Emerging Properties in Self-Supervised Vision Transformers

Mathilde Caron[1,2] Hugo Touvron[1,3] Ishan Misra[1] Hervé Jegou[1]
Julien Mairal[2] Piotr Bojanowski[1] Armand Joulin[1]

[1] Facebook AI Research [2] Inria* [3] Sorbonne University

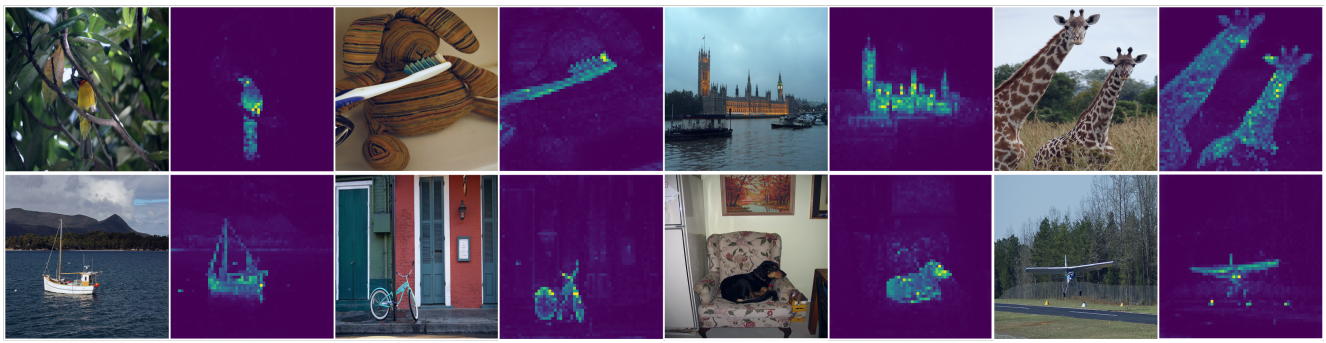


Figure 1: **Self-attention from a Vision Transformer with 8×8 patches trained with no supervision.** We look at the self-attention of the `[CLS]` token on the heads of the last layer. This token is not attached to any label nor supervision. These maps show that the model automatically learns class-specific features leading to unsupervised object segmentations.

Abstract

*In this paper, we question if self-supervised learning provides new properties to Vision Transformer (ViT) [19] that stand out compared to convolutional networks (convnets). Beyond the fact that adapting self-supervised methods to this architecture works particularly well, we make the following observations: first, self-supervised ViT features contain explicit information about the semantic segmentation of an image, which does not emerge as clearly with supervised ViTs, nor with convnets. Second, these features are also excellent k-NN classifiers, reaching 78.3% top-1 on ImageNet with a small ViT. Our study also underlines the importance of momentum encoder [33], multi-crop training [10], and the use of small patches with ViTs. We implement our findings into a simple self-supervised method, called DINO, which we interpret as a form of self-**di**stillation with **no** labels. We show the synergy between DINO and ViTs by achieving 80.1% top-1 on ImageNet in linear evaluation with ViT-Base.*

*Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.
Correspondence: mathilde@fb.com
Code: https://github.com/facebookresearch/dino

1. Introduction

Transformers [70] have recently emerged as an alternative to convolutional neural networks (convnets) for visual recognition [19, 69, 83]. Their adoption has been coupled with a training strategy inspired by natural language processing (NLP), that is, pretraining on large quantities of data and finetuning on the target dataset [18, 55]. The resulting Vision Transformers (ViT) [19] are competitive with convnets but, they have not yet delivered clear benefits over them: they are computationally more demanding, require more training data, and their features do not exhibit unique properties.

In this paper, we question whether the muted success of Transformers in vision can be explained by the use of supervision in their pretraining. Our motivation is that one of the main ingredients for the success of Transformers in NLP was the use of self-supervised pretraining, in the form of close procedure in BERT [18] or language modeling in GPT [55]. These self-supervised pretraining objectives use the words in a sentence to create pretext tasks that provide a richer learning signal than the supervised objective of predicting a single label per sentence. Similarly, in images, image-level supervision often reduces the rich visual information contained in an image to a single concept selected from a predefined set of a few thousand categories of objects [60].

While the self-supervised pretext tasks used in NLP are

自监督视觉Transformer中的新兴特性

玛蒂尔德·卡隆[1,2] 雨果·图夫龙[1,3] 伊山·米斯拉[1] 埃尔韦·热古[1] 朱利安·梅拉尔[2] 彼得·博亚诺夫斯基[1] 阿尔芒·儒兰[1]

1 Facebook人工智能研究院 [2] 法国国家信息与自动化研究所* [3] 索邦大学

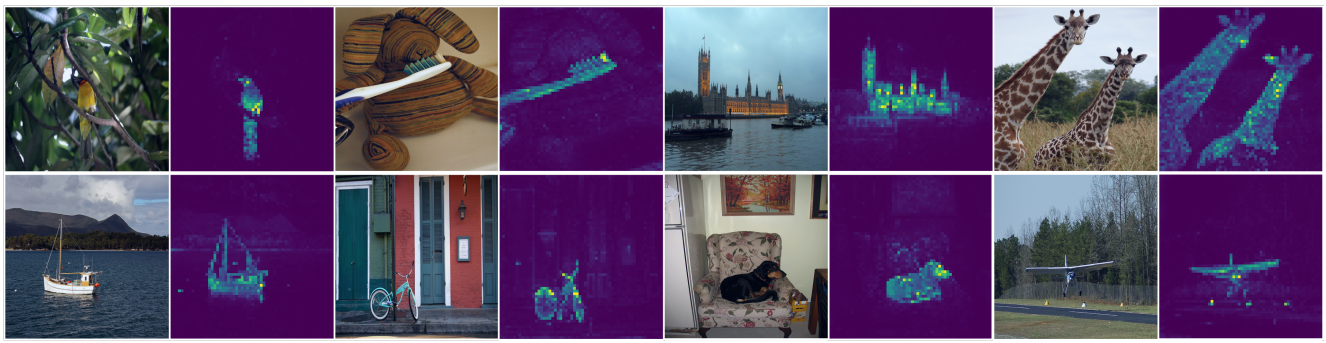


图1：来自一个无监督训练的Vision Transformer的自注意力机制，该模型使用8 × 8图像块。我们观察了最后一层头部中[CLS]标记的自注意力情况。该标记未与任何标签或监督信号相关联。这些注意力图表明，模型能自动学习类别特定的特征，从而实现无监督的对象分割。

摘要

*In this paper, we question if self-supervised learning provides new properties to Vision Transformer (ViT) [19] that stand out compared to convolutional networks (convnets). Beyond the fact that adapting self-supervised methods to this architecture works particularly well, we make the following observations: first, self-supervised ViT features contain explicit information about the semantic segmentation of an image, which does not emerge as clearly with supervised ViTs, nor with convnets. Second, these features are also excellent k-NN classifiers, reaching 78.3% top-1 on ImageNet with a small ViT. Our study also underlines the importance of momentum encoder [33], multi-crop training [10], and the use of small patches with ViTs. We implement our findings into a simple self-supervised method, called DINO, which we interpret as a form of self-**di**stillation with **no** labels. We show the synergy between DINO and ViTs by achieving 80.1% top-1 on ImageNet in linear evaluation with ViT-Base.*

*Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.
Correspondence: mathilde@fb.com
Code: https://github.com/facebookresearch/dino

1. 引言

Transformer模型[70]近期作为卷积神经网络(convnets)的替代方案在视觉识别领域崭露头角[19,69,83]。其应用策略借鉴了自然语言处理(NLP)的范式：即先在大规模数据上进行预训练，再针对目标数据集进行微调[18,55]。由此产生的视觉Transformer(ViT)[19]虽能与卷积网络分庭抗礼，但尚未展现出明显优势：它们计算需求更高、需要更多训练数据，且其特征并不具备独特属性。

本文探讨了Transformer在视觉领域表现平平的原因，是否可归咎于其预训练过程中采用的监督学习方式。我们的研究动机源于：Transformer在自然语言处理（NLP）领域大获成功的关键因素之一，正是采用了自监督预训练方法——如BERT[18]中的完形填空任务或GPT[55]中的语言建模。这些自监督预训练目标利用句子中的词汇构建前置任务，相比仅预测句子单一标签的监督目标，能提供更丰富的学习信号。同理，在图像领域，图像级监督往往将蕴含丰富视觉信息的图像简化为从预定义的数千种物体类别中选择单一概念[60]，{v*}这一处理方式可能削弱了模型的视觉理解能力。

虽然NLP中使用的自监督预训练任务是

text specific, many existing self-supervised methods have shown their potential on images with convnets [10, 12, 30, 33]. They typically share a similar structure but with different components designed to avoid trivial solutions (collapse) or to improve performance [16]. In this work, inspired from these methods, we study the impact of self-supervised pretraining on ViT features. Of particular interest, we have identified several interesting properties that do not emerge with supervised ViTs, nor with convnets:

- Self-supervised ViT features explicitly contain the scene layout and, in particular, object boundaries, as shown in Figure 1. This information is directly accessible in the self-attention modules of the last block.
- Self-supervised ViT features perform particularly well with a basic nearest neighbors classifier (k-NN) *without any finetuning, linear classifier nor data augmentation*, achieving 78.3% top-1 accuracy on ImageNet.

The emergence of segmentation masks seems to be a property shared across self-supervised methods. However, the good performance with k-NN only emerge when combining certain components such as momentum encoder [33] and multi-crop augmentation [10]. Another finding from our study is the importance of using smaller patches with ViTs to improve the quality of the resulting features.

Overall, our findings about the importance of these components lead us to design a simple self-supervised approach that can be interpreted as a form of knowledge **di**stillation [35] with **no** labels. The resulting framework, DINO, simplifies self-supervised training by directly predicting the output of a teacher network—built with a momentum encoder—by using a standard cross-entropy loss. Interestingly, our method can work with only a centering and sharpening of the teacher output to avoid collapse, while other popular components such as predictor [30], advanced normalization [10] or contrastive loss [33] add little benefits in terms of stability or performance. Of particular importance, our framework is flexible and works on both convnets and ViTs without the need to modify the architecture, nor adapt internal normalizations [58].

We further validate the synergy between DINO and ViT by outperforming previous self-supervised features on the ImageNet linear classification benchmark with 80.1% top-1 accuracy with a ViT-Base with small patches. We also confirm that DINO works with convnets by matching the state of the art with a ResNet-50 architecture. Finally, we discuss different scenarios to use DINO with ViTs in case of limited computation and memory capacity. In particular, training DINO with ViT takes just two 8-GPU servers over 3 days to achieve 76.1% on ImageNet linear benchmark, which outperforms self-supervised systems based on convnets of comparable sizes with significantly reduced compute requirements [10, 30].

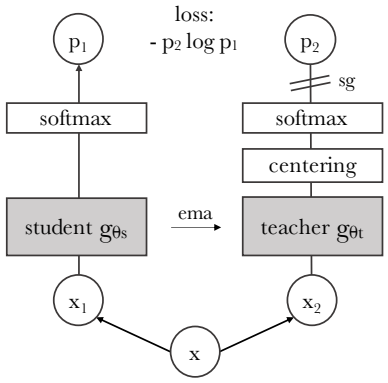


Figure 2: **Self-distillation with no labels.** We illustrate DINO in the case of one single pair of views (x_1, x_2) for simplicity. The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch. Each networks outputs a K dimensional feature that is normalized with a temperature softmax over the feature dimension. Their similarity is then measured with a cross-entropy loss. We apply a stop-gradient (sg) operator on the teacher to propagate gradients only through the student. The teacher parameters are updated with an exponential moving average (ema) of the student parameters.

2. Related work

Self-supervised learning. A large body of work on self-supervised learning focuses on discriminative approaches coined *instance classification* [12, 20, 33, 73], which considers each image a different class and trains the model by discriminating them up to data augmentations. However, explicitly learning a classifier to discriminate between all images [20] does not scale well with the number of images. Wu *et al*. [73] propose to use a noise contrastive estimator (NCE) [32] to compare instances instead of classifying them. A caveat of this approach is that it requires comparing features from a large number of images simultaneously. In practice, this requires large batches [12] or memory banks [33, 73]. Several variants allow automatic grouping of instances in the form of clustering [2, 8, 9, 36, 42, 74, 80, 85].

Recent works have shown that we can learn unsupervised features without discriminating between images. Of particular interest, Grill *et al*. [30] propose a metric-learning formulation called BYOL, where features are trained by matching them to representations obtained with a momentum encoder. Methods like BYOL work even without a momentum encoder, at the cost of a drop of performance [16, 30]. Several other works echo this direction, showing that one can match more elaborate representations [26, 27], train features matching them to a uniform distribution [6] or by using whitening [23, 81]. Our approach takes its inspiration from BYOL but operates with a different similarity matching loss

文本特定而言，许多现有的自监督方法已在卷积神经网络（convnet）处理的图像上展现出潜力[10, 12, 30, 33]。这些方法通常共享相似的结构，但通过设计不同组件来避免平凡解（崩溃）或提升性能[16]。本工作中，受这些方法启发，我们研究了自监督预训练对视觉Transformer（ViT）特征的影响。尤为值得注意的是，我们发现了若干在监督式ViT或卷积网络中均未显现的有趣特性：

- 自监督ViT特征明确包含了场景布局，尤其是物体边界，如图1所示。这些信息可直接从最后一个模块的自注意力机制中获取。

- 自监督ViT特征与基础的最近邻分类器(k-NN) *without any finetuning, linear classifier nor data augmentation* 配合表现尤为出色，在ImageNet上实现了78.3%的top-1准确率。

分割掩码的出现似乎是自监督方法共有的特性。然而，仅当结合动量编码器[33]和多裁剪增强[10]等特定组件时，k-NN才能展现出良好性能。我们研究的另一发现是，使用更小的补丁对ViTs提升特征质量的重要性。

总体而言，我们对这些组件重要性的发现促使我们设计了一种简单的自监督方法，可视为无需标签的知识蒸馏[35]形式。由此产生的框架DINO通过直接预测教师网络的输出——该网络由动量编码器构建——并采用标准交叉熵损失，简化了自监督训练。有趣的是，我们的方法仅需对教师输出进行中心化和锐化处理即可避免崩溃，而其他流行组件如预测器[30]、高级归一化[10]或对比损失[33]在稳定性或性能方面带来的提升甚微。尤为重要的是，我们的框架具有灵活性，可同时适用于卷积网络和视觉变换器(ViTs)，既无需修改架构，也不必调整内部归一化[58]。

我们进一步验证了DINO与ViT之间的协同效应，在使用小补丁的ViT-Base模型上，以80.1%的ImageNet线性分类基准top-1准确率超越了以往的自监督特征表现。同时，我们证实DINO同样适用于卷积网络，通过ResNet-50架构达到了与当前最优技术相当的水平。最后，我们探讨了在计算和内存资源有限的情况下，如何利用DINO与ViT结合的不同应用场景。特别是，使用ViT训练DINO仅需两台8-GPU服务器运行3天，即可在ImageNet线性基准上取得76{v*}1%的成绩，这一表现超越了基于同等规模卷积网络的自监督系统，同时显著降低了计算需求[10,30]。

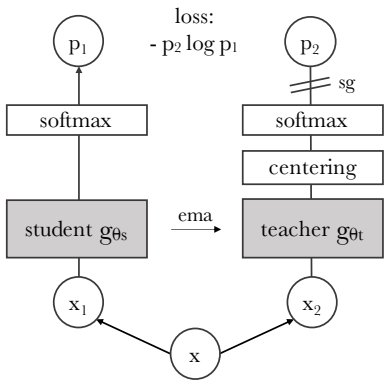


图2：无标签的自蒸馏。为简化起见，我们以单视图对（x_1，x_2）为例说明DINO方法。模型将输入图像的两种不同随机变换分别输入学生网络和教师网络。两网络结构相同但参数不同。教师网络的输出会以批次均值进行中心化处理。每个网络输出一个K维特征，该特征在特征维度上通过温度调节的softmax进行归一化。随后通过交叉熵损失衡量二者的相似性。我们在教师网络应用停止梯度算子(sg)，确保梯度仅通过学生网络传播。教师网络参数通过学生网络参数的指数移动平均(ema)进行更新。

2. 相关工作

自监督学习。大量关于自监督学习的研究聚焦于被称为*instance classification*的判别式方法[12, 20, 33, 73]，这类方法将每张图像视为不同类别，并通过数据增强后的图像判别来训练模型。然而，显式学习分类器以区分所有图像[20]的方法难以随图像数量扩展。Wu *et al* [73]提出采用噪声对比估计(NCE)[32]进行实例间对比而非分类。该方法的局限在于需要同时比较大量图像特征，实践中需依赖大批次训练[12]或记忆库机制[33, 73]。若干改进方案通过聚类形式实现了实例的自动分组[2, 8, 9, 36, 42, 74, 80, 85]。

近期研究表明，我们无需区分图像即可学习无监督特征。其中Grill *et al* [30]提出的BYOL度量学习框架尤为引人注目，该方法通过将特征与动量编码器获得的表征进行匹配来训练特征。即使没有动量编码器，类似BYOL的方法仍能工作，但会以性能下降为代价[16,30]。其他多项研究也呼应了这一方向，表明可以通过匹配更复杂的表征[26,27]、将特征训练至与均匀分布匹配[6]或采用白化技术[23,81]来实现目标。我们的方法受BYOL启发，但采用了不同的相似性匹配损失函数。

and uses the exact same architecture for the student and the teacher. That way, our work completes the interpretation initiated in BYOL of self-supervised learning as a form of Mean Teacher self-distillation [65] with no labels.

Self-training and knowledge distillation. Self-training aims at improving the quality of features by propagating a small initial set of annotations to a large set of unlabeled instances. This propagation can either be done with hard assignments of labels [41, 78, 79] or with a soft assignment [76]. When using soft labels, the approach is often referred to as knowledge distillation [7, 35] and has been primarily designed to train a small network to mimic the output of a larger network to compress models. Xie *et al*. [76] have shown that distillation could be used to propagate soft pseudo-labels to unlabelled data in a self-training pipeline, drawing an essential connection between self-training and knowledge distillation. Our work builds on this relation and extends knowledge distillation to the case where no labels are available. Previous works have also combined self-supervised learning and knowledge distillation [25, 63, 13, 47], enabling self-supervised model compression and performance gains. However, these works rely on a *pre-trained* fixed teacher while our teacher is dynamically built during training. This way, knowledge distillation, instead of being used as a post-processing step to self-supervised pre-training, is directly cast as a self-supervised objective. Finally, our work is also related to *codistillation* [1] where student and teacher have the same architecture and use distillation during training. However, the teacher in *codistillation* is also distilling from the student, while it is updated with an average of the student in our work.

3. Approach

3.1. SSL with Knowledge Distillation

The framework used for this work, DINO, shares the same overall structure as recent self-supervised approaches [10, 16, 12, 30, 33]. However, our method shares also similarities with knowledge distillation [35] and we present it under this angle. We illustrate DINO in Figure 2 and propose a pseudo-code implementation in Algorithm 1.

Knowledge distillation is a learning paradigm where we train a student network g_{θ_s} to match the output of a given teacher network g_{θ_t}, parameterized by θ_s and θ_t respectively. Given an input image x, both networks output probability distributions over K dimensions denoted by P_s and P_t. The probability P is obtained by normalizing the output of the network g with a softmax function. More precisely,

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s)}, \qquad (1)$$

with $\tau_s > 0$ a temperature parameter that controls the sharpness of the output distribution, and a similar formula holds for P_t with temperature τ_t. Given a fixed teacher network g_{θ_t}, we learn to match these distributions by minimizing the cross-entropy loss w.r.t. the parameters of the student network θ_s:

$$\min_{\theta_s} H(P_t(x), P_s(x)), \qquad (2)$$

where $H(a, b) = -a \log b$.

In the following, we detail how we adapt the problem in Eq. (2) to self-supervised learning. First, we construct different distorted views, or crops, of an image with multi-crop strategy [10]. More precisely, from a given image, we generate a set V of different views. This set contains two *global* views, x_1^g and x_2^g and several *local* views of smaller resolution. All crops are passed through the student while only the *global* views are passed through the teacher, therefore encouraging "local-to-global" correspondences. We minimize the loss:

$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x')). \qquad (3)$$

This loss is general and can be used on any number of views, even only 2. However, we follow the standard setting for multi-crop by using 2 global views at resolution 224^2 covering a large (for example greater than 50%) area of the original image, and several local views of resolution 96^2 covering only small areas (for example less than 50%) of the original image. We refer to this setting as the basic parametrization of DINO, unless mentioned otherwise.

Both networks share the same architecture g with different sets of parameters θ_s and θ_t. We learn the parameters θ_s by minimizing Eq. (3) with stochastic gradient descent.

Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

并且为学生和教师采用了完全相同的架构。这样一来，我们的工作完成了对自监督学习作为一种无标签的Mean Teacher自蒸馏[65]形式的解释，这一解释始于BYOL。

自训练与知识蒸馏。自训练旨在通过将少量初始标注信息传播至大量无标签实例，以提升特征质量。这种传播可通过硬标签分配[41, 78, 79]或软标签分配[76]实现。当采用软标签时，该方法常被称为知识蒸馏[7, 35]，其最初设计目的是训练小型网络模仿大型网络的输出以实现模型压缩。Xie *et al* [76]证明了蒸馏可用于在自训练流程中向无标签数据传播软伪标签，从而揭示了自训练与知识蒸馏之间的本质联系。我们的工作基于这一关联，将知识蒸馏扩展至完全无标签的场景。先前研究也结合了自监督学习与知识蒸馏[25, 63, 13, 47]，实现了自监督模型压缩与性能提升。但这些方法依赖*pre-trained*固定教师模型，而我们的教师模型在训练过程中动态构建。由此，知识蒸馏不再作为自监督预训练的后处理步骤，而是直接作为自监督目标。此外，我们的工作也与*codistilla- tion* [1]相关，其中师生网络结构相同并在训练中使用蒸馏。但*codistillation*中的教师模型同时从学生模型蒸馏知识，而我们的教师模型通过学生模型的参数平均进行更新。

3. 方法

3.1. 基于知识蒸馏的SSL

本工作采用的框架DINO，其整体结构与近期自监督方法[10, 16, 12, 30, 33]相同。然而，我们的方法也与知识蒸馏[35]有相似之处，并从这个角度进行阐述。图2展示了DINO的示意图，算法1则提供了伪代码实现。

知识蒸馏是一种学习范式，其中我们训练一个学生网络g_{θ_s}以匹配给定教师网络g_{θ_t}的输出，这两个网络分别由参数θ_s和θ_t参数化。给定输入图像x，两个网络都会输出在K维度上的概率分布，分别记为P_s和P_t。概率P是通过使用softmax函数对网络g的输出进行归一化得到的。更准确地说，

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K}\exp(g_{\theta_s}(x)^{(k)}/\tau_s)}, \qquad (1)$$

其中 $\tau_s > 0$ 是一个控制温度的参数

算法1 DINO PyTorch伪代码（不含多裁剪）

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

输出分布的锐度，对于温度τ_t下的P_t也有类似的公式成立。给定一个固定的教师网络g_{θ_t}，我们通过最小化学生网络θ_s参数相对于交叉熵损失来学习匹配这些分布：

$$\min_{\theta_s} H(P_t(x), P_s(x)), \qquad (2)$$

其中 $H(a,b) = -a$ 对数 b。

以下，我们将详细阐述如何将方程(2)中的问题适配到自监督学习中。首先，我们采用多裁剪策略[10]为图像构建不同的扭曲视图或裁剪区域。具体而言，对于给定图像，我们生成一组不同视图V。该集合包含两个*global*视图——x_1^g和x_2^g，以及若干分辨率较小的*local*视图。所有裁剪区域都会通过学生网络处理，而只有*global*视图会通过教师网络传递，从而促进"局部到全局"的对应关系。我们最小化以下损失函数：

$$\min_{\theta_s} \sum_{x\in\{x_1^g, x_2^g\}} \sum_{\substack{x'\in V \\ x'\neq x}} H(P_t(x), P_s(x')). \qquad (3)$$

这一损失函数具有通用性，可适用于任意数量的视图，即便是仅有两个视图的情况。不过，我们遵循多裁剪的标准设置：使用两个224^2分辨率的全局视图覆盖原始图像的大部分区域（例如大于50%的面积），以及若干个96^2分辨率的局部视图仅覆盖原始图像的小部分区域（例如小于50%的面积）。除非另有说明，我们将此设置称为DINO的基本参数化配置。

两个网络共享相同的架构g，但参数集θ_s和θ_t不同。我们通过随机梯度下降最小化方程(3)来学习参数θ_s。

Table 1: **Networks configuration.** "Blocks" is the number of Transformer blocks, "dim" is channel dimension and "heads" is the number of heads in multi-head attention. "# tokens" is the length of the token sequence when considering 224^2 resolution inputs, "# params" is the total number of parameters (without counting the projection head) and "im/s" is the inference time on a NVIDIA V100 GPU with 128 samples per forward.

model	blocks	dim	heads	#tokens	#params	im/s
ResNet-50	–	2048	–	–	23M	1237
ViT-S/16	12	384	6	197	21M	1007
ViT-S/8	12	384	6	785	21M	180
ViT-B/16	12	768	12	197	85M	312
ViT-B/8	12	768	12	785	85M	63

Teacher network. Unlike knowledge distillation, we do not have a teacher g_{θ_t} given *a priori* and hence, we build it from past iterations of the student network. We study different update rules for the teacher in Section 5.2 and show that freezing the teacher network over an epoch works surprisingly well in our framework, while copying the student weight for the teacher fails to converge. Of particular interest, using an exponential moving average (EMA) on the student weights, i.e., a momentum encoder [33], is particularly well suited for our framework. The update rule is $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$, with λ following a cosine schedule from 0.996 to 1 during training [30]. Originally the momentum encoder has been introduced as a substitute for a queue in contrastive learning [33]. However, in our framework, its role differs since we do not have a queue nor a contrastive loss, and may be closer to the role of the mean teacher used in self-training [65]. Indeed, we observe that this teacher performs a form of model ensembling similar to Polyak-Ruppert averaging with an exponential decay [51, 59]. Using Polyak-Ruppert averaging for model ensembling is a standard practice to improve the performance of a model [38]. We observe that this teacher has better performance than the student throughout the training, and hence, guides the training of the student by providing target features of higher quality. This dynamic was not observed in previous works [30, 58].

Network architecture. The neural network g is composed of a backbone f (ViT [19] or ResNet [34]), and of a projection head h: $g = h \circ f$. The features used in downstream tasks are the backbone f output. The projection head consists of a 3-layer multi-layer perceptron (MLP) with hidden dimension 2048 followed by ℓ_2 normalization and a weight normalized fully connected layer [61] with K dimensions, which is similar to the design from SwAV [10]. We have tested other projection heads and this particular design appears to work best for DINO (Appendix C). We do not use a predictor [30, 16], resulting in the exact same architecture in both student and teacher networks. Of particular interest, we note that unlike standard convnets, ViT architectures do not use batch normalizations (BN) by default. Therefore, when applying DINO to ViT we do not use any BN also in the projection heads, making the system *entirely BN-free*.

Avoiding collapse. Several self-supervised methods differ by the operation used to avoid collapse, either through contrastive loss [73], clustering constraints [8, 10], predictor [30] or batch normalizations [30, 58]. While our framework can be stabilized with multiple normalizations [10], it can also work with only a centering and sharpening of the momentum teacher outputs to avoid model collapse. As shown experimentally in Section 5.3, centering prevents one dimension to dominate but encourages collapse to the uniform distribution, while the sharpening has the opposite effect. Applying both operations balances their effects which is sufficient to avoid collapse in presence of a momentum teacher. Choosing this method to avoid collapse trades stability for less dependence over the batch: the centering operation only depends on first-order batch statistics and can be interpreted as adding a bias term c to the teacher: $g_t(x) \leftarrow g_t(x) + c$. The center c is updated with an exponential moving average, which allows the approach to work well across different batch sizes as shown in Section 5.5:

$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i), \tag{4}$$

where $m > 0$ is a rate parameter and B is the batch size. Output sharpening is obtained by using a low value for the temperature τ_t in the teacher softmax normalization.

3.2. Implementation and evaluation protocols

In this section, we provide the implementation details to train with DINO and present the evaluation protocols used in our experiments.

Vision Transformer. We briefly describe the mechanism of the Vision Transformer (ViT) [19, 70] and refer to Vaswani *et al*. [70] for details about Transformers and to Dosovitskiy *et al*. [19] for its adaptation to images. We follow the implementation used in `DeiT` [69]. We summarize the configuration of the different networks used in this paper in Table 1. The ViT architecture takes as input a grid of non-overlapping contiguous image patches of resolution $N \times N$. In this paper we typically use $N = 16$ ("/16") or $N = 8$ ("/8"). The patches are then passed through a linear layer to form a set of embeddings. We add an extra learnable token to the sequence [18, 19]. The role of this token is to aggregate information from the entire sequence and we attach the projection head h at its output. We refer to this token as the class token `[CLS]` for consistency with

表1：网络配置。“Blocks”表示Transformer块的数量，“dim”为通道维度，“heads”是多头注意力中的头数。“# tokens”是在考虑224^2分辨率输入时的令牌序列长度，“# params”是参数总数（不计算投影头），“im/s”是在NVIDIA V100 GPU上每次前向传播处理128个样本的推理时间。

model	blocks	dim	heads	#tokens	#params	im/s
ResNet-50	–	2048	–	–	23M	1237
ViT-S/16	12	384	6	197	21M	1007
ViT-S/8	12	384	6	785	21M	180
ViT-B/16	12	768	12	197	85M	312
ViT-B/8	12	768	12	785	85M	63

教师网络。与知识蒸馏不同，我们并未预先给定教师网络g_{θ_t}基于*a priori*，而是从学生网络的历史迭代中构建它。在第5.2节中，我们探讨了教师网络的不同更新规则，结果表明：在我们的框架中，冻结教师网络一个周期的做法表现惊人地好，而直接复制学生权重会导致训练无法收敛。特别值得注意的是，对学生权重采用指数移动平均（EMA）——即动量编码器[33]——与我们的框架尤为契合。其更新规则为$\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$，，其中$\lambda$遵循余弦调度从0.996逐步增至1[30]。动量编码器最初是作为对比学习中队列机制的替代方案提出的[33]，但在我们的框架中，由于既无队列结构也无对比损失，其作用更接近于自训练中使用的均值教师[65]。实际上，我们发现该教师网络执行了一种类似带指数衰减的Polyak-Ruppert平均的模型集成方法[51,59]。采用Polyak-Ruppert平均进行模型集成是提升模型性能的常规手段[38]。我们观察到，在整个训练过程中，该教师网络的性能始终优于学生网络，因此能通过提供更高质量的目标特征来指导学生网络的训练。这一动态特性在先前研究中未被观察到[30,58]。

网络架构。该神经网络g由一个主干网络f (ViT [19]或ResNet [34])以及一个投影头h组成：$g = h \circ f$。在下游任务中使用的特征是主干网络f的输出。投影头包含一个隐藏层维度为2048的三层多层感知机（MLP），随后进行ℓ_2归一化处理，再接一个权重归一化的全连接层[61]，其维度为K，这一设计与SwAV [10]类似。我们测试了其他投影头设计，发现此特定架构在DINO中表现最佳（附录C）。我们未采用预测器[30,16]，因此架构与...（保持公式记号{v*}不变）

学生和教师网络。特别值得注意的是，与标准卷积网络不同，ViT架构默认不使用批量归一化（BN）。因此，在将DINO应用于ViT时，我们同样不在投影头中使用任何BN，使得系统{v*}。

避免崩溃。多种自监督方法通过不同的操作来防止模型崩溃，这些操作包括对比损失[73]、聚类约束[8,10]、预测器[30]或批量归一化[30,58]。虽然我们的框架可以通过多重归一化[10]实现稳定，但仅需对动量教师输出进行中心化和锐化处理，同样能避免模型崩溃。如第5.3节实验所示，中心化能防止单一维度主导，但会促使输出趋于均匀分布；而锐化则产生相反效果。同时应用这两种操作可平衡其效应，在动量教师存在时足以避免崩溃。选择此方法以稳定性换取对批次的较低依赖：中心化操作仅依赖一阶批次统计量，可视为向教师添加偏置项c：$g_t(x) \leftarrow g_t(x) + c$。中心值$c$通过指数移动平均更新，使得该方法能适应不同批次规模（如第5.5节所示）：

$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i), \tag{4}$$

其中m >0为速率参数，B为批次大小。通过在教师softmax归一化中使用较低的τ_t温度值，可获得输出锐化效果。

3.2. 实施与评估协议

在本节中，我们将详细介绍使用DINO进行训练的实施方案，并阐述实验中采用的评估协议。

视觉Transformer。我们简要描述视觉Transformer（ViT）[19,70]的机制，关于Transformer的细节请参考Vaswani *et al*。[70]，其图像适应方法请参阅Dosovitskiy *et al*。[19]。我们遵循DeiT[69]中的实现方式。本文所用不同网络的配置总结于表1。ViT架构的输入是一个不重叠的连续图像块网格，分辨率为$N \times N$。本文通常采用N =16（“/16”）或N =8（“/8”）。这些图像块随后通过线性层形成一组嵌入向量。我们在序列中添加了一个额外的可学习令牌[18,19]，该令牌的作用是从整个序列中聚合信息，并在其输出端连接投影头h。为保持一致性，我们将此令牌称为类别令牌[CLS]。

previous works[18, 19, 69], even though it is not attached to any label nor supervision in our case. The set of patch tokens and `[CLS]` token are fed to a standard Transformer network with a "pre-norm" layer normalization [11, 39]. The Transformer is a sequence of self-attention and feed-forward layers, paralleled with skip connections. The self-attention layers update the token representations by looking at the other token representations with an attention mechanism [4].

Implementation details. We pretrain the models on the ImageNet dataset [60] without labels. We train with the adamw optimizer [44] and a batch size of 1024, distributed over 16 GPUs when using ViT-S/16. The learning rate is linearly ramped up during the first 10 epochs to its base value determined with the following linear scaling rule [29]: $lr = 0.0005 * \text{batchsize}/256$. After this warmup, we decay the learning rate with a cosine schedule [43]. The weight decay also follows a cosine schedule from 0.04 to 0.4. The temperature τ_s is set to 0.1 while we use a linear warm-up for τ_t from 0.04 to 0.07 during the first 30 epochs. We follow the data augmentations of BYOL [30] (color jittering, Gaussian blur and solarization) and multi-crop [10] with a bicubic interpolation to adapt the position embeddings to the scales [19, 69]. The code and models to reproduce our results is publicly available.

Evaluation protocols. Standard protocols for self-supervised learning are to either learn a linear classifier on frozen features [82, 33] or to finetune the features on downstream tasks. For linear evaluations, we apply random resize crops and horizontal flips augmentation during training, and report accuracy on a central crop. For finetuning evaluations, we initialize networks with the pretrained weights and adapt them during training. However, both evaluations are sensitive to hyperparameters, and we observe a large variance in accuracy between runs when varying the learning rate for example. We thus also evaluate the quality of features with a simple weighted nearest neighbor classifier (k-NN) as in [73]. We freeze the pretrain model to compute and store the features of the training data of the downstream task. The nearest neighbor classifier then matches the feature of an image to the k nearest stored features that votes for the label. We sweep over different number of nearest neighbors and find that 20 NN is consistently working the best for most of our runs. This evaluation protocol does not require any other hyperparameter tuning, nor data augmentation and can be run with only one pass over the downstream dataset, greatly simplifying the feature evaluation.

Table 2: **Linear and k-NN classification on ImageNet.** We report top-1 accuracy for linear and k-NN evaluations on the validation set of ImageNet for different self-supervised methods. We focus on ResNet-50 and ViT-small architectures, but also report the best results obtained across architectures. * are run by us. We run the k-NN evaluation for models with official released weights. The throughput (im/s) is calculated on a NVIDIA V100 GPU with 128 samples per forward. Parameters (M) are of the feature extractor.

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	**75.3**	65.7
DINO	RN50	23	1237	**75.3**	**67.5**
Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [30]	ViT-S	21	1007	71.4	66.6
MoCov2* [15]	ViT-S	21	1007	72.7	64.4
SwAV* [10]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	**77.0**	**74.5**
Comparison across architectures					
SCLR [12]	RN50w4	375	117	76.8	69.3
SwAV [10]	RN50w2	93	384	77.3	67.3
BYOL [30]	RN50w2	93	384	77.4	–
DINO	ViT-B/16	85	312	78.2	76.1
SwAV [10]	RN50w5	586	76	78.5	67.1
BYOL [30]	RN50w4	375	117	78.6	–
BYOL [30]	RN200w2	250	123	79.6	73.9
DINO	ViT-S/8	21	180	79.7	**78.3**
SCLRv2 [13]	RN152w3+SK	794	46	79.8	73.1
DINO	ViT-B/8	85	63	**80.1**	77.4

4. Main Results

We first validate the DINO framework used in this study with the standard self-supervised benchmark on ImageNet. We then study the properties of the resulting features for retrieval, object discovery and transfer-learning.

4.1. Comparing with SSL frameworks on ImageNet

We consider two different settings: comparison with the same architecture and across architectures.

Comparing with the same architecture. In top panel of Table 2, we compare DINO with other self-supervised methods with the same architecture, either a ResNet-50 [34] or a ViT-small (which follows the design of DeiT-S [69]). The choice of ViT-S is motivated by its similarity with ResNet-50 along several axes: number of parameters (21M vs 23M),

先前的工作[18, 19, 69]中，尽管在我们的案例中它并未附加任何标签或监督。补丁标记集和[CLS]标记被输入到一个采用“预归一化”层标准化[11, 39]的标准Transformer网络中。该Transformer由一系列自注意力层和前馈层组成，并伴有跳跃连接并行。自注意力层通过注意力机制[4]查看其他标记表示来更新标记表示。

实现细节。我们在无标签的ImageNet数据集[60]上对模型进行预训练。采用adamw优化器[44]，批处理大小为1024，当使用ViT-S/16架构时，分布在16块GPU上进行训练。学习率在前10个epoch内线性上升至其基础值，该值通过以下线性缩放规则[29]确定：$lr = 0.0005 * \text{batchsize}/256$。预热阶段结束后，采用余弦调度[43]衰减学习率。权重衰减同样遵循余弦调度，从0.04降至0.4。温度参数τ_s设为0.1，同时在前30个epoch内对τ_t进行线性预热，从0.04升至0.07。数据增强方面遵循BYOL[30]的方法（色彩抖动、高斯模糊和曝光处理）及多裁剪策略[10]，并通过双三次插值调整位置嵌入以适应不同尺度[19,69]。用于复现我们结果的代码与模型已公开提供。

评估协议。自监督学习的标准协议通常包括在冻结特征上训练线性分类器[82, 33]或对下游任务进行特征微调。在线性评估中，我们在训练时应用随机尺寸裁剪和水平翻转增强，并报告中心裁剪区域的准确率。对于微调评估，我们用预训练权重初始化网络，并在训练过程中进行调整。然而，这两种评估都对超参数敏感，例如我们观察到学习率变化时运行间准确率存在较大波动。因此，我们还采用如[73]中所述的加权最近邻分类器(k-NN)来评估特征质量：冻结预训练模型以计算并存储下游任务训练数据的特征，该分类器将图像特征与存储的k个最近邻特征进行匹配，通过投票决定标签。我们测试了不同数量的最近邻，发现20 NN在大多数实验中都表现最佳。该评估方案无需其他超参数调优或数据增强，仅需单次遍历下游数据集即可完成，极大简化了特征评估流程。

表2：ImageNet上的线性和k-NN分类。我们报告了不同自监督方法在ImageNet验证集上进行的线性和k-NN评估的top-1准确率。研究主要聚焦ResNet-50和ViT-small架构，同时也列出了跨架构取得的最佳结果。*由我们亲自运行。对于官方发布权重的模型，我们执行了k-NN评估。吞吐量（im/s）基于NVIDIA V100 GPU计算，每次前向传播处理128个样本。参数数量（M）指特征提取器的规模。

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	**75.3**	65.7
DINO	RN50	23	1237	**75.3**	**67.5**
Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [30]	ViT-S	21	1007	71.4	66.6
MoCov2* [15]	ViT-S	21	1007	72.7	64.4
SwAV* [10]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	**77.0**	**74.5**
Comparison across architectures					
SCLR [12]	RN50w4	375	117	76.8	69.3
SwAV [10]	RN50w2	93	384	77.3	67.3
BYOL [30]	RN50w2	93	384	77.4	–
DINO	ViT-B/16	85	312	78.2	76.1
SwAV [10]	RN50w5	586	76	78.5	67.1
BYOL [30]	RN50w4	375	117	78.6	–
BYOL [30]	RN200w2	250	123	79.6	73.9
DINO	ViT-S/8	21	180	79.7	**78.3**
SCLRv2 [13]	RN152w3+SK	794	46	79.8	73.1
DINO	ViT-B/8	85	63	**80.1**	77.4

4. 主要结果

我们首先通过ImageNet上的标准自监督基准验证了本研究中使用的DINO框架。随后，我们探究了所得特征在检索、对象发现及迁移学习中的特性。

4.1. 与ImageNet上的SSL框架对比

我们考虑两种不同的设置：同架构内的比较与跨架构的比较。

与相同架构进行比较。在表2的上半部分，我们将DINO与其他采用相同架构的自监督方法进行对比，这些架构要么是ResNet-50[34]，要么是ViT-small（遵循DeiT-S[69]的设计）。选择ViT-S的动机在于其与ResNet-50在多个维度上的相似性：参数量（21M vs 23M），

Table 3: **Image retrieval.** We compare the performance in retrieval of off-the-shelf features pretrained with supervision or with DINO on ImageNet and Google Landmarks v2 (GLDv2) dataset. We report mAP on revisited Oxford and Paris. Pretraining with DINO on a landmark dataset performs particularly well. For reference, we also report the best retrieval method with off-the-shelf features [57].

			$\mathcal{R}$Ox		$\mathcal{R}$Par	
Pretrain	Arch.	Pretrain	M	H	M	H
Sup. [57]	RN101+R-MAC	ImNet	49.8	18.5	74.0	**52.1**
Sup.	ViT-S/16	ImNet	33.5	8.9	63.0	37.2
DINO	ResNet-50	ImNet	35.4	11.1	55.9	27.5
DINO	ViT-S/16	ImNet	41.8	13.7	63.1	34.4
DINO	ViT-S/16	GLDv2	**51.5**	**24.3**	**75.3**	51.6

throughput (1237/sec VS 1007 im/sec) and supervised performance on ImageNet with the training procedure of [69] (79.3% VS 79.8%). We explore variants of ViT-S in Appendix D. First, we observe that DINO performs on par with the state of the art on ResNet-50, validating that DINO works in the standard setting. When we switch to a ViT architecture, DINO outperforms BYOL, MoCov2 and SwAV by +3.5% with linear classification and by +7.9% with k-NN evaluation. More surprisingly, the performance with a simple k-NN classifier is almost on par with a linear classifier (74.5% versus 77.0%). This property emerges only when using DINO with ViT architectures, and does not appear with other existing self-supervised methods nor with a ResNet-50.

Comparing across architectures. On the bottom panel of Table 2, we compare the best performance obtained across architectures. The interest of this setting is not to compare methods directly, but to evaluate the limits of a ViT trained with DINO when moving to larger architectures. While training a larger ViT with DINO improves the performance, reducing the size of the patches ("/8" variants) has a bigger impact on the performance. While reducing the patch size do not add parameters, it still leads to a significant reduction of running time, and larger memory usage. Nonetheless, a base ViT with 8×8 patches trained with DINO achieves 80.1% top-1 in linear classification and 77.4% with a k-NN classifier with $10\times$ less parameters and $1.4\times$ faster run time than previous state of the art [13].

4.2. Properties of ViT trained with SSL

We evaluate properties of the DINO features in terms of nearest neighbor search, retaining information about object location and transferability to downstream tasks.

Table 4: **Copy detection.** We report the mAP performance in copy detection on Copydays "strong" subset [21]. For reference, we also report the performance of the multigrain model [5], trained specifically for particular object retrieval.

Method	Arch.	Dim.	Resolution	mAP
Multigrain [5]	ResNet-50	2048	224^2	75.1
Multigrain [5]	ResNet-50	2048	largest side 800	82.5
Supervised [69]	ViT-B/16	1536	224^2	76.4
DINO	ViT-B/16	1536	224^2	81.7
DINO	ViT-B/8	1536	320^2	**85.5**

4.2.1 Nearest neighbor retrieval with DINO ViT

The results on ImageNet classification have exposed the potential of our features for tasks relying on nearest neighbor retrieval. In this set of experiments, we further consolidate this finding on landmark retrieval and copy detection tasks.

Image Retrieval. We consider the revisited [53] Oxford and Paris image retrieval datasets [50]. They contain 3 different splits of gradual difficulty with query/database pairs. We report the Mean Average Precision (mAP) for the Medium (M) and Hard (H) splits. In Table 3, we compare the performance of different *off-the-shelf* features obtained with either supervised or DINO training. We freeze the features and directly apply k-NN for retrieval. We observe that DINO features outperform those trained on ImageNet with labels.

