类别级信息。

F. 不同损失之间的细节

表II展示了不同损失函数之间的关系。为方便起见，本文给出了对齐项的梯度幅值。对齐项表示该方法如何学习两个相关表示之间的信息。我们还给出了每个损失函数对应的梯度幅值上界。表II阐明了我们的方法与其他损失函数之间的关系。由于 $\tau \ll \|q_1\|$ （已在第III-C节中分析），$L_{similarity}$ 的上界小于 L_{ce} 。 $L_{similarity}$ 的梯度幅值上界与我们的方法相似，这也是我们的方法与 L_{ce}之间的核心区别。这一点使得$L_{similarity}$ 可以被我们的方法替代。此外，我们的方法与 L_{ce} 的对齐项相似，这表明我们的方法可以利用交叉熵损失来学习类内实例之间的相似性。因此，我们的方法可以在训练初期使用 $L_{similarity}$，并在经过若干轮次后用 L_{iccl} 替换 $L_{similarity}$，以学习类内信息。

IV. 实验

在本节中，我们针对自监督表示学习的模型设计开展了一系列实验。

A. 基线设置

我们的方法可以轻松地与 BYOL 和 SimSiam 结合。我们采用 BYOL 的设置作为基线。具体而言，所有数据集的默认温度 τ_2 均为 0.07。 τ_1 默认为 0.1。所有实验均使用余弦衰减学习率调度方案 [42] 。所有增强策略和初始化方法均与 BYOL 相同。对于 ResNet [43]，我们将每个残差块最后一个批归一化（BN） [45] 层中的缩放参数初始化为 0 [44]。为加快训练速度，我们的所有模型均采用混合精度训练。为确保比较公平，每种方法的增强策略和初始化设置保持一致。增强和初始化的详细信息见附录。

1) Imagenette 设置：我们使用 Imagenette [40] 开展基础实验。按照 BYOL 的设置，我们使用 LARS [46]，基准学习率 (lr) = 2.0，训练 1000 轮次，权重衰减为 = 1e-6，动量为 = 0.9，批量大小为 = 256。根据[19]， lr 为（基础 lr） $\times \frac{BatchSize}{256}$ 。骨干网络为 ResNet-18。投影器是一个 3 层多层感知机（MLP），预测器是一个 2 层 MLP。输出维度为 512。此处我们不使用动量编码器。

2）ImageNet 设置：我们使用 ImageNet [47] 来验证我们的表征。我们使用 LARS，基础学习率为 lr = 0.3，权重衰减为 = 1e-6，动量为 = 0.9, $\lambda_r = 5$，批量大小为 = 1024。骨干网络为 ResNet-50。投影器是一个输出维度为 256 的 3 层多层感知机。预测器是一个输出维度为 256 的 2 层多层感知机。动量编码器的动量为 0.99。

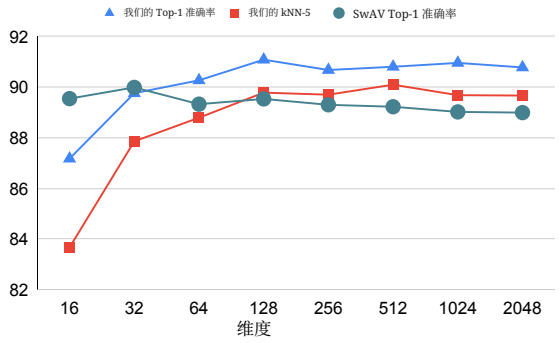


图5。**输出维度的对比结果**。kNN-5 表示当 K 为 5 时，K近邻的结果。

表III**超参数分析-参数 λ_r 和 τ_2**。 结果为 Imagenette 上的 Top-1准确率（%），取三次独立实验的平均值。

τ_2	λ_r 0	0.5	1.0	2.5	5.0	7.5	10.0
0.05	90.77	90.90	91.04	90.91	90.95	90.72	90.76
0.07	91.18	91.15	91.21	91.14	91.08	90.71	90.71
0.1	90.74	90.93	91.02	90.89	91.05	90.79	90.56

3）线性评估：给定预训练网络，我们使用冻结特征（从 ResNet 进行平均池化后）训练监督线性分类器。对于 Imagenette，分类器使用基础学习率 lr = 0.2、权重衰减 = 0、动量 = 0.9、轮次 = 100 和批量大小 = 4096。优化器为带 Nesterov 动量的随机梯度下降。对于 ImageNet，我们遵循 [10] 中的设置。线性分类器训练使用基础学习率 = 0.3，并采用余弦衰减调度，训练 100 轮次；权重衰减为 = 1e-6，动量为 =0.9，批量大小为 = 256，使用 SGD 优化器。

B. 超参数分析

1) 超参数 λ_r： 在我们的方法中，我们使用 λ_r 来规范 q_1的均匀性。表 III 展示了不同 λ_r下的结果。由于与 $L_{similarity}$存在相关性，我们的方法可能较少受到错误聚类的影响。实例级学习降低了对聚类生成的梯度的依赖。与 SwAV 和 DINO 相比，我们的方法无需施加任何平衡机制。当 $\lambda_r = 0$时，我们的方法表现良好。相比之下，SwAV 依赖 Sinkhorn-Knopp 算法，而 DINO 依赖中心化。

2) 超参数 τ_2： 图 2 展示了不同 τ下的结果。随着 τ 减小，性能会变好。这一现象与（14）一致。不恰当的 τ 会放大或缩小 $\|\nabla_{q_1} L\|$的幅值。表 III 分析了 τ_2的影响。我们发现， $\tau_2 = 0.07$在该数据集上能取得更好的结果。事实上， τ_2 只会影响 $\|\mathbf{P}(z_2)\|$。 $\|\mathbf{P}(z_2)\|$ 只会受到特征的维度和 τ_2的影响。因此， $\tau_2 = 0.07$可能适用于其他数据集。

3）输出维度的超参数：图5展示了不同维度下的结果。我们的方法表现稳定。

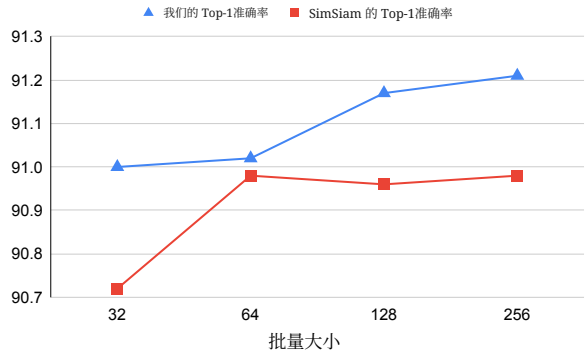


图 6. **不同批量大小的实验。**

表 IV**T的消融研究** $f(i,\ z_2)$ **在 (4) 中。** SK 是 Sinkhorn-Knopp 算法的简称。中心化表示 DINO 中的中心化过程。这里我们只改变 $f(i,\ z_2)$。因此，我们使用 $\tilde{p}$ 而不是 p 作为 $Softmax$的输入。

