Exploring Simple Siamese Representation Learning

探索简单的孪生表示学习

Xinlei Chen Kaiming He
Xinlei Chen Kaiming He
Facebook AI Research (FAIR)
Facebook AI Research (FAIR)

Abstract

摘要

Siamese networks have become a common structure in various recent models for unsupervised visual representation learning. These models maximize the similarity between two augmentations of one image, subject to certain conditions for avoiding collapsing solutions. In this paper, we report surprising empirical results that simple Siamese networks can learn meaningful representations even using none of the following: (i) negative sample pairs, (ii) large batches, (iii) momentum encoders. Our experiments show that collapsing solutions do exist for the loss and structure, but a stop-gradient operation plays an essential role in preventing collapsing. We provide a hypothesis on the implication of stop-gradient, and further show proof-of-concept experiments verifying it. Our "SimSiam" method achieves competitive results on ImageNet and downstream tasks. We hope this simple baseline will motivate people to rethink the roles of Siamese architectures for unsupervised representation learning. Code will be made available.

孪生网络已成为最近多种无监督视觉表示学习模型中的一种常见结构。这些模型在一定条件下最大化一幅图像的两个增强版本之间的相似性，以避免崩溃解。在本文中，我们报告了令人惊讶的实证结果，表明简单的孪生网络即使在不使用以下任一项的情况下也能学习到有意义的表示:(i) 负样本对，(ii) 大批量，(iii) 动量编码器。我们的实验表明，损失和结构确实存在崩溃解，但停止梯度操作在防止崩溃中起着至关重要的作用。我们提供了关于停止梯度的影响的假设，并进一步展示了验证其的概念验证实验。我们的 "SimSiam" 方法在 ImageNet 和下游任务上取得了竞争性的结果。我们希望这个简单的基线能够激励人们重新思考孪生架构在无监督表示学习中的作用。代码将会公开。

1. Introduction

1. 引言

Recently there has been steady progress in un-/self-supervised representation learning, with encouraging results on multiple visual tasks (e.g., [2, 17, 8, 15, 7]). Despite various original motivations, these methods generally involve certain forms of Siamese networks [4]. Siamese networks are weight-sharing neural networks applied on two or more inputs. They are natural tools for comparing (including but not limited to "contrasting") entities. Recent methods define the inputs as two augmentations of one image, and maximize the similarity subject to different conditions.

最近，在无监督/自监督表示学习方面取得了稳步进展，在多个视觉任务上取得了令人鼓舞的结果 (例如，[2, 17, 8, 15, 7])。尽管有各种原始动机，这些方法通常涉及某种形式的孪生网络 [4]。孪生网络是应用于两个或多个输入的权重共享神经网络。它们是比较 (包括但不限于 "对比") 实体的自然工具。最近的方法将输入定义为一幅图像的两个增强版本，并在不同条件下最大化相似性。

An undesired trivial solution to Siamese networks is all outputs "collapsing" to a constant. There have been several general strategies for preventing Siamese networks from collapsing. Contrastive learning [16], e.g., instantiated in SimCLR [8], repulses different images (negative pairs) while attracting the same image's two views (positive pairs). The negative pairs preclude constant outputs from the solution space. Clustering [5] is another way of avoiding constant output, and SwAV [7] incorporates online clustering into Siamese networks. Beyond contrastive learning and clustering, BYOL [15] relies only on positive pairs but it does not collapse in case a momentum encoder is used.

对于孪生网络，一个不希望出现的简单解决方案是所有输出 "崩溃" 到一个常数。为了防止孪生网络崩溃，已经提出了几种通用策略。例如，对比学习 [16]，如在 SimCLR [8] 中实例化的那样，能够排斥不同的图像 (负样本对)，同时吸引同一图像的两个视图 (正样本对)。负样本对阻止了解空间中的常数输出。

聚类 [5] 是避免常数输出的另一种方法，而 SwAV [7] 将在线聚类纳入孪生网络。除了对比学习和聚类，BYOL [15] 仅依赖正样本对，但在使用动量编码器的情况下不会崩溃。

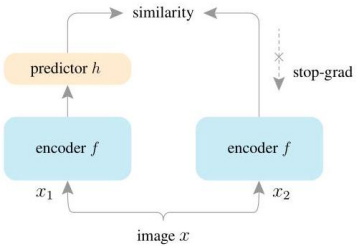


Figure 1. SimSiam architecture. Two augmented views of one image are processed by the same encoder network f (a backbone plus a projection MLP). Then a prediction MLP h is applied on one side, and a stop-gradient operation is applied on the other side. The model maximizes the similarity between both sides. It uses neither negative pairs nor a momentum encoder.

图 1. SimSiam 架构。一个图像的两个增强视图由同一个编码器网络 f (一个主干加上一个投影 MLP) 处理。然后在一侧应用预测 MLP h ，而在另一侧应用停止梯度操作。该模型最大化两个侧面的相似性。它既不使用负样本对，也不使用动量编码器。

In this paper, we report that simple Siamese networks can work surprisingly well with none of the above strategies for preventing collapsing. Our model directly maximizes the similarity of one image's two views, using neither negative pairs nor a momentum encoder. It works with typical batch sizes and does not rely on large-batch training. We illustrate this "SimSiam" method in Figure 1.

在本文中，我们报告简单的孪生网络在没有上述防止崩溃的策略的情况下也能出奇地有效。我们的模型直接最大化一个图像的两个视图之间的相似性，既不使用负样本对，也不使用动量编码器。它适用于典型的批量大小，并且不依赖于大批量训练。我们在图 1 中说明了这种 "SimSiam" 方法。

Thanks to the conceptual simplicity, SimSiam can serve as a hub that relates several existing methods. In a nutshell, our method can be thought of as "BYOL without the momentum encoder". Unlike BYOL but like SimCLR and SwAV, our method directly shares the weights between the two branches, so it can also be thought of as "SimCLR without negative pairs", and "SwAV without online clustering". Interestingly, SimSiam is related to each method by removing one of its core components. Even so, SimSiam does not cause collapsing and can perform competitively.

由于概念上的简单性，SimSiam 可以作为一个枢纽，将几种现有方法联系起来。简而言之，我们的方法可以被视为 "没有动量编码器的 BYOL"。与 BYOL 不同，但与 SimCLR 和 SwAV 类似，我们的方法在两个分支之间直接共享权重，因此也可以被视为 "没有负对的 SimCLR" 和 "没有在线聚类的 SwAV"。有趣的是，SimSiam 通过去除每种方法的核心组件之一与每种方法相关联。即便如此，SimSiam 并不会导致崩溃，并且能够具有竞争力地执行。

We empirically show that collapsing solutions do exist, but a stop-gradient operation (Figure 1) is critical to prevent such solutions. The importance of stop-gradient suggests that there should be a different underlying optimization problem that is being solved. We hypothesize that there are implicitly two sets of variables, and SimSiam behaves like alternating between optimizing each set. We provide proof-of-concept experiments to verify this hypothesis.

我们通过实验证明崩溃解决方案确实存在，但停止梯度操作 (图 1) 对防止此类解决方案至关重要。停止梯度的重要性表明应该存在一个不同的基础优化问题正在被解决。我们假设隐含地有两组变量，SimSiam 的行为类似于在优化每组之间交替进行。我们提供了概念验证实验来验证这一假设。

Our simple baseline suggests that the Siamese architectures can be an essential reason for the common success of the related methods. Siamese networks can naturally introduce inductive biases for modeling invariance, as by definition "invariance" means that two observations of the same concept should produce the same outputs. Analogous to convolutions [25], which is a successful inductive bias via weight-sharing for modeling translation-invariance, the weight-sharing Siamese networks can model invariance w.r.t. more complicated transformations (e.g., augmentations). We hope our exploration will motivate people to rethink the fundamental roles of Siamese architectures for unsupervised representation learning.

我们的简单基线表明，西梅网络架构可能是相关方法普遍成功的一个重要原因。西梅网络可以自然地引入归纳偏差以建模不变性，因为根据定义，"不变性" 意味着同一概念的两个观察应产生相同的输出。类似于卷积 [25]，这是一种通过权重共享建模平移不变性的成功归纳偏差，权重共享的西梅网络可以针对更复杂的变换 (例如，增强) 建模不变性。我们希望我们的探索能够激励人们重新思考西梅架构在无监督表示学习中的基本作用。

2. Related Work

2. 相关工作

Siamese networks. Siamese networks [4] are general models for comparing entities. Their applications include signature [4] and face [34] verification, tracking [3], one-shot learning [23], and others. In conventional use cases, the inputs to Siamese networks are from different images, and the comparability is determined by supervision.

暗示网络。暗示网络 [4] 是用于比较实体的通用模型。它们的应用包括签名 [4] 和人脸 [34] 验证、跟踪 [3]、一次性学习 [23] 等。在传统用例中，暗示网络的输入来自不同的图像，比较性由监督决定。

Contrastive learning. The core idea of contrastive learning [16] is to attract the positive sample pairs and repulse the negative sample pairs. This methodology has been recently popularized for un-/self-supervised representation learning [36, 30, 20, 37, 21, 2, 35, 17, 29, 8, 9] . Simple and effective instantiations of contrastive learning have been developed using Siamese networks [37, 2, 17, 8, 9] .

对比学习。对比学习 [16] 的核心思想是吸引正样本对并排斥负样本对。这种方法最近在无监督/自监督表示学习 [36, 30, 20, 37, 21, 2, 35, 17, 29, 8, 9] 中得到了广泛应用。使用暗示网络 [37, 2, 17, 8, 9] 开发了简单而有效的对比学习实例。

In practice, contrastive learning methods benefit from a large number of negative samples [36, 35, 17, 8] . These samples can be maintained in a memory bank [36]. In a Siamese network, MoCo [17] maintains a queue of negative samples and turns one branch into a momentum encoder to improve consistency of the queue. SimCLR [8] directly uses negative samples coexisting in the current batch, and it requires a large batch size to work well.

在实践中，对比学习方法受益于大量的负样本 [36, 35, 17, 8] 。这些样本可以保存在一个记忆库中 [36]。在暗示网络中，MoCo [17] 维护一个负样本队列，并将一个分支转变为动量编码器，以提高队列的一致性。SimCLR [8] 直接使用当前批次中共存的负样本，并且需要较大的批量大小才能良好工作。

Clustering. Another category of methods for unsupervised representation learning are based on clustering [5, 6, 1, 7] . They alternate between clustering the representations and learning to predict the cluster assignment. SwAV [7] incorporates clustering into a Siamese network, by computing the assignment from one view and predicting it from another view. SwAV performs online clustering under a balanced partition constraint for each batch, which is solved by the Sinkhorn-Knopp transform [10].

聚类。另一类用于无监督表示学习的方法基于聚类 [5, 6, 1, 7] 。它们在聚类表示和学习预测聚类分配之间交替进行。SwAV [7] 将聚类纳入暗示网络，通过从一个视图计算分配并从另一个视图进行预测。SwAV 在每个批次下执行在线聚类，遵循平衡划分约束，该约束通过 Sinkhorn-Knopp 变换 [10] 解决。

While clustering-based methods do not define negative exemplars, the cluster centers can play as negative prototypes. Like contrastive learning, clustering-based methods require either a memory bank [5, 6, 1] , large batches [7] , or a queue [7] to provide enough samples for clustering.