An advantage of SSL approaches is that they can be trained on any dataset, without requiring any form of annotations. We train DINO on the 1.2M clean set from Google Landmarks v2 (GLDv2) [72], a dataset of landmarks designed for retrieval purposes. DINO ViT features trained on GLDv2 are remarkably good, outperforming previously published methods based on off-the-shelf descriptors [68, 57].

Copy detection. We also evaluate the performance of ViTs trained with DINO on a copy detection task. We report the mean average precision on the "strong" subset of the INRIA Copydays dataset [21]. The task is to recognize images that have been distorted by blur, insertions, print and scan, etc. Following prior work [5], we add 10k distractor images randomly sampled from the YFCC100M dataset [66]. We perform copy detection directly with cosine similarity on the features obtained from our pretrained network. The features are obtained as the concatenation of the output `[CLS]` token and of the GeM pooled [54] output patch tokens. This results in a 1536d descriptor for ViT-B. Following [5], we apply whitening on the features. We learn this transformation on an extra 20K random images from YFCC100M, distincts from the distractors. Table 4 shows that ViT trained with DINO is very competitive on copy detection.

表3：图像检索。我们比较了在ImageNet和Google Landmarks v2（GLDv2）数据集上通过监督学习或DINO预训练的现成特征在检索任务中的性能表现。数据展示了在Revisited Oxford和Paris数据集上的mAP值。特别值得注意的是，在标志性数据集上使用DINO进行预训练表现尤为出色。作为参考，我们还列出了使用现成特征的最佳检索方法[57]。

			$\mathcal{R}$Ox		$\mathcal{R}$Par	
Pretrain	Arch.	Pretrain	M	H	M	H
Sup. [57]	RN101+R-MAC	ImNet	49.8	18.5	74.0	**52.1**
Sup.	ViT-S/16	ImNet	33.5	8.9	63.0	37.2
DINO	ResNet-50	ImNet	35.4	11.1	55.9	27.5
DINO	ViT-S/16	ImNet	41.8	13.7	63.1	34.4
DINO	ViT-S/16	GLDv2	**51.5**	**24.3**	**75.3**	51.6

吞吐量（1237次/秒 VS 1007次/秒）以及在ImageNet上采用[69]训练流程的监督性能（79.3% VS 79.8%）。我们在附录D中探索了ViT-S的变体。首先，我们观察到DINO与ResNet-50上的现有技术表现相当，验证了DINO在标准设置下的有效性。当转向ViT架构时，DINO在线性分类上以+3.5%的优势超越BYOL、MoCov2和SwAV，在k-NN评估中以+7.9%的优势领先。更令人惊讶的是，使用简单的k-NN分类器时，其性能几乎与线性分类器持平（74.5%对比77.0%）。这一特性仅在将DINO与ViT架构结合使用时出现，其他现有自监督方法或ResNet-50均未展现此特性。

跨架构比较。在表2的下半部分，我们对比了不同架构中获得的最佳性能。这一设置的关注点并非直接比较方法，而是评估当转向更大架构时，使用DINO训练的ViT的极限。虽然用DINO训练更大的ViT能提升性能，但减小补丁尺寸（“/8”变体）对性能的影响更为显著。尽管缩小补丁尺寸不会增加参数数量，却会显著减少运行时间，同时增加内存占用。尽管如此，一个基础ViT采用8×8补丁尺寸并通过DINO训练，在线性分类中达到了80.1%的top-1准确率，使用k-NN分类器时为77.4%，且相比之前的最先进技术[13]，参数减少了10×倍，运行时间快了1.4×倍。

4.2. 基于自监督学习训练的ViT特性

我们通过最近邻搜索评估DINO特征的属性，保留关于物体位置的信息以及向下游任务的可迁移性。

表4：复制检测。我们报告了在Copydays“强”子集[21]上进行复制检测的mAP性能。作为参考，我们还报告了专为特定对象检索训练的多粒度模型[5]的性能。

Method	Arch.	Dim.	Resolution	mAP
Multigrain [5]	ResNet-50	2048	224^2	75.1
Multigrain [5]	ResNet-50	2048	largest side 800	82.5
Supervised [69]	ViT-B/16	1536	224^2	76.4
DINO	ViT-B/16	1536	224^2	81.7
DINO	ViT-B/8	1536	320^2	**85.5**

4.2.1 使用DINO ViT进行最近邻检索

ImageNet分类任务的结果揭示了我们的特征在依赖最近邻检索的任务中的潜力。在这一系列实验中，我们进一步在地标检索和复制检测任务上巩固了这一发现。

图像检索。我们考量了经过修订的[53]牛津与巴黎图像检索数据集[50]。这些数据集包含3种难度递增的分割方式，每种均提供查询/数据库配对。我们报告了中等(M)和困难(H)分割下的平均精度均值(mAP)。在表3中，我们对比了通过监督学习或DINO训练获得的不同*off-the-shelf*特征的性能表现。我们固定这些特征并直接采用k-NN进行检索。实验观察到，DINO特征的表现优于基于ImageNet标签训练所得特征。

SSL方法的一个优势在于，它们可以在任何数据集上进行训练，无需任何形式的标注。我们在Google Landmarks v2（GLDv2）[72]的120万张精选图像上训练了DINO，这是一个专为检索任务设计的地标数据集。基于GLDv2训练的DINO ViT特征表现出色，超越了此前发表的基于现成描述符的方法[68, 57]。

复制检测。我们还评估了使用DINO训练的ViT在复制检测任务上的表现。我们在INRIA Copydays数据集的“强”子集[21]上报告了平均精度均值。该任务旨在识别经过模糊、插入、打印和扫描等扭曲处理的图像。遵循先前工作[5]，我们从YFCC100M数据集[66]中随机抽取10k张干扰图像加入测试集。我们直接利用预训练网络提取的特征，通过余弦相似度进行复制检测。特征由输出[CLS]标记与经过GeM池化[54]处理的输出补丁标记拼接而成，最终为ViT-B生成1536维描述符。按照[5]的方法，我们对特征应用白化处理，这一变换通过在YFCC100M中额外选取的20K张随机图像（与干扰图像不同）上进行学习。表4显示，经DINO训练的ViT在复制检测任务上极具竞争力。

Table 5: **DAVIS 2017 Video object segmentation.** We evaluate the quality of frozen features on video instance tracking. We report mean region similarity $\mathcal{J}_m$ and mean contour-based accuracy $\mathcal{F}_m$. We compare with existing self-supervised methods and a supervised ViT-S/8 trained on ImageNet. Image resolution is 480p.

Method	Data	Arch.	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
Supervised					
ImageNet	INet	ViT-S/8	66.0	63.9	68.1
STM [48]	I/D/Y	RN50	81.8	79.2	84.3
Self-supervised					
CT [71]	VLOG	RN50	48.7	46.4	50.0
MAST [40]	YT-VOS	RN18	65.5	63.3	67.6
STC [37]	Kinetics	RN18	67.6	64.8	70.2
DINO	INet	ViT-S/16	61.8	60.2	63.4
DINO	INet	ViT-B/16	62.3	60.7	63.9
DINO	INet	ViT-S/8	**69.9**	**66.6**	**73.1**
DINO	INet	ViT-B/8	**71.4**	**67.9**	**74.9**

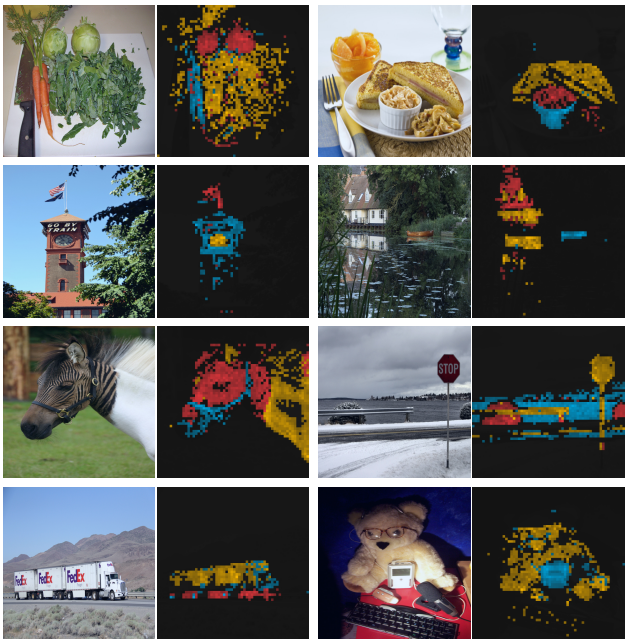


Figure 3: **Attention maps from multiple heads.** We consider the heads from the last layer of a ViT-S/8 trained with DINO and display the self-attention for `[CLS]` token query. Different heads, materialized by different colors, focus on different locations that represents different objects or parts (more examples in Appendix).

4.2.2 Discovering the semantic layout of scenes

As shown qualitatively in Figure 1, our self-attention maps contain information about the segmentation of an image. In this study, we measure this property on a standard benchmark as well as by directly probing the quality of masks generated from these attention maps.

Video instance segmentation. In Tab. 5, we evaluate the output patch tokens on the DAVIS-2017 video instance segmentation benchmark [52]. We follow the experimental protocol in Jabri *et al*. [37] and segment scenes with a nearest-neighbor between consecutive frames; we thus do not train any model on top of the features, nor finetune any weights for the task. We observe in Tab. 5 that even though our training objective nor our architecture are designed for dense tasks, the performance is competitive on this benchmark. Since the network is not finetuned, the output of the model must have retained some spatial information. Finally, for this dense recognition task, the variants with small patches ("/8") perform much better (+9.1% $(\mathcal{J}\&\mathcal{F})_m$ for ViT-B).

Probing the self-attention map. In Fig. 3, we show that different heads can attend to different semantic regions of an image, even when they are occluded (the bushes on the third row) or small (the flag on the second row). Visualizations are obtained with 480p images, resulting in sequences of 3601 tokens for ViT-S/8. In Fig. 4, we show that a supervised ViT does not attend well to objects in presence of clutter both qualitatively and quantitatively. We report the Jaccard similarity between the ground truth and segmentation masks obtained by thresholding the self-attention map to keep 60% of the mass. Note that the self-attention maps are smooth and not optimized to produce a mask. Nonetheless, we see a clear difference between the supervised or DINO models with a significant gap in terms of Jaccard similarities. Note that self-supervised convnets also contain information about segmentations but it requires dedicated methods to extract it from their weights [31].

4.2.3 Transfer learning on downstream tasks

In Tab. 6, we evaluate the quality of the features pretrained with DINO on different downstream tasks. We compare with features from the same architectures trained with supervision on ImageNet. We follow the protocol used in Touvron *et al*. [69] and finetune the features on each downstream task. We observe that for ViT architectures, self-supervised pretraining transfers better than features trained with supervision, which is consistent with observations made on convolutional networks [10, 33, 62]. Finally, self-supervised pretraining greatly improves results on ImageNet (+1-2%).

5. Ablation Study of DINO

In this section, we empirically study DINO applied to ViT. The model considered for this entire study is ViT-S. We also refer the reader to Appendix for additional studies.

5.1. Importance of the Different Components

We show the impact of adding different components from self-supervised learning on ViT trained with our framework.

表5：DAVIS 2017视频对象分割。我们评估了冻结特征在视频实例跟踪中的质量，报告了平均区域相似度$\mathcal{J}_m$和基于轮廓的平均准确率$\mathcal{F}_m$。我们与现有的自监督方法及在ImageNet上训练的监督式ViT-S/8进行了对比，图像分辨率为480p。

Method	Data	Arch.	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
Supervised					
ImageNet	INet	ViT-S/8	66.0	63.9	68.1
STM [48]	I/D/Y	RN50	81.8	79.2	84.3
Self-supervised					
CT [71]	VLOG	RN50	48.7	46.4	50.0
MAST [40]	YT-VOS	RN18	65.5	63.3	67.6
STC [37]	Kinetics	RN18	67.6	64.8	70.2
DINO	INet	ViT-S/16	61.8	60.2	63.4
DINO	INet	ViT-B/16	62.3	60.7	63.9
DINO	INet	ViT-S/8	**69.9**	**66.6**	**73.1**
DINO	INet	ViT-B/8	**71.4**	**67.9**	**74.9**

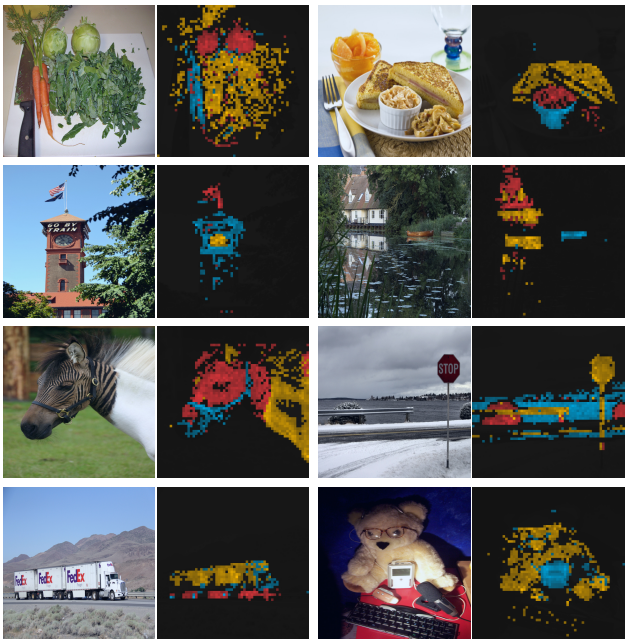


图3：多头注意力图。我们考察了使用DINO训练的ViT-S/8最后一层的多个注意力头，并展示了以[CLS]标记为查询的自注意力分布。不同颜色代表的注意力头聚焦于不同位置，这些位置对应着不同的物体或部件（更多示例见附录）。

4.2.2 探索场景的语义布局

如图1定性所示，我们的自注意力图包含了图像分割的信息。在本研究中，我们通过标准基准测试以及直接检验从这些注意力图生成的掩码质量，来衡量这一特性。

视频实例分割。在表5中，我们在DAVIS-2017视频实例分割基准[52]上评估了输出补丁标记的性能。我们遵循Jabri *et al*的实验协议[37]，通过在连续帧间进行最近邻匹配来分割场景；因此，我们没有在特征之上训练任何模型，也没有针对该任务微调任何权重。从表5中可以看出，尽管我们的训练目标和架构并非为密集任务设计，但在该基准测试中表现依然具有竞争力。由于网络未经微调，模型的输出必然保留了部分空间信息。最后，对于这一密集识别任务，采用小补丁尺寸的变体（“/8”）表现显著更优（ViT-B的+9.1% $(\mathcal{J}\&\mathcal{F})_m$）。

探究自注意力图。在图3中，我们展示了不同注意力头能够关注图像的不同语义区域，即使这些区域被遮挡（第三行的灌木丛）或尺寸较小（第二行的旗帜）。可视化结果基于480p分辨率图像生成，对ViT-S/8模型产生了3601个令牌的序列。图4则表明，在有杂乱背景的情况下，监督式ViT模型在定性和定量层面均未能有效关注目标物体。我们通过设定阈值保留自注意力图60%的质量，计算了真实标注与由此获得的分割掩模之间的杰卡德相似系数。需注意自注意力图本身是平滑的，并未针对生成掩模进行优化。尽管如此，监督模型与DINO模型在杰卡德相似度上仍存在显著差距。值得注意的是，自监督卷积网络同样包含分割信息，但需要专门方法从其权重中提取[31]。

4.2.3 下游任务的迁移学习

在表6中，我们评估了用DINO预训练的特征在不同下游任务上的质量。我们与同一架构在ImageNet上通过监督学习训练得到的特征进行了比较。我们遵循了Touvron *et al* [69]中采用的方案，在每个下游任务上对这些特征进行微调。我们观察到，对于ViT架构而言，自监督预训练比监督训练的特征具有更好的迁移性能，这与在卷积网络上的观察结果一致[10, 33, 62]。最后，自监督预训练显著提升了ImageNet上的性能（提升+1-2%）。

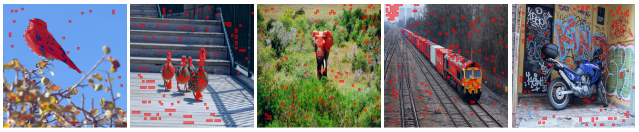
5. DINO的消融研究

在本节中，我们实证研究了DINO应用于ViT的效果。整个研究考虑的模型为ViT-S。更多相关研究可参阅附录部分。

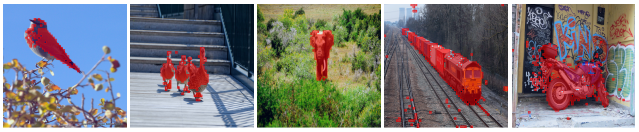
5.1. 不同组件的重要性

我们展示了在采用我们框架训练的ViT中，加入自监督学习不同组件所带来的影响。

Supervised



DINO



	Random	Supervised	DINO
ViT-S/16	22.0	27.3	45.9
ViT-S/8	21.8	23.7	44.7

Figure 4: **Segmentations from supervised versus DINO.** We visualize masks obtained by thresholding the self-attention maps to keep 60% of the mass. On top, we show the resulting masks for a ViT-S/8 trained with supervision and DINO. We show the best head for both models. The table at the bottom compares the Jaccard similarity between the ground truth and these masks on the validation images of PASCAL VOC12 dataset.

Table 6: **Transfer learning by finetuning pretrained models on different datasets.** We report top-1 accuracy. Self-supervised pretraining with DINO transfers better than supervised pretraining.

	Cifar_{10}	Cifar_{100}	INat_{18}	INat_{19}	Flwrs	Cars	INet
ViT-S/16							
Sup. [69]	**99.0**	89.5	70.7	76.6	98.2	92.1	79.9
DINO	**99.0**	**90.5**	**72.0**	**78.2**	**98.5**	**93.0**	**81.5**
ViT-B/16							
Sup. [69]	99.0	90.8	**73.2**	77.7	98.4	92.1	81.8
DINO	**99.1**	**91.7**	72.6	**78.6**	**98.8**	**93.0**	**82.8**

In Table 7, we report different model variants as we add or remove components. First, we observe that in the absence of momentum, our framework does not work (row 2) and more advanced operations, SK for example, are required to avoid collapse (row 9). However, with momentum, using SK has little impact (row 3). In addtition, comparing rows 3 and 9 highlights the importance of the momentum encoder for performance. Second, in rows 4 and 5, we observe that multi-crop training and the cross-entropy loss in DINO are important components to obtain good features. We also observe that adding a predictor to the student network has little impact (row 6) while it is critical in BYOL to prevent collapse [16, 30]. For completeness, we propose in Appendix B an extended version of this ablation study.

Importance of the patch size. In Fig. 5, we compare the k-NN classification performance of ViT-S models trained

Table 7: **Important component for self-supervised ViT pretraining.** Models are trained for 300 epochs with ViT-S/16. We study the different components that matter for the k-NN and linear ("Lin.") evaluations. For the different variants, we highlight the differences from the default DINO setting. The best combination is the momentum encoder with the multicrop augmentation and the cross-entropy loss. We also report results with BYOL [30], MoCo-v2 [15] and SwAV [10].

	Method	Mom.	SK	MC	Loss	Pred.	k-NN	Lin.
1	DINO	✓	✗	✓	CE	✗	72.8	76.1
2		✗	✗	✓	CE	✗	0.1	0.1
3		✓	✓	✓	CE	✗	72.2	76.0
4		✓	✗	✗	CE	✗	67.9	72.5
5		✓	✗	✓	MSE	✗	52.6	62.4
6		✓	✗	✓	CE	✓	71.8	75.6
7	BYOL	✓	✗	✗	MSE	✓	66.6	71.4
8	MoCov2	✓	✗	✗	INCE	✗	62.0	71.6
9	SwAV	✗	✓	✓	CE	✗	64.7	71.8

SK: Sinkhorn-Knopp, MC: Multi-Crop, Pred.: Predictor
CE: Cross-Entropy, MSE: Mean Square Error, INCE: InfoNCE

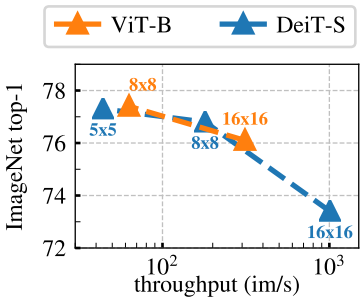


Figure 5: **Effect of Patch Size.** k-NN evaluation as a function of the throughputs for different input patch sizes with ViT-B and ViT-S. Models are trained for 300 epochs.

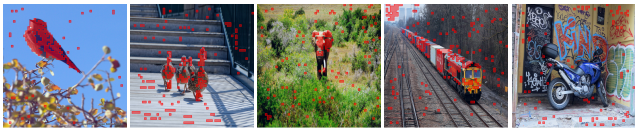
with different patch sizes, 16×16, 8×8 and 5×5. We also compare to ViT-B with 16×16 and 8×8 patches. All the models are trained for 300 epochs. We observe that the performance greatly improves as we decrease the size of the patch. It is interesting to see that performance can be greatly improved without adding additional parameters. However, the performance gain from using smaller patches comes at the expense of throughput: when using 5×5 patches, the throughput falls to 44 im/s, *vs* 180 im/s for 8×8 patches.

5.2. Impact of the choice of Teacher Network

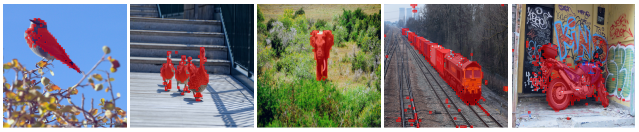
In this ablation, we experiment with different teacher network to understand its role in DINO. We compare models trained for 300 epochs using the k-NN protocol.

