

2
2
0
2
y
a
M
5
1
V
C
s
c
l
2
v
5
6
2
8
0
6
0
1
2
:
v
X
r
a

迈向工业异常检测的全面召回

卡尔斯滕·罗特^{1,*}、拉塔·佩穆拉²、华金·塞佩达²、伯恩哈德·舍尔科普夫²、托马斯·布罗克斯²、
彼得·格勒² ¹蒂宾根大学、²亚马逊AWS

摘要

*Being able to spot defective parts is a critical component in large-scale industrial manufacturing. A particular challenge that we address in this work is the cold-start problem: fit a model using nominal (non-defective) example images only. While handcrafted solutions per class are possible, the goal is to build systems that work well simultaneously on many different tasks automatically. The best performing approaches combine embeddings from ImageNet models with an outlier detection model. In this paper, we extend on this line of work and propose **PatchCore**, which uses a maximally representative memory bank of nominal patch-features. PatchCore offers competitive inference times while achieving state-of-the-art performance for both detection and localization. On the challenging, widely used MVTec AD benchmark PatchCore achieves an image-level anomaly detection AUROC score of up to 99.6%, more than halving the error compared to the next best competitor. We further report competitive results on two additional datasets and also find competitive results in the few samples regime. Code: github.com/amazon-research/patchcore-inspection.*

1. 引言

检测图像中异常模式的能力是人类认知中根深蒂固的特征。人类只需见过少量正常样本，就能区分数据中的预期变化与异常值。在本研究中，我们探讨这一问题的计算实现形式——针对工业图像数据视觉检测的 $\{v^*\}$ 异常检测。该需求广泛存在于工业场景中：正常样本的图像容易获取，但完整定义预期缺陷变体却成本高昂且复杂。这一任务自然可归为分布外检测问题，即模型需要区分来自训练数据分布的样本与超出其支持域的样本。工业视觉缺陷分类尤其困难，因为误差可能

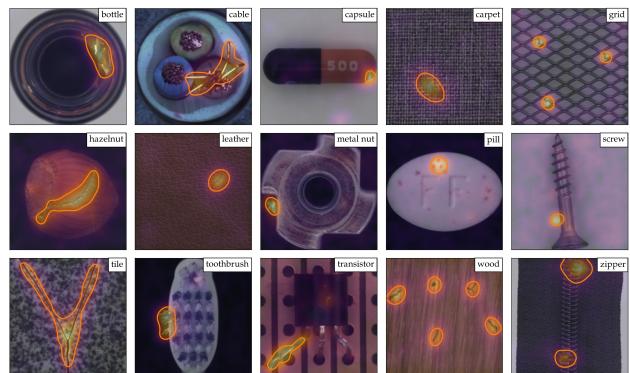


图1. MVTec基准数据集示例。图像上叠加了来自Patch-Core的分割结果。橙色边界表示异常轮廓，对应实际分割图中诸如玻璃破损、划痕、灼痕或结构变化等异常，并以蓝-橙渐变色呈现。

从细微的变化（如细小的划痕）到较大的结构缺陷（如部件缺失）均有体现[5]。图1展示了MVTec AD基准测试中的一些示例以及我们提出方法的结果。现有关于冷启动工业视觉异常检测的研究依赖于通过自动编码方法[12, 36, 44]、生成对抗网络[2, 39, 43]或其他无监督适应方法[42, 56]来学习正常分布模型。最近，[4, 10]提出直接利用ImageNet分类任务中的通用深度表征，而无需针对目标分布进行适应。尽管缺乏适应性调整，这些模型仍能提供强大的异常检测性能，甚至能对缺陷进行精确的空间定位。这些技术背后的核心原理在于测试样本与正常样本之间的特征匹配，同时利用深度特征表征的多尺度特性：细微的、细粒度的缺陷分割由高分辨率特征覆盖，而结构偏差和完整的图像级异常检测则应由更高抽象层次的特征覆盖。由于这种方法不具备适应性，其固有缺陷在于较高抽象层次上的匹配置信度有限：来自ImageNet训练的高层抽象特征与目标数据重合度较低。

* Work done during a research internship at Amazon AWS.

¹Commonly also dubbed one-class classification (OCC).