SK	中心化	$p(i\|z_2,\ \tau=\tau_1)$	$p(i\|z_2,\ \tau=\tau_2<\tau_1)$
90.85	90.49	90.74	91.21

尽管该数据集只有 10 个类别，但我们的方法在维度较大时仍能取得良好结果。相比之下，我们发现，在 I magenette 中，某些基于聚类的方法在原型数量过多时无法取得良好结果。这表明，$L_{similarity}$ 与L_{iccl} 之间的关系有助于减轻错误聚类的影响。

4) 批量大小的超参数：图 6 展示了不同批量大小下的结果。我们遵循 BYOL，累积 N 步的梯度，因此学习率保持不变。正如我们推测的那样，我们的方法受批量大小的影响较小，这与基于对比学习的方法类似。

C. 损失函数消融实验

首先，如图 4 所示，$\|p\|$ 对于建立与 $L_{similarity}$ 的关系至关重要。我们的损失与 L_{ce} 之间的主要区别在于 $Softmax$ 的输入。根据梯度分析，$\|p\|$ 是输入 $Softmax$ 的更合适形式。这一观点对我们的方法至关重要。

我们选择不同的方法生成概率分布，以监督 q_1 的更新。表 IV 展示了结果。Sinkhorn-Knopp 算法和中心化会根据数据批次调整概率分布。尽管采用了较小的 τ_2，$\|\mathbf{P}(z_2)\|$ 的幅值仍可能较小。然而，我们的方法没有中心化机制，这可能导致 $\|\mathbf{P}(z_2)\|$ 较大。此外，这种重新调整可能会扰乱伪标签，干扰训练。因此，当 $\tau_2=\tau_1$ 时，监督概率可能不够尖锐，从而无法捕捉不同图像之间的相似性。

D. 其他方法比较

我们首先在 Imagenette 中将我们的方法与其他自监督方法进行比较。表 VI 展示了不同自监督方法的结果。我们发现，SimSiam、MoCo 和 BYOL 可以轻松扩展到该数据集，这表明这些方法能够适应不同的数据集。BarlowTwins 关注不同通道之间的相关性。我们发现，该方法与这些基于聚类的方法相似。较大的维度并不适用于该数据集。SwAV 和 DINO 是基于聚类的方法。超参数是针对 ImageNet 设置的。我们发现，这些超参数在该数据集中并不十分有效。我们调整了原型数量，并选择最佳结果。然而，其性能仍然差于基于对比学习的方法的结果。SwAV 和 DINO 都严重依赖均匀性正则化。在 ImageNet 和 Imagenette 中，如果没有均匀性正则化，这些方法可能会失效。我们的方法通过 (15) 利用实例级信息和类别级信息。因此，我们的方法可能较少受到错误聚类问题的影响。此外，τ_1 和 τ_2 在 SwAV 和 DINO 中是手动设置的。在我们的方法中，自适应的 τ_1 可以产生具有竞争力的结果。这一点验证了 (14) 以及对梯度的分析。

基于 Imagenette 中使用的超参数，我们在 ImageNet 上进行了实验。我们发现，这些超参数在 ImageNet 上仍然适用。表 V 展示了 ImageNet 上的结果。我们从损失函数和维度的角度分析结果。所有骨干网络均为 ResNet-50。τ_1 的设置固定为 0.1。这可能是超参数方面的问题，我们将在未来寻找合适的超参数。

1) 损失函数：传统的对比学习方法使用相似性损失。对于 SimSiam，作者发现用 L_{ce} 替换 $L_{similarity}$ 可能会导致较差的结果（67.3 和 63.2）。这些基于对比学习的方法的结果可能较少受到维度的影响。然而，使用 InfoNCE [2] 的方法可能依赖较大的批量大小。MoCo 需要较大的队列大小或批量大小才能保持良好的结果。BYOL 在 ImageNet 和 Imagenette 上都很稳定。总体而言，相似性损失可能是捕获实例级信息的一种好方法。BarlowTwins 试图利用不同维度之间的相关性。这种方法可能受益于较大的维度。当输出维度为 8192 时，BarlowTwins 的结果具有竞争力。然而，当输出维度变小时，性能可能会大幅下降 [10]。因此，该超参数可能适用于 ImageNet，但不适用于其他数据集。SwAV 和 DINO 都利用 L_{ce} 进行在线聚类。正如我们在命题 1 中提出的，L_{ce} 可能有助于关联不同的实例。我们的方法建立了 $L_{similarity}$ 与 L_{ce} 之间的关系。因此，我们的方法也能捕获类别级信息。从这个角度来看，我们的方法可能比其他方法利用更多信息。当类别级信息难以捕获时，该方法可以利用实例级信息来提供稳健的训练。

2) 维度：在实验中，我们发现这些基于聚类的方法可能对数据集规模较为敏感。这些方法中原型数量这一超参数在极小数据集上可能无法发挥良好作用。在 ImageNet 上，这些方法仍然需要较高的维度。减少原型数量或数据多样性

表V**不同自监督-学习方法在 ImageNet 上的结果**。所有结果均使用两个 224×224 视图进行预训练。所有方法均使用 ResNet-50 作为骨干网络。我们不使用任何技巧（例如 SwAV 中的多裁剪和 SimSiam 中固定学习率）。* 表示训练 400 轮次的结果。‡ 表示使用多裁剪的结果（这一技巧可能会提升结果）。排名前四的自监督方法以带下划线标示。

方法	基本损失	批量大小	维度	Top-1准确率（%） 100 轮次	300 轮次
对比方法					
SimCLR [8]	InfoNCE	4096	2048	66.5	*69.8
MoCo v2 [48]	InfoNCE	256（队列大小为 65536）	256	67.4	*71.1
MoCo v3 [23]	InfoNCE	1024	256	68.1	**72.3**
BYOL [18]	$L_{similarity}$	1024	256	66.0	72.2
SimSiam [19]	$L_{similarity}$	256	2048	67.3	*70.8
	L_{ce}	256	2048	63.2	-
BarlowTwins [10]	BarlowTwins 损失	1024	8192	67.4	71.4
基于聚类的方法					
SeLa [30]	L_{ce}	4096	3000	61.5	*67.2
SwAV [31]	L_{ce}	256（队列大小为 4096）	3000	66.5	*70.7
DINO [11]	L_{ce}	1024	65536	67.8	‡72.1
Ours	L_{ce}	1024	256	**68.2**	71.7
Ours	L_{ce}	1024	512	68.1	71.5