虽然基于聚类的方法不定义负样本，但聚类中心可以作为负原型。与对比学习类似，基于聚类的方法需要记忆库 [5, 6, 1] 、大批量 [7] 或队列 [7] 来提供足够的样本进行聚类。

BYOL. BYOL [15] directly predicts the output of one view from another view. It is a Siamese network in which one branch is a momentum encoder. [1] It is hypothesized in [15] that the momentum encoder is important for BYOL to avoid collapsing, and it reports failure results if removing the momentum encoder (0.3% accuracy, Table 5 in [15]). [2] Our empirical study challenges the necessity of the momentum encoder for preventing collapsing. We discover that the stop-gradient operation is critical. This discovery can be obscured with the usage of a momentum encoder, which is always accompanied with stop-gradient (as it is not updated by its parameters' gradients). While the moving-average behavior may improve accuracy with an appropriate momentum coefficient, our experiments show that it is not directly related to preventing collapsing.

BYOL。BYOL [15] 直接预测一个视图的输出来自另一个视图。它是一个孪生网络，其中一个分支是动量编码器。[1] 在 [15] 中假设动量编码器对于 BYOL 避免崩溃是重要的，并且如果去除动量编码器则报告失败结果 (0.3% 准确率，见 [15] 的表 5)。[2] 我们的实证研究挑战了动量编码器在防止崩溃中的必要性。我们发现停止梯度操作是关键。这一发现可能会因使用动量编码器而被掩盖，因为动量编码器总是伴随停止梯度 (因为它不受其参数梯度的更新)。虽然移动平均行为可能通过适当的动量系数提高准确性，但我们的实验表明它与防止崩溃并没有直接关系。

Algorithm 1 SimSiam Pseudocode, PyTorch-like

算法 1 SimSiam 伪代码，类似 PyTorch

f: backbone + projection mlp

f: 主干 + 投影 MLP

h: prediction mlp

h: 预测 MLP

for x in loader: # load a minibatch x with n samples
对于 x 在加载器中: # 加载一个包含 n 个样本的小批量 x
x1, x2 = aug(x), aug(x) # random augmentation
x1, x2 = aug(x), aug(x) # 随机增强
z1, z2 = f(x1), f(x2) # projections, n-by-d
z1, z2 = f(x1), f(x2) # 投影，n-by-d
p1, p2 = h(z1), h(z2) # predictions, n-by-d
p1, p2 = h(z1), h(z2) # 预测，n-by-d
L = D (p1, z2) /2 + D (p2, z1) /2# loss
L = D (p1, z2) /2 + D (p2, z1) /2# 损失
L.backward( ) # back-propagate
L.backward( ) # 反向传播
update(f, h) # SGD update
update(f, h) # SGD 更新
def D(p, z): # negative cosine similarity
def D(p, z): # 负余弦相似度
z = z.detach( ) # stop gradient
z = z.detach( ) # 停止梯度
p = normalize(p, dim=1) # 12-normalize
p = normalize(p, dim=1) # 12-归一化
z = normalize (z , dim=1) # 12-normalize)
z = 归一化 (z , dim=1) # 12-normalize)
return - (p*z).sum(dim=1).mean( )
return - (p*z).sum(dim=1).mean( )

3. Method

3. 方法

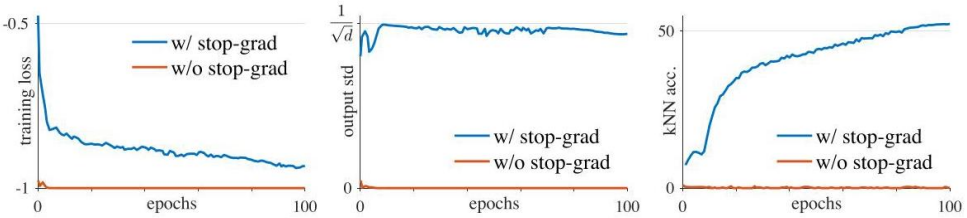
Our architecture (Figure 1) takes as input two randomly augmented views x_1 and x_2 from an image x . The two views are processed by an encoder network f consisting of a backbone (e.g., ResNet [19]) and a projection MLP head [8]. The encoder f shares weights between the two views. A prediction MLP head [15], denoted as h , transforms the output of one view and matches it to the other view. Denoting the two output vectors as $p_1 \triangleq h\left(f\left(x_1\right)\right)$ and $z_2 \triangleq f\left(x_2\right)$, we minimize their negative cosine similarity:

我们的架构 (图 1) 以两个随机增强的视图 x_1 和 x_2 作为输入，来自于一张图像 x 。这两个视图通过一个编码器网络 f 进行处理，该网络由一个主干 (例如，ResNet [19]) 和一个投影 MLP 头 [8] 组成。编码器 f 在两个视图之间共享权重。一个预测 MLP 头 [15]，记作 h ，将一个视图的输出转换并匹配到另一个视图。将两个输出向量记作 $p_1 \triangleq h\left(f\left(x_1\right)\right)$ 和 $z_2 \triangleq f\left(x_2\right)$ ，我们最小化它们的负余弦相似度:

$$\mathcal{D}\left(p_1, z_2\right)=-\frac{p_1}{\left\|p_1\right\|_2} \cdot \frac{z_2}{\left\|z_2\right\|_2}, \tag{1}$$

where $\|\cdot\|_2$ is ℓ_2 -norm. This is equivalent to the mean squared error of ℓ_2 -normalized vectors [15], up to a scale

其中 $\|\cdot\|_2$ 是 ℓ_2 -范数。这等价于 ℓ_2 -归一化向量的均方误差 [15]，只差一个比例因子。



	acc. (%)
w/ stop-grad	67.7±0.1
w/o stop-grad	0.1

Figure 2. SimSiam with vs . without stop-gradient. Left plot: training loss. Without stop-gradient it degenerates immediately. Middle plot: the per-channel std of the ℓ_2 -normalized output, plotted as the averaged std over all channels. Right plot: validation accuracy of a kNN classifier [36] as a monitor of progress. Table: ImageNet linear evaluation ("w/ stop-grad" is mean±std over 5 trials).

图 2. SimSiam 与 vs . 无停止梯度。左图: 训练损失。没有停止梯度时，它立即退化。中间图: 经过 ℓ_2 归一化输出的每通道标准差，绘制为所有通道的平均标准差。右图:kNN 分类器 [36] 的验证准确率，作为进展的监测。表:ImageNet 线性评估 ("w/ stop-grad" 是 5 次试验的均值 ± 标准差)。

of 2. Following [15], we define a symmetrized loss as:

根据 [15]，我们定义一个对称化损失为:

$$\mathcal{L} = \frac{1}{2}\mathcal{D}\left(p_1, z_2\right) + \frac{1}{2}\mathcal{D}\left(p_2, z_1\right). \tag{2}$$

This is defined for each image, and the total loss is averaged over all images. Its minimum possible value is -1 .

这是针对每个图像定义的，总损失在所有图像上取平均。其最小可能值为 -1。

An important component for our method to work is a stop-gradient (stopgrad) operation (Figure 1). We implement it by modifying (1) as:

我们方法的一个重要组成部分是停止梯度 (stopgrad) 操作 (图 1)。我们通过将 (1) 修改为:

$$\mathcal{D}\left(p_1, \text{stopgrad}\left(z_2\right)\right) \ . \tag{3}$$

This means that z_2 is treated as a constant in this term. Similarly, the form in (2) is implemented as:

这意味着 z_2 在此项中被视为常量。类似地，(2) 中的形式实现为:

$$\mathcal{L} = \frac{1}{2}\mathcal{D}\left(p_1, \text{stopgrad}\left(z_2\right)\right) + \frac{1}{2}\mathcal{D}\left(p_2, \text{stopgrad}\left(z_1\right)\right).$$

(4)

Here the encoder on x_2 receives no gradient from z_2 in the first term, but it receives gradients from p_2 in the second term (and vice versa for x_1).

在这里，编码器在 x_2 上在第一项中没有来自 z_2 的梯度，但在第二项中接收来自 p_2 的梯度 (反之亦然，对于 x_1)。

The pseudo-code of SimSiam is in Algorithm 1.

SimSiam 的伪代码在算法 1 中。

Baseline settings. Unless specified, our explorations use the following settings for unsupervised pre-training:

基线设置。除非另有说明，我们的探索使用以下无监督预训练设置:

[1] MoCo [17] and BYOL [15] do not directly share the weights between the two branches, though in theory the momentum encoder should converge to the same status as the trainable encoder. We view these models as Siamese networks with "indirect" weight-sharing.

[1] MoCo [17] 和 BYOL [15] 并不直接在两个分支之间共享权重，尽管理论上动量编码器应该收敛到与可训练编码器相同的状态。我们将这些模型视为具有 "间接" 权重共享的孪生网络。

[2] In BYOL' s arXiv v3 update, it reports 66.9% accuracy with 300-epoch pre-training when removing the momentum encoder and increasing the predictor' s learning rate by 10× . Our work was done concurrently with this arXiv update. Our work studies this topic from different perspectives, with better results achieved.

[2] 在 BYOL 的 arXiv v3 更新中，它报告了在去除动量编码器并将预测器的学习率提高 10× 后，300 轮预训练的 66.9% 准确率。我们的工作与此 arXiv 更新同时进行。我们的工作从不同的角度研究了这个主题，并取得了更好的结果。

- Optimizer. We use SGD for pre-training. Our method does not require a large-batch optimizer such as LARS [38] (unlike [8, 15, 7]). We use a learning rate of $lr\times$ BatchSize/256 (linear scaling [14]), with a base $lr = 0.05$. The learning rate has a cosine decay schedule [27, 8] . The weight decay is 0.0001 and the SGD momentum is 0.9 .

- 优化器。我们使用 SGD 进行预训练。我们的方法不需要像 LARS [38] 这样的较大批量优化器 (与 [8, 15, 7] 不同)。我们使用的学习率为 $lr\times$ BatchSize/256(线性缩放 [14])，基础 $lr =$ 为 0.05。学习率具有余弦衰减计划 [27, 8] 。权重衰减为 0.0001，SGD 动量为 0.9。

The batch size is 512 by default, which is friendly to typical 8-GPU implementations. Other batch sizes also work well (Sec. 4.3). We use batch normalization (BN) [22] synchronized across devices, following [8, 15, 7].

批量大小默认为 512，这对典型的 8-GPU 实现友好。其他批量大小也表现良好 (第 4.3 节)。我们使用跨设备同步的批量归一化 (BN)[22]，遵循 [8, 15, 7]。

- Projection MLP. The projection MLP (in f) has BN applied to each fully-connected (fc) layer, including its output fc. Its output fc has no ReLU. The hidden fc is 2048-d. This MLP has 3 layers.