工业环境所需的抽象特征。此外，这些方法在测试时可利用的名义上下文，实际上受限于可提取的高层特征表示数量较少。

本文提出*PatchCore*作为一种有效的解决方案，其通过以下方式实现：(1) 最大化测试时可用的正常信息，(2) 减少对ImageNet类别的偏向性，(3) 保持较高的推理速度。基于图像中只要存在单个异常块即可被判定为异常的事实[14, 56]，*Patch-Core*通过利用局部聚合的中层特征块达成这一目标。使用中层网络块特征使得*PatchCore*能在高分辨率下以最小偏向性处理ImageNet类别，同时通过局部邻域的特征聚合确保保留足够的空间上下文信息。这构建了一个庞大的记忆库，使*PatchCore*能在测试时最优利用可用的正常上下文信息。最后，为提升实际适用性，*PatchCore*额外引入针对正常特征库的贪心核心集子采样方法[1]作为关键要素，既降低了提取的块级记忆库中的冗余度，又显著减少了存储内存与推理时间，使得*Patch-Core*对实际工业应用场景极具吸引力。

在多样化的MVTec AD [5] 以及专业的磁砖缺陷（MTD）[26] 工业异常检测基准上进行的大量实验，展示了*PatchCore*在工业异常检测方面的强大能力。它在MVTec AD和MTD上实现了最先进的图像级检测分数，在MVTec AD上取得了近乎完美的成绩（AUROC高达99.6%），将先前方法的检测误差降低了一半以上，同时实现了最先进的工业异常定位性能。*Patch-Core*在实现这些成果的同时，保持了快速的推理速度，且无需在目标数据集上进行训练。这使得*PatchCore*在工业异常检测的实际应用中极具吸引力。此外，进一步的实验展示了*PatchCore*的高样本效率，在使用仅一小部分正常训练数据的情况下，其性能即可与现有异常检测方法相媲美。

2. 相关工作

大多数异常检测模型依赖于学习名义数据固有表征的能力。例如，这可以通过使用自编码模型来实现[44]。为了促进对名义特征分布的更好估计，基于高斯混合模型[60]、生成对抗训练目标[2,39,43]、对预定义物理增强的不变性[25]、隐藏特征对重建数据重新引入的鲁棒性[29]、原型记忆库[21]、注意力引导[52]、结构化目标[7,59]或受限表征空间[38]的扩展方法已被提出

其他无监督表示学习方法同样可以被利用，例如通过生成对抗网络（GANs）[13]、学习预测预定义的几何变换[20]或通过归一化流[42]。在获得相应的正常表示和新的测试表示后，异常检测可以简化为计算重构误差[44]、与k最近邻的距离[18]，或在这些特征基础上微调单类分类模型，如OC-SVMs[46]或SVDD[50, 56]。对于大多数这些方法，异常定位自然基于像素级重构误差；基于显著性的方法（如GradCAM[47]或XRAI[28]）也可用于异常分割[42, 45, 52]。

工业异常检测。尽管通过学习的正常表示进行通用异常检测的文献浩如烟海，但工业图像数据仍面临其独特挑战[5]。针对这些挑战，自[4]开创的近期研究显示，通过使用在大型外部自然图像数据集（如ImageNet[16]）上预训练的模型，无需对当前数据进行任何适配即可实现最先进的检测性能。这催生了其他依赖更好利用预训练特征的工业异常检测方法，例如SPADE[10]——该方法利用包含多层级特征的记忆库，实现基于kNN[18]的细粒度异常分割与图像级异常检测。类似地，[14]近期提出的PaDiM采用局部约束词袋方法[8]，通过估计图像块层级的特征分布矩（均值与协方差）来计算块级马氏距离[33]。该方法与[40]在全图像层面的研究思路相似。为更好应对自然图像预训练与工业图像数据间的分布偏移，可进行后续适配，例如通过师生知识蒸馏[24]（如[6,45]所示）或在预训练网络特征上训练归一化流[17,30,42]来实现。

在*PatchCore*中使用的具体组件与SPADE和PaDiM最为相关。SPADE利用从预训练骨干网络中提取的正常特征构建记忆库，并采用不同的方法进行图像级和像素级异常检测。*PatchCore*同样使用了记忆库，但采用了邻域感知的块级特征，这对于实现更高性能至关重要，因为它保留了更多的正常上下文信息，并融入了更合适的归纳偏置。此外，记忆库通过核心集子采样来确保在更高性能下保持较低的推理成本。核心集在基础的kNN和kMeans方法[22]或混合模型[19]中已有长期应用，通过寻找最能近似给定集合结构的子集，以显著降低的成本实现近似求解[1, 9]。最近，基于核心集的方法也进入了深度学习领域，例如用于网络剪枝[34]、主动学习[48]以及提升数据有效性等方面。