表 VI：**在 IMAGENETTE 上不同方法的结果**。†表示默认设置无法迁移至 Imagenette 实验的方法，此处报告其最佳结果。 ‡表示未使用均匀性正则化的方法（SwAV 中的 Sinkhorn-Knopp、DINO 中的中心化，以及我们的方法中的 λ_r ）。

方法	Top-1准确率 (%)	KNN-5 Top-1准确率 (%)
对比方法		
SimSiam	**90.98**	89.90
MoCo	90.39	88.81
BYOL	90.87	**90.26**
BarlowTwins†	87.06	87.31
基于聚类的方法		
SwAV†	89.99	88.61
DINO†	88.86	88.41
‡SwAV	fail	-
‡DINO	fail	-
‡我们的方法	91.18	89.98
我们的方法（固定 τ_1）	91.21	**90.31**
我们的方法（自适应 τ_1）	**91.23**	89.94

可能会影响这些方法。然而，如图5和表 V 所示，我们的方法与这些基于对比学习的方法相似。当类别数为 10 时，我们的方法在维度为 2048 时可以取得有竞争力的结果。当类别数为 1000 时，我们的方法在维度为 256 时也可以取得有竞争力的结果。此外，使用 L_{ce} 有助于发现不同图像之间的相似性。相反，SimSiam 使用 L_{ce} 时表现较差。 L_{ce} 与 $L_{similarity}$ 之间建立的关系增强了所学信息。

E. 迁移到其他任务

继 SimSiam [19], [53],之后，我们进行了若干迁移学习实验。在表 VII 中，我们比较了表示通过迁移学习获得的质量。我们在 VOC [54]数据集上微调参数。实验设置遵循 [55] 中的代码库。我们发现，我们的预训练模型与某些自监督方法相比，结果并不具竞争力。我们推测，我们的方法试图学习类别级一致性 [56]。事实上，我们的目标是在自监督训练过程中学习类别信息（图像的类别）。这一尝试可能会削弱目标检测的性能，并导致中层信息被丢弃。然而，正如其他自监督方法中所提到的，自监督训练可能比监督学习取得更好的结果。

表 VIII 展示了 COCO 数据集上的迁移学习结果。对于 COCO 数据集，我们只在 MoCo 的 GitHub 上找到了代码库。因此，我们沿用了 MoCo 的设置。COCO 数据集上的性能与 VOC 数据集上的性能一致。我们的方法性能接近最佳方法。事实上，我们发现，MoCo 在 COCO 和 VOC 上使用的学习率可能不适用于我们的预训练模型。我们可能会在后续版本中为迁移学习寻找合适的超参数。

V. 结论

我们的方法在概念上与 SwAV 和 DINO 类似。这些方法都利用特征级交叉熵进行无监督学习。然而，SwAV 和 DINO 需要采用一些方法来平衡概率分布。例如，SwAV 使用 Sinkhorn-Knopp 算法来平衡批次中所有实例的概率分布。DINO 则通过中心化来累积概率分布的偏差。中心化机制可能会改变后续训练中不同原型的强度。在 SwAV 和 DINO 中，作者强调了保持均匀性的重要性。然而，在本文中，我们降低了对均匀性机制的依赖。从梯度的视角出发，我们发现

表VII**迁移学习**。VOC 07 检测：Faster R-CNN[49] 在 VOC 2007 训练集上微调，并在 VOC 2007 测试集上评估。VOC 07+12 检测：Faster R-CNN 在 VOC 2007 训练集 + 2012 上微调，并在 VOC 2007 测试集上评估。使用视觉 Transformer[50] 作为骨干网络的方法可能与 Faster R-CNN 不兼容。我们还在附录中提供了 COCO[51] 上的迁移学习结果。

方法	VOC 07 检测			VOC 07+12 检测		
	AP_{all}	AP_{50}	AP_{75}	AP_{all}	AP_{50}	AP_{75}
监督学习	42.4	74.4	42.7	53.5	81.3	58.5
SimCLR	46.8	75.9	50.1	55.5	81.8	61.4
MoCo v2	**48.5**	77.1	**52.5**	**57.0**	82.5	63.3
BYOL	47.0	77.1	49.9	55.3	81.4	61.1
SwAV	46.5	75.5	49.6	55.4	81.5	61.4
SimSiam（最优）	**48.5**	**77.3**	**52.5**	**57.0**	82.4	**63.7**
BarlowTwins	-	-	-	56.8	**82.6**	63.4
Ours	47.2	75.7	51.4	55.5	81.9	61.5

表 VIII**迁移 学习**。 COCO目标检测和COCO实例分割：掩膜区域卷积神经网络 C-4 [52]（2倍训练计划）在 COCO 2017训练集上微调， [51], 在 COCO 2017验证集上评估。

方法	COCO目标检测			COCO实例分割		
	AP	AP_{50}	AP_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
1倍训练计划						
监督学习	38.2	58.2	41.2	33.3	54.7	35.2
SimCLR	37.9	57.7	40.9	33.3	54.6	35.3
MoCo	**39.2**	58.8	42.5	34.3	55.5	36.6
BYOL	37.9	57.8	40.9	33.2	54.3	35.0
SwAV	37.6	57.6	40.3	33.1	54.2	35.1
SimSiam	**39.2**	**59.3**	42.1	**34.4**	**56.0**	**36.7**
BarlowTwins	**39.2**	59.0	**42.5**	34.3	**56.0**	36.5
Ours	38.4	58.3	41.2	33.2	54.7	35.0
2倍训练计划						
监督学习	40.0	59.9	43.1	34.7	56.5	36.9
MoCo	40.7	60.5	44.1	35.4	57.3	37.6
Ours	39.9	59.8	43.2	34.9	56.6	37.1

一种损失函数，其行为可能与相似性损失相似。这一点对于摆脱均匀性机制至关重要。

我们的方法使用的损失函数与基于对比学习的方法完全不同。然而，通过推导梯度，可以发现该方法与相似性损失相关。这一视角有助于我们的方法保持基于对比学习的方法所具有的鲁棒性。与那些直接将q_1输入 $Softmax$ 的方法相比，合理的梯度也提供了一种稳定、平滑的训练视角。如图 4 所示， $Softmax$的输入对于建立这种相关性至关重要。事实上，我们的方法可以解释为通过概率分布来执行相似性损失。这一视角解释了为什么我们的方法受维度和均匀性正则化的影响较小。通过最大化实例的概率分布，该方法可以隐式地学习图像之间的相似性。