Building different teachers from the student. In Fig. 6(right), we compare different strategies to build the teacher from previous instances of the student besides the

Supervised



DINO



	Random	Supervised	DINO
ViT-S/16	22.0	27.3	45.9
ViT-S/8	21.8	23.7	44.7

图4：监督学习与DINO的分割效果对比。我们通过阈值化自注意力图并保留60%的质量来可视化得到的掩码。上方展示了使用监督学习和DINO训练的ViT-S/8模型生成的结果掩码，两者均选取了最佳注意力头进行展示。底部表格比较了PASCAL VOC12数据集验证图像上这些掩码与真实标注之间的Jaccard相似度。

表6：在不同数据集上微调预训练模型的迁移学习效果。我们报告了top-1准确率。采用DINO进行自监督预训练比监督预训练具有更好的迁移效果。

	Cifar_{10}	Cifar_{100}	INat_{18}	INat_{19}	Flwrs	Cars	INet
ViT-S/16							
Sup. [69]	**99.0**	89.5	70.7	76.6	98.2	92.1	79.9
DINO	**99.0**	**90.5**	**72.0**	**78.2**	**98.5**	**93.0**	**81.5**
ViT-B/16							
Sup. [69]	99.0	90.8	**73.2**	77.7	98.4	92.1	81.8
DINO	**99.1**	**91.7**	72.6	**78.6**	**98.8**	**93.0**	**82.8**

在表7中，我们报告了添加或移除组件时不同模型变体的表现。首先，我们观察到在没有动量机制的情况下，我们的框架无法正常工作（第2行），此时需要更高级的操作（例如SK）来避免崩溃（第9行）。然而，当引入动量机制后，使用SK的影响微乎其微（第3行）。此外，对比第3行与第9行结果凸显了动量编码器对性能的关键作用。其次，通过第4行和第5行可见，多裁剪训练和DINO中的交叉熵损失是获取优质特征的重要组件。我们还发现，在学生网络中添加预测器影响甚微（第6行），而这在BYOL中却是防止崩溃的关键要素[16,30]。为完整起见，我们在附录B中提供了该消融研究的扩展版本。

补丁尺寸的重要性。在图5中，我们比较了经过训练的ViT-S模型的k-NN分类性能

表7：自监督ViT预训练中的重要组件。所有模型均采用ViT-S/16架构训练300个周期。我们研究了影响{v*}-NN和线性（"Lin."）评估的关键要素，各变体与默认DINO设置的差异已标出。最佳组合为动量编码器配合多裁剪增强和交叉熵损失。同时列出了BYOL[30]、MoCo-v2[15]和SwAV[10]的对比结果。

Method	Mom.	SK	MC	Loss	Pred.	k-NN	Lin.
1 DINO	✓	✗	✓	CE	✗	72.8	76.1
2	✗	✗	✓	CE	✗	0.1	0.1
3	✓	✓	✓	CE	✗	72.2	76.0
4	✓	✗	✗	CE	✗	67.9	72.5
5	✓	✗	✓	MSE	✗	52.6	62.4
6	✓	✗	✓	CE	✓	71.8	75.6
7 BYOL	✓	✗	✗	MSE	✓	66.6	71.4
8 MoCov2	✓	✗	✗	INCE	✗	62.0	71.6
9 SwAV	✗	✓	✓	CE	✗	64.7	71.8

SK: Sinkhorn-Knopp, MC: 多裁剪, Pred.: 预测器, CE: 交叉熵, MSE: 均方误差, INCE: 信息噪声对比估计

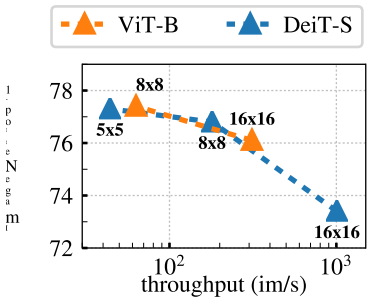


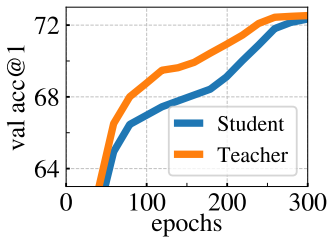
图5：补丁大小的影响。k-NN评估作为不同输入补丁大小下ViT-B和ViT-S吞吐量的函数。模型训练了300个周期。

采用不同的补丁尺寸，16×16、8×8和5×5。我们还与使用16×16和8×8补丁的ViT-B进行了比较。所有模型均训练了300个周期。我们观察到，随着补丁尺寸的减小，性能显著提升。有趣的是，无需增加额外参数即可大幅提升性能。然而，使用更小补丁带来的性能提升是以吞吐量为代价的：当使用5×5补丁时，吞吐量降至44 im/s，而8×8补丁则为180 im/s。

5.2. 教师网络选择的影响

在此消融实验中，我们尝试不同的教师网络以理解其在DINO中的作用。我们比较了使用k-NN协议训练300个周期的模型。

构建与学生不同的教师模型。在图6（右）中，我们比较了除现有方法外，基于学生模型先前实例构建教师模型的不同策略。



Teacher	Top-1
Student copy	0.1
Previous iter	0.1
Previous epoch	66.6
Momentum	72.8

Figure 6: Top-1 accuracy on ImageNet validation with k-NN classifier. **(left)** Comparison between the performance of the momentum teacher and the student during training. **(right)** Comparison between different types of teacher network. The momentum encoder leads to the best performance but is not the only viable option.

momentum teacher. First we consider using the student network from a previous epoch as a teacher. This strategy has been used in a memory bank [73] or as a form of clustering hard-distillation [8, 2, 14]. Second, we consider using the student network from the previous iteration, as well as a copy of the student for the teacher. In our setting, using a teacher based on a recent version of the student does not converge. This setting requires more normalizations to work. Interestingly, we observe that using a teacher from the previous epoch does not collapse, providing performance in the k-NN evaluation competitive with existing frameworks such as MoCo-v2 or BYOL. While using a momentum encoder clearly provides superior performance to this naive teacher, this finding suggests that there is a space to investigate alternatives for the teacher.

Analyzing the training dynamic. To further understand the reasons why a momentum teacher works well in our framework, we study its dynamic during the training of a ViT in the left panel of Fig. 6. A key observation is that this teacher constantly outperforms the student during the training, and we observe the same behavior when training with a ResNet-50 (Appendix D). This behavior has not been observed by other frameworks also using momentum [33, 30], nor when the teacher is built from the previous epoch. We propose to interpret the momentum teacher in DINO as a form of Polyak-Ruppert averaging [51, 59] with an exponentially decay. Polyak-Ruppert averaging is often used to simulate model ensembling to improve the performance of a network at the end of the training [38]. Our method can be interpreted as applying Polyak-Ruppert averaging during the training to constantly build a model ensembling that has superior performances. This model ensembling then guides the training of the student network [65].

5.3. Avoiding collapse

We study the complementarity role of centering and target sharpening to avoid collapse. There are two forms of collapse: regardless of the input, the model output is uniform along all the dimensions or dominated by one dimension. The centering avoids the collapse induced by a dominant dimension, but encourages an uniform output. Sharpening induces the opposite effect. We show this complementarity by decomposing the cross-entropy H into an entropy h and the Kullback-Leibler divergence ("KL") D_{KL}:

$$H(P_t, P_s) = h(P_t) + D_{KL}(P_t|P_s). \quad (5)$$

A KL equal to zero indicates a constant output, and hence a collapse. In Fig. 7, we plot the entropy and KL during training with and without centering and sharpening. If one operation is missing, the KL converges to zero, indicating a collapse. However, the entropy h converges to different values: 0 with no centering and $-\log(1/K)$ with no sharpening, indicating that both operations induce different form of collapse. Applying both operations balances these effects (see study of the sharpening parameter τ_t in Appendix D).

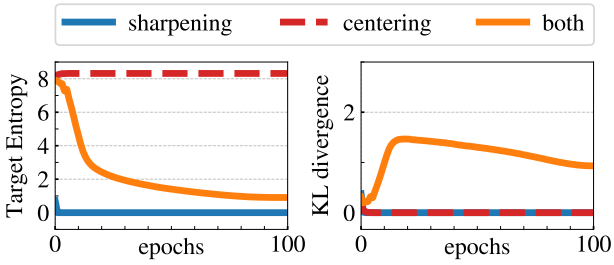


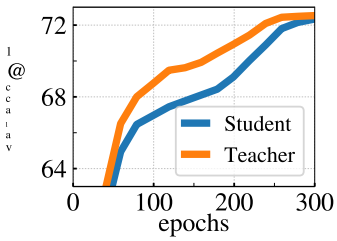
Figure 7: **Collapse study. (left)**: evolution of the teacher's target entropy along training epochs; **(right)**: evolution of KL divergence between teacher and student outputs.

Table 8: **Time and memory requirements.** We show total running time and peak memory per GPU ("mem.") when running ViT-S/16 DINO models on two 8-GPU machines. We report top-1 ImageNet val acc with linear evaluation for several variants of multi-crop, each having a different level of compute requirement.

	100 epochs		300 epochs		
multi-crop	top-1	time	top-1	time	mem.
2×224^2	67.8	15.3h	72.5	45.9h	9.3G
$2\times224^2 + 2\times96^2$	71.5	17.0h	74.5	51.0h	10.5G
$2\times224^2 + 6\times96^2$	73.8	20.3h	75.9	60.9h	12.9G
$2\times224^2 + 10\times96^2$	74.6	24.2h	76.1	72.6h	15.4G

5.4. Compute requirements

In Tab. 8, we detail the time and GPU memory requirements when running ViT-S/16 DINO models on two 8-GPU machines. We report results with several variants of multi-crop training, each having a different level of compute requirement. We observe in Tab. 8 that using multi-crop improves the accuracy / running-time tradeoff for DINO runs.



Teacher	Top-1
Student copy	0.1
Previous iter	0.1
Previous epoch	66.6
Momentum	72.8

图6：使用k-NN分类器在ImageNet验证集上的Top-1准确率。（左）动量教师模型与学生在训练期间的性能对比。（右）不同类型教师网络的比较。动量编码器带来了最佳性能，但并非唯一可行的选择。

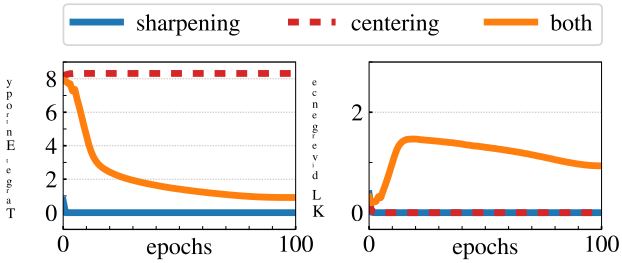


图7：坍塌研究。（左）：教师目标熵随训练周期的演变；（右）：教师与学生输出间KL散度的演变。

动量教师。首先，我们考虑使用前一周期（epoch）的学生网络作为教师。这一策略已在记忆库[73]或作为聚类硬蒸馏的一种形式[8,2,14]中被采用。其次，我们尝试使用前一次迭代的学生网络及其副本作为教师。在我们的实验设置中，基于学生网络近期版本构建的教师模型无法收敛，该设置需要更多标准化处理才能生效。有趣的是，我们观察到使用前一周期生成的教师模型不会崩溃，其在{v*}近邻评估中表现与MoCo-v2、BYOL等现有框架相当。虽然动量编码器的使用明显优于这种朴素教师方案，但这一发现表明存在探索替代性教师模型的研究空间。

分析训练动态。为了进一步理解动量教师在我们的框架中表现优异的原因，我们研究了ViT训练过程中其动态变化（图6左面板）。一个关键观察是：该教师在训练期间持续优于学生网络，这一现象在使用ResNet-50训练时同样存在（附录D）。其他同样采用动量的框架[33,30]或基于前一轮epoch构建教师模型时均未观察到该行为。我们提出将DINO中的动量教师解释为带有指数衰减的Polyak-Ruppert平均[51,59]形式。Polyak-Ruppert平均通常用于模拟模型集成，以提升网络在训练结束时的性能[38]。我们的方法可视为在训练过程中持续应用Polyak-Ruppert平均，不断构建具有更优性能的模型集成，继而指导学生网络的训练[65]。

5.3. 避免崩溃

我们研究了中心化与目标锐化的互补作用，以避免崩溃。存在两种形式的坍缩：无论输入如何，模型在所有维度上的输出都是均匀的，或由单一维度主导。中心化避免了由主导维度引发的坍缩，但会促使输出趋于均匀。锐化则产生相反的效果。我们通过将交叉熵H分解为熵h和Kullback-Leibler散度（“KL”）D_{KL}来展示这种互补性：

$$H(P_t, P_s) = h(P_t) + D_{KL}(P_t|P_s). \quad (5)$$

KL等于零表示输出恒定，即发生了崩溃。在图7中，我们绘制了训练过程中带与不带中心化和锐化操作时的熵与KL值。若缺少任一操作，KL会收敛至零，表明崩溃发生。然而，熵h会收敛至不同值：无中心化时为0，无锐化时为$-\log(1/K)$，这说明两种操作会引发不同形式的崩溃。同时应用这两种操作可平衡这些效应（参见附录D中关于锐化参数τ_t的研究）。

表8：时间与内存需求。我们展示了在两台配备8-GPU的机器上运行ViT-S/16 DINO模型时的总运行时间及每块GPU的峰值内存（“内存”）。针对多裁剪的几种变体，每种变体具有不同的计算需求，我们报告了线性评估下的ImageNet验证集top-1准确率。

	100 epochs		300 epochs		
multi-crop	top-1	time	top-1	time	mem.
2×224^2	67.8	15.3h	72.5	45.9h	9.3G
$2\times224^2 + 2\times96^2$	71.5	17.0h	74.5	51.0h	10.5G
$2\times224^2 + 6\times96^2$	73.8	20.3h	75.9	60.9h	12.9G
$2\times224^2 + 10\times96^2$	74.6	24.2h	76.1	72.6h	15.4G

5.4. 计算需求

在表8中，我们详细列出了在两台配备8块GPU的机器上运行ViT-S/16 DINO模型所需的时间与显存开销。我们汇报了采用不同计算资源需求级别的多裁剪训练变体结果。通过表8可观察到，使用多裁剪技术优化了DINO运行的精度与耗时权衡关系。

For example, the performance is 72.5% after 46 hours of training without multi-crop (i.e. 2×224^2) while DINO in $2\times224^2+10\times96^2$ crop setting reaches 74.6% in 24 hours only. This is an improvement of $+2\%$ while requiring $2\times$ less time, though the memory usage is higher ($15.4G$ versus $9.3G$). We observe that the performance boost brought with multi-crop cannot be caught up by more training in the 2×224^2 setting, which shows the value of the "local-to-global" augmentation. Finally, the gain from adding more views diminishes (+.2% form $6\times$ to $10\times$ 96^2 crops) for longer trainings.

Overall, training DINO with Vision Transformers achieves 76.1 top-1 accuracy using two 8-GPU servers for 3 days. This result outperforms state-of-the-art self-supervised systems based on convolutional networks of comparable sizes with a significant reduction of computational requirements [30, 10]. Our code is available to train self-supervised ViT on a limited number of GPUs.

5.5. Training with small batches

bs	128	256	512	1024
top-1	57.9	59.1	59.6	59.9

Table 9: **Effect of batch sizes.** Top-1 with k-NN for models trained for 100 epochs without multi-crop.

In Tab. 9, we study the impact of the batch size on the features obtained with DINO. We also study the impact of the smooth parameter m used in the centering update rule of Eq. 4 in Appendix D. We scale the learning rate linearly with the batch size [29]: $lr = 0.0005 * \text{batchsize}/256$. Tab. 9 confirms that we can train models to high performance with small batches. Results with the smaller batch sizes ($bs = 128$) are slightly below our default training setup of $bs = 1024$, and would certainly require to re-tune hyperparameters like the momentum rates for example. Note that the experiment with batch size of 128 runs on only 1 GPU. We have explored training a model with a batch size of 8, reaching 35.2% after 50 epochs, showing the potential for training large models that barely fit an image per GPU.

6. Conclusion

In this work, we have shown the potential of self-supervised pretraining a standard ViT model, achieving performance that are comparable with the best convnets specifically designed for this setting. We have also seen emerged two properties that can be leveraged in future applications: the quality of the features in k-NN classification has a potential for image retrieval where ViT are already showing promising results [22]. The presence of information about the scene layout in the features can also benefit weakly supervised image segmentation. However, the main result of this paper is that we have evidences that self-supervised learning could be the key to developing a BERT-like model based on ViT. In the future, we plan to explore if pretraining a large ViT model with DINO on random uncurated images could push the limits of visual features [28].

Acknowledgement. We thank Mahmoud Assran, Matthijs Douze, Allan Jabri, Jure Zbontar, Alaaeldin El-Nouby, Y-Lan Boureau, Kaiming He, Thomas Lucas as well as the Thoth and FAIR teams for their help, support and discussions around this project. Julien Mairal was funded by the ERC grant number 714381 (SOLARIS project) and by ANR 3IA MIAI@Grenoble Alpes (ANR-19-P3IA-0003).

References

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018. 3

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 2, 9

[3] Mahmoud Assran, Nicolas Ballas, Lluis Castrejon, and Michael Rabbat. Recovering petaflops in contrastive semi-supervised learning of visual representations. *preprint arXiv:2006.10803*, 2020. 14

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *preprint arXiv:1409.0473*, 2014. 5

[5] Maxim Berman, Hervé Jégou, Vedaldi Andrea, Iasonas Kokkinos, and Matthijs Douze. MultiGrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019. 6

[6] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, 2017. 2

[7] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *SIGKDD*, 2006. 3

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2, 4, 9, 16

[9] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019. 2, 16

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 1, 2, 3, 4, 5, 7, 8, 10, 14, 15, 16, 17, 18

[11] Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. The best of both worlds: Combining recent advances in neural machine translation. *preprint arXiv:1804.09849*, 2018. 5

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *preprint arXiv:2002.05709*, 2020. 2, 3, 5, 16, 17

例如，在不使用多裁剪（即2×224^2）的情况下，经过46小时训练后性能达到72.5%，而DINO在$2\times224^2+10\times96^2$的裁剪设置下仅用24小时就达到了74.6%。这一结果提升了+2%，同时所需时间减少了2×，尽管内存使用量更高（15.4G对比9.3G）。我们观察到，在2×224^2设置下，通过增加训练时间无法弥补多裁剪带来的性能提升，这体现了“局部到全局”增强策略的价值。最后，随着训练时间的延长，增加更多视图带来的增益逐渐减弱（从$6\times$到10×96^2裁剪仅提升+.2%）。

总体而言，使用Vision Transformers训练DINO在双8-GPU服务器上运行3天即可实现76.1的top-1准确率。这一结果超越了基于同等规模卷积网络的最先进自监督系统，同时显著降低了计算需求[30, 10]。我们提供的代码支持在有限数量GPU上训练自监督ViT模型。

5.5. 小批量训练

bs	128	256	512	1024
top-1	57.9	59.1	59.6	59.9

表9：批次大小的影响。使用k-NN的Top-1准确率，模型训练100个周期且未采用多裁剪。

在表9中，我们研究了批量大小对DINO所获特征的影响，同时探讨了附录D式4中心化更新规则中平滑参数m的作用。我们根据批量大小线性调整学习率[29]：$lr = 0.0005 * \text{批量大小}/256$。表9证实了即使采用小批量也能训练出高性能模型。较小批量（$bs=128$）的结果略低于默认训练设置$bs=1024$，此时可能需要重新调整超参数（例如动量率）。需注意批量128的实验仅在1块GPU上运行。我们还尝试了批量8的训练，50个周期后达到35.2%的精度，这表明即使每块GPU仅能容纳单张图像，仍具备训练大模型的潜力。

6. 结论

在这项工作中，我们展示了自监督预训练标准ViT模型的潜力，其性能可与专为此场景设计的最佳卷积网络相媲美。我们还发现了两个可在未来应用中加以利用的特性：在k-NN分类中特征的质量显示出图像检索的潜力——ViT在此领域已展现出令人瞩目的成果[22]。此外，特征中包含的场景布局信息也有助于弱监督图像分割。然而，本文的主要成果在于，我们获得了证据表明自监督学习可能是开发基于ViT。未来，我们计划探索是否通过在随机未筛选图像上使用DINO预训练大型ViT模型，能够突破视觉特征的极限[28]。

致谢。我们感谢Mahmoud Assran、Matthijs Douze、Allan Jabri、Jure Zbontar、Alaaeldin El-Nouby、Y-Lan Boureau、Kaiming He、Thomas Lucas以及Thoth和FAIR团队对本项目的帮助、支持与讨论。Julien Mairal的研究工作获得了ERC资助（项目编号714381，SOLARIS项目）及法国国家研究署（ANR）3IA MIAI@Grenoble Alpes计划（编号ANR-19-P3IA-0003）的资助。

参考文献

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, 以及 Geoffrey E Hinton。通过在线蒸馏进行大规模分布式神经网络训练。*arXiv preprint arXiv:1804.03235*，2018年。3 [2] Yuki Markus Asano, Christian Rupprecht, 和 Andrea Vedaldi。通过同步聚类与表征学习实现自标注。收录于*ICLR*，2020年。2, 9 [3] Mahmoud Assran, Nicolas Ballas, Lluis Castrejon, 和 Michael Rabbat。在视觉表征的对比半监督学习中恢复千万亿次浮点运算。*preprint arXiv:2006.10803*，2020年。14 [4] Dzmitry Bahdanau, Kyunghyun Cho, 和 Yoshua Bengio。通过联合学习对齐与翻译实现神经机器翻译。*preprint arXiv:1409.0473*，2014年。5 [5] Maxim Berman, Hervé Jégou, Vedaldi Andrea, Iasonas Kokkinos, 和 Matthijs Douze。MultiGrain：面向类别与实例的统一图像嵌入方法。*arXiv preprint arXiv:1902.05509*，2019年。6 [6] Piotr Bojanowski 和 Armand Joulin。通过预测噪声进行无监督学习。收录于*ICML*，2017年。2 [7] Cristian Buciluǎ, Rich Caruana, 和 Alexandru Niculescu-Mizil。模型压缩。收录于*SIGKDD*，2006年。3 [8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, 和 Matthijs Douze。深度聚类用于视觉特征的无监督学习。收录于*ECCV*，2018年。2, 4, 9, 16 [9] Mathilde Caron, Piotr Bojanowski, Julien Mairal, 和 Armand Joulin。非精选数据上图像特征的无监督预训练。收录于*ICCV*，2019年。2, 16

[10] Mathilde Caron、Ishan Misra、Julien Mairal、Priya Goyal、Piotr Bojanowski 和 Armand Joulin。通过对比聚类分配的无监督视觉特征学习。发表于*NeurIPS*，2020年。1, 2, 3, 4, 5, 7, 8, 10, 14, 15, 16, 17, 18

[11] Mia Xu Chen、Orhan Firat、Ankur Bapna、Melvin Johnson、Wolfgang Macherey、George Foster、Llion Jones、Niki Parmar、Mike Schuster、Zhifeng Chen等。两全其美：结合神经机器翻译的最新进展。*preprint arXiv:1804.09849*，2018年。5

[12] 陈霆（Ting Chen）、西蒙·科恩布利斯（Simon Kornblith）、穆罕默德·诺鲁齐（Mohammad Norouzi）与杰弗里·辛顿（Geoffrey Hinton）。视觉表示对比学习的简易框架。*preprint arXiv:2002.05709*，2020年。第2、3、5、16、17页