- 投影 MLP。投影 MLP(在 f 中) 对每个全连接 (fc) 层应用了 BN，包括其输出 fc。其输出 fc 没有 ReLU。隐藏 fc 的维度为 2048。这种 MLP 有 3 层。

- Prediction MLP. The prediction MLP(h)has BN applied to its hidden fc layers. Its output fc does not have BN (ablation in Sec. 4.4) or ReLU. This MLP has 2 layers. The dimension of h ' s input and output(zandp)is $d = 2048$, and h ' s hidden layer' s dimension is 512, making h a bottleneck structure (ablation in supplement).

- 预测 MLP。预测 MLP(h) 对其隐藏 fc 层应用了 BN。其输出 fc 没有 BN(第 4.4 节的消融实验) 或 ReLU。这种 MLP 有 2 层。h 的输入和输出维度 (z 和 p) 为 $d = 2048$，而 h 的隐藏层维度为 512，使得 h 成为一个瓶颈结构 (补充材料中的消融实验)。

We use ResNet-50 [19] as the default backbone. Other implementation details are in supplement. We perform 100- epoch pre-training in ablation experiments.

我们使用 ResNet-50 [19] 作为默认骨干网络。其他实现细节见补充材料。我们在消融实验中进行了 100 个周期的预训练。

Experimental setup. We do unsupervised pre-training on the 1000-class ImageNet training set [11] without using labels. The quality of the pre-trained representations is evaluated by training a supervised linear classifier on frozen representations in the training set, and then testing it in the validation set, which is a common protocol. The implementation details of linear classification are in supplement.

实验设置。我们在不使用标签的情况下对 1000 类 ImageNet 训练集 [11] 进行无监督预训练。通过在训练集中对冻结表示训练一个监督线性分类器，然后在验证集上测试，这是评估预训练表示质量的常见协议。线性分类的实现细节见补充材料。

4. Empirical Study

4. 实证研究

In this section we empirically study the SimSiam behaviors. We pay special attention to what may contribute to the model's non-collapsing solutions.

在本节中，我们实证研究 SimSiam 的行为。我们特别关注可能导致模型非崩溃解的因素。

4.1. Stop-gradient

4.1. 停止梯度

Figure 2 presents a comparison on "with vs. without stop-gradient". The architectures and all hyper-parameters are kept unchanged, and stop-gradient is the only difference.

图 2 展示了“有与没有停止梯度”的比较。架构和所有超参数保持不变，停止梯度是唯一的区别。

Figure 2 (left) shows the training loss. Without stop-gradient, the optimizer quickly finds a degenerated solution and reaches the minimum possible loss of -1 . To show that the degeneration is caused by collapsing, we study the standard deviation (std) of the ℓ_2 -normalized output $z/\left\|z\right\|_2$. If the outputs collapse to a constant vector, their std over all samples should be zero for each channel. This can be observed from the red curve in Figure 2 (middle).

图 2(左) 显示了训练损失。在没有停止梯度的情况下，优化器迅速找到一个退化解，并达到了可能的最小损失 -1。为了表明退化是由崩溃引起的，我们研究了 ℓ_2 归一化输出 $z/\left\|z\right\|_2$ 的标准差 (std)。如果输出崩溃为一个常量向量，则它们在每个通道上的所有样本的标准差应该为零。这可以从图 2(中) 中的红色曲线观察到。

As a comparison, if the output z has a zero-mean isotropic Gaussian distribution, we can show that the std of $z/\left\|z\right\|_2$ is $\frac{1}{\sqrt{d}}$.[3] The blue curve in Figure 2 (middle) shows

作为比较，如果输出 z 具有零均值各向同性高斯分布，我们可以证明 $z/\left\|z\right\|_2$ 的标准差是 $\frac{1}{\sqrt{d}}$.[3] 。图 2(中) 中的蓝色曲线显示了这一点。

	pred. MLP h	acc. (%)
baseline	lr with cosine decay	67.7
(a)	no pred. MLP	0.1
(b)	fixed random init.	1.5
(c)	lr not decayed	68.1

	预测 MLP h	准确率 (%)
基线	lr 采用余弦衰减	67.7
(a)	无预测 MLP	0.1
(b)	固定随机初始化。	1.5
(c)	学习率未衰减	68.1

Table 1. Effect of prediction MLP (ImageNet linear evaluation accuracy with 100-epoch pre-training). In all these variants, we use the same schedule for the encoder f (lr with cosine decay).

表 1. 预测 MLP 的效果 (ImageNet 线性评估准确率，经过 100 轮预训练)。在所有这些变体中，我们对编码器 f 使用相同的调度 (学习率与余弦衰减)。

that with stop-gradient, the std value is near $\frac{1}{\sqrt{d}}$. This indicates that the outputs do not collapse, and they are scattered on the unit hypersphere.

在停止梯度的情况下，标准差值接近 $\frac{1}{\sqrt{d}}$ 。这表明输出没有崩溃，而是散布在单位超球面上。

Figure 2 (right) plots the validation accuracy of a k-nearest-neighbor (kNN) classifier [36]. This kNN classifier can serve as a monitor of the progress. With stop-gradient, the kNN monitor shows a steadily improving accuracy.

图 2(右) 绘制了 k 最近邻 (kNN) 分类器的验证准确率 [36]。这个 kNN 分类器可以作为进展的监测工具。在停止梯度的情况下，kNN 监测器显示出准确率稳步提高。

The linear evaluation result is in the table in Figure 2. SimSiam achieves a nontrivial accuracy of 67.7% . This result is reasonably stable as shown by the std of 5 trials. Solely removing stop-gradient, the accuracy becomes 0.1%, which is the chance-level guess in ImageNet.

线性评估结果见图 2 中的表格。SimSiam 达到了非平凡的准确率 67.7% 。这一结果相对稳定，如 5 次试验的标准差所示。仅仅去除止梯度，准确率降至 0.1%，这是 ImageNet 中的随机猜测水平。

Discussion. Our experiments show that there exist collapsing solutions. The collapse can be observed by the minimum possible loss and the constant outputs. [4] The existence of the collapsing solutions implies that it is insufficient for our method to prevent collapsing solely by the architecture designs (e.g., predictor, BN, ℓ_2 -norm). In our comparison, all these architecture designs are kept unchanged, but they do not prevent collapsing if stop-gradient is removed.

---

[3] Here is an informal derivation: denote $z/\left\|z\right\|_2$ as z' , that is, $z'_i = z_i / \left(\sum_{j=1}^{d} z_j^2\right)^{\frac{1}{2}}$ for the i -th channel. If z_j is subject to an i.i.d Gaussian distribution: $z_j \sim \mathcal{N}(0,1), \forall j$, then $z'_i \approx z_i / d^{\frac{1}{2}}$ and $\text{std}\left[z'_i\right] \approx 1/d^{\frac{1}{2}}$.

[3] 这里是一个非正式的推导: 将 $z/\left\|z\right\|_2$ 表示为 z' ，即 $z'_i = z_i / \left(\sum_{j=1}^{d} z_j^2\right)^{\frac{1}{2}}$ 对于第 i 个通道。如果 z_j 服从独立同分布的高斯分布: $z_j \sim \mathcal{N}(0,1), \forall j$ ，那么 $z'_i \approx z_i / d^{\frac{1}{2}}$ 和 $\text{std}\left[z'_i\right] \approx 1/d^{\frac{1}{2}}$ 。

讨论。我们的实验表明存在崩溃解。崩溃可以通过最小可能损失和恒定输出观察到。[4] 崩溃解的存在意味着，仅通过架构设计 (例如，预测器，BN, ℓ_2 -范数) 不足以防止崩溃。在我们的比较中，所有这些架构设计保持不变，但如果去除止梯度，它们并不能防止崩溃。

The introduction of stop-gradient implies that there should be another optimization problem that is being solved underlying. We propose a hypothesis in Sec. 5.

引入止梯度意味着应该存在另一个正在解决的优化问题。我们在第 5 节提出了一个假设。

4.2. Predictor

4.2. 预测器

In Table 1 we study the predictor MLP's effect.

在表 1 中，我们研究了预测器 MLP 的效果。

The model does not work if removing h (Table 1a), i.e., h is the identity mapping. Actually, this observation can be expected if the symmetric loss (4) is used. Now the loss is $\frac{1}{2}\mathcal{D}(z_1, \text{stopgrad}(z_2)) + \frac{1}{2}\mathcal{D}(z_2, \text{stopgrad}(z_1))$. Its gradient has the same direction as the gradient of $\mathcal{D}(z_1, z_2)$, with the magnitude scaled by 1/2 . In this case, using stop-gradient is equivalent to removing stop-gradient and scaling the loss by 1/2. Collapsing is observed (Table 1a).

如果去除 h (表 1a)，模型将无法工作，即 h 是恒等映射。实际上，如果使用对称损失 (4)，这一观察是可以预期的。现在损失为 $\frac{1}{2}\mathcal{D}(z_1, \text{stopgrad}(z_2)) + \frac{1}{2}\mathcal{D}(z_2, \text{stopgrad}(z_1))$ 。它的梯度与 $\mathcal{D}(z_1, z_2)$ 的梯度方向相同，大小由 1/2 缩放。在这种情况下，使用止梯度相当于去除止梯度并将损失缩放为 1/2。观察到崩溃 (表 1a)。

We note that this derivation on the gradient direction is valid only for the symmetrized loss. But we have observed that the asymmetric variant (3) also fails if removing h , while it can work if h is kept (Sec. 4.6). These experiments suggest that h is helpful for our model.

我们注意到，这一关于梯度方向的推导仅对对称损失有效。但我们观察到，如果去除 h ，不对称变体 (3) 也会失败，而如果保留 h ，则可以正常工作 (第 4.6 节)。这些实验表明 h 对我们的模型是有帮助的。

If h is fixed as random initialization, our model does not work either (Table 1b). However, this failure is not about collapsing. The training does not converge, and the loss remains high. The predictor h should be trained to adapt to the representations.

如果 h 被固定为随机初始化，我们的模型也无法工作 (表 1b)。然而，这种失败并不是因为崩溃。训练没有收敛，损失保持在高位。预测器 h 应该被训练以适应表示。

batch size	64	128	256	512	1024	2048	4096
acc. (%)	66.1	67.3	68.1	68.1	68.0	67.9	64.0

批量大小	64	128	256	512	1024	2048	4096
准确率 (%)	66.1	67.3	68.1	68.1	68.0	67.9	64.0

Table 2. Effect of batch sizes (ImageNet linear evaluation accuracy with 100-epoch pre-training).

表 2. 批量大小的影响 (ImageNet 线性评估准确率，经过 100 轮预训练)。

case		proj. MLP's BN		pred. MLP's BN		
		hidden	output	hidden	output	acc. (%)
(a)	none	-	-	-	-	34.6
(b)	hidden-only	✓	-	✓	-	67.4
(c)	default	✓	✓	✓	-	68.1
(d)	all	✓	✓	✓	✓	unstable

情况		proj. MLP 的 BN		pred. MLP 的 BN		
		隐藏层	输出	隐藏层	输出	准确率 (%)
(a)	无	-	-	-	-	34.6
(b)	仅隐藏	✓	-	✓	-	67.4
(c)	默认	✓	✓	✓	-	68.1
(d)	所有	✓	✓	✓	✓	不稳定

Table 3. Effect of batch normalization on MLP heads (Ima-geNet linear evaluation accuracy with 100-epoch pre-training).