致谢

本研究得到中国国家重点研发计划（项目编号：2022YFF0711404）、江苏省自然科学基金（项目编号：BK20201250）、国家自然科学基金（项目编号：62172090）、中国人工智能学会-华为昇思开放基金、南京市海外学者科技创新计划（项目编号：1109012301）以及东南大学启动研究基金（项目编号：RF1028623097）的资助。我们感谢东南大学大数据计算中心为本文的数值计算提供设施支持。桂杰是本文的通讯作者。

APPENDIX
DERIVATION OF EQUATIONS IN MAIN PAPER

正文中公式（*9*）和公式（*10*）的推导

对于损失函数:

$$L_{similarity} = -\langle \tilde{q}_1, \tilde{z}_2 \rangle, \tag{18}$$

我们可以得到

$$\tilde{q}_1 = \frac{q_1}{\|q_1\|},$$
$$and \quad \frac{\partial L}{\partial \tilde{q}_1} = -\tilde{z}_2. \tag{19}$$

这里我们用 $p^{(i)}$ 来索引向量 p 中的第 i^{th} 个元素。然后，我们可以计算 q_1 的梯度:

$$\frac{\partial \tilde{q}_1^{(i)}}{\partial q_1^{(i)}} = \frac{1}{\|q_1\|} - q_1^{(i)} \frac{q_1^{(i)}}{\|q_1\|^3} = \frac{1}{\|q_1\|}(1 - \tilde{q}_1^{(i)} \cdot \tilde{q}_1^{(i)}), \tag{20}$$

$$\frac{\partial \tilde{q}_1^{(j)}}{\partial q_1^{(i)}} = -q_1^{(j)} \frac{q_1^{(i)}}{\|q_1\|^3} = \frac{1}{\|q_1\|}(-\tilde{q}_1^{(j)} \cdot \tilde{q}_1^{(i)}), \tag{21}$$

$$\begin{aligned}\frac{\partial L}{\partial q_1^{(i)}} &= \sum_j \frac{\partial L}{\partial \tilde{q}_1^{(j)}} \frac{\partial \tilde{q}_1^{(j)}}{\partial q_1^{(i)}} \\ &= \frac{\partial L}{\partial \tilde{q}_1^{(i)}} \frac{\partial \tilde{q}_1^{(i)}}{\partial q_1^{(i)}} + \sum_{j!=i} \frac{\partial L}{\partial \tilde{q}_1^{(j)}} \frac{\partial \tilde{q}_1^{(j)}}{\partial q_1^{(i)}} \\ &= \frac{1}{\|q_1\|}(\frac{\partial L}{\partial \tilde{q}_1^{(i)}} - \tilde{q}_1^{(i)}(\sum_j \tilde{q}_1^{(j)} \cdot \frac{\partial L}{\partial \tilde{q}_1^{(j)}})) \\ &= \frac{1}{\|q_1\|}(\frac{\partial L}{\partial \tilde{q}_1^{(i)}} - \tilde{q}_1^{(i)} \cdot \langle \tilde{q}_1, \frac{\partial L}{\partial \tilde{q}_1} \rangle) \\ &= \frac{1}{\|q_1\|}(-\tilde{z}_2^{(i)} - \tilde{q}_1^{(i)} \cdot \langle \tilde{q}_1, -\tilde{z}_2 \rangle). \end{aligned} \tag{22}$$

因此，主论文中的公式（9）得证。根据上述公式，我们可以得到

$$\begin{aligned}(\frac{\partial L}{\partial q_1^{(i)}})^2 =& \frac{1}{\|q_1\|^2}(\tilde{z}_2^{(i)^2} + \tilde{q}_1^{(i)^2} \cdot \langle \tilde{q}_1, -\tilde{z}_2 \rangle^2 \\ & - 2(-\tilde{z}_2^{(i)} \cdot \tilde{q}_1^{(i)}) \langle \tilde{q}_1, -\tilde{z}_2 \rangle), \end{aligned} \tag{23}$$

$$\begin{aligned}\sum_i (\frac{\partial L}{\partial q_1^{(i)}})^2 =& \frac{1}{\|q_1\|^2}(\| - \tilde{z}_2\|^2 + \langle \tilde{q}_1, -\tilde{z}_2 \rangle^2 - 2\langle \tilde{q}_1, -\tilde{z}_2 \rangle^2) \\ =& \frac{1}{\|q_1\|^2}(\| - \tilde{z}_2\|^2 - \langle \tilde{q}_1, -\tilde{z}_2 \rangle^2) \\ =& \frac{1}{\|q_1\|^2}(1 - \langle \tilde{q}_1, \tilde{z}_2 \rangle^2). \end{aligned} \tag{24}$$

正文中的公式（10）如上所述成立。

正文中公式（*11*）和公式（*12*）的推导

对于损失函数：

$$L_{ce} = -\sum_i p(i|z_2, \tau) \log p(i|q_1, \tau),$$

$$where \quad p(i|x, \tau) = \frac{\exp(\frac{x^{(i)}}{\tau})}{\sum_{j=1} \exp(\frac{x^{(j)}}{\tau})}, \tag{25}$$

我们可以得到另一种交叉熵损失：

$$\begin{aligned}L_{ce} &= -\sum_i p(i|z_2, \tau) \log \frac{\exp(\frac{q_1^{(i)}}{\tau})}{\sum_{j=1} \exp(\frac{q_1^{(j)}}{\tau})} \\ &= -\sum_i p(i|z_2, \tau)(\frac{q_1^{(i)}}{\tau} - log \sum_{j=1} \exp(\frac{q_1^{(j)}}{\tau})) \\ &= \sum_i p(i|z_2, \tau) \cdot log \sum_{j=1} \exp(\frac{q_1^{(j)}}{\tau}) - \sum_i p(i|z_2, \tau) \frac{q_1^{(i)}}{\tau} \\ &= log \sum_{j=1} \exp(\frac{q_1^{(j)}}{\tau}) - \sum_i p(i|z_2, \tau) \frac{q_1^{(i)}}{\tau}. \end{aligned} \tag{26}$$