[13] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 3, 5, 6, 14

[14] Weijie Chen, Shiliang Pu, Di Xie, Shicai Yang, Yilu Guo, and Luojun Lin. Unsupervised image classification for deep representation learning. *arXiv preprint arXiv:2006.11480*, 2020. 9, 15

[15] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *preprint arXiv:2003.04297*, 2020. 5, 8, 14, 15, 18

[16] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *preprint arXiv:2011.10566*, 2020. 2, 3, 4, 8, 14, 16, 18

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 15

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *preprint arXiv:1810.04805*, 2018. 1, 4, 5, 19

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *preprint arXiv:2010.11929*, 2020. 1, 4, 5, 13

[20] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 2016. 2

[21] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR*, 2009. 6

[22] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training vision transformers for image retrieval. *preprint arXiv:2102.05644*, 2021. 10

[23] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. *preprint arXiv:2007.06346*, 2020. 2

[24] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 13

[25] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. 2021. 3

[26] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *CVPR*, 2020. 2

[27] Spyros Gidaris, Andrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. Online bag-of-visual-words generation for unsupervised representation learning. *arXiv preprint arXiv:2012.11552*, 2020. 2, 5

[28] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *preprint arXiv:2103.01988*, 2021. 10

[29] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *preprint arXiv:1706.02677*, 2017. 5, 10

[30] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 2, 3, 4, 5, 8, 9, 10, 14, 15, 16, 18

[31] Shir Gur, Ameen Ali, and Lior Wolf. Visualization of supervised and self-supervised neural networks via attribution guided factorization. *preprint arXiv:2012.02166*, 2020. 7

[32] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, 2010. 2

[33] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 3, 4, 5, 7, 9, 16

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 5

[35] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *preprint arXiv:1503.02531*, 2015. 2, 3

[36] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *ICML*, 2019. 2

[37] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. 2020. 7

[38] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *preprint arXiv:1412.2007*, 2014. 4, 9

[39] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. Opennmt: Open-source toolkit for neural machine translation. *preprint arXiv:1701.02810*, 2017. 5

[40] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, 2020. 7

[41] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 3

[42] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. *ICLR*, 2021. 2

[43] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *preprint arXiv:1608.03983*, 2016. 5

[44] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 5

[13] 陈挺、Simon Kornblith、Kevin Swersky、Mohammad Norouzi与Geoffrey Hinton。大规模自监督模型是强大的半监督学习器。发表于*NeurIPS*，2020年。3, 5, 6, 14 [14] 陈伟杰、濮世亮、谢迪、杨世才、郭一璐、林罗军。基于深度表征学习的无监督图像分类。*arXiv preprint arXiv:2006.11480*，2020年。9, 15 [15] 陈新雷、范浩祺、Ross Girshick、何恺明。动量对比学习改进基线方法。*preprint arXiv:2003.04297*，2020年。5, 8, 14, 15, 18 [16] 陈新雷、何恺明。探索简单孪生表征学习。*preprint arXiv:2011.10566*，2020年。2, 3, 4, 8, 14, 16, 18 [17] Marco Cuturi。Sinkhorn距离：最优传输的极速计算。发表于*NeurIPS*，2013年。15 [18] Jacob Devlin、张明伟、Kenton Lee、Kristina Toutanova。BERT：面向语言理解的深度双向Transformer预训练。*preprint arXiv:1810.04805*，2018年。1, 4, 5, 19 [19] Alexey Dosovitskiy等。一幅图像等价于16x16词汇：大规模图像识别的Transformer。*preprint arXiv:2010.11929*，2020年。1, 4, 5, 13 [20] Alexey Dosovitskiy、Philipp Fischer、Jost Tobias Springenberg、Martin Riedmiller、Thomas Brox。基于范例卷积神经网络的判别式无监督特征学习。*TPAMI*，2016年。2 [21] Matthijs Douze、Hervé Jégou等。GIST描述符在网络规模图像搜索中的评估。发表于*CIVR*，2009年。6 [22] Alaaeldin El-Nouby、Natalia Neverova、Ivan Laptev、Hervé Jégou。面向图像检索的视觉Transformer训练。*preprint arXiv:2102.05644*，2021年。10 [23] Aleksandr Ermolov等。自监督表征学习中的白化技术。*preprint arXiv:2007.06346*，2020年。2 [24] Mark Everingham等。PASCAL视觉对象分类（VOC）挑战赛。*IJCV*，2010年。13 [25] 方志远、王建峰、王丽娟、张磊、杨业周、刘子成。SEED：视觉表征的自监督蒸馏。2021年。3 [26] Spyros Gidaris等。通过预测视觉词袋学习表征。发表于*CVPR*，2020年。2 [27] Spyros Gidaris等。无监督表征学习的在线视觉词袋生成。*arXiv preprint arXiv:2012.11552*，2020年。2, 5 [28] Priya Goyal等。真实场景中视觉特征的自监督预训练。*preprint arXiv:2103.01988*，2021年。10 [29] Priya Goyal、Piotr Dollár、Ross Girshick、Pieter Noordhuis、Lukasz Wesolowski、Aapo Kyrola、Andrew Tulloch、贾扬清和何恺明。精确的大批量SGD：1小时内训练ImageNet。*preprint arXiv:1706.02677*，2017年。5, 10 [30] Jean-Bastien Grill、Florian Strub、Florent Altché、Corentin Tallec、Pierre H Richemond、Elena Buchatskaya、Carl Doersch、Bernardo Avila Pires、Zhaohan Daniel Guo、Mohammad Gheshlaghi Azar、Bilal Piot、Koray Kavukcuoglu、Rémi Munos和Michal Valko。自引导潜在空间：自监督学习的新方法。收录于*NeurIPS*，2020年。2, 3, 4, 5, 8, 9, 10, 14, 15, 16, 18 [31] Shir Gur、Ameen Ali和Lior Wolf。通过归因引导分解可视化监督与自监督神经网络。*preprint arXiv:2012.02166*，2020年。7 [32] Michael Gutmann和Aapo Hyvärinen。噪声对比估计：非归一化统计模型的新估计原理。收录于*International Conference on Artificial Intelligence and Statistics*，2010年。2 [33] 何恺明、范浩琪、吴育昕、谢赛宁和Ross Girshick。动量对比无监督视觉表示学习。收录于*CVPR*，2020年。1, 2, 3, 4, 5, 7, 9, 16 [34] 何恺明、张翔宇、任少卿和孙剑。深度残差学习用于图像识别。收录于*CVPR*，2016年。4, 5 [35] Geoffrey Hinton、Oriol Vinyals和Jeff Dean。神经网络中的知识蒸馏。*preprint arXiv:1503.02531*，2015年。2, 3 [36] 黄佳波、董琪、龚曙光和朱夏恬。通过邻域发现的无监督深度学习。收录于*ICML*，2019年。2 [37] Allan Jabri、Andrew Owens和Alexei A Efros。时空对应作为对比随机游走。2020年。7 [38] Sébastien Jean、Kyunghyun Cho、Roland Memisevic和Yoshua Bengio。论神经机器翻译中极大目标词汇的使用。*preprint arXiv:1412.2007*，2014年。4, 9 [39] Guillaume Klein、Yoon Kim、Yuntian Deng、Jean Senellart和Alexander M Rush。Opennmt：神经机器翻译开源工具包。*preprint arXiv:1701.02810*，2017年。5 [40] 赖梓航、卢Erika和谢伟迪。Mast：记忆增强的自监督跟踪器。收录于*CVPR*，2020年。7 [41] Dong-Hyun Lee等。伪标签：深度神经网络简单高效的半监督学习方法。收录于*Workshop on challenges in representation learning, ICML*，2013年。3 [42] 李俊楠、周盼、熊才明和Steven C.H. Hoi。原型对比无监督表示学习。*ICLR*，2021年。2 [43] Ilya Loshchilov和Frank Hutter。带热重启的随机梯度下降。*preprint arXiv:1608.03983*，2016年。5 [44] Ilya Loshchilov和Frank Hutter。修复Adam中的权重衰减正则化。2018年。5

[45] Julien Mairal. Cyanure: An open-source toolbox for empirical risk minimization for python, c++, and soon more. *preprint arXiv:1912.08165*, 2019. 13, 14

[46] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. 13

[47] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018. 3

[48] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 7

[49] Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V Le. Meta pseudo labels. *preprint arXiv:2003.10580*, 2020. 14

[50] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 6

[51] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 4, 9, 17

[52] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *preprint arXiv:1704.00675*, 2017. 7

[53] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. 2018. 6

[54] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 6

[55] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 1

[56] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020. 13

[57] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 6

[58] Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *preprint arXiv:2010.10241*, 2020. 2, 4

[59] David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, 1988. 4, 9

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 1, 5, 13

[61] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *NeurIPS*, 2016. 4, 16

[62] Mert Bulent Sariyildiz, Yannis Kalantidis, Diane Larlus, and Karteek Alahari. Concept generalization in visual representation learning. *arXiv preprint arXiv:2012.05649*, 2020. 7

[63] Zhiqiang Shen, Zechun Liu, Jie Qin, Lei Huang, Kwang-Ting Cheng, and Marios Savvides. S2-bnn: Bridging the gap between self-supervised real and 1-bit neural networks via guided distribution calibration. *arXiv preprint arXiv:2102.08946*, 2021. 3

[64] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 14

[65] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *preprint arXiv:1703.01780*, 2017. 3, 4, 9, 17

[66] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015. 6

[67] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *NeurIPS*, 2020. 5

[68] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015. 6

[69] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *preprint arXiv:2012.12877*, 2020. 1, 4, 5, 6, 7, 8, 13, 17

[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 4

[71] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 7

[72] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. 2020. 6

[73] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 2, 4, 5, 9, 18

[74] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016. 2

[75] Qizhe Xie, Zihang Dai Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. *preprint arXiv:1904.12848*, 2020. 14

[76] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 3

[77] Haohang Xu, Xiaopeng Zhang, Hao Li, Lingxi Xie, Hongkai Xiong, and Qi Tian. Seed the views: Hierarchical semantic alignment for contrastive representation learning. *arXiv preprint arXiv:2012.02733*, 2021. 16

[45] Julien Mairal. Cyanure：一个开源的Python、C++（及后续更多语言）经验风险最小化工具箱。*preprint arXiv:1912.08165*，2019年。13, 14 [46] Maria-Elena Nilsback与Andrew Zisserman. 大规模花卉种类自动分类。载于*2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*，2008年。13 [47] Mehdi Noroozi、Ananth Vinjimoor、Paolo Favaro与Hamed Pirsiavash. 通过知识迁移增强自监督学习。载于*CVPR*，2018年。3 [48] Seoung Wug Oh、Joon-Young Lee、Ning Xu与Seon Joo Kim. 使用时空记忆网络的视频对象分割。载于*ICCV*，2019年。7 [49] Hieu Pham、Qizhe Xie、Zihang Dai与Quoc V Le. 元伪标签。*preprint arXiv:2003.10580*，2020年。14 [50] James Philbin、Ondrej Chum、Michael Isard、Josef Sivic与Andrew Zisserman. 量化中的迷失：改进大规模图像数据库中的特定对象检索。载于*CVPR*，2008年。6 [51] Boris T Polyak与Anatoli B Juditsky. 通过平均加速随机逼近。*SIAM journal on control and optimization*，30(4):838–855，1992年。4, 9, 17 [52] Jordi Pont-Tuset、Federico Perazzi、Sergi Caelles、Pablo Arbeláez、Alex Sorkine-Hornung与Luc Van Gool. 2017年DAVIS视频对象分割挑战赛。*preprintarXiv:1704.00675*，2017年。7 [53] Filip Radenović、Ahmet Iscen、Giorgos Tolias、Yannis Avrithis与Ondřej Chum. 重访牛津与巴黎：大规模图像检索基准测试。2018年。6 [54] Filip Radenović、Giorgos Tolias与Ondřej Chum. 无需人工标注的CNN图像检索微调。*IEEE transactions on pattern analysis and machine intelligence*，2018年。6 [55] Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei与Ilya Sutskever. 语言模型是无监督多任务学习者。1 [56] Ilija Radosavovic、Raj Prateek Kosaraju、Ross Girshick、Kaiming He与Piotr Dollár. 设计网络设计空间。载于*CVPR*，2020年。13 [57] Jerome Revaud、Jon Almazán、Rafael S Rezende与Cesar Roberto de Souza. 以平均精度学习：用列表式损失训练图像检索。载于*ICCV*，2019年。6 [58] Pierre H Richemond、Jean-Bastien Grill、Florent Altché、Corentin Tallec、Florian Strub、Andrew Brock、Samuel Smith、Soham De、Razvan Pascanu、Bilal Piot等. BYOL无需批量统计仍可工作。*preprint arXiv:2010.10241*，2020年。2, 4 [59] David Ruppert. 缓慢收敛Robbins-Monro过程的高效估计。技术报告，1988年。4, 9 [60] Olga Russakovsky、Jia Deng、Hao Su、Jonathan Krause、Sanjeev Satheesh、Sean Ma、Zhiheng Huang、Andrej Karpathy、Aditya Khosla、Michael Bernstein、Alexander C Berg与李飞飞. ImageNet大规模视觉识别挑战赛。*IJCV*，2015年。1, 5, 13 [61] Tim Salimans与Diederik P Kingma. 权重归一化：加速深度神经网络训练的简单重参数化方法。*NeurIPS*，2016年。4, 16

[62] Mert Bulent Sariyildiz, Yannis Kalantidis, Diane Larlus, 和 Karteek Alahari。视觉表示学习中的概念泛化。*arXiv preprint arXiv:2012.05649*，2020年。7 [63] 沈志强, 刘泽春, 秦杰, 黄磊, Kwang-Ting Cheng, 和 Marios Savvides。S2-BNN：通过引导分布校准弥合自监督实值与1位神经网络间的差距。*arXiv preprint arXiv:2102.08946*，2021年。3 [64] Kihyuk Sohn, David Berthelot, 李春良, 张子浩, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, 张翰, 和 Colin Raffel。FixMatch：利用一致性与置信度简化半监督学习。载于*NeurIPS*，2020年。14 [65] Antti Tarvainen 和 Harri Valpola。均值教师是更好的榜样：权重平均一致性目标提升半监督深度学习效果。*preprint arXiv:1703.01780*，2017年。3, 4, 9, 17 [66] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, 和李立佳。YFCC100M：多媒体研究的新数据。*arXiv preprint arXiv:1503.01817*，2015年。6 [67] 田永龙, 孙晨, Ben Poole, Dilip Krishnan, Cordelia Schmid, 和 Phillip Isola。什么构成了对比学习的良好视角。*NeurIPS*，2020年。5 [68] Giorgos Tolias, Ronan Sicre, 和 Hervé Jégou。基于CNN激活积分最大池化的特定对象检索。*arXiv preprint arXiv:1511.05879*，2015年。6 [69] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, 和 Hervé Jégou。训练数据高效的图像Transformer及注意力蒸馏。*preprint arXiv:2012.12877*，2020年。1, 4, 5, 6, 7, 8, 13, 17 [70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, 和 Illia Polosukhin。注意力机制就是你所需要的一切。载于*NeurIPS*，2017年。1, 4 [71] 王晓龙, Allan Jabri, 和 Alexei A Efros。从时间循环一致性中学习对应关系。载于*CVPR*，2019年。7 [72] Tobias Weyand, Andre Araujo, 曹秉毅, 和 Jack Sim。Google地标数据集v2——实例级识别与检索的大规模基准。2020年。6 [73] 吴志荣, 熊元君, 余星乐, 和林达华。通过非参数实例判别进行无监督特征学习。载于*CVPR*，2018年。2, 4, 5, 9, 18 [74] 谢俊元, Ross Girshick, 和 Ali Farhadi。无监督深度嵌入用于聚类分析。载于*ICML*，2016年。2 [75] 谢启哲, 戴子航, Eduard Hovy, 梁明涛, 和 Quoc V. Le。无监督数据增强用于一致性训练。*preprint arXiv:1904.12848*，2020年。14 [76] 谢启哲, 梁明涛, Eduard Hovy, 和 Quoc V Le。带噪声学生的自训练提升ImageNet分类。载于*CVPR*，2020年。3 [77] 徐浩航, 张晓鹏, 李浩, 谢凌曦, 熊红凯, 和田奇。播种视角：对比表示学习的层次语义对齐。*arXiv preprint arXiv:2012.02733*，2021年。16

[78] Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert. Iterative pseudo-labeling for speech recognition. *preprint arXiv:2005.09267*, 2020. 3

[79] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *preprint arXiv:1905.00546*, 2019. 3

[80] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016. 2

[81] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 2, 5

[82] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 5

[83] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020. 1

[84] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, 2014. 13

[85] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, 2019. 2

Appendix

A. Additional Results

k-NN classification. In Tab. 10, we evaluate the frozen representations given by ResNet-50 or ViT-small pre-trained with DINO with two evaluation protocols: linear or k-NN. For both evaluations, we extract representations from a pre-trained network without using any data augmentation. Then, we perform classification either with weighted k-NN or with a linear regression learned with `cyanure` library [45]. In Tab. 10 we see that ViT-S accuracies are better than accuracies obtained with RN50 both with a linear or a k-NN classifier. However, the performance gap when using the k-NN evaluation is much more significant than when considering linear evaluation. For example on ImageNet 1%, ViT-S outperforms ResNet-50 by a large margin of $+14.1\%$ with k-NN evaluation. This suggests that transformers architectures trained with DINO might offer more model flexibility that benefits the k-NN evaluation. K-NN classifiers have the great advantage of being fast and light to deploy, without requiring any domain adaptation. Overall, ViT trained with DINO provides features that combine particularly well with k-NN classifiers.

Self-supervised ImageNet pretraining of ViT. In this experiment, we study the impact of pretraining a supervised ViT model with our method. In Tab. 11, we compare the performance of supervised ViT models that are initialized with different pretraining or guided during training with an additional pretrained convnet. The first set of models are pretrained with and without supervision on the large curated dataset composed of 300M images. The second set of models are trained with hard knowledge distillation from a pretrained supervised RegNetY [56]. The last set of models do not use any additional data nor models, and are initialized either randomly or after a pretraining with DINO on ImageNet. Compare to random initialization, pretraining with DINO leads to a performance gain of +1%. This is not caused by a longer training since pretraining with supervision instead of DINO does not improve performance. Using self-supervised pretraining reduces the gap with models pretrained on extra data or distilled from a convnet.

Table 10: **k-NN and linear evaluation for ViT-S/16 and ResNet-50 pre-trained with DINO.** We use ImageNet-1k [60] ("Inet"), Places205 [84], PASCAL VOC [24] and Oxford-102 flowers ("FLOWERS") [46]. ViT trained with DINO provides features that are particularly k-NN friendly.

	Logistic			k-NN		
	RN50	ViT-S	Δ	RN50	ViT-S	Δ
Inet 100%	72.1	75.7	3.6	67.5	74.5	7.0
Inet 10%	67.8	72.2	4.4	59.3	69.1	9.8
Inet 1%	55.1	64.5	9.4	47.2	61.3	14.1
Pl. 10%	53.4	52.1	-1.3	46.9	48.6	1.7
Pl. 1%	46.5	46.3	-0.2	39.2	41.3	2.1
VOC07	88.9	89.2	0.3	84.9	88.0	3.1
FLOWERS	95.6	96.4	0.8	87.9	89.1	1.2
Average Δ			2.4			**5.6**

Table 11: **ImageNet classification with different pretraining.** Top-1 accuracy on ImageNet for supervised ViT-B/16 models using different pretrainings or using an additional pretrained convnet to guide the training. The methods use different image resolution ("res.") and training procedure ("tr. proc."), i.e., data augmentation and optimization. "MPP" is *Masked Patch Prediction*.

Pretraining				
method	data	res.	tr. proc.	Top-1
Pretrain on additional data				
MMP	JFT-300M	384	[19]	79.9
Supervised	JFT-300M	384	[19]	84.2
Train with additional model				
Rand. init.	-	224	[69]	83.4
No additional data nor model				
Rand. init.	-	224	[19]	77.9
Rand. init.	-	224	[69]	81.8
Supervised	ImNet	224	[69]	81.9
DINO	ImNet	224	[69]	82.8

[78] 徐千桐、Tatiana Likhomanenko、Jacob Kahn、Awni Hannun、Gabriel Synnaeve与Ronan Collobert。语音识别中的迭代伪标注技术。*preprint arXiv:2005.09267*，2020年。3 [79] I Zeki Yalniz、Hervé Jégou、Kan Chen、Manohar Paluri及Dhruv Mahajan。十亿级半监督学习在图像分类中的应用。*preprint arXiv:1905.00546*，2019年。3 [80] 杨建伟、Devi Parikh与Dhruv Batra。深度表示与图像簇的联合无监督学习。收录于*CVPR*，2016年。2 [81] Jure Zbontar、李静、Ishan Misra、Yann LeCun及Stéphane Deny。Barlow双胞胎：通过冗余减少实现自监督学习。*arXiv preprint arXiv:2103.03230*，2021年。2,5 [82] Richard Zhang、Phillip Isola与Alexei A Efros。多彩图像上色技术。收录于*ECCV*，2016年。5 [83] 赵恒爽、贾佳亚与Vladlen Koltun。探索自注意力机制在图像识别中的应用。收录于*CVPR*，2020年。1 [84] Bolei Zhou、Agata Lapedriza、肖建雄、Antonio Torralba及Aude Oliva。基于Places数据库的场景识别深度特征学习。收录于*NeurIPS*，2014年。13 [85] 庄承旭、Alex Lin Zhai与Daniel Yamins。视觉嵌入无监督学习的局部聚合方法。收录于*ICCV*，2019年。2

附录

A. 补充结果

k-最近邻（NN）分类。在表10中，我们评估了由ResNet-50或ViT-small通过DINO预训练提供的冻结表示，采用两种评估协议：线性或k-NN。对于这两种评估，我们从预训练网络中提取表示，不使用任何数据增强。然后，我们通过加权k-NN或使用cyanure库[45]学习的线性回归进行分类。在表10中可以看到，无论是线性分类器还是k-NN分类器，ViT-S的准确率都优于RN50。然而，使用k-NN评估时的性能差距比线性评估更为显著。例如，在ImageNet 1%数据集上，ViT-S以+14.1%的大幅度领先于ResNet-50（基于k-NN评估）。这表明，通过DINO训练的Transformer架构可能提供了更大的模型灵活性，这对k-NN评估尤为有利。K-NN分类器具有部署快速、轻量级的巨大优势，且无需任何领域适应。总体而言，DINO训练的ViT所提供的特征与k-NN分类器结合得尤为出色。

表10：k-NN及线性评估结果，针对使用DINO预训练的ViT-S/16与ResNet-50模型。我们采用的数据集包括ImageNet-1k [60]（简称"Inet"）、Places205 [84]、PASCAL VOC [24]以及Oxford-102花卉数据集（标记为"FLOWERS"）[46]。经DINO训练的ViT模型所提供的特征尤其适合k-NN方法。

	Logistic			k-NN		
	RN50	ViT-S	Δ	RN50	ViT-S	Δ
Inet 100%	72.1	75.7	3.6	67.5	74.5	7.0
Inet 10%	67.8	72.2	4.4	59.3	69.1	9.8
Inet 1%	55.1	64.5	9.4	47.2	61.3	14.1
Pl. 10%	53.4	52.1	-1.3	46.9	48.6	1.7
Pl. 1%	46.5	46.3	-0.2	39.2	41.3	2.1
VOC07	88.9	89.2	0.3	84.9	88.0	3.1
FLOWERS	95.6	96.4	0.8	87.9	89.1	1.2
Average Δ			2.4			**5.6**

表11：采用不同预训练方法的ImageNet分类结果。展示了监督式ViT-B/16模型在使用不同预训练方法或额外预训练卷积网络指导训练时的ImageNet Top-1准确率。各方法采用不同图像分辨率（“分辨率”）和训练流程（“训练流程”），即数据增强与优化策略。其中“MPP”指代*Masked Patch Prediction*。

Pretraining				
method	data	res.	tr. proc.	Top-1
Pretrain on additional data				
MMP	JFT-300M	384	[19]	79.9
Supervised	JFT-300M	384	[19]	84.2
Train with additional model				
Rand. init.	-	224	[69]	83.4
No additional data nor model				
Rand. init.	-	224	[19]	77.9
Rand. init.	-	224	[69]	81.8
Supervised	ImNet	224	[69]	81.9
DINO	ImNet	224	[69]	82.8

ViT的自监督ImageNet预训练。在本实验中，我们研究了使用我们的方法对监督式ViT模型进行预训练的影响。在表11中，我们比较了不同预训练初始化或训练过程中额外使用预训练卷积网络引导的监督式ViT模型的性能。第一组模型是在大规模精选数据集（包含3亿张图像）上，分别进行了有监督和无监督的预训练。第二组模型通过从预训练的有监督RegNetY[56]中进行硬知识蒸馏来训练。最后一组模型既不使用额外数据也不依赖其他模型，其初始化方式包括随机初始化或在ImageNet上使用DINO预训练后初始化。与随机初始化相比，DINO预训练带来了+1%的性能提升。这一提升并非源于更长的训练时间，因为若采用有监督预训练替代DINO，性能并无改善。采用自监督预训练缩小了与基于额外数据预训练或从卷积网络蒸馏所得模型之间的性能差距。

Table 12: **Low-shot learning on ImageNet with frozen ViT features.** We train a logistic regression on frozen features (FROZEN). Note that this FROZEN evaluation is performed *without any finetuning nor data augmentation*. We report top-1 accuracy. For reference, we show previously published results that uses finetuning and semi-supervised learning.