表 3. 批量归一化对 MLP 头的影响 (ImageNet 线性评估准确率，经过 100 轮预训练)。

We also find that h with a constant lr (without decay) can work well and produce even better results than the baseline (Table 1c). A possible explanation is that h should adapt to the latest representations, so it is not necessary to force it converge (by reducing lr) before the representations are sufficiently trained. In many variants of our model, we have observed that h with a constant lr provides slightly better results. We use this form in the following subsections.

我们还发现，具有恒定 lr 的 h (没有衰减) 可以很好地工作，并且产生比基线更好的结果 (表 1c)。一个可能的解释是 h 应该适应最新的表示，因此在表示尚未充分训练之前，不必强制其收敛 (通过降低 lr)。在我们模型的许多变体中，我们观察到具有恒定 lr 的 h 提供了稍微更好的结果。我们在以下小节中使用这种形式。

4.3. Batch Size

4.3. 批量大小

Table 2 reports the results with a batch size from 64 to 4096. When the batch size changes, we use the same linear scaling rule ($lr \times$ BatchSize/256) [14] with base $lr = 0.05$. We use 10 epochs of warm-up [14] for batch sizes ≥ 1024 . Note that we keep using the same SGD optimizer (rather than LARS [38]) for all batch sizes studied.

表 2 报告了批量大小从 64 到 4096 的结果。当批量大小变化时，我们使用相同的线性缩放规则 ($lr \times$ BatchSize/256) [14] ，基数为 $lr = 0.05$ 。我们对批量大小 ≥ 1024 使用 10 轮的热身 [14]。请注意，我们对所有研究的批量大小保持使用相同的 SGD 优化器 (而不是 LARS [38])。

Our method works reasonably well over this wide range of batch sizes. Even a batch size of 128 or 64 performs decently, with a drop of 0.8% or 2.0% in accuracy. The results are similarly good when the batch size is from 256 to 2048, and the differences are at the level of random variations.

我们的方法在这一广泛的批量大小范围内表现得相当不错。即使批量大小为 128 或 64，性能也相当不错，准确率下降了 0.8% 或 2.0% 。当批量大小在 256 到 2048 之间时，结果同样良好，差异处于随机变动的水平。

This behavior of SimSiam is noticeably different from SimCLR [8] and SwAV [7]. All three methods are Siamese networks with direct weight-sharing, but SimCLR and SwAV both require a large batch (e.g., 4096) to work well.

SimSiam 的这种行为与 SimCLR [8] 和 SwAV [7] 明显不同。这三种方法都是具有直接权重共享的孪生网络，但 SimCLR 和 SwAV 都需要较大的批量 (例如，4096) 才能良好运行。

We also note that the standard SGD optimizer does not work well when the batch is too large (even in supervised learning [14, 38]), and our result is lower with a 4096 batch. We expect a specialized optimizer (e.g., LARS [38]) will help in this case. However, our results show that a specialized optimizer is not necessary for preventing collapsing.

我们还注意到，当批量过大时，标准的 SGD 优化器表现不佳 (即使在监督学习 [14, 38] 中也是如此)，而我们的结果在 4096 批量下较低。我们预计专门的优化器 (例如，LARS [38]) 在这种情况下会有所帮助。然而，我们的结果表明，防止崩溃并不需要专门的优化器。

4.4. Batch Normalization

4.4. 批量归一化

Table 3 compares the configurations of BN on the MLP heads. In Table 3a we remove all BN layers in the MLP heads (10-epoch warmup [14] is used specifically for this entry). This variant does not cause collapse, although the accuracy is low (34.6%) . The low accuracy is likely because of optimization difficulty. Adding BN to the hidden layers (Table 3b) increases accuracy to 67.4%.

表 3 比较了 MLP 头部上 BN 的配置。在表 3a 中，我们移除了 MLP 头部的所有 BN 层 (此条目专门使用 10 轮的预热 [14])。这种变体不会导致崩溃，尽管准确率较低 (34.6%) 。低准确率可能是由于优化困难。将 BN 添加到隐藏层 (表 3b) 将准确率提高到 67.4%。

[4] We note that a chance-level accuracy (0.1%) is not sufficient to indicate collapsing. A model with a diverging loss, which is another pattern of failure, may also exhibit a chance-level accuracy.

[4] 我们注意到，偶然水平的准确率 (0.1%) 并不足以表明崩溃。具有发散损失的模型，这是一种失败模式，也可能表现出偶然水平的准确率。

Further adding BN to the output of the projection MLP (i.e., the output of f) boosts accuracy to 68.1% (Table 3c), which is our default configuration. In this entry, we also find that the learnable affine transformation (scale and offset [22]) in f ' s output BN is not necessary, and disabling it leads to a comparable accuracy of 68.2% .

进一步将 BN 添加到投影 MLP 的输出 (即 f 的输出) 将准确率提升到 68.1% (表 3c)，这是我们的默认配置。在这一条目中，我们还发现 f 的输出 BN 中的可学习仿射变换 (缩放和偏移 [22]) 并不是必需的，禁用它会导致相当的准确率 68.2% 。

Adding BN to the output of the prediction MLP h does not work well (Table 3d). We find that this is not about collapsing. The training is unstable and the loss oscillates.

将 BN 添加到预测 MLP 的输出 h 并不奏效 (表 3d)。我们发现这并不是关于崩溃的问题。训练不稳定，损失值波动。

In summary, we observe that BN is helpful for optimization when used appropriately, which is similar to BN's behavior in other supervised learning scenarios. But we have seen no evidence that BN helps to prevent collapsing: actually, the comparison in Sec. 4.1 (Figure 2) has exactly the same BN configuration for both entries, but the model collapses if stop-gradient is not used.

总之，我们观察到，当 BN 被适当地使用时，它对优化是有帮助的，这与 BN 在其他监督学习场景中的表现相似。但我们没有看到 BN 有助于防止崩溃的证据: 实际上，4.1 节中的比较 (图 2) 对于两个条目使用了完全相同的 BN 配置，但如果不使用停止梯度，模型会崩溃。

4.5. Similarity Function

4.5. 相似性函数

Besides the cosine similarity function (1), our method also works with cross-entropy similarity. We modify $\mathcal{D}$ as: $\mathcal{D}(p_1, z_2) = -\operatorname{softmax}(z_2) \cdot \log \operatorname{softmax}(p_1)$. Here the softmax function is along the channel dimension. The output of softmax can be thought of as the probabilities of belonging to each of d pseudo-categories.

除了余弦相似性函数 (1) 外，我们的方法还适用于交叉熵相似性。我们将 $\mathcal{D}$ 修改为: $\mathcal{D}(p_1, z_2) = -\operatorname{softmax}(z_2) \cdot \log \operatorname{softmax}(p_1)$ 。这里的 softmax 函数沿着通道维度进行。softmax 的输出可以被视为属于每个 d 伪类别的概率。

We simply replace the cosine similarity with the cross-entropy similarity, and symmetrize it using (4). All hyper-parameters and architectures are unchanged, though they may be suboptimal for this variant. Here is the comparison:

我们简单地用交叉熵相似性替换余弦相似性，并使用 (4) 进行对称化。所有超参数和架构保持不变，尽管它们可能对这个变体并不是最优的。以下是比较:

	cosine	cross-entropy
acc. (%)	68.1	63.2

The cross-entropy variant can converge to a reasonable result without collapsing. This suggests that the collapsing prevention behavior is not just about the cosine similarity.

交叉熵变体可以在不崩溃的情况下收敛到合理的结果。这表明，防止崩溃的行为不仅仅与余弦相似性有关。

This variant helps to set up a connection to SwAV [7], which we discuss in Sec. 6.2.

这个变体有助于建立与 SwAV [7] 的联系，我们将在 6.2 节中讨论。

4.6. Symmetrization

4.6. 对称化

Thus far our experiments have been based on the symmetrized loss (4). We observe that SimSiam's behavior of preventing collapsing does not depend on symmetrization. We compare with the asymmetric variant (3) as follows:

到目前为止，我们的实验基于对称化损失 (4)。我们观察到，SimSiam 防止崩溃的行为并不依赖于对称化。我们将其与非对称变体 (3) 进行比较，如下所示:

	sym.	asym.	asym. 2×
acc. (%)	68.1	64.8	67.3

The asymmetric variant achieves reasonable results. Symmetrization is helpful for boosting accuracy, but it is not related to collapse prevention. Symmetrization makes one more prediction for each image, and we may roughly compensate for this by sampling two pairs for each image in the asymmetric version ("2 ×"). It makes the gap smaller.

非对称变体取得了合理的结果。对称化有助于提高准确性，但与防止崩溃无关。对称化为每个图像做出更多的预测，我们可以通过在非对称版本中为每个图像抽样两个对来大致补偿这一点（“2 ×”）。这使得差距变小。

4.7. Summary

4.7. 总结

We have empirically shown that in a variety of settings, SimSiam can produce meaningful results without collapsing. The optimizer (batch size), batch normalization, similarity function, and symmetrization may affect accuracy, but we have seen no evidence that they are related to collapse prevention. It is mainly the stop-gradient operation that plays an essential role.

我们通过实证研究表明，在多种环境下，SimSiam 能够在不崩溃的情况下产生有意义的结果。优化器(批量大小)、批量归一化、相似性函数和对称化可能会影响准确性，但我们没有看到它们与防止崩溃相关的证据。主要是停止梯度操作发挥了重要作用。

5. Hypothesis

5. 假设

We discuss a hypothesis on what is implicitly optimized by SimSiam, with proof-of-concept experiments provided.

我们讨论了一个关于 SimSiam 隐式优化内容的假设，并提供了概念验证实验。

5.1. Formulation

5.1. 公式化

Our hypothesis is that SimSiam is an implementation of an Expectation-Maximization (EM) like algorithm. It implicitly involves two sets of variables, and solves two underlying sub-problems. The presence of stop-gradient is the consequence of introducing the extra set of variables.

我们的假设是 SimSiam 是期望最大化 (EM) 类算法的一个实现。它隐式涉及两组变量，并解决两个潜在的子问题。停止梯度的存在是引入额外变量集的结果。

We consider a loss function of the following form:

我们考虑以下形式的损失函数:

$$\mathcal{L}(\theta, \eta) = \mathbb{E}_{x,\mathcal{T}} \left[\left\| \mathcal{F}_\theta\left(\mathcal{T}(x)\right) - \eta_x \right\|_2^2 \right]. \tag{5}$$

$\mathcal{F}$ is a network parameterized by θ.$\mathcal{T}$ is the augmentation. x is an image. The expectation $\mathbb{E}[\cdot]$ is over the distribution of images and augmentations. For the ease of analysis, here we use the mean squared error $\|\cdot\|_2^2$, which is equivalent to the cosine similarity if the vectors are ℓ_2 -normalized. We do not consider the predictor yet and will discuss it later.