这就是正文中的公式（12）。 q_1 的梯度可通过以下方式计算

$$\begin{aligned}\frac{\partial L_{ce}}{\partial {q_1}^{(i)}} &= \frac{1}{\tau} \frac{exp(\frac{q_1^{(i)}}{\tau})}{\sum_{j=1} \exp(\frac{q_1^{(j)}}{\tau})} - \frac{1}{\tau} p(i|z_2, \tau) \\ &= \frac{1}{\tau}(p(i|q_1, \tau) - p(i|z_2, \tau)). \end{aligned} \tag{27}$$

因此，主论文中的公式（11）得以成立。

主论文中公式（*14*）的推导

对于损失函数

$$L_{mce} = -\sum_i p(i|z_2, \tau) \frac{\tilde{q}_1^{(i)}}{\tau}, \tag{28}$$

我们有

$$\frac{\partial L_{mce}}{\partial \tilde{q}_1^{(i)}} = -\frac{1}{\tau} p(i|z_2, \tau). \tag{29}$$

$$\begin{aligned}\frac{\partial L_{mce}}{\partial q_1^{(i)}} &= \sum_j \frac{\partial L_{mce}}{\partial \tilde{q}_1^{(j)}} \frac{\partial \tilde{q}_1^{(j)}}{\partial q_1^{(i)}} \\ &= \frac{1}{\|q_1\|}(\frac{\partial L_{mce}}{\partial \tilde{q}_1^{(i)}} - \tilde{q}_1^{(i)} \cdot \langle \tilde{q}_1, \frac{\partial L_{mce}}{\partial \tilde{q}_1} \rangle), \end{aligned} \tag{30}$$

$$\begin{aligned}\|\frac{L_{mce}}{\partial q_1}\|^2 &= \frac{1}{\tau^2 \|q_1\|^2}(\|\mathbf{P}(z_2)\|^2 - \langle \tilde{q}_1, \mathbf{P}(z_2) \rangle^2) \\ &= \frac{\|\mathbf{P}(z_2)\|^2}{\tau^2 \|q_1\|^2}(1 - \langle \tilde{q}_1, \frac{\mathbf{P}(z_2)}{\|\mathbf{P}(z_2)\|} \rangle^2). \end{aligned} \tag{31}$$

主论文中公式（*15*）的 τ_1 说明

对于损失函数

$$L_{iccl} = (\log \sum_j \exp \frac{\tilde{q}_1^{(j)}}{\tau_1}) - (\sum_i p(i|z_2, \tau_2) \frac{\tilde{q}_1^{(i)}}{\tau_1}), \tag{32}$$

我们有

$$\frac{\partial L_{iccl}}{\partial \tilde{q}_1^{(i)}} = \frac{1}{\tau_1}(p(i|q_1, \tau_1) - p(i|z_2, \tau_2)), \tag{33}$$

$$\begin{aligned}\frac{\partial L_{iccl}}{\partial q_1^{(i)}} &= \sum_j \frac{\partial L_{iccl}}{\partial \tilde{q}_1^{(j)}} \frac{\partial \tilde{q}_1^{(j)}}{\partial q_1^{(i)}} \\ &= \frac{1}{\|q_1\|}(\frac{\partial L_{iccl}}{\partial \tilde{q}_1^{(i)}} - \tilde{q}_1^{(i)} \cdot \langle \tilde{q}_1, \frac{\partial L_{iccl}}{\partial \tilde{q}_1} \rangle), \end{aligned} \tag{34}$$

$$\|\frac{L_{iccl}}{\partial q_1}\|^2 = \frac{\|\mathbf{P}(q_1) - \mathbf{P}(z_2)\|^2}{\tau_1^2 \|q_1\|^2}(1 - \langle \tilde{q}_1, \frac{\mathbf{P}(q_1) - \mathbf{P}(z_2)}{\|\mathbf{P}(q_1) - \mathbf{P}(z_2)\|} \rangle^2). \tag{35}$$

在主论文中，我们设置 $\tau_1 = \|\mathbf{P}(z_2)\|$。 τ_1 的值是自适应的。然而，随着$\mathbf{P}(q_1)$逐渐接近$\mathbf{P}(z_2)$，梯度的幅度可能会消失。因此，我们为 τ_1提供了另一种设置。默认的 τ_1 是一个超参数（例如，DINO [1]中的 $\tau_1 = 0.1$ ）。为了使 τ_1 能够针对不同实例进行自适应，我们将 τ_1 设为 $min(\tau_1, \|\mathbf{P}(z_2)\|)$。

实现细节

代码已开源。详情请参阅 **Code/README.md**。我们提供了多种方法在 ImageNet 和 Imagenette 上的配置文件，便于复现不同方法的结果。

初始化。

对于 ResNet 骨干网络，卷积层的权重采用 He初始化，偏置初始化为 0。其他组件（例如投影器、预测器）的全连接层权重和偏差采用 Xavier初始化 [57]。这些设置遵循 BYOL 中的详细说明。

数据增强。

在自监督训练期间，我们使用以下图像增强方法（类似 PyTorch 的代码）。

- $RandomResizedCrop$ 面积比例在 0.08 到 1.0 之间均匀采样，宽高比在 3/4 到 4/3 之间按对数采样。
- $Resize$将图像块调整为 224x224 的目标尺寸。
- $RandomHorizontalFlip$ 以 0.5 的概率对图像进行处理。
- $ColorJitter$ 通过参数 {0.4, 0.4, 0.4, 0.1}调整图像的{亮度、对比度、饱和度和色调} 。此数据增强操作以 0.8 的概率随机应用。
- $RandomGrayscale$以 0.2 的概率对图像进行处理。
- $GaussianBlur$ 使用标准差为 [0.1, 2.0] 的高斯核处理图像。对于两个独立的变换，此数据增强操作分别以 1.0 和 0.1 的概率随机应用。
- $Solarization$ 以 0.2 的概率对图像应用其中一种变换。
- $ToTensor$。将 [0, 255] 的值缩放至 [0.0, 1.0]。
- $Normalize$ 具有估计均值和标准差的图像。

优化器和学习率。

在 Imagenette 上进行实验时，我们使用 LARS， lr =2.0, 训练 1000 轮次，权重衰减为 = 1e-6，动量为 = 0.9，批量大小为 = 256。对于使用动量编码器的方法，动量编码器的动量值为 0.996。这些设置与 BYOL 的 GitHub 仓库相同。在 ImageNet 上进行实验时，我们使用 LARS，基准学习率为 lr = 0.3，权重衰减为 =1e-6，动量为 = 0.9，批量大小为 = 1024。动量编码器的动量值为 0.99。在线性评估中，线性分类器训练使用基准学习率 = 0.3，并采用余弦衰减调度，训练 100 轮次，权重衰减为 = 1e-6，动量为= 0.9，批量大小为 = 256 ，优化器采用 SGD。我们还尝试了 LARS 优化器，基准学习率为 lr = 0.1，权重衰减为 = 0，动量为 = 0.9，轮次为 = 100，批量大小为 = 4096。该优化器可能取得有竞争力的结果。