Method	Arch	Param.	Top 1 1%	Top 1 10%
Self-supervised pretraining with finetuning				
UDA [75]	RN50	23	–	68.1
SimCLRv2 [13]	RN50	23	57.9	68.4
BYOL [30]	RN50	23	53.2	68.8
SwAV [10]	RN50	23	53.9	70.2
SimCLRv2 [16]	RN50w4	375	63.0	74.4
BYOL [30]	RN200w2	250	71.2	77.7
Semi-supervised methods				
SimCLRv2+KD [13]	RN50	23	60.0	70.5
SwAV+CT [3]	RN50	23	–	70.8
FixMatch [64]	RN50	23	–	71.5
MPL [49]	RN50	23	–	73.9
SimCLRv2+KD [13]	RN152w3+SK	794	76.6	80.9
Frozen self-supervised features				
DINO -FROZEN	ViT-S/16	21	64.5	72.2

Low-shot learning on ImageNet. We evaluate the features obtained with DINO applied on ViT-S on low-shot learning. In Tab. 12, we report the validation accuracy of a logistic regression trained on frozen features (FROZEN) with 1% and 10% labels. The logistic regression is trained with the `cyanure` library [45]. When comparing models with a similar number of parameters and image/sec, we observe that our features are on par with state-of-the-art semi-supervised models. Interestingly, this performance is obtained by training a multi-class logistic regression on *frozen features, without data augmentation nor finetuning*.

B. Methodology Comparison

We compare the performance of different self-supervised frameworks, MoCo-v2 [15], SwAV [10] and BYOL [30] when using convnet or ViT. In Tab. 13, we see that when trained with ResNet-50 (convnet), DINO performs on par with SwAV and BYOL. However, DINO unravels its potential with ViT, outperforming MoCo-v2, SwAV and BYOL by large margins (+4.3% with linear and +6.2% with k-NN evaluations). In the rest of this section, we perform ablations to better understand the performance of DINO applied to ViT. In particular, we provide a detailed comparison with methods that either use a momentum encoder, namely MoCo-v2 and BYOL, and methods that use multi-crop, namely SwAV.

Table 13: **Methodology comparison for DEIT-small and ResNet-50.** We report ImageNet linear and k-NN evaluations validation accuracy after 300 epochs pre-training. All numbers are run by us and match or outperform published results.

	ResNet-50		ViT-small	
Method	Linear	k-NN	Linear	k-NN
MoCo-v2	71.1	62.9	71.6	62.0
BYOL	72.7	65.4	71.4	66.6
SwAV	74.1	65.4	71.8	64.7
DINO	**74.5**	**65.6**	**76.1**	**72.8**

Relation to MoCo-v2 and BYOL. In Tab. 14, we present the impact of ablating components that differ between DINO, MoCo-v2 and BYOL: the choice of loss, the predictor in the student head, the centering operation, the batch normalization in the projection heads, and finally, the multi-crop augmentation. The loss in DINO is a cross-entropy on sharpened softmax outputs (`CE`) while MoCo-v2 uses the InfoNCE contrastive loss (`INCE`) and BYOL a mean squared error on l2-normalized outputs (`MSE`). No sharpening is applied with the `MSE` criterion. Though, DINO surprisingly still works when changing the loss function to `MSE`, but this significantly alters the performance (see rows (1, 2) and (4, 9)). We also observe that adding a predictor has little impact (1, 3). However, in the case of BYOL, the predictor is critical to prevent collapse (7, 8) which is consistent with previous studies [16, 30]. Interestingly, we observe that the teacher output centering avoids collapse without predictor nor batch normalizations in BYOL (7, 9), though with a significant performance drop which can likely be explained by the fact that our centering operator is designed to work in combination with sharpening. Finally, we observe that multi-crop works particularly well with DINO and MoCo-v2, removing it hurts performance by $2-4\%$ (1 versus 4 and, 5 versus 6). Adding multi-crop to BYOL does not work out-of-the-box (7, 10) as detailed in Appendix E and further adaptation may be required.

Relation to SwAV. In Tab. 15, we evaluate the differences between DINO and SwAV: the presence of the momentum encoder and the operation on top of the teacher output. In absence of the momentum, a copy of the student with a stop-gradient is used. We consider three operations on the teacher output: `Centering`, `Sinkhorn-Knopp` or a `Softmax` along the batch axis. The `Softmax` is similar to a single Sinkhorn-Knopp iteration as detailed in the next paragraph. First, these ablations show that using a momentum encoder significantly improves the performance for ViT (3 versus 6, and 2 versus 5). Second, the momentum encoder also avoids collapse when using only centering (row 1). In the absence

表12：使用冻结ViT特征在ImageNet上的少样本学习。我们在冻结特征上训练逻辑回归（FROZEN）。请注意，此FROZEN评估是在*without any fine- tuning nor data augmentation*条件下进行的。我们报告了top-1准确率。作为参考，我们还展示了先前发表的采用微调和半监督学习的结果。

Method	Arch	Param.	Top 1 1%	Top 1 10%
Self-supervised pretraining with finetuning				
UDA [75]	RN50	23	–	68.1
SimCLRv2 [13]	RN50	23	57.9	68.4
BYOL [30]	RN50	23	53.2	68.8
SwAV [10]	RN50	23	53.9	70.2
SimCLRv2 [16]	RN50w4	375	63.0	74.4
BYOL [30]	RN200w2	250	71.2	77.7
Semi-supervised methods				
SimCLRv2+KD [13]	RN50	23	60.0	70.5
SwAV+CT [3]	RN50	23	–	70.8
FixMatch [64]	RN50	23	–	71.5
MPL [49]	RN50	23	–	73.9
SimCLRv2+KD [13]	RN152w3+SK	794	76.6	80.9
Frozen self-supervised features				
DINO -FROZEN	ViT-S/16	21	64.5	72.2

ImageNet上的小样本学习。我们评估了在ViT-S上应用DINO所获得的特征在小样本学习中的表现。表12中，我们报告了使用1%和10%标签训练冻结特征（FROZEN）逻辑回归的验证准确率。该逻辑回归训练采用了cyanure库[45]。在比较参数数量和图像处理速度相近的模型时，我们发现我们的特征与最先进的半监督模型性能相当。值得注意的是，这一性能是通过在*frozen features, without data augmentation nor finetuning*上训练多类逻辑回归实现的。

B. 方法比较

我们比较了不同自监督框架（MoCo-v2 [15]、SwAV [10] 和 BYOL [30]）在使用卷积网络或ViT时的性能表现。如表13所示，当采用ResNet-50（卷积网络）训练时，DINO与SwAV和BYOL表现相当。然而，DINO在ViT架构上展现出其潜力，以显著优势超越MoCo-v2、SwAV和BYOL（线性评估提升+4.3%，k-NN评估提升+6.2%）。在本节剩余部分，我们通过消融实验深入分析DINO应用于ViT的性能表现，特别针对使用动量编码器的方法（即MoCo-v2和BYOL）与采用多裁剪策略的方法（即SwAV）进行了详细对比。

表13：DEIT-small与ResNet-50的方法论对比。我们报告了300轮预训练后ImageNet线性评估及k-NN验证准确率。所有数据均由我们复现，达到或优于已发表结果。

	ResNet-50		ViT-small	
Method	Linear	k-NN	Linear	k-NN
MoCo-v2	71.1	62.9	71.6	62.0
BYOL	72.7	65.4	71.4	66.6
SwAV	74.1	65.4	71.8	64.7
DINO	**74.5**	**65.6**	**76.1**	**72.8**

与MoCo-v2和BYOL的关系。在表14中，我们展示了消融DINO、MoCo-v2和BYOL之间差异组件的影响：损失函数的选择、学生头中的预测器、中心化操作、投影头中的批量归一化，以及多裁剪增强。DINO中的损失函数是对锐化后的softmax输出进行的交叉熵（CE），而MoCo-v2使用的是InfoNCE对比损失（INCE），BYOL则是对l2归一化输出的均方误差（MSE）。使用MSE准则时不应用锐化。然而，令人惊讶的是，当将损失函数改为MSE时，DINO仍然有效，但这显著改变了性能（见行(1,2)和(4,9)）。我们还观察到，添加预测器影响不大（1,3）。但在BYOL的情况下，预测器对于防止崩溃至关重要（7,8），这与之前的研究一致[16,30]。有趣的是，我们发现教师输出的中心化在BYOL中无需预测器或批量归一化也能避免崩溃（7,9），尽管性能显著下降，这可能是因为我们的中心化操作设计为与锐化结合使用。最后，我们注意到多裁剪在DINO和MoCo-v2中效果特别好，移除它会降低性能2−4%（1对4，以及5对6）。直接将多裁剪应用于BYOL并不能开箱即用（7,10），如附录E所述，可能需要进一步的适配。

与SwAV的关系。在表15中，我们评估了DINO与SwAV之间的差异：动量编码器的存在以及对教师输出进行的操作。在没有动量的情况下，使用带有停止梯度的学生模型副本。我们考虑对教师输出进行三种操作：沿批次轴进行中心化处理、Sinkhorn-Knopp算法或Softmax运算。Softmax类似于单次Sinkhorn-Knopp迭代，具体细节将在下一段说明。首先，这些消融实验表明，使用动量编码器显著提升了ViT的性能（比较第3与第6行，以及第2与第5行）。其次，动量编码器还能在仅使用中心化处理时避免模型崩溃（第1行）。当缺乏

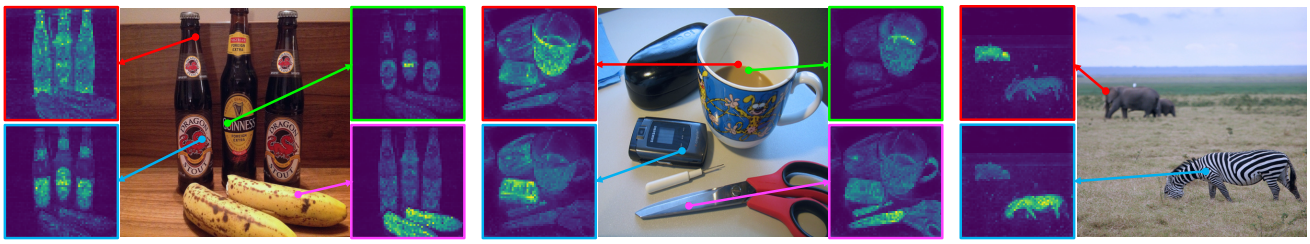


Figure 8: **Self-attention for a set of reference points.** We visualize the self-attention module from the last block of a ViT-S/8 trained with DINO. The network is able to separate objects, though it has been trained with no supervision at all.

Table 14: **Relation to MoCo-v2 and BYOL.** We ablate the components that differ between DINO, MoCo-v2 and BYOL: the loss function (cross-entropy, `CE`, versus InfoNCE, `INCE`, versus mean-square error, `MSE`), the multi-crop training, the centering operator, the batch normalization in the projection heads and the student predictor. Models are run for 300 epochs with ViT-S/16. We report top-1 accuracy on ImageNet linear evaluation.

	Method	Loss	multi-crop	Center.	BN	Pred.	Top-1
1	DINO	`CE`	✓	✓			76.1
2	–	`MSE`	✓	✓			62.4
3	–	`CE`	✓	✓		✓	75.6
4	–	`CE`		✓			72.5
5	MoCov2	`INCE`			✓		71.4
6		`INCE`	✓		✓		73.4
7	BYOL	`MSE`			✓	✓	71.4
8	–	`MSE`			✓		0.1
9	–	`MSE`		✓			52.6
10	–	`MSE`	✓		✓	✓	64.8

Table 15: **Relation to SwAV.** We vary the operation on the teacher output between centering, a softmax applied over the batch dimension and the Sinkhorn-Knopp algorithm. We also ablate the Momentum encoder by replacing it with a hard copy of the student with a stop-gradient as in SwAV. Models are run for 300 epochs with ViT-S/16. We report top-1 accuracy on ImageNet linear evaluation.

	Method	Momentum	Operation	Top-1
1	DINO	✓	`Centering`	76.1
2	–	✓	`Softmax(batch)`	75.8
3	–	✓	`Sinkhorn-Knopp`	76.0
4	–		`Centering`	0.1
5	–		`Softmax(batch)`	72.2
6	SwAV		`Sinkhorn-Knopp`	71.8

of momentum, centering the outputs does not work (4) and more advanced operations are required (5, 6). Overall, these ablations highlight the importance of the momentum encoder, not only for performance but also to stabilize training, removing the need for normalization beyond centering.

Details on the `Softmax(batch)` variant. The iterative Sinkhorn-Knopp algorithm [17] used in SwAV [10] is implemented simply with the following PyTorch style code.

```
# x is n-by-K
# tau is Sinkhorn regularization param
x = exp(x / tau)
for _ in range(num_iters): # 1 iter of Sinkhorn
  # total weight per dimension (or cluster)
  c = sum(x, dim=0, keepdim=True)
  x /= c

  # total weight per sample
  n = sum(x, dim=1, keepdim=True)
  # x sums to 1 for each sample (assignment)
  x /= n
```

When performing a single Sinkhorn iteration (`num_iters=1`) the implementation can be highly simplified into only two lines of code, which is our `softmax(batch)` variant:

```
x = softmax(x / tau, dim=0)
x /= sum(x, dim=1, keepdim=True)
```

We have seen in Tab. 15 that this highly simplified variant of SwAV works competitively with SwAV. Intuitively, the `softmax` operation on the batch axis allows to select for each dimension (or "cluster") its best matches in the batch.

Validating our implementation. We observe in Tab. 13 that our reproduction of BYOL, MoCo-v2, SwAV matches or outperforms the corresponding published numbers with ResNet-50. Indeed, we obtain 72.7% for BYOL while [30] report 72.5% in this 300-epochs setting. We obtain 71.1% for MoCo after 300 epochs of training while [15] report 71.1% after 800 epochs of training. Our improvement compared to the implementation of [15] can be explained by the use of a larger projection head (3-layer, use of batch-normalizations and projection dimension of 256).

Relation to other works. DINO is also related to UIC [14] that use outputs from the previous epoch as hard

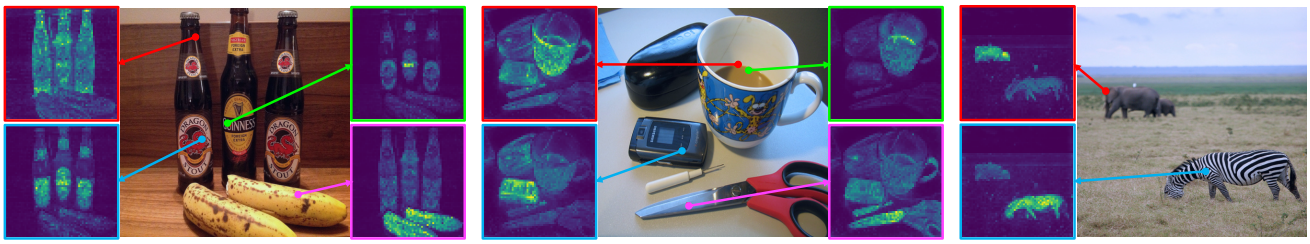


图8：一组参考点的自注意力机制。我们可视化了一个由DINO训练的ViT-S/8最后一层中的自注意力模块。尽管网络完全是在无监督的情况下训练的，但它仍能够区分不同的物体。

表14：与MoCo-v2和BYOL的关系。我们分析了DINO、MoCo-v2和BYOL之间差异的组成部分：损失函数（交叉熵CE、InfoNCE INCE与均方误差MSE）、多裁剪训练、中心化操作、投影头中的批量归一化以及学生预测器。所有模型均采用ViT-S/16架构运行300个周期，并报告ImageNet线性评估的top-1准确率。

	Method	Loss	multi-crop	Center.	BN	Pred.	Top-1
1	DINO	CE	✓	✓			76.1
2	–	MSE	✓	✓			62.4
3	–	CE	✓	✓		✓	75.6
4	–	CE		✓			72.5
5	MoCov2	INCE			✓		71.4
6		INCE	✓		✓		73.4
7	BYOL	MSE			✓	✓	71.4
8	–	MSE			✓		0.1
9	–	MSE		✓			52.6
10	–	MSE	✓		✓	✓	64.8

表15：与SwAV的关系。我们对教师输出的操作进行了变化，包括中心化处理、在批次维度上应用softmax以及Sinkhorn-Knopp算法。同时，我们移除了动量编码器，如SwAV中那样，用带有停止梯度的学生模型硬拷贝替代。所有模型均采用ViT-S/16架构训练300个周期，并报告ImageNet线性评估的top-1准确率。

	Method	Momentum	Operation	Top-1
1	DINO	✓	Centering	76.1
2	–	✓	Softmax(batch)	75.8
3	–	✓	Sinkhorn-Knopp	76.0
4	–		Centering	0.1
5	–		Softmax(batch)	72.2
6	SwAV		Sinkhorn-Knopp	71.8

动量的情况下，仅仅对输出进行中心化处理并不奏效（4），需要更高级的操作（5、6）。总体而言，这些消融实验凸显了动量编码器的重要性，不仅对性能至关重要，还能稳定训练过程。

消除了除中心化外对归一化的需求。

关于Softmax(batch)变体的详细信息。SwAV[10]中采用的迭代Sinkhorn-Knopp算法[17]通过以下PyTorch风格代码简洁实现。

```
# x is n-by-K
# tau is Sinkhorn regularization param
x = exp(x / tau)
for _ in range(num_iters): # 1 iter of Sinkhorn
  # total weight per dimension (or cluster)
  c = sum(x, dim=0, keepdim=True)
  x /= c

  # total weight per sample
  n = sum(x, dim=1, keepdim=True)
  # x sums to 1 for each sample (assignment)
  x /= n
```

当执行单次Sinkhorn迭代（num iters{v*}1）时，实现可以大幅简化为仅两行代码，这就是我们的softmax(batch)变体：

```
x = softmax(x / tau, 维度=0) x /= sum(x, 维度
=1, 保持维度=True)
```

我们在表15中看到，SwAV这种高度简化的变体与SwAV本身表现相当。直观上，沿批次轴进行的softmax操作能够为每个维度（或“簇”）选择批次中与之最匹配的样本。

验证我们的实现。我们在表13中观察到，对于ResNet-50，我们对BYOL、MoCo-v2、SwAV的复现结果与已发表数据相当或更优。具体而言，在300轮训练设置下，我们实现的BYOL达到72.7%，而文献[30]报告的结果为72.5%。MoCo经过300轮训练后，我们获得71.1%的准确率，而文献[15]在800轮训练后报告的结果为71.1%。相较于文献[15]的实现，我们的改进可归因于采用了更大的投影头（3层结构、使用批量归一化及256维投影空间）。

与其他工作的关系。DINO也与UIC[14]相关，后者使用前一时期的输出作为硬

pseudo-labels for "unsupervised classification". However, we use centering to prevent collapse while UIC resorts to balance sampling techniques as in [8]. Our work can be interpreted as a soft UIC variant with momentum teacher.

The concurrent work CsMI [77] also exhibits strong performance with simple k-NN classifiers on ImageNet, even with convnets. As DINO, CsMI combines a momentum network and multi-crop training, which we have seen are both crucial for good k-NN performance in our experiments with ViTs. We believe studying this work would help us identifying more precisely the components important for good k-NN performance and leave this investigation for future work.

C. Projection Head

Similarly to other self-supervised frameworks, using a projection head [12] improves greatly the accuracy of our method. The projection head starts with a n-layer multilayer perceptron (MLP). The hidden layers are 2048d and are with gaussian error linear units (GELU) activations. The last layer of the MLP is without GELU. Then we apply a ℓ_2 normalization and a weight normalized fully connected layer [16, 61] with K dimensions. This design is inspired from the projection head with a "prototype layer" used in SwAV [10]. We do not apply batch normalizations.