$\mathcal{F}$ 是一个由 θ.$\mathcal{T}$ 参数化的网络，x 是一种增强。$\mathbb{E}[\cdot]$ 是对图像和增强分布的期望。为了便于分析，这里我们使用均方误差 $\|\cdot\|_2^2$ ，当向量经过 ℓ_2 归一化时，它等价于余弦相似度。我们尚未考虑预测器，稍后将讨论。

In (5), we have introduced another set of variables which we denote as η . The size of η is proportional to the number of images. Intuitively, η_x is the representation of the image x , and the subscript x means

using the image index to access a sub-vector of η.η is not necessarily the output of a network; it is the argument of an optimization problem.

在 (5) 中，我们引入了另一组变量，记作 η 。η 的大小与图像数量成正比。直观上，η_x 是图像 x 的表示，下标 x 表示使用图像索引访问 η.η 的子向量并不一定是网络的输出；它是一个优化问题的参数。

With this formulation, we consider solving:

在这种表述下，我们考虑解决:

$$\min_{\theta,\eta} \mathcal{L}(\theta,\eta) \tag{6}$$

Here the problem is w.r.t. both θ and η . This formulation is analogous to k-means clustering [28]. The variable θ is analogous to the clustering centers: it is the learnable parameters of an encoder. The variable η_x is analogous to the assignment vector of the sample x (a one-hot vector in k-means): it is the representation of x .

这里的问题涉及到 θ 和 η 。这种表述类似于 k-means 聚类 [28]。变量 θ 类似于聚类中心: 它是编码器的可学习参数。变量 η_x 类似于样本 x 的分配向量 (在 k-means 中是一个独热向量): 它是 x 的表示。

Also analogous to k-means, the problem in (6) can be solved by an alternating algorithm, fixing one set of variables and solving for the other set. Formally, we can alternate between solving these two subproblems:

同样类似于 k-means，(6) 中的问题可以通过交替算法解决，固定一组变量并求解另一组。形式上，我们可以在解决这两个子问题之间交替:

$$\theta^t \leftarrow \arg\min_{\theta} \mathcal{L}\left(\theta,\eta^{t-1}\right) \tag{7}$$

$$\eta^t \leftarrow \arg\min_{\eta} \mathcal{L}\left(\theta^t,\eta\right) \tag{8}$$

Here t is the index of alternation and " $\leftarrow$ " means assigning.

这里 t 是交替的索引，" $\leftarrow$ " 意味着分配。

Solving for θ . One can use SGD to solve the sub-problem (7). The stop-gradient operation is a natural consequence, because the gradient does not back-propagate to η^{t-1} which is a constant in this subproblem.

求解 θ 。可以使用 SGD 来解决子问题 (7)。停止梯度操作是一个自然的结果，因为梯度不会反向传播到 η^{t-1} ，在这个子问题中它是一个常量。

Solving for η . The sub-problem (8) can be solved independently for each η_x . Now the problem is to minimize: $\mathbb{E}_{\mathcal{T}}\left[\left\|\mathcal{F}_{\theta^t}(\mathcal{T}(x))-\eta_x\right\|_2^2\right]$ for each image x , noting that the expectation is over the distribution of augmentation $\mathcal{T}$. Due to the mean squared error, [5] it is easy to solve it by:

求解 η 。子问题 (8) 可以独立地为每个 η_x 解决。现在的问题是最小化: 对于每个图像 x 的 $\mathbb{E}_{\mathcal{T}}\left[\left\|\mathcal{F}_{\theta^t}(\mathcal{T}(x))-\eta_x\right\|_2^2\right]$ ，注意期望是关于增强分布 $\mathcal{T}$ 的。由于均方误差，[5] 可以很容易地通过以下方式解决:

$$\eta_x^t \leftarrow \mathbb{E}_{\mathcal{T}}\left[\mathcal{F}_{\theta^t}(\mathcal{T}(x))\right] \tag{9}$$

This indicates that η_x is assigned with the average representation of x over the distribution of augmentation.

这表明 η_x 被分配为 x 在增强分布上的平均表示。

One-step alternation. SimSiam can be approximated by one-step alternation between (7) and (8). First, we approximate (9) by sampling the augmentation only once, denoted as $\mathcal{T}'$, and ignoring $\mathbb{E}_{\mathcal{T}}[\cdot]$:

一步交替。SimSiam 可以通过 (7) 和 (8) 之间的一步交替来近似。首先，我们通过仅采样一次增强来近似 (9)，记作 $\mathcal{T}'$ ，并忽略 $\mathbb{E}_{\mathcal{T}}[\cdot]$:

$$\eta_x^t \leftarrow \mathcal{F}_{\theta^t}(\mathcal{T}'(x)) \tag{10}$$

Inserting it into the sub-problem (7), we have:

将其插入子问题 (7)，我们得到:

$$\theta^{t+1} \leftarrow \arg\min_{\theta} \mathbb{E}_{x,\mathcal{T}}\left[\left\|\mathcal{F}_{\theta}(\mathcal{T}(x))-\mathcal{F}_{\theta^t}(\mathcal{T}'(x))\right\|_2^2\right]. \tag{11}$$

Now θ^t is a constant in this sub-problem, and $\mathcal{T}'$ implies another view due to its random nature. This formulation exhibits the Siamese architecture. Second, if we implement (11) by reducing the loss with one SGD step, then we can approach the SimSiam algorithm: a Siamese network naturally with stop-gradient applied.

现在 θ^t 在这个子问题中是一个常数，而 $\mathcal{T}'$ 由于其随机性意味着另一个视图。这个公式展示了西雅图架构。其次，如果我们通过减少一次 SGD 步骤的损失来实现 (11)，那么我们可以接近 SimSiam 算法: 一个自然应用停止梯度的西雅图网络。

Predictor. Our above analysis does not involve the predictor h . We further assume that h is helpful in our method because of the approximation due to (10).

预测器。我们上述的分析不涉及预测器 h 。我们进一步假设 h 在我们的方法中是有帮助的，因为 (10) 带来的近似。

By definition, the predictor h is expected to minimize: $\mathbb{E}_z\left[\left\|h\left(z_1\right)-z_2\right\|_2^2\right]$. The optimal solution to h should satisfy: $h\left(z_1\right)=\mathbb{E}_z\left[z_2\right]=\mathbb{E}_{\mathcal{T}}\left[f\left(\mathcal{T}\left(x\right)\right)\right]$ for any image x . This term is similar to the one in (9). In our approximation in (10), the expectation $\mathbb{E}_{\mathcal{T}}\left[\cdot\right]$ is ignored. The usage of h may fill this gap. In practice, it would be unrealistic to actually compute the expectation $\mathbb{E}_{\mathcal{T}}$. But it may be possible for a neural network (e.g., the preditor h) to learn to predict the expectation, while the sampling of $\mathcal{T}$ is implicitly distributed across multiple epochs.

根据定义，预测器 h 期望最小化: $\mathbb{E}_z\left[\left\|h\left(z_1\right)-z_2\right\|_2^2\right]$ 。对 h 的最优解应满足: 对于任何图像 x ，都有 $h\left(z_1\right)=\mathbb{E}_z\left[z_2\right]=\mathbb{E}_{\mathcal{T}}\left[f\left(\mathcal{T}\left(x\right)\right)\right]$ 。这个项与 (9) 中的项类似。在我们在 (10) 中的近似中，期望 $\mathbb{E}_{\mathcal{T}}\left[\cdot\right]$ 被忽略。使用 h 可能填补这个空白。在实践中，实际计算期望 $\mathbb{E}_{\mathcal{T}}$ 是不现实的。但对于神经网络 (例如，预测器 h) 来说，学习预测期望是可能的，同时 $\mathcal{T}$ 的采样在多个周期中隐式分布。

Symmetrization. Our hypothesis does not involve symmetrization. Symmetrization is like denser sampling $\mathcal{T}$ in (11). Actually, the SGD optimizer computes the empirical expectation of $\mathbb{E}_{x,\mathcal{T}}\left[\cdot\right]$ by sampling a batch of images and one pair of augmentations $(\mathcal{T}_1,\mathcal{T}_2)$. In principle, the empirical expectation should be more precise with denser sampling. Symmetrization supplies an extra pair $(\mathcal{T}_2,\mathcal{T}_1)$. This explains that symmetrization is not necessary for our method to work, yet it is able to improve accuracy, as we have observed in Sec. 4.6.

对称化。我们的假设不涉及对称化。对称化类似于在 (11) 中的更密集采样 $\mathcal{T}$ 。实际上，SGD 优化器通过对一批图像和一对增强进行采样来计算 $\mathbb{E}_{x,\mathcal{T}}\left[\cdot\right]$ 的经验期望 $(\mathcal{T}_1,\mathcal{T}_2)$ 。原则上，经验期望在更密集的采样下应该更精确。对称化提供了额外的一对 $(\mathcal{T}_2,\mathcal{T}_1)$ 。这解释了对称化对于我们的方法并不是必要的，但正如我们在第 4.6 节中观察到的，它能够提高准确性。

5.2. Proof of concept

5.2. 概念验证

We design a series of proof-of-concept experiments that stem from our hypothesis. They are methods different with SimSiam, and they are designed to verify our hypothesis.

我们设计了一系列源于我们假设的概念验证实验。它们与 SimSiam 方法不同，旨在验证我们的假设。

Multi-step alternation. We have hypothesized that the SimSiam algorithm is like alternating between (7) and (8), with an interval of one step of SGD update. Under this hypothesis, it is likely for our formulation to work if the interval has multiple steps of SGD.

多步交替。我们假设 SimSiam 算法类似于在 (7) 和 (8) 之间交替，间隔为一次 SGD 更新步骤。在这一假设下，如果间隔有多个 SGD 步骤，我们的公式很可能会有效。

In this variant, we treat t in (7) and (8) as the index of an outer loop; and the sub-problem in (7) is updated by an inner loop of k SGD steps. In each alternation, we pre-compute the η_x required for all k SGD steps using (10) and cache them in memory. Then we perform k SGD steps to update θ . We use the same architecture and hyper-parameters as SimSiam. The comparison is as follows:

在这个变体中，我们将 (7) 和 (8) 中的 t 视为外循环的索引；而 (7) 中的子问题通过 k SGD 步骤的内循环进行更新。在每次交替中，我们使用 (10) 预计算所有 k SGD 步骤所需的 η_x 并将其缓存到内存中。然后我们执行 k SGD 步骤来更新 θ 。我们使用与 SimSiam 相同的架构和超参数。比较如下:

	1-step	10-step	100-step	1-epoch
acc. (%)	68.1	68.7	68.9	67.0

Here, "1-step" is equivalent to SimSiam, and "1-epoch" denotes the k steps required for one epoch. All multi-step variants work well. The 10-/100-step variants even achieve better results than SimSiam, though at the cost of extra pre-computation. This experiment suggests that the alternating optimization is a valid formulation, and SimSiam is a special case of it.

在这里，"1 步" 相当于 SimSiam，而 "1 个周期" 表示一个周期所需的 k 步骤。所有多步变体表现良好。10 步/100 步的变体甚至取得了比 SimSiam 更好的结果，尽管这需要额外的预计算。这个实验表明交替优化是一种有效的形式，而 SimSiam 是其特例。

Expectation over augmentations. The usage of the predictor h is presumably because the expectation $\mathbb{E}_{\mathcal{T}}[\cdot]$ in (9) is ignored. We consider another way to approximate this expectation, in which we find h is not needed.