目标检测的超参数。

我们使用 Detectron2 库训练检测模型，并严格遵循 MoCo 的评估设置。我们使用 Faster R-CNN C-4 检测模型。骨干网络由我们的预训练模型初始化。对于 VOC 07 和 VOC07+12，训练共进行 24K 次迭代，批量大小为 16，使用 8 个 GPU 和同步批量归一化。模型的初始学习率为 0.1，在 18K 次和 22K 次迭代后分别降低 10 倍。我们采用 1000 次迭代的线性预热 。对于 COCO 数据集，我们在 COCO 2017训练集上训练 Mask R-CNN C-4 骨干网络。我们使用 0.03 的学习率，其他参数与 MoCo 的 GitHub 中设置相同。

参考文献

[1] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, 《通过互信息估计与最大化学习深度表征》，载于国际学习表征会议，2018年。[2] A. v. d. Oord, Y. Li, and O. Vinyals, 《通过对比预测编码进行表征学习》，*arXiv*预印本 *arXiv:1807.03748*，2018年。[3] P. Bachman 等, 《通过最大化不同视图间的互信息来学习表征》，载于神经信息处理系统进展（*NeurIPS*），2019年，第15 535–15 545页。[4] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin, 《在未经整理的数据上对图像特征进行无监督预训练》，载于*IEEE/CVF* 国际计算机视觉会议论文集，2019年，第2959–2968页。[5] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, 《S4l：自监督半监督学习》，载于*IEEE/CVF* 国际计算机视觉会议论文集，2019年，第1476–1485页。[6] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, 《用于无监督视觉表征学习的动量对比》，载于*IEEE/CVF*计算机视觉与模式识别会议论文集，2020年，第9729–9738页。[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, 《语言模型是少样本学习者》，载于神经信息处理系统进展（*NeurIPS*），H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin 编，2020年。[8] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, 《用于视觉表征对比学习的简单框架》，载于第*37*届国际机器学习会议论文集，*ICML 2020*，第119卷，2020年，第1597–1607页。[9] O. Henaff, 《通过对比预测编码实现数据高效图像识别》，载于国际机器学习会议。PMLR，2020年，第4182–4192页。[10] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, 《Barlow双胞胎：通过冗余减少实现自监督学习》，载于第*38 2021, 18 24 2021, th*届国际机器学习会议，*ICML*——*7*月线上活动，《机器学习研究论文集》系列，M. Meila and T. Zhang 编，第139卷。PMLR，2021年，第12 310–12 320页。[在线]。可访问：http://proceedings.mlr.press/v139/ zbontar21a.html [11] M. Caron, H. Touvron, I. Misra, H. J´egou, J. Mairal, P. Bojanowski, and A. Joulin, 《自监督视觉变换器中的涌现特性》，*arXiv* 预印本 *arXiv:2104.14294*，2021年。[12] K. He, X. Chen, S. Xie, Y. Li, P. Doll´ar, and R. B. Girshick, 《掩码自编码器是可扩展的视觉学习器》，*CoRR*，第abs/2111 06377, 卷。。2021 [ ] 2111 在线。可访问：https://arxiv.org/abs/ 。 06377 [13] P. Bojanowski and A. Joulin, 《通过预测噪声进行无监督学习》，载于国际机器学习会议。PMLR，2017年，第517–526页。[14] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, 《利用样本卷积神经网络进行判别式无监督特征学习》，*IEEE*模式分析与机器智能汇刊，第38卷，第9期，第1734–1747页，2015年。[15] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, 《通过非参数实例判别进行无监督特征学习》，载于*IEEE*计算机视觉与模式识别会议论文集，2018年，第3733–3742页。[16] Y. Cao, Z. Xie, B. Liu, Y. Lin, Z. Zhang, and H. Hu, 《用于无监督视觉特征学习的参数化实例分类》，载于神经信息处理系统进展（*NeurIPS*），H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin 编，2020年。[17] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, 《什么样的视图有利于对比学习? 》，载于神经信息处理系统进展（*NeurIPS*），2020年。