BN-free system. Unlike standard convnets, ViT architectures do not use batch normalizations (BN) by default. Therefore, when applying DINO to ViT we do not use any BN also in the projection heads. In this table we evaluate the impact of adding BN in the heads. We observe that adding BN in the projection heads has little impact, showing that BN is not important in our framework. *Overall, when applying DINO to ViT, we do not use any BN anywhere, making the system entirely BN-free.* This is a great advantage of DINO + ViT to work at state-of-the-art performance without requiring any BN. Indeed, training with BN typically slows down trainings considerably, especially when these BN modules need to be synchronized across processes [33, 10, 9, 30].

ViT-S, 100 epochs	heads w/o BN	heads w/ BN
k-NN top-1	69.7	68.6

L2-normalization bottleneck in projection head. We illustrate the design of the projection head with or without l2-normalization bottleneck in Fig. 9. We evaluate the accuracy of DINO models trained with or without l2-normalization bottleneck and we vary the number of linear layers in the projection head. With l2 bottleneck, the total number of linear layers is $n + 1$ (n from the MLP and 1 from the weight normalized layer) while without bottleneck the total number of linear layers is n in the head. In this table, we report ImageNet top-1 k-NN evaluation accuracy after 100 epochs pre-training with ViT-S/16. The output dimensionality K is set to 4096 in this experiment. We observe that DINO training fails without the l2-normalization bottleneck when increasing the depth of the projection head. L2-normalization bottleneck stabilizes the training of DINO with deep projection head. We observe that increasing the depth of the projection head improves accuracy. Our default is to use a total of 4 linear layers: 3 are in the MLP and one is after the l2 bottleneck.

# proj. head linear layers	1	2	3	4
w/ l2-norm bottleneck	–	62.2	68.0	69.3
w/o l2-norm bottleneck	61.6	62.9	0.1	0.1

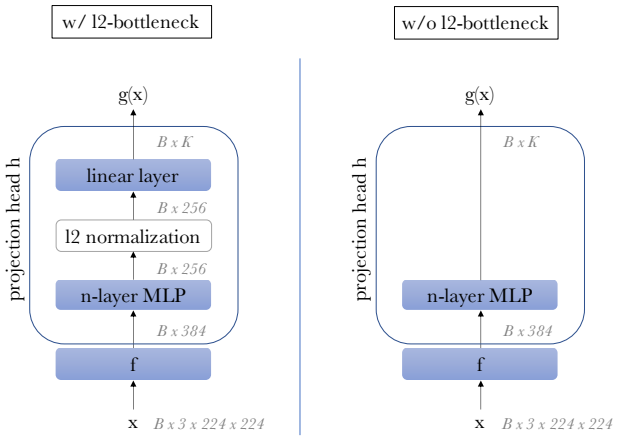


Figure 9: **Projection head design w/ or w/o l2-norm bottleneck.**

Output dimension. In this table, we evaluate the effect of varying the output dimensionality K. We observe that a large output dimensionality improves the performance. We note that the use of l2-normalization bottleneck permits to use a large output dimension with a moderate increase in the total number of parameters. Our default is to use K equals to 65536 and $d = 256$ for the bottleneck.

K	1024	4096	16384	65536	262144
k-NN top-1	67.8	69.3	69.2	69.7	69.1

GELU activations. By default, the activations used in ViT are gaussian error linear units (GELU). Therefore, for consistency within the architecture, we choose to use GELU also in the projection head. We evaluate the effect of using ReLU instead of GELU in this table and observe that changing the activation unit to ReLU has relatively little impact.

ViT-S, 100 epochs	heads w/ GELU	heads w/ ReLU
k-NN top-1	69.7	68.9

伪标签用于“无监督分类”。然而，我们采用中心化防止崩溃，而UIC则如[8]中所述依赖平衡采样技术。我们的工作可被视作一种带有动量教师的软性UIC变体。

同期研究CsMI[77]同样展现出强大的性能，即便使用卷积网络，在ImageNet上仅凭简单的k-NN分类器就能取得优异表现。与DINO类似，CsMI结合了动量网络和多裁剪训练策略——我们在ViT实验中发现这两者对实现良好的k-NN分类性能都至关重要。我们认为深入研究该工作将有助于更精确地识别影响{v*}近邻算法性能的关键组件，故将此项探索留待未来工作。

C. 投影头

与其他自监督框架类似，使用投影头[12]能大幅提升我们方法的准确率。该投影头起始于一个n层的多层感知机（MLP）。其隐藏层为2048维，并采用高斯误差线性单元（GELU）激活函数。MLP的最后一层未使用GELU。随后我们施加ℓ_2归一化处理，并接入一个权重归一化的全连接层[16,61]，其维度为K。这一设计灵感来源于SwAV[10]中带有“原型层”的投影头结构。我们未采用批量归一化操作。

无BN系统。与标准卷积网络不同，ViT架构默认不使用批量归一化（BN）。因此——

ViT-S, 100 epochs	heads w/o BN	heads w/ BN
k-NN top-1	69.7	68.6

因此，在将DINO应用于ViT时，我们同样未在投影头中使用任何BN层。本表中我们评估了在投影头中添加BN的影响。观察到添加BN对结果影响甚微，这表明BN在我们的框架中并不重要。

Overall, when applying DINO to ViT, we do not use any BN anywhere, making the system entirely BN-free. 这是DINO的一大优势+——ViT无需任何BN即可达到最先进的性能。事实上，使用BN训练通常会显著拖慢训练速度，尤其是当这些BN模块需要在多进程间同步时[33, 10, 9, 30]。

投影头中的L2归一化瓶颈。我们在图9中展示了带有或不带L2归一化瓶颈的投影头设计。我们评估了准确率

# proj. head linear layers	1	2	3	4
w/ l2-norm bottleneck	–	62.2	68.0	69.3
w/o l2-norm bottleneck	61.6	62.9	0.1	0.1

采用或不采用l2归一化瓶颈训练的DINO模型，我们改变了投影头中线性层的数量。使用l2瓶颈时，总的

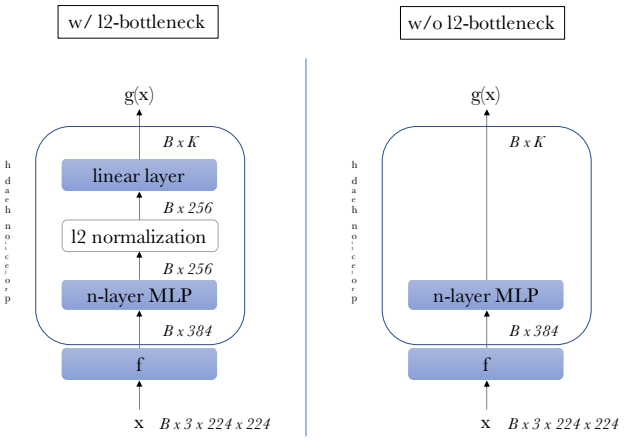


图9：投影头设计（含/不含l2-范数瓶颈）。

线性层的数量为$n+$（其中n来自MLP，1层来自权重归一化层），而在无瓶颈结构时，头部的线性层总数是n。本表中，我们报告了使用ViT-S/16进行100轮预训练后的ImageNet top-1 k-NN评估准确率。本实验中输出维度K设为4096。我们观察到，当增加投影头深度时，若缺少l2归一化瓶颈，DINO训练会失败。L2归一化瓶颈能稳定具有深层投影头的DINO训练。同时发现增加投影头深度可提升准确率。默认配置共使用4个线性层：其中3层位于MLP内，1层置于l2瓶颈之后。

输出维度。在本表中，我们评估了不同输出维度K的影响。观察到

K	1024	4096	16384	65536	262144
k-NN top-1	67.8	69.3	69.2	69.7	69.1

大的输出维度能提升性能。我们注意到，使用l2归一化瓶颈允许在参数总量适度增加的情况下采用较大的输出维度。默认情况下，我们使用K等于65536，瓶颈处的$d=$为256。

GELU激活函数。默认情况下，ViT中使用的激活函数是高斯误差线性单元（GELU）。因此，为了保持一

ViT-S, 100 epochs	heads w/ GELU	heads w/ ReLU
k-NN top-1	69.7	68.9

为了保持架构内的一致性，我们选择在投影头中也使用GELU。我们在本表中评估了使用ReLU替代GELU的效果，观察到将激活单元更改为ReLU的影响相对较小。

D. Additional Ablations

We have detailed in the main paper that the combination of centering and sharpening is important to avoid collapse in DINO. We ablate the hyperparameters for these two operations in the following. We also study the impact of training length and some design choices for the ViT networks.

Online centering. We study the impact of the smoothing parameters in the update rule for the center c used in the output of the teacher network. The convergence is robust

m	0	0.9	0.99	0.999
k-NN top-1	69.1	69.7	69.4	0.1

to a wide range of smoothing, and the model only collapses when the update is too slow, i.e., $m = 0.999$.

Sharpening. We enforce sharp targets by tuning the teacher softmax temperature parameter τ_t. In this table, we observe that a temperature lower than 0.06 is required to avoid collapse. When the temperature is higher than 0.06,

τ_t	0	0.02	0.04	0.06	0.08	$0.04 \rightarrow 0.07$
k-NN top-1	43.9	66.7	69.6	68.7	0.1	69.7

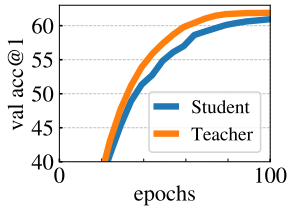
the training loss consistently converges to $ln(K)$. However, we have observed that using higher temperature than 0.06 does not collapse if we start the training from a smaller value and increase it during the first epochs. In practice, we use a linear warm-up for τ_t from 0.04 to 0.07 during the first 30 epochs of training. Finally, note that $\tau \rightarrow 0$ (extreme sharpening) correspond to the `argmax` operation and leads to one-hot hard distributions.

Longer training. We observe in this table that longer training improves the performance of DINO applied to ViT-Small. This observation is consistent with self-supervised results

DINO ViT-S	100-ep	300-ep	800-ep
k-NN top-1	70.9	72.8	74.5

obtained with convolutional architectures [12]. We note that in our experiments with BYOL on ViT-S, training longer than 300 epochs has been leading to worse performance compare our 300 epochs run. For this reason we report BYOL for 300 epochs in Tab. 2 while SwAV, MoCo-v2 and DINO are trained for 800 epochs.

The teacher outperforms the student. We have shown in Fig. 6 that the momentum teacher outperforms the student with ViT and we show in this Figure that it is also the case with ResNet-50. The fact that the teacher continually outperforms the student further encourages the interpretation of DINO as a form of Mean Teacher [65] self-distillation. Indeed, as motivated in Tarvainen et al. [65], weight averaging



usually produces a better model than the individual models from each iteration [51]. By aiming a target obtained with a teacher better than the student, the student's representations improve. Consequently, the teacher also improves since it is built directly from the student weights.

Self-attention maps from supervised versus self-supervised learning. We evaluate the masks obtained by thresholding the self-attention maps to keep 80% of the mass. We compare the Jaccard similarity between the

ViT-S/16 weights	
Random weights	22.0
Supervised	27.3
DINO	45.9
DINO w/o multicrop	45.1
MoCo-v2	46.3
BYOL	47.8
SwAV	46.8

ground truth and these masks on the validation images of PASCAL VOC12 dataset for different ViT-S trained with different frameworks. The properties that self-attention maps from ViT explicitly contain the scene layout and, in particular, object boundaries is observed across different self-supervised methods.

Impact of the number of heads in ViT-S. We study the impact of the number of heads in ViT-S on the accuracy and throughput (images processed per second at inference time on a singe V100 GPU). We find that increasing the number

# heads	dim	dim/head	# params	im/sec	k-NN
6	384	64	21	1007	72.8
8	384	48	21	971	73.1
12	384	32	21	927	73.7
16	384	24	21	860	73.8

of heads improves the performance, at the cost of a slighlty worse throughput. In our paper, all experiments are run with the default model DeiT-S [69], i.e. with 6 heads only.

E. Multi-crop

In this Appendix, we study a core component of DINO: multi-crop training [10].

D. 额外消融实验

我们在主论文中详细阐述了，为避免DINO中的坍塌现象，中心化与锐化的结合至关重要。以下我们将对这两项操作的超参数进行消融分析。同时，我们还将研究训练时长的影响以及ViT网络部分设计选择的效果。

在线中心化。我们研究了教师网络输出中使用的中心c更新规则中平滑参数的影响。收敛性表现稳健

m	0	0.9	0.99	0.999
k-NN top-1	69.1	69.7	69.4	0.1

适用于广泛的平滑处理，只有当更新速度过慢时模型才会崩溃，即$m = 0.999$。

锐化。我们通过调整教师softmax温度参数τ_t来强制实现锐利的目标。在此表中，我们观察到，温度需低于0.06以避免崩溃。当温度高于0.06时，

τ_t	0	0.02	0.04	0.06	0.08	$0.04 \rightarrow 0.07$
k-NN top-1	43.9	66.7	69.6	68.7	0.1	69.7

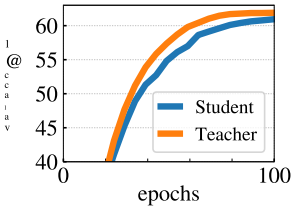
训练损失始终收敛于$ln(K)$。然而，我们观察到，如果从较小值开始训练并在最初几个周期内逐步提高温度，使用高于0.06的温度并不会导致崩溃。实际应用中，我们在前30个训练周期内对τ_t采用从0.04到0.07的线性预热策略。最后需注意，$\tau \rightarrow$为0（极端锐化）对应arg max操作，会导致独热硬分布。

更长的训练时间。我们在此表中观察到，延长训练时间能提升DINO应用于ViT-Small架构时的性能。这一发现与自监督学习的结果一致。

DINO ViT-S	100-ep	300-ep	800-ep
k-NN top-1	70.9	72.8	74.5

通过卷积架构获得[12]。我们注意到，在ViT-S上进行的BYOL实验中，训练超过300个周期会导致性能下降，与我们的300周期运行相比。因此，在表2中我们报告了BYOL的300周期结果，而SwAV、MoCo-v2和DINO则训练了800个周期。

教师模型的表现优于学生模型。我们在图6中展示了采用ViT架构的动量教师模型超越了学生模型，而本图进一步表明，这一优势同样存在于ResNet-50架构中。教师模型持续领先学生模型的现象，更强化了将DINO解读为一种Mean Teacher[65]自蒸馏形式的合理性。事实上，正如Tarvainen等人[65]所论证的，权重平均



通常能产生比每次迭代中的单个模型更好的模型[51]。通过以优于学生的教师所获得的目标为指引，学生的表征能力得以提升。相应地，由于教师模型直接基于学生权重构建，其性能也会随之提高。

监督学习与自监督学习中的自注意力图对比。我们通过设定阈值对自注意力图进行二值化处理，保留80%的质量区域，进而评估所得掩码。在此基础上，我们比较了

ViT-S/16 weights	
Random weights	22.0
Supervised	27.3
DINO	45.9
DINO w/o multicrop	45.1
MoCo-v2	46.3
BYOL	47.8
SwAV	46.8

在PASCAL VOC12数据集的验证图像上，针对不同框架训练的不同ViT-S模型，我们观察到了真实标注与这些掩码之间的关系。研究发现，无论采用何种自监督方法，ViT生成的自注意力图都明确包含了场景布局信息，尤其是物体边界这一特性。

ViT-S中头数的影响。我们研究了ViT-S中头数对准确率和吞吐量（在单块V100 GPU上推理时每秒处理的图像数量）的影响。发现增加头数

# heads	dim	dim/head	# params	im/sec	k-NN
6	384	64	21	1007	72.8
8	384	48	21	971	73.1
12	384	32	21	927	73.7
16	384	24	21	860	73.8

增加注意力头的数量可以提升性能，但会略微降低吞吐量。在我们的论文中，所有实验均采用默认模型DeiT-S[69]进行，即仅使用6个注意力头。

E. 多作物种植

在本附录中，我们研究了DINO的一个核心组件：多裁剪训练[10]。

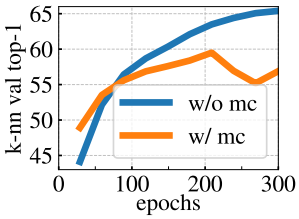
Range of scales in multi-crop. For generating the different views, we use the `RandomResizedCrop` method from `torchvision.transforms` module in PyTorch. We sample two global views with scale range $(s, 1)$ before resizing them to 224^2 and 6 local views with scale sampled in the range $(0.05, s)$ resized to 96^2 pixels. Note that we arbitrarily choose to have non-overlapping scaling range for the global and local views following the original design of SwAV. However, the ranges could definitely be overlapping and experimenting with finer hyperparameters search could lead to a more optimal setting. In this table, we vary the parameter s that controls the range of scales used in multi-crop and find the optimum to be around 0.3 in our experiments. We note that this is higher than the parameter used in SwAV which is of 0.14.

$(0.05, s)$, $(s, 1)$, s:	0.08	0.16	0.24	0.32	0.48
k-NN top-1	65.6	68.0	69.7	69.8	69.5

Multi-crop in different self-supervised frameworks. We compare different recent self-supervised learning frameworks, namely MoCo-v2 [15], BYOL [30] and SwAV [10] with ViT-S/16 architecture. For fair comparisons, all models are pretrained either with two 224^2 crops or with multi-crop [10] training, i.e. two 224^2 crops and six 96^2 crops for each image. We report k-NN and linear probing evaluations after 300 epochs of training. Multi-crop does not benefit all frameworks equally, which has been ignored in benchmarks considering only the two crops setting [16]. The effectiveness of multi-crop depends on the considered framework, which positions multi-crop as a core component of a model and not a simple "add-ons" that will boost any framework the same way. Without multi-crop, DINO has better accuracy than other frameworks, though by a moderate margin (1%). Remarkably, DINO benefits the most from multi-crop training (+3.4% in linear eval). Interestingly, we also observe that the ranking of the frameworks depends on the evaluation protocol considered.

crops	2×224^2		$2 \times 224^2 + 6 \times 96^2$	
eval	k-NN	linear	k-NN	linear
BYOL	66.6	71.4	59.8	64.8
SwAV	60.5	68.5	64.7	71.8
MoCo-v2	62.0	71.6	65.4	73.4
DINO	**67.9**	**72.5**	**72.7**	**75.9**

Training BYOL with multi-crop. When applying multi-crop to BYOL with ViT-S, we observe the transfer performance is higher than the baseline without multi-crop for the first training epochs. However, the transfer performance growth rate is slowing down and declines after a certain amount of training. We have performed learning rate, weight decay, multi-crop parameters sweeps for this setting and systematically observe the same pattern. More precisely, we experiment with $\{1e^{-5}, 3e^{-5}, 1e^{-4}, 3e^{-4}, 1e^{-3}, 3e^{-3}\}$ for learning rate base values, with $\{0.02, 0.05, 0.1\}$ for weight decay and with different number of small crops: $\{2, 4, 6\}$. All our runs are performed with synchronized batch normalizations in the heads. When using a low learning rate, we did not observe the performance break point, i.e. the transfer performance was improving continually during training, but the overall accuracy was low. We have tried a run with multi-crop training on ResNet-50 where we also observe the same behavior. Since integrating multi-crop training to BYOL is not the focus of this study we did not push that direction further. However, we believe this is worth investigating why multi-crop does not combine well with BYOL in our experiments and leave this for future work.



F. Evaluation Protocols

F.1 k-NN classification

Following the setting of Wu *et al*. [73], we evaluate the quality of features with a simple weighted k Nearest Neighbor classifier. We freeze the pretrained model to compute and store the features of the training data of the downstream task. To classify a test image x, we compute its representation and compare it against all stored training features T. The representation of an image is given by the output `[CLS]` token: it has dimensionality $d = 384$ for ViT-S and $d = 768$ for ViT-B. The top k NN (denoted $\mathcal{N}_k$) are used to make a prediction via weighted voting. Specifically, the class c gets a total weight of $\sum_{i \in \mathcal{N}_k} \alpha_i \mathbf{1}_{c_i=c}$, where α_i is a contribution weight. We use $\alpha_i = \exp(T_i x / \tau)$ with τ equals to 0.07 as in [73] which we do not tune. We evaluate different values for k and find that $k = 20$ is consistently leading to the best accuracy across our runs. This evaluation protocol does not require hyperparameter tuning, nor data augmentation and can be run with only one pass over the downstream dataset.