对增强的期望。使用预测器 h 的原因可能是因为在 (9) 中忽略了期望 $\mathbb{E}_{\mathcal{T}}[\cdot]$ 。我们考虑另一种近似这种期望的方法，在这种方法中，我们发现 h 是不需要的。

In this variant, we do not update η_x directly by the assignment (10); instead, we maintain a moving-average: $\eta_x^t \leftarrow m * \eta_x^{t-1} + (1-m) * \mathcal{F}_{\theta^t}(\mathcal{T}'(x))$, where m is a momentum coefficient (0.8 here). This computation is similar to maintaining the memory bank as in [36]. This moving-average provides an approximated expectation of multiple views. This variant has 55.0% accuracy without the predictor h . As a comparison, it fails completely if we remove h but do not maintain the moving average (as shown in Table 1a). This proof-of-concept experiment supports that the usage of predictor h is related to approximating $\mathbb{E}_{\mathcal{T}}[\cdot]$.

在这个变体中，我们并不直接通过赋值 (10) 更新 η_x ；相反，我们保持一个移动平均: $\eta_x^t \leftarrow m * \eta_x^{t-1} + (1-m) * \mathcal{F}_{\theta^t}(\mathcal{T}'(x))$ ，其中 m 是动量系数 (这里为 0.8)。这个计算类似于在 [36] 中维护记忆库。这个移动平均提供了多个视图的近似期望。这个变体在没有预测器 h 的情况下具有 55.0% 的准确性。作为比较，如果我们去掉 h 但不维护移动平均 (如表 1a 所示)，则完全失败。这个概念验证实验支持了预测器 h 的使用与近似 $\mathbb{E}_{\mathcal{T}}[\cdot]$ 相关。

method	batch size	negative pairs	momentum encoder	100 ep	200 ep	400 ep	800ep
SimCLR (repro.+)	4096	✓		66.5	68.3	69.8	70.4
MoCo v2 (repro.+)	256	✓	✓	67.4	69.9	71.0	72.2
BYOL (repro.)	4096		✓	66.5	70.6	73.2	74.3
SwAV (repro.+)	4096			66.5	69.1	70.7	71.8
SimSiam	256			68.1	70.0	70.8	71.3

方法	批量大小	负样本对	动量编码器	100 轮	200 轮	400 轮	800ep
SimCLR (重现 +)	4096	✓		66.5	68.3	69.8	70.4
MoCo v2 (重现 +)	256	✓	✓	67.4	69.9	71.0	72.2
BYOL (重现)	4096		✓	66.5	70.6	73.2	74.3
SwAV (重现 +)	4096			66.5	69.1	70.7	71.8
SimSiam	256			68.1	70.0	70.8	71.3

Table 4. Comparisons on ImageNet linear classification. All are based on ResNet-50 pre-trained with two 224×224 views. Evaluation is on a single crop. All competitors are from our reproduction, and "+" denotes improved reproduction vs. original papers (see supplement).

表 4. 在 ImageNet 线性分类上的比较。所有基于用两个 224×224 视图预训练的 ResNet-50。评估是在单个裁剪上进行的。所有竞争者均来自我们的再现，"+" 表示与原始论文相比的改进再现 (见补充材料)。

pre-train	VOC 07 detection			VOC 07+12 detection			COCO detection			COCO instance seg.		
	AP50	AP	AP75	AP50	AP	AP75	AP50	AP	AP75	AP_{50}^{mask}	APmask	APmask
scratch	35.9	16.8	13.0	60.2	33.8	33.1	44.0	26.4	27.8	46.9	29.3	30.8
ImageNet supervised	74.4	42.4	42.7	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR (repro.+)	75.9	46.8	50.1	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo v2 (repro.+)	77.1	48.5	52.5	82.3	57.0	63.3	58.8	39.2	42.5	55.5	34.3	36.6
BYOL (repro.)	77.1	47.0	49.9	81.4	55.3	61.1	57.8	37.9	40.9	54.3	33.2	35.0
SwAV (repro.+)	75.5	46.5	49.6	81.5	55.4	61.4	57.6	37.6	40.3	54.2	33.1	35.1
SimSiam, base	75.5	47.0	50.2	82.0	56.4	62.8	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam, optimal	77.3	48.5	52.5	82.4	57.0	63.7	59.3	39.2	42.1	56.0	34.4	36.7

[5] If we use the cosine similarity, we can approximately solve it by ℓ_2 - normalizing $\mathcal{F}$ ' s output and η_x .

[5] 如果我们使用余弦相似度，我们可以通过 ℓ_2 近似解决它 - 对 $\mathcal{F}$ 的输出和 η_x 进行归一化。

预训练	VOC 07 检测			VOC 07+12 检测			COCO 检测			COCO 实例分割		
	AP50	AP	AP75	AP50	AP	AP75	AP50	AP	AP75	AP_{50}^{mask}	APmask	APmask
从头开始	35.9	16.8	13.0	60.2	33.8	33.1	44.0	26.4	27.8	46.9	29.3	30.8
ImageNet 监督学习	74.4	42.4	42.7	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR (重现 +)	75.9	46.8	50.1	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo v2 (重现 +)	77.1	48.5	52.5	82.3	57.0	63.3	58.8	39.2	42.5	55.5	34.3	36.6
BYOL (重现)	77.1	47.0	49.9	81.4	55.3	61.1	57.8	37.9	40.9	54.3	33.2	35.0
SwAV (重现 +)	75.5	46.5	49.6	81.5	55.4	61.4	57.6	37.6	40.3	54.2	33.1	35.1
SimSiam, 基础	75.5	47.0	50.2	82.0	56.4	62.8	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam, 最优	77.3	48.5	52.5	82.4	57.0	63.7	59.3	39.2	42.1	56.0	34.4	36.7

Table 5. Transfer Learning. All unsupervised methods are based on 200-epoch pre-training in ImageNet. VOC 07 detection: Faster R-CNN [32] fine-tuned in VOC 2007 trainval, evaluated in VOC 2007 test; VOC 07+12 detection: Faster R-CNN fine-tuned in VOC 2007 trainval + 2012 train, evaluated in VOC 2007 test; COCO detection and COCO instance segmentation: Mask R-CNN [18] (1 × schedule) fine-tuned in COCO 2017 train, evaluated in COCO 2017 val. All Faster/Mask R-CNN models are with the C4-backbone [13]. All VOC results are the average over 5 trials. Bold entries are within 0.5 below the best.

表 5. 迁移学习。所有无监督方法均基于在 ImageNet 上进行的 200 轮预训练。VOC 07 检测: 在 VOC 2007 trainval 中微调的 Faster R-CNN [32]，在 VOC 2007 测试中评估；VOC 07+12 检测: 在 VOC 2007 trainval + 2012 train 中微调的 Faster R-CNN，在 VOC 2007 测试中评估；COCO 检测和 COCO 实例分割: 在 COCO 2017 train 中微调的 Mask R-CNN [18] (1 × 计划)，在 COCO 2017 val 中评估。所有 Faster/Mask R-CNN 模型均采用 C4-backbone [13]。所有 VOC 结果为 5 次试验的平均值。粗体条目低于最佳值 0.5 以内。

5.3. Discussion

5.3. 讨论

Our hypothesis is about what the optimization problem can be. It does not explain why collapsing is prevented. We point out that SimSiam and its variants' non-collapsing behavior still remains as an empirical observation.

我们的假设是关于优化问题可能是什么。它并没有解释为什么会防止崩溃。我们指出，SimSiam 及其变体的非崩溃行为仍然是一个经验观察。

Here we briefly discuss our understanding on this open question. The alternating optimization provides a different trajectory, and the trajectory depends on the initialization. It is unlikely that the initialized η , which is the output of a randomly initialized network, would be a constant. Starting from this initialization, it may be difficult for the alternating optimizer to approach a constant η_x for all x , because the method does not compute the gradients w.r.t. η jointly for all x . The optimizer seeks another trajectory (Figure 2 left), in which the outputs are scattered (Figure 2 middle).

在这里，我们简要讨论我们对这个开放问题的理解。交替优化提供了不同的轨迹，而轨迹依赖于初始化。初始化的 η ，即随机初始化网络的输出，不太可能是一个常数。从这个初始化开始，交替优化器可能难以接近所有 x 的常数 η_x ，因为该方法并未对所有 x 共同计算相对于 η 的梯度。优化器寻求另一种轨迹 (图 2 左)，其中输出是分散的 (图 2 中)。

6. Comparisons

6. 比较

6.1. Result Comparisons

6.1. 结果比较

ImageNet. We compare with the state-of-the-art frameworks in Table 4 on ImageNet linear evaluation. For fair comparisons, all competitors are based on our reproduction, and "+" denotes improved reproduction vs. the original papers (see supplement). For each individual method, we follow the hyper-parameter and augmentation recipes in its original paper. [6] All entries are based on a standard ResNet- 50, with two 224 × 224 views used during pre-training.

ImageNet。我们在表 4 中对 ImageNet 线性评估的最先进框架进行了比较。为了公平比较，所有竞争者均基于我们的重现，“+”表示与原始论文相比的改进重现 (见补充材料)。对于每个单独的方法，我们遵循其原始论文中的超参数和增强配方。[6] 所有条目均基于标准的 ResNet-50，在预训练期间使用了两个 224×224 视图。

Table 4 shows the results and the main properties of the methods. SimSiam is trained with a batch size of 256, using neither negative samples nor a momentum encoder. Despite it simplicity, SimSiam achieves competitive results. It has the highest accuracy among all methods under 100-epoch pre-training, though its gain of training longer is smaller. It has better results than SimCLR in all cases.

表 4 显示了这些方法的结果和主要特性。SimSiam 以 256 的批量大小进行训练，既不使用负样本也不使用动量编码器。尽管其简单性，SimSiam 仍然取得了具有竞争力的结果。在 100 个周期的预训练下，它在所有方法中具有最高的准确性，尽管其延长训练的收益较小。在所有情况下，它的结果都优于 SimCLR。

Transfer Learning. In Table 5 we compare the representation quality by transferring them to other tasks, including VOC [12] object detection and COCO [26] object detection and instance segmentation. We fine-tune the pre-trained models end-to-end in the target datasets. We use the public codebase from MoCo [17] for all entries, and search the fine-tuning learning rate for each individual method. All methods are based on 200-epoch pre-training in ImageNet using our reproduction.

迁移学习。在表 5 中，我们通过将表示转移到其他任务来比较表示质量，包括 VOC [12] 目标检测和 COCO [26] 目标检测及实例分割。我们在目标数据集中对预训练模型进行端到端的微调。我们对所有条目使用 MoCo [17] 的公共代码库，并为每个单独的方法搜索微调学习率。所有方法均基于在 ImageNet 上使用我们的重现进行的 200 个周期的预训练。

Table 5 shows that SimSiam's representations are transferable beyond the ImageNet task. It is competitive among these leading methods. The "base" SimSiam in Table 5 uses the baseline pre-training recipe as in our ImageNet experiments. We find that another recipe of $lr = 0.5$ and $wd = 1\text{e}-5$ (with similar ImageNet accuracy) can produce better results in all tasks (Table 5, "SimSiam, optimal").