小批量覆盖用于改进GAN训练[49]或表征学习[41]。后三种方法通过贪心核心集选择机制取得了成功。由于我们的目标是近似记忆库特征空间覆盖，我们同样为 *PatchCore* 采用了贪心核心集机制。最后，我们在图像级异常检测和异常分割中采用的块级方法，与 PaDiM 相关，旨在提高异常检测的灵敏度。我们利用高效的块特征记忆库，测试时所有待评估块均可平等访问，而 PaDiM 将块级异常检测限制于针对每个块计算的马氏距离度量。通过这种方式，*PatchCore* 对图像对齐的依赖降低，同时能利用更大的正常上下文估计异常。此外，与 PaDiM 不同，输入图像在训练和测试时无需保持相同尺寸。最后，*PatchCore* 利用局部感知的块特征评分来考虑局部空间方差，并减少对 ImageNet 类别的偏向。

3. 方法

PatchCore 方法由几个部分按顺序组成：局部补丁特征聚合到记忆库中（§3.1），用于提高效率的核心集约简方法（§3.2），以及最终实现检测与定位决策的完整算法（§3.3）。

3.1. 局部感知的补丁特征

我们使用 \mathcal{X}_N 表示训练时可用的所有正常图像集合（ $\forall x \in \mathcal{X}_N : y_x = 0$ ），其中 $y_x \in \{0, 1\}$ 表示图像 x 属于正常（0）或异常（1）。相应地，我们将 \mathcal{X}_T 定义为测试时提供的样本集合，其中 $\forall x \in \mathcal{X}_T : y_x \in \{0, 1\}$ 。遵循[4]、[10] 和 [14] 的方法，*PatchCore* 采用在 ImageNet 上预训练的网络 ϕ 。由于特定网络层级的特征具有重要作用，我们使用 $\phi_{i,j} = \phi_j(x_i)$ 表示数据集 \mathcal{X} 中图像 $x_i \in \mathcal{X}$ （在预训练网络 ϕ 第 j 层级所提取的特征。除非另有说明，按照现有文献惯例， j 索引类 ResNet 架构[23]（如 ResNet-50 或 WideResnet-50[57]）的特征图，其中 $j \in \{1, 2, 3, 4\}$ 对应各空间分辨率模块的最终输出。

一种特征表示的选择可以是网络特征层次结构中的最后一层。这在[4]或[10]中有所实现，但引入了以下两个问题。首先，它丢失了更多局部化的名义信息[14]。由于测试时遇到的异常类型未知 *a priori*，这对下游异常检测性能产生了不利影响。其次，ImageNet 预训练网络中非常深层和抽象的特征偏向于自然图像分类任务，这与冷启动工业异常检测任务及手头评估数据的重叠度极低。

因此，我们建议使用包含 *intermediate* 或 *mid-level* 特征表示的补丁级特征记忆库 \mathcal{M} ，以利用提供的训练上下文，避免特征过于通用或过度偏向 ImageNet 分类。在类似 ResNet 架构的具体情况下，这指的是例如 $j \in [2, 3]$ 。为了形式化补丁表示，我们扩展了先前引入的符号。假设特征图 $\phi_{i,j} \in \mathbb{R}^{c^* \times h^* \times w^*}$ 是一个深度为 c^* 、高度为 h^* 、宽度为 w^* 的三维张量。然后，我们使用 $\phi_{i,j}(h, w) = \phi_j(x_i, h, w) \in \mathbb{R}^{c^*}$ 表示在位置 $h \in \{1, \dots, h^*\}$ 和 $w \in \{1, \dots, w^*\}$ 处的 c^* 维特征切片。假设每个 $\phi_{i,j}$ 的感受野大小大于一，这实际上与图像补丁的特征表示相关。理想情况下，每个补丁表示在足够大的感受野上操作，以考虑对局部空间变化具有鲁棒性的有意义异常上下文。虽然这可以通过步幅池化和进一步深入网络层次来实现，但由此产生的补丁特征变得更加 ImageNet 特定，因此对于当前的异常检测任务相关性降低，同时训练成本增加且有效特征图分辨率下降。

这促使我们在构建每个补丁级特征表示时采用局部邻域聚合，