[18] J. Grill、F. Strub、F. Altch´e、C. Tallec、P. H. Richemond、E. Buchatskaya、C. Doersch、B. ´A. Pires、Z. Guo、M. G. Azar、B. Piot、K. Kavukcuoglu、R. Munos 和 M. Valko，“自举潜在表示——一种自监督学习的新方法”，载于《神经信息处理系统进展》（NeurIPS），2020年。[19] X. Chen 和 K. He，“探索简单的孪生表示学习”，载于IEEE计算机视觉与模式识别会议（CVPR 2021），线上活动，2021年6月19日至25日。计算机视觉基金会/IEEE，2021年，第15 750–15 758页。[20] P. H. Richemond、J.-B. Grill、F. Altch´e、C. Tallec、F. Strub、A. Brock、S. Smith、S. De、R. Pascanu、B. Piot 等，“BYOL即使没有批次统计量也能工作”，arXiv预印本 arXiv:2010.10241，2020年。[21] Y. Tian、L. Yu、X. Chen 和 S. Ganguli，“通过双深度网络理解自监督学习”，arXiv预印本 arXiv:2010 00578，2020年。.[22] Y. Tian、X. Chen 和 S. Ganguli，“无需对比对理解自监督学习动态”，载于第38届国际机器学习会议（ICML）论文集，2021年7月18日至24日，线上活动，丛书《机器学习研究论文集》，M. Meila 和 T. Zhang 编，第139卷。PMLR，2021年，第10 268–10 278页。 [在线]。获取地址：http://proceedings.mlr.press/v139/ tian21a.html[23] X. Chen、S. Xie 和 K. He，“自监督训练视觉变换器的实证研究”，arXiv预印本 arXiv:2104.02057，2021年。[24] I. Misra 和 L. v. d. Maaten，“自监督学习前文本不变表征”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2020年，第6707–6717页。[25] J. Yang、D. Parikh 和 D. Batra，“深度表征与图像簇的联合无监督学习”，载于IEEE计算机视觉与模式识别会议论文集，2016年，第5147–5156页。[26]J. Xie、R. B. Girshick 和 A. Farhadi，“ 用于聚类分析的无监督深度嵌入 ”，载于第 33 届国际机器学习会议（ICML 2016）论文集，纽约市，纽约州，美国，2016年6月19日至24日，丛书《JMLR研讨会与会议论文集》，M. Balcan 和 K. Q. Weinberger 编，第48卷。JMLR.org，2016年，第478–487页。 [在线]。获取地址：http://proceedings.mlr.press/v48/xieb16.html [27] X. Yan、I. Misra、A. Gupta、D. Ghadiyaram 和 D. Mahajan，“ClusterFit：提升视觉表征的泛化能力”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2020年，第6509–6518页。[28] J. Huang、Q. Dong、S. Gong 和 X. Zhu，“通过邻域发现进行无监督深度学习”，载于国际机器学习会议。PMLR，2019年，第2849–2858页。[29] M. Caron、P. Bojanowski、A. Joulin 和 M. Douze，“用于无监督视觉特征学习的深度聚类”，载于欧洲计算机视觉会议（ECCV）论文集，2018年，第132–149页。[30] Y. M. Asano 等，“通过同步聚类和表征学习进行自标记”，载于第8届国际学习表征会议（ICLR 2020），2020年。[31] M. Caron、I. Misra、J. Mairal、P. Goyal、P. Bojanowski 和 A. Joulin，“通过对比簇分配进行视觉特征的无监督学习”，载于《神经信息处理系统进展》（NeurIPS），2020年。[32] Y. Li、P. Hu、Z. Liu、D. Peng、J. T. Zhou 和 X. Peng，“对比聚类”，载于2021年AAAI人工智能会议（AAAI），2021年。[33] N. Zhao、Z. Wu、R. W. H. Lau 和 S. Lin，“为自监督表示学习提炼定位信息”，载于第35届AAAI人工智能会议（AAAI 2021）、第33届人工智能创新应用会议（IAAI 2021）、第11届人工智能教育进展研讨会（EAAI 2021），线上活动，2021年2月2日至9日。AAAI出版社，2021年，第10 990–10 998页。 [在线]。获取地址：https://ojs.aaai.org/index.php/ AAAI/article/view/17312[34] M. Cuturi，“Sinkhorn距离：极速计算最优传输”，《神经信息处理系统进展》（NeurIPS），第26卷，第2292–2300页，2013年。[35] R. Hadsell、S. Chopra 和 Y. LeCun，“通过学习不变映射进行降维”，载于2006年IEEE计算机学会计算机视觉与模式识别会议（CVPR'06），第2卷。IEEE，2006年，第1735–1742页。[36] T. Wang 和 P. Isola，“通过超球面上的对齐与均匀性理解对比表示学习”，载于第37届国际机器学习会议（ICML 2020）论文集，2020年7月13日至18日，线上活动，机器学习研究论文集丛书，第119卷，2020年，第9929–9939页。[37] T. Chen、S. Kornblith、K. Swersky、M. Norouzi 和 G. E. Hinton，“大型自监督模型是强大的半监督学习器”，载于《神经信息处理系统进展》（NeurIPS），H. Larochelle、M. Ranzato、R. Hadsell、M. Balcan 和 H. Lin 编，2020年。[38] G. Hinton、O. Vinyals 和 J. Dean，“提炼神经网络中的知识”，arXiv 预印本 arXiv:1503.02531，2015年。[39] L. Van der Maaten 和 G. Hinton，“使用 t-SNE 可视化数据。”《机器学习研究杂志》，第9卷，第11期，2008年。[40] F. J. Howard，“Imagenette 数据集”，https: //github.com/fastai/imagenette.，2020年。[41] K. He、X. Zhang、S. Ren 和 J. Sun，“深入研究整流器：在 ImageNet 分类上超越人类水平的表现”，载于《IEEE 国际计算机视觉会议论文集》，2015年，第1026–1034页。[42] I. Loshchilov 和 F. Hutter，“SGDR：带热重启的随机梯度下降”，载于第五届国际学习表征会议（ICLR 2017），法国土伦，2017年4月24日至26日，会议论文集。OpenReview.net，2017年。 [在线]。网址：https://openreview.net/forum?id=Skq89Scxx [43] K. He、X. Zhang、S. Ren 和 J. Sun，“深度残差网络中的恒等映射”，载于《欧洲计算机视觉会议》。施普林格，2016年，第630–645页。[44] P. Goyal、P. Doll´ar、R. Girshick、P. Noordhuis、L. Wesolowski、A. Kyrola、A. Tulloch、Y. Jia 和 K. He，“准确的大型小批量随机梯度下降：1小时内完成 ImageNet 训练”，arXiv 预印本 arXiv:1706.02677，2017年。[45] S. Ioffe 和 C. Szegedy，“批归一化：通过减少内部协变量偏移加速深度网络训练”，载于《第32届国际机器学习会议论文集》，ICML 2015，法国里尔，2015年7月6日至11日，2015年。[46] Y. You、I. Gitman 和 B. Ginsburg，“卷积网络的大批量训练”，arXiv 预印本 arXiv:1708.03888，2017年。[47] A. Krizhevsky、I. Sutskever 和 G. E. Hinton，“使用深度卷积神经网络进行 ImageNet 分类”，《神经信息处理系统进展》（NeurIPS），第25卷，第1097–1105页，2012年。[48] X. Chen、H. Fan、R. Girshick 和 K. He，“通过动量对比学习改进基线”，arXiv 预印本 arXiv:2003.04297，2020年。[49] S. Ren、K. He、R. Girshick 和 J. Sun，“Faster R-CNN：借助区域提议网络实现实时目标检测”，《神经信息处理系统进展》（NeurIPS），第28卷，第91–99页，2015年。[50] A. Dosovitskiy、L. Beyer、A. Kolesnikov、D. Weissenborn、X. Zhai、T. Unterthiner、M. Dehghani、M. Minderer、G. Heigold、S. Gelly、J. Uszkoreit 和 N. Houlsby，“一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer”，载于第九届国际学习表征会议（ICLR 2021），线上活动，奥地利，2021年5月3日至7日。OpenReview.net，2021年。 [在线]。网址：https://openreview.net/forum?id=YicbFdNTTy [51] T.-Y. Lin、M. Maire、S. Belongie、J. Hays、P. Perona、D. Ramanan、P. Doll´ar 和 C. L. Zitnick，“Microsoft COCO：上下文中的常见物体”，载于《欧洲计算机视觉会议》。施普林格，2014年，第740–755页。[52] K. He、G. Gkioxari、P. Doll´ar 和 R. B. Girshick，“掩膜区域卷积神经网络”，CoRR，第abs/1703.06870卷，2017年。 [在线]。网址：http://arxiv.org/abs/ 1703 06870 。P. Goyal、D. Mahajan、A. Gupta 和 I. Misra[53]，“自监督视觉表征学习的规模扩展与基准测试”，载于《IEEE/CVF 国际计算机视觉会议论文集》，2019年，第6391–6400页。[54] M. Everingham、L. Van Gool、C. K. Williams、J. Winn 和 A. Zisserman，“PASCAL 视觉目标类别（VOC）挑战赛”，《国际计算机视觉杂志》，第88卷，第2期，第303–338页，2010年。[55] Y. Wu、A. Kirillov、F. Massa、W.-Y. Lo 和 R. Girshick，“Detectron2”，https://github.com/facebookresearch/detectron2，2019年。[56] L. Ericsson、H. Gouk 和 T. M. Hospedales，“自监督模型的迁移效果如何？”，载于《IEEE/CVF 计算机视觉与模式识别会议论文集》，2021年，第5414–5423页。[57] X. Glorot 和 Y. Bengio，“理解深度前馈神经网络训练的难点”，载于第十三届国际人工智能与统计学会议论文集，AISTATS 2010，15，2010年，13，意大利撒丁岛基亚拉古纳度假村，5月——JMLR 会议论文集，Y. W. Teh 和 D. M. Titterington 编，第9卷。JMLR.org，2010年，第249–256页。 [在线]。网址：http://proceedings.mlr.press/v9/glorot10a.html