表 5 显示 SimSiam 的表示可以超越 ImageNet 任务进行迁移。它在这些领先方法中具有竞争力。表 5 中的“基础” SimSiam 使用与我们在 ImageNet 实验中相同的基线预训练配方。我们发现另一种 $lr = 0.5$ 和 $wd = 1\text{e}-5$ 的配方 (具有类似的 ImageNet 准确性) 可以在所有任务中产生更好的结果 (表 5，“SimSiam，最优”)。

We emphasize that all these methods are highly successful for transfer learning-in Table 5, they can surpass or be on par with the ImageNet supervised pre-training counterparts in all tasks. Despite many design differences, a common structure of these methods is the Siamese network. This comparison suggests that the Siamese structure is a core factor for their general success.

我们强调所有这些方法在迁移学习中都非常成功——在表 5 中，它们在所有任务中都能超越或与 ImageNet 监督预训练的对应方法相媲美。尽管设计上有许多差异，这些方法的一个共同结构是孪生网络。这一比较表明，孪生结构是它们普遍成功的核心因素。

6.2. Methodology Comparisons

6.2. 方法比较

Beyond accuracy, we also compare the methodologies of these Siamese architectures. Our method plays as a hub to connect these methods. Figure 3 abstracts these methods. The "encoder" subsumes all layers that can be shared between both branches (e.g., backbone, projection MLP [8], prototypes [7]). The components in red are those missing in SimSiam. We discuss the relations next.

除了准确性，我们还比较了这些孪生架构的方法论。我们的方法作为一个中心来连接这些方法。图 3 抽象了这些方法。“编码器”包含所有可以在两个分支之间共享的层 (例如，主干、投影 MLP [8]、原型 [7])。红色部分是 SimSiam 中缺失的组件。我们接下来讨论这些关系。

Relation to SimCLR [8]. SimCLR relies on negative samples ("dissimilarity") to prevent collapsing. SimSiam can be thought of as "SimCLR without negatives".

与 SimCLR [8] 的关系。SimCLR 依赖于负样本 (“不相似性”) 来防止崩溃。SimSiam 可以被视为“没有负样本的 SimCLR”。

---

[6] In our BYOL reproduction, the 100,200(400),800-epoch recipes follow the100,300,1000-epoch recipes in [15]: lr is $\{0.45, 0.3, 0.2\}$, wd is $\{1\text{e}-6, 1\text{e}-6, 1.5\text{e}-6\}$, and momentum coefficient is $\{0.99, 0.99, 0.996\}$.

[6] 在我们的 BYOL 重现中，100,200(400),800-epoch 的配方遵循 [15] 中的 100,300,1000-epoch 配方: lr 是 $\{0.45, 0.3, 0.2\}$, wd 是 $\{1\text{e}-6, 1\text{e}-6, 1.5\text{e}-6\}$ ，动量系数是 $\{0.99, 0.99, 0.996\}$ 。

To have a more thorough comparison, we append the prediction MLP h and stop-gradient to SimCLR. [7] Here is the ablation on our SimCLR reproduction:

为了进行更全面的比较，我们将预测 MLP h 和停止梯度附加到 SimCLR。[7] 这是我们 SimCLR 重现的消融实验:

SimCLR	w/ predictor	w/ pred. & stop-grad
66.5	66.4	66.0

SimCLR	带预测器	带预测器和停止梯度
66.5	66.4	66.0

Neither the stop-gradient nor the extra predictor is necessary or helpful for SimCLR. As we have analyzed in Sec. 5, the introduction of the stop-gradient and extra predictor is presumably a consequence of another underlying optimization problem. It is different from the contrastive learning problem, so these extra components may not be helpful.

对于 SimCLR 来说，停止梯度和额外的预测器既不是必要的也没有帮助。正如我们在第 5 节中分析的，引入停止梯度和额外预测器可能是另一个潜在优化问题的结果。这与对比学习问题不同，因此这些额外组件可能没有帮助。

Relation to SwAV [7]. SimSiam is conceptually analogous to "SwAV without online clustering". We build up this connection by recasting a few components in SwAV. (i) The shared prototype layer in SwAV can be absorbed into the Siamese encoder. (ii) The prototypes were weight-normalized outside of gradient propagation in [7]; we instead implement by full gradient computation [33]. [8] (iii) The similarity function in SwAV is cross-entropy. With these abstractions, a highly simplified SwAV illustration is shown in Figure 3.

与 SwAV 的关系 [7]。SimSiam 在概念上类似于 "没有在线聚类的 SwAV"。我们通过重新构建 SwAV 中的几个组件来建立这种联系。(i) SwAV 中的共享原型层可以被吸收到 Siamese 编码器中。(ii) 在 [7] 中，原型是在梯度传播之外进行权重归一化的；我们则通过完整的梯度计算来实现 [33]。[8] (iii) SwAV 中的相似性函数是交叉熵。通过这些抽象，图 3 显示了一个高度简化的 SwAV 说明。

SwAV applies the Sinkhorn-Knopp (SK) transform [10] on the target branch (which is also symmetrized [7]). The SK transform is derived from online clustering [7]: it is the outcome of clustering the current batch subject to a balanced partition constraint. The balanced partition can avoid collapsing. Our method does not involve this transform.

SwAV 在目标分支上应用 Sinkhorn-Knopp (SK) 变换 [10](该变换也进行了对称化 [7])。SK 变换源于在线聚类 [7]: 它是对当前批次进行聚类的结果，受限于平衡分区约束。平衡分区可以避免崩溃。我们的方法不涉及此变换。

We study the effect of the prediction MLP h and stop-gradient on SwAV. Note that SwAV applies stop-gradient on the SK transform, so we ablate by removing it. Here is the comparison on our SwAV reproduction:

我们研究了预测 MLP h 和停止梯度对 SwAV 的影响。注意，SwAV 在 SK 变换上应用了停止梯度，因此我们通过去除它来进行消融实验。以下是我们对 SwAV 复现的比较:

SwAV	w/ predictor	remove stop-grad
66.5	65.2	NaN

SwAV	带预测器	移除停止梯度
66.5	65.2	NaN

Adding the predictor does not help either. Removing stop-gradient (so the model is trained end-to-end) leads to divergence. As a clustering-based method, SwAV is inherently an alternating formulation [7]. This may explain why stop-gradient should not be removed from SwAV.

添加预测器也没有帮助。去除停止梯度 (因此模型是端到端训练的) 导致了发散。作为一种基于聚类的方法，SwAV 本质上是一种交替形式 [7]。这可能解释了为什么停止梯度不应从 SwAV 中去除。

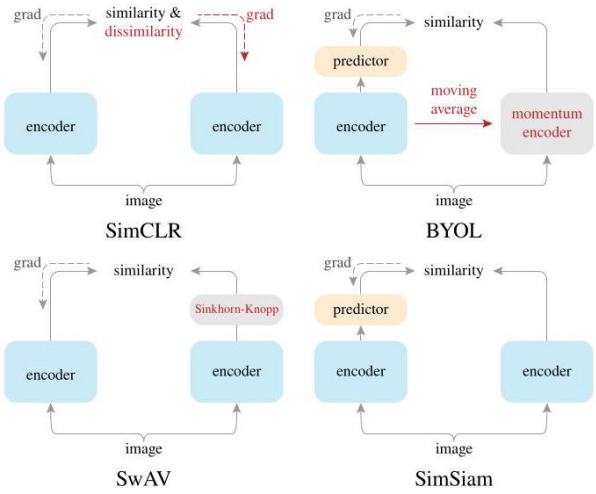


Figure 3. Comparison on Siamese architectures. The encoder includes all layers that can be shared between both branches. The dash lines indicate the gradient propagation flow. In BYOL, SwAV, and SimSiam, the lack of a dash line implies stop-gradient, and their symmetrization is not illustrated for simplicity. The components in red are those missing in SimSiam.

图 3. Siamese 架构的比较。编码器包括可以在两个分支之间共享的所有层。虚线表示梯度传播流。在 BYOL、SwAV 和 SimSiam 中，缺少虚线意味着停止梯度，并且为了简化起见，它们的对称化没有被说明。红色组件是 SimSiam 中缺失的部分。

Relation to BYOL [15]. Our method can be thought of as "BYOL without the momentum encoder", subject to many implementation differences. The momentum encoder may be beneficial for accuracy (Table 4), but it is not necessary for preventing collapsing. Given our hypothesis in Sec. 5, the η sub-problem (8) can be solved by other optimizers, e.g., a gradient-based one. This may lead to a temporally smoother update on η . Although not directly related, the momentum encoder also produces a smoother version of η . We believe that other optimizers for solving (8) are also plausible, which can be a future research problem.

与 BYOL 的关系 [15]。我们的方法可以被视为 "没有动量编码器的 BYOL"，尽管存在许多实现上的差异。动量编码器可能对准确性有益 (表 4)，但对于防止崩溃并不是必需的。根据我们在第 5 节中的假设，η 子问题 (8) 可以通过其他优化器解决，例如基于梯度的优化器。这可能导致 η 的更新更加平滑。虽然与之没有直接关系，但动量编码器也会产生 η 的更平滑版本。我们相信，解决 (8) 的其他优化器也是可行的，这可以成为未来的研究课题。

7. Conclusion

7. 结论

We have explored Siamese networks with simple designs. The competitiveness of our minimalist method suggests that the Siamese shape of the recent methods can be a core reason for their effectiveness. Siamese networks are natural and effective tools for modeling invariance, which is a focus of representation learning. We hope our study will attract the community's attention to the fundamental role of Siamese networks in representation learning.

我们探讨了具有简单设计的孪生网络。我们极简方法的竞争力表明，最近方法的孪生形状可能是其有效性的核心原因。孪生网络是建模不变性的自然且有效的工具，而不变性是表示学习的一个重点。我们希望我们的研究能够引起社区对孪生网络在表示学习中基础性作用的关注。

References

参考文献

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. arXiv:1911.05371, 2019.

[2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. arXiv:1906.00910, 2019.

[3] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional Siamese networks for object tracking. In ECCV, 2016.

[4] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "Siamese" time delay neural network. In NeurIPS, 1994.

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In ECCV, 2018.

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

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. arXiv:2006.09882, 2020.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. arXiv:2002.05709, 2020.

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. arXiv:2003.04297, 2020.

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

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In CVPR, 2009.

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. IJCV, 2010.

[13] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron, 2018.

[14] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noord-huis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. arXiv:1706.02677, 2017.

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Do-ersch, 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. arXiv:2006.07733v1, 2020.

[16] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In CVPR, 2006.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. arXiv:1911.05722, 2019.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Gir-shick. Mask R-CNN. In ICCV, 2017.

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

[20] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In ICLR, 2019.

[21] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. arXiv:1905.09272v2, 2019.

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.

---

[7] We append the extra predictor to one branch and stop-gradient to the other branch, and symmetrize this by swapping.
[7] 我们将额外的预测器附加到一个分支，并对另一个分支停止梯度，并通过交换使其对称化。
[8] This modification produces similar results as original SwAV, but it can enable end-to-end propagation in our ablation.
[8] 该修改产生的结果与原始 SwAV 相似，但它可以在我们的消融实验中实现端到端传播。

[23] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In ICML deep learning workshop, 2015.