F.2 Linear classification

Following common practice in self-supervised learning, we evaluate the representation quality with a linear classifier. The projection head is removed, and we train a supervised linear classifier on top of frozen features. This linear classifier is trained with SGD and a batch size of 1024 during

多作物尺度范围。为了生成不同的视图，我们采用了PyTorch中torchvision.transforms模块的RandomResizedCrop方法。在({v*} 1)的尺度范围内，我们首先采样两个全局视图

$(0.05, s)$, $(s, 1)$, s:	0.08	0.16	0.24	0.32	0.48
k-NN top-1	65.6	68.0	69.7	69.8	69.5

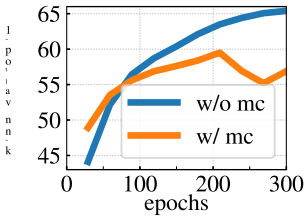
将它们调整为224^2像素，并采用6个局部视图，其尺度在$(0.05, s)$范围内采样后调整为96^2像素。需要注意的是，我们遵循SwAV的原始设计，任意选择了全局视图与局部视图之间不重叠的尺度范围。然而，这些范围完全可以重叠，通过更精细的超参数搜索实验可能会得到更优的设置。在此表中，我们调整了控制多裁剪尺度范围的参数s，并在实验中找到了最佳值约为0.3。我们注意到，这一数值高于SwAV中使用的0.14参数值。

多裁剪在不同自监督框架中的应用。我们比较了近期几种自监督学习框架，即MoCo-v2 [15]、BYOL [30]和SwAV [10]，均采用ViT-S/16架构。为确保公平比较，所有模型

crops	2×224^2		$2 \times 224^2 + 6 \times 96^2$	
eval	k-NN	linear	k-NN	linear
BYOL	66.6	71.4	59.8	64.8
SwAV	60.5	68.5	64.7	71.8
MoCo-v2	62.0	71.6	65.4	73.4
DINO	**67.9**	**72.5**	**72.7**	**75.9**

预训练采用两种方式：一是使用两幅224^2尺寸的裁剪图像，二是采用多裁剪[10]训练策略，即每幅图像包含两幅224^2尺寸裁剪和六幅96^2尺寸裁剪。我们报告了300个训练周期后的k-NN和线性探测评估结果。多裁剪策略并非对所有框架均等有益，这一点在仅考虑双裁剪设置的基准测试[16]中被忽视了。多裁剪的有效性取决于所采用的框架，它应被视为模型的核心组成部分，而非能以相同方式提升所有框架性能的简单“附加组件”。若不采用多裁剪，DINO的准确率仍以微弱优势（1%）领先其他框架。值得注意的是，DINO从多裁剪训练中获益最大（线性评估提升+3.4%）。有趣的是，我们还发现框架的排名会随评估协议的不同而变化。

使用多裁剪训练BYOL。当将多裁剪技术应用于采用ViT-S架构的BYOL时，我们观察到在初始训练周期中，迁移性能高于未使用多裁剪的基线。然而，迁移性能的增长速率逐渐放缓，并在达到一定阶段后开始下降。



训练量。我们针对这一设置进行了学习率、权重衰减和多裁剪参数的全面扫描，并系统性地观察到了相同的模式。具体而言，我们尝试了学习率基值为{$1e^{-5}$、$3e^{-5}$、$1e^{-4}$、$3e^{-4}$、$1e^{-3}$、$3e^{-3}$}，权重衰减为{0.02、0.05、0.1}，以及不同数量的小裁剪：{2、4、6}。所有实验均在头部使用同步批量归一化进行。当采用较低学习率时，我们未观察到性能拐点，即迁移性能在训练过程中持续提升，但整体准确率较低。我们还在ResNet-50上尝试了多裁剪训练，同样观察到了相同现象。由于将多裁剪训练整合到BYOL并非本研究重点，我们未对此方向深入探索。但我们认为，值得探究为何多裁剪在实验中与BYOL配合不佳，这将留待未来工作解决。

F. 评估协议

F.1 k-NN分类

遵循Wu *et al.*[73]的设置，我们采用一个简单的加权k最近邻分类器来评估特征质量。冻结预训练模型以计算并存储下游任务训练数据的特征。对于测试图像x的分类，我们计算其表征并与所有存储的训练特征T进行比较。图像的表征由输出[CLS]标记给出：ViT-S的维度为d =384，ViT-B则为d =768。通过加权投票进行预测时，选取前k个最近邻（记为$\mathcal{N}_k$）。具体而言，类别c获得的总权重为$\sum_{i \in \mathcal{N}_k} \alpha_i \mathbf{1}_{c_i=c}$，其中$\alpha_i$为贡献权重。我们使用$\alpha_i$ =exp($T_i x/\tau$)，其中τ取固定值0.07（如[73]所述，未作调整）。通过评估不同k值，发现k =20在各次实验中始终能取得最佳准确率。该评估方案无需超参数调优或数据增强，且仅需对下游数据集进行一次遍历即可完成。

F.2 线性分类

遵循自监督学习的常见做法，我们通过线性分类器评估表示质量。移除投影头后，在冻结特征之上训练一个有监督的线性分类器。该线性分类器使用SGD优化器进行训练，批量大小为1024。

100 epochs on ImageNet. We do not apply weight decay. For each model, we sweep the learning rate value. During training, we apply only random resizes crops (with default parameters from PyTorch `RandomResizedCrop`) and horizontal flips as data augmentation. We report central-crop top-1 accuracy. When evaluating convnets, the common practice is to perform global average pooling on the final feature map before the linear classifier. In the following, we describe how we adapt this design when evaluating ViTs.

ViT-S representations for linear eval. Following the *feature-based* evaluations in BERT [18], we concatenate the `[CLS]` tokens from the l last layers. We experiment

concatenate l last layers	1	2	4	6
representation dim	384	768	1536	2304
ViT-S/16 linear eval	76.1	76.6	77.0	77.0

with the concatenation of a different number l of layers and similarly to [18] we find $l = 4$ to be optimal.

ViT-B representations for linear eval. With ViT-B we did not find that concatenating the representations from the last l layers to provide any performance gain, and consider the final layer only ($l = 1$). In this setting, we adapt the

pooling strategy	`[CLS]` tok. only	concatenate `[CLS]` tok. and avgpooled patch tok.
representation dim	768	1536
ViT-B/16 linear eval	78.0	78.2

pipeline used in convnets with global average pooling on the output patch tokens. We concatenate these pooled features to the final `[CLS]` output token.

G. Self-Attention Visualizations

We provide more self-attention visualizations in Fig. 8 and in Fig. 10. The images are randomly selected from COCO validation set, and are not used during training of DINO. In Fig. 8, we show the self-attention from the last layer of a DINO ViT-S/8 for several reference points.

H. Class Representation

As a final visualization, we propose to look at the distribution of ImageNet concepts in the feature space from DINO. We represent each ImageNet class with the average feature vector for its validation images. We reduce the dimension of these features to 30 with PCA, and run t-SNE with a perplexity of 20, a learning rate of 200 for 5000 iterations. We present the resulting class embeddings in Fig. 11. Our model recovers structures between classes: similar animal species are grouped together, forming coherent clusters of birds (top) or dogs, and especially terriers (far right).

在ImageNet上训练100个周期。我们不应用权重衰减。对于每个模型，我们都会对学习率值进行扫描。训练期间，仅采用随机尺寸裁剪（使用PyTorch RandomResizedCrop的默认参数）和水平翻转作为数据增强手段。我们报告中心裁剪的top-1准确率。评估卷积网络时，常规做法是在线性分类器前对最终特征图执行全局平均池化。下文将说明我们在评估ViTs时如何调整这一设计。

ViT-S表示的线性评估。遵循BERT[18]中的*feature-based*评估方法，我们将l最后几层的[CLS]标记进行拼接。我们实验

concatenate l last layers	1	2	4	6
representation dim	384	768	1536	2304
ViT-S/16 linear eval	76.1	76.6	77.0	77.0

通过连接不同数量的层l，与[18]类似，我们发现l =4是最优的。

ViT-B的线性评估表示。在使用ViT-B时，我们发现将最后l层的表示进行拼接并未带来任何性能提升，因此仅考虑最后一层（l = 1）。在此设置下，我们调整了

pooling strategy	[CLS] tok. only	concatenate [CLS] tok. and avgpooled patch tok.
representation dim	768	1536
ViT-B/16 linear eval	78.0	78.2

在卷积网络中使用的流程，对输出补丁标记进行全局平均池化。我们将这些池化后的特征与最终的[CLS]输出标记连接起来。

G. 自注意力可视化

我们在图8和图10中提供了更多的自注意力可视化示例。这些图像是从COCO验证集中随机选取的，并未在DINO的训练过程中使用。图8展示了DINO ViT-S/8最后一层针对若干参考点的自注意力分布情况。

H. 类别表示

作为最后的可视化展示，我们建议观察DINO特征空间中ImageNet概念的分布情况。我们通过计算每个ImageNet类别验证图像的平均特征向量来代表该类别。首先使用PCA将这些特征降维至30维，然后运行t-SNE算法（困惑度设为20，学习率为200，迭代5000次）。图11展示了最终得到的类别嵌入结果。我们的模型成功还原了类别间的结构关系：相似的动物物种被归为一组，形成了连贯的鸟类集群（顶部）或犬类集群，尤其是梗犬类（最右侧）。

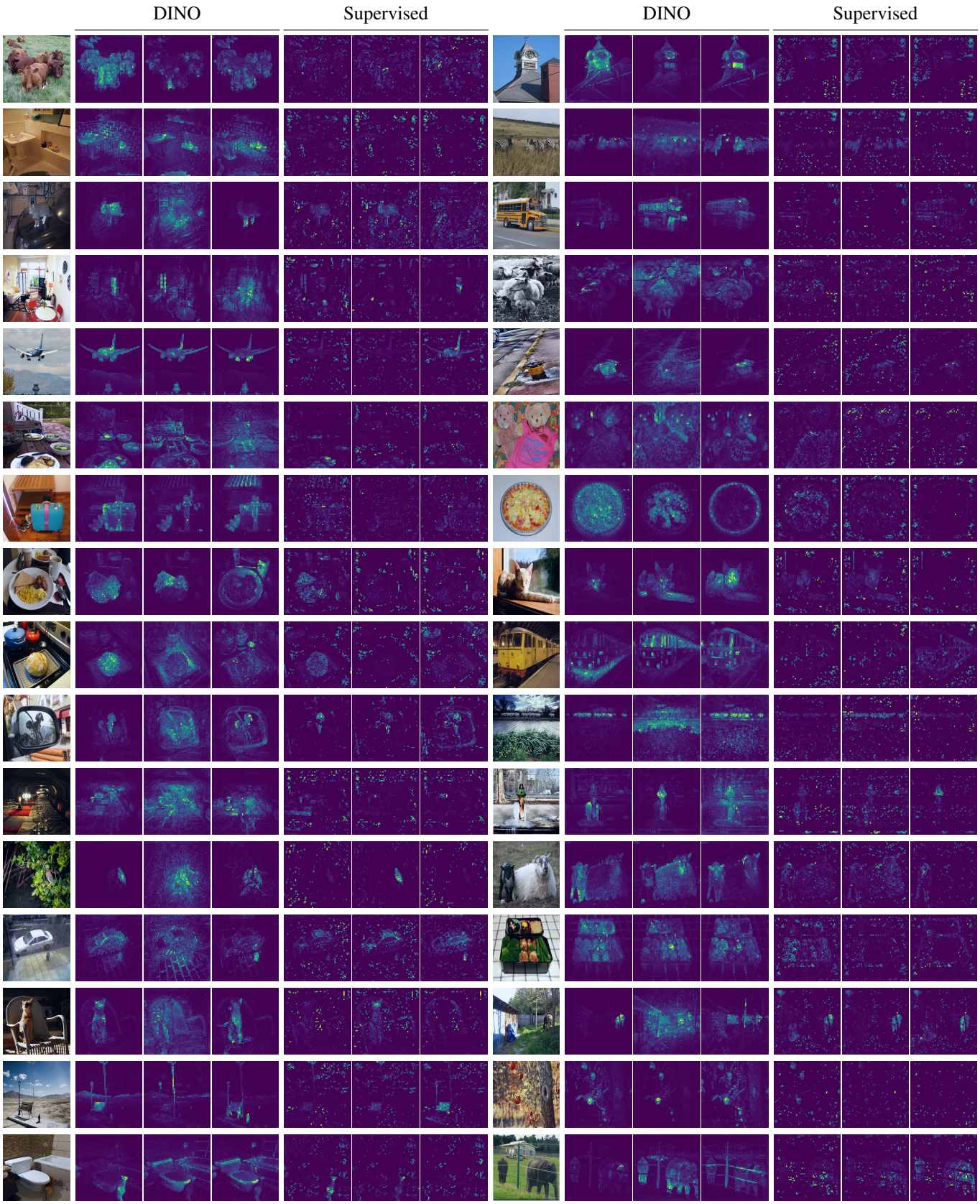


Figure 10: **Self-attention heads from the last layer.** We look at the attention map when using the `[CLS]` token as a query for the different heads in the last layer. Note that the `[CLS]` token is not attached to any label or supervision.

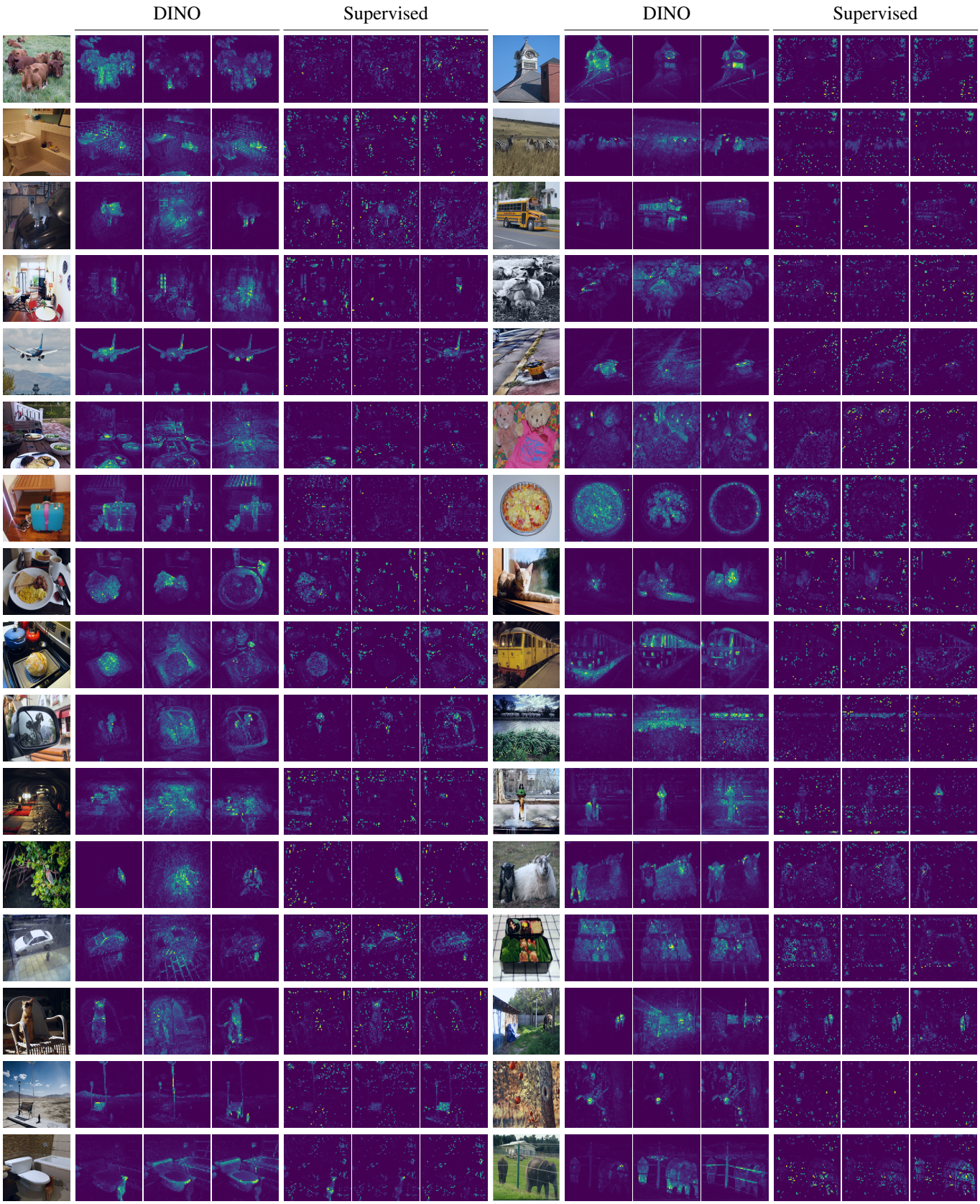


图10：最后一层的自注意力头。我们观察了当使用[CLS]标记作为最后一层不同注意力头的查询时的注意力分布图。请注意，[CLS]标记并未与任何标签或监督信号相关联。

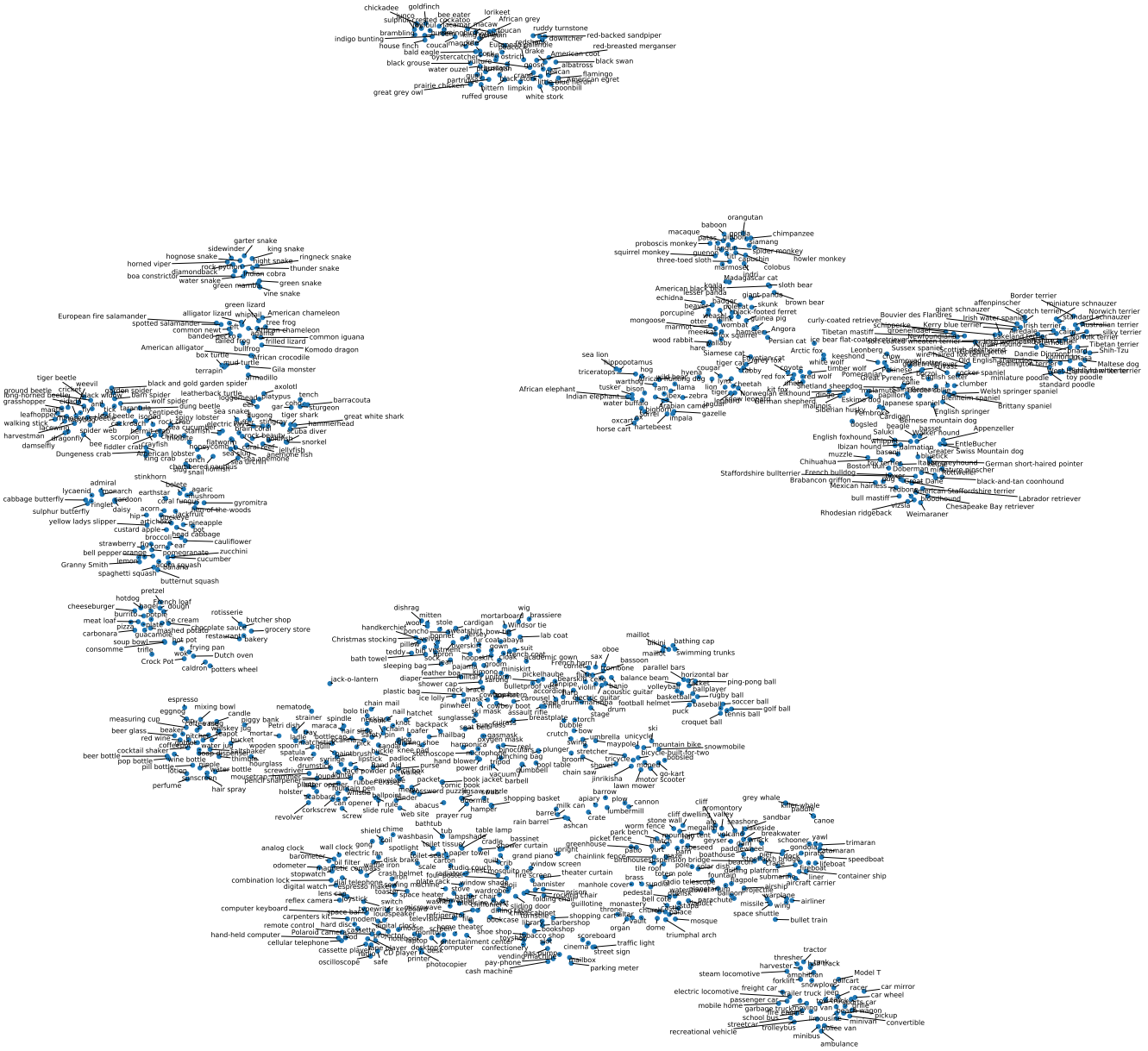


Figure 11: t-SNE visualization of ImageNet classes as represented using DINO. For each class, we obtain the embedding by taking the average feature for all images of that class in the validation set.

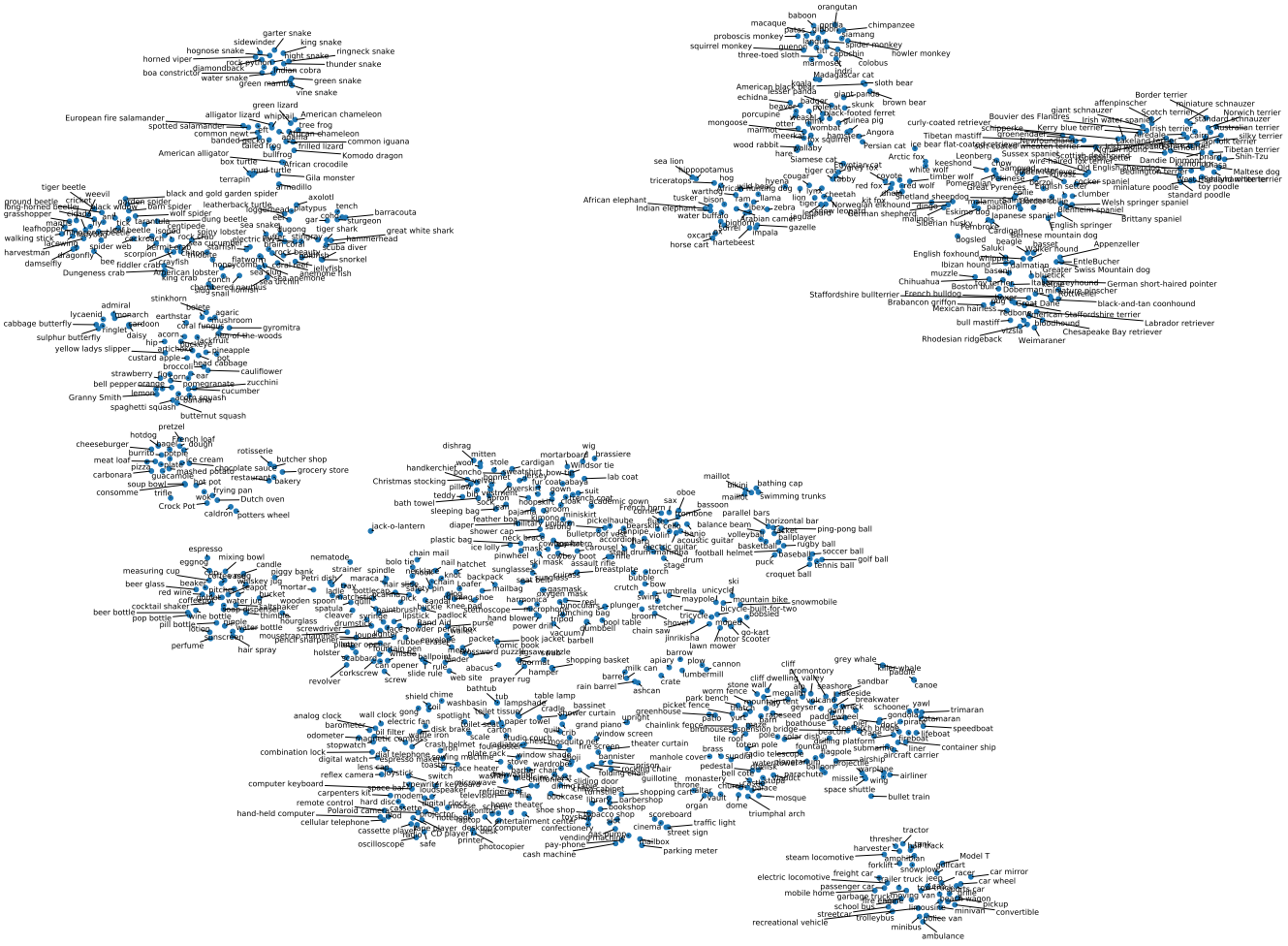


图11：使用DINO表示的ImageNet类别的t-SNE可视化。对于每个类别，我们通过计算验证集中该类别所有图像的平均特征来获得嵌入向量。