传记部分



Jidong Ge 于2007年获得南京大学计算机科学博士学位。他现任南京大学软件学院副教授。目前的研究兴趣包括机器学习、深度学习、图像处理、软件工程、自然语言处理和流程挖掘。其研究成果已发表在100多篇国际期刊和会议论文中，包括《IEEE 软件工程汇刊》（TSE）、《IEEE/ACM 汇刊》

网络（TNET）、IEEE 并行与分布式系统汇刊（TPDS）、IEEE 移动计算汇刊（TMC）、IEEE 服务计算汇刊（TSC）、ACM 数据知识发现汇刊（TKDD）、IEEE/ACM 音频、语音与语言处理汇刊（TASLP）、《自动化软件工程》（国际期刊）、《计算机网络》、《并行与分布式计算杂志》（JPDC）、未来一代计算机系统（FGCS）、系统与软件杂志（JSS）、信息科学、《网络与计算机应用杂志》（JNCA）、《软件杂志：演化与过程》（JSEP）、专家系统与应用（ESA）、AAAI、ICSE、ESEC/FSE、ASE、IWQoS 和 GlobeCom。



Yuxiang Liu 毕业于南京大学软件学院，获学士学位。目前，他正在南京大学软件学院攻读硕士学位。他的研究兴趣包括计算机视觉和深度学习。



桂杰（高级会员（2016年））现任东南大学网络空间安全学院教授。他于2004年在中国南京河海大学获得计算机科学理学学士学位，2007年在中国合肥中国科学院合肥物质科学研究院获得计算机应用技术理学硕士学位，并于2010年在中国合肥中国科学技术大学获得模式识别与智能系统博士学位。他是 Neurocomputing 的副主编、高级

IEEE 和 ACM 会员，同时也是中国计算机学会杰出会员。他在 IEEE 模式分析与机器智能汇刊、IEEE 神经网络与学习系统汇刊、IEEE 控制论汇刊、IEEE 图像处理汇刊、IEEE 电路与视频技术汇刊、IEEE 系统、人类与控制论汇刊、KDD、AAAI 和 ACM 多媒体等国际期刊和会议上发表了 60 多篇论文。他曾担任 NeurIPS 和 ICML 等多个会议的领域主席、高级程序委员会委员或程序委员会委员。他的研究兴趣包括机器学习、模式识别和图像处理。



Lanting Fang 是中国南京东南大学网络空间安全学院的讲师。她于2018年获东南大学博士学位。她目前的研究兴趣包括自然语言处理、信息抽取、社交媒体和情感分析。



Ming Lin 是 Amazon.com LCC 的高级应用科学家。他的研究兴趣包括深度学习的数学基础和统计机器学习，以及它们在深度学习加速、计算机视觉和移动人工智能中的应用。在加入亚马逊之前，他于2018年4月至2022年7月担任阿里巴巴集团达摩院（美国）的算法工程师。他于2015年9月至2018年4月在密歇根大学医学院担任研究调查员。他曾在网络空间安全学院担任博士后研究员

2014年7月至2015年9月，他在卡内基梅隆大学计算机科学系任职。他于2014年获得清华大学计算机科学博士学位。攻读博士学位期间，他曾于2013年12月至2014年7月在密歇根州立大学和卡内基梅隆大学担任访问学者。



James Tin-Yau Kwok（IEEE会士）于1996年获得香港科技大学计算机科学博士学位。他现任香港科技大学计算机科学与工程学系教授。目前的研究兴趣包括核方法、机器学习、模式识别和人工神经网络。他于2004年获得IEEE杰出论文奖，并获颁中国教育部自然科学二等奖

。他曾担任多个国际会议的程序委员会共同主席，并于2006年至2012年担任IEEE《神经网络与学习系统汇刊》副主编。他目前是Neurocomputing的副主编。



LiGuo Huang 现任美国德克萨斯州达拉斯南卫理公会大学（SMU）计算机科学与工程系副教授。她在南加州大学（USC）计算机科学系及系统与软件工程中心（CSSE）获得博士学位（2006年）和硕士学位。她目前的研究方向包括过程建模、仿真与改进、系统与软件质量保证、信息可信性、挖掘系统与软件工程代码库，

基于利益相关者/价值的软件工程以及软件度量。她的研究得到美国国家科学基金会、美国国防部（DoD）和美国国家安全局的支持。她积极参与发起了系统与软件工程基于利益相关者/价值的集成研究。



Bin Luo 是南京大学软件学院教授。他的主要研究兴趣包括机器学习、深度学习、图像处理、软件工程、自然语言处理和流程挖掘。他的研究成果已发表于100多篇国际期刊论文和会议论文集，包括 IEEE并行与分布式系统汇刊、IEEE移动计算汇刊、IEEE服务计算汇刊、IEEE/ACM音频、语音与语言处理汇刊、ACM知识发现与数据挖掘汇刊、ACM交互式智能系统汇刊、未来一代计算机系统、系统与软件杂志、信息科学、专家系统与应用、专家系统等。他负责领导南京大学应用软件工程研究所。