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. Tech Report, 2009.

[25] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1989.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In ECCV. 2014.

[27] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In ICLR, 2017.

[28] James MacQueen et al. Some methods for classification and analysis of multivariate observations. 1967.

[29] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. arXiv:1912.01991, 2019.

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv:1807.03748, 2018.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In NeurIPS, 2019.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NeurIPS, 2015.

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

[34] Yaniv Taigman, Ming Yang, MarcAurelio Ranzato, and Lior Wolf. DeepFace: Closing the gap to human-level performance in face verification. In CVPR, 2014.

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. arXiv:1906.05849, 2019.

[36] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In CVPR, 2018.

[37] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In CVPR, 2019.

[38] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. arXiv:1708.03888, 2017.

A. Implementation Details

A. 实现细节

Unsupervised pre-training. Our implementation follows the practice of existing works [36, 17, 8, 9, 15].

无监督预训练。我们的实现遵循现有工作的做法 [36, 17, 8, 9, 15]。

Data augmentation. We describe data augmentation using the PyTorch [31] notations. Geometric augmentation is RandomResizedCrop with scale in $[0.2, 1.0]$ [36] and RandomHorizontalFlip. Color augmentation is ColorJitter with {brightness, contrast, saturation, hue} strength of {0.4, 0.4, 0.4, 0.1} with an applying probability of 0.8, and RandomGrayscale with an applying probability of 0.2 . Blurring augmentation [8] has a Gaussian kernel with std in $[0.1, 2.0]$.

数据增强。我们使用 PyTorch [31] 的符号描述数据增强。几何增强是 RandomResizedCrop，缩放范围在 $[0.2, 1.0]$ [36] 之间，以及 RandomHorizontalFlip。颜色增强是 ColorJitter，亮度、对比度、饱和度和色调的强度分别为 {0.4, 0.4, 0.4, 0.1}，应用概率为 0.8，以及 RandomGrayscale，应用概率为 0.2。模糊增强 [8] 使用标准差在 $[0.1, 2.0]$ 的高斯核。

Initialization. The convolution and fc layers follow the default PyTorch initializers. Note that by default PyTorch initializes fc layers' weight and bias by a uniform distribution $\mathcal{U}\left(-\sqrt{k}, \sqrt{k}\right)$ where $k = \frac{1}{\text{in\_channels}}$. Models with substantially different fc initializers (e.g., a fixed std of 0.01) may not converge. Moreover, similar to the implementation of [8], we initialize the scale parameters as 0 [14] in the last BN layer for every residual block.

初始化。卷积层和全连接层遵循默认的 PyTorch 初始化器。请注意，默认情况下，PyTorch 通过均匀分布 $\mathcal{U}\left(-\sqrt{k}, \sqrt{k}\right)$ 初始化全连接层的权重和偏差，其中 $k = \frac{1}{\text{in\_channels}}$ 。具有显著不同全连接初始化

器的模型 (例如，固定标准差为 0.01) 可能无法收敛。此外，类似于 [8] 的实现，我们在每个残差块的最后 BN 层中将尺度参数初始化为 0 [14]。

Weight decay. We use a weight decay of 0.0001 for all parameter layers, including the BN scales and biases, in the SGD optimizer. This is in contrast to the implementation of [8, 15] that excludes BN scales and biases from weight decay in their LARS optimizer.

权重衰减。我们在 SGD 优化器中对所有参数层 (包括 BN 的尺度和偏差) 使用 0.0001 的权重衰减。这与 [8, 15] 的实现形成对比，后者在其 LARS 优化器中排除了 BN 的尺度和偏差的权重衰减。

Linear evaluation. Given the pre-trained network, we train a supervised linear classifier on frozen features, which are from ResNet' s global average pooling layer (pool_5) . The linear classifier training uses base $lr = 0.02$ with a cosine decay schedule for 90 epochs, weight decay $= 0$, momentum $= 0.9$, batch size $= 4096$ with a LARS optimizer [38]. We have also tried the SGD optimizer following [17] with base $lr = 30.0$, weight decay $= 0$, momentum $= 0.9$, and batch size $= 256$, which gives $\sim 1\%$ lower accuracy. After training the linear classifier, we evaluate it on the center 224×224 crop in the validation set.

线性评估。给定预训练网络，我们在冻结特征上训练一个监督线性分类器，这些特征来自 ResNet 的全局平均池化层 (pool_5) 。线性分类器训练使用基学习率 $lr = 0.02$ ，采用余弦衰减计划，持续 90 个周期，权重衰减 $= 0$ ，动量 $= 0.9$ ，批量大小 $= 4096$ ，并使用 LARS 优化器 [38]。我们还尝试了遵循 [17] 的 SGD 优化器，使用基学习率 $lr = 30.0$ ，权重衰减 $= 0$ ，动量 $= 0.9$ ，和批量大小 $= 256$ ，但结果 $\sim 1\%$ 的准确性较低。训练完线性分类器后，我们在验证集的中心 224×224 裁剪上对其进行评估。

B. Additional Ablations on ImageNet

B. 关于 ImageNet 的额外消融实验

The following table reports the SimSiam results vs. the output dimension d :

下表报告了 SimSiam 结果与输出维度 d 的关系:

output d	256	512	1024	2048
acc. (%)	65.3	67.2	67.5	68.1

输出 d	256	512	1024	2048
准确率 (%)	65.3	67.2	67.5	68.1

It benefits from a larger d and gets saturated at $d = 2048$. This is unlike existing methods [36, 17, 8, 15] whose accuracy is saturated when d is 256 or 512 .

它受益于更大的 d ，并在 $d = 2048$ 达到饱和。这与现有方法 [36, 17, 8, 15] 不同，后者的准确性在 d 为 256 或 512 时达到饱和。

In this table, the prediction MLP's hidden layer dimension is always 1/4 of the output dimension. We find that this bottleneck structure is more robust. If we set the hidden dimension to be equal to the output dimension, the training can be less stable or fail in some variants of our exploration. We hypothesize that this bottleneck structure, which behaves like an auto-encoder, can force the predictor to digest the information. We recommend to use this bottleneck structure for our method.

在此表中，预测 MLP 的隐藏层维度始终为输出维度的 1/4 。我们发现这种瓶颈结构更为稳健。如果我们将隐藏维度设置为等于输出维度，训练可能会不够稳定或在我们探索的某些变体中失败。我们假设这种瓶颈结构，类似于自编码器，可以迫使预测器消化信息。我们建议在我们的方法中使用这种瓶颈结构。

	SimCLR			MoCo v2		BYOL			SwAV
epoch	200	800	1000	200	800	300	800	1000	400
origin	66.6	68.3	69.3	67.5	71.1	72.5	-	74.3	70.1
repro.	68.3	70.4	-	69.9	72.2	72.4	74.3	-	70.7

	SimCLR			MoCo v2		BYOL			SwAV
纪元	200	800	1000	200	800	300	800	1000	400
起源	66.6	68.3	69.3	67.5	71.1	72.5	-	74.3	70.1
重现	68.3	70.4	-	69.9	72.2	72.4	74.3	-	70.7

Table C.1. Our reproduction vs. original papers' results. All are based on ResNet-50 pre-trained with two 224×224 crops.

表 C.1。我们的重现结果与原始论文的结果对比。所有结果均基于使用两个 224×224 裁剪的 ResNet-50 预训练模型。

C. Reproducing Related Methods

C. 重现相关方法

Our comparison in Table 4 is based on our reproduction of the related methods. We re-implement the related methods as faithfully as possible following each individual paper. In addition, we are able to improve SimCLR, MoCo v2, and SwAV by small and straightforward modifications: specifically, we use 3 layers in the projection MLP in SimCLR and SwAV (vs. originally 2), and use symmetrized loss for MoCo v2 (vs. originally asymmetric). Table C. 1 compares our reproduction of these methods with the original papers' results (if available). Our reproduction has better results for SimCLR, MoCo v2, and SwAV (denoted as "+" in Table 4), and has at least comparable results for BYOL.

我们在表 4 中的比较基于我们对相关方法的再现。我们尽可能忠实地重新实现相关方法，遵循每篇论文的具体内容。此外，我们能够通过小而简单的修改来改进 SimCLR、MoCo v2 和 SwAV: 具体而言，我们在 SimCLR 和 SwAV 的投影 MLP 中使用 3 层 (而最初为 2 层)，并对 MoCo v2 使用对称损失 (而最初为不对称)。表 C.1 将我们对这些方法的再现与原始论文的结果进行了比较 (如果有的话)。我们的再现结果在 SimCLR、MoCo v2 和 SwAV 上表现更佳 (在表 4 中标记为"+")，并且在 BYOL 上至少具有可比的结果。

D. CIFAR Experiments

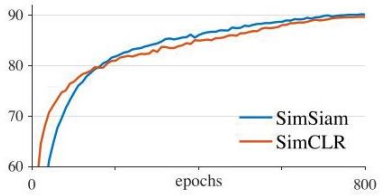
D. CIFAR 实验

We have observed similar behaviors of SimSiam in the CIFAR-10 dataset [24]. The implementation is similar to that in ImageNet. We use SGD with base $lr = 0.03$ and a cosine decay schedule for 800 epochs, weight decay $= 0.0005$, momentum $= 0.9$, and batch size $= 512$. The input image size is 32×32 . We do not use blur augmentation. The backbone is the CIFAR variant of ResNet-18 [19], followed by a 2-layer projection MLP. The outputs are 2048-d.

我们在 CIFAR-10 数据集 [24] 中观察到了 SimSiam 的类似行为。其实现与 ImageNet 中的类似。我们使用 SGD，基础 $lr = 0.03$ 和一个余弦衰减计划进行 800 个周期，权重衰减 $= 0.0005$，动量 $= 0.9$ 和批量大小 $= 512$ 。输入图像大小为 32×32 。我们不使用模糊增强。主干网络是 CIFAR 变体的 ResNet-18 [19]，后接一个 2 层投影 MLP。输出为 2048 维。

Figure D. 1 shows the kNN classification accuracy (left) and the linear evaluation (right). Similar to the ImageNet observations, SimSiam achieves a reasonable result and does not collapse. We compare with SimCLR [8] trained with the same setting. Interestingly, the training curves are similar between SimSiam and SimCLR. SimSiam is slightly better by 0.7% under this setting.

图 D.1 显示了 kNN 分类准确率 (左) 和线性评估 (右)。与 ImageNet 的观察结果类似，SimSiam 达到了合理的结果并且没有崩溃。我们与在相同设置下训练的 SimCLR [8] 进行了比较。有趣的是，SimSiam 和 SimCLR 之间的训练曲线相似。在这种设置下，SimSiam 略微优于 0.7% 。



	acc. (%)
SimCLR	91.1
SimSiam	91.8

Figure D.1. CIFAR-10 experiments. Left: validation accuracy of kNN classification as a monitor during pre-training. Right: linear evaluation accuracy. The backbone is ResNet-18.

图 D.1. CIFAR-10 实验。左:kNN 分类的验证准确率作为预训练期间的监测。右: 线性评估准确率。主干网络为 ResNet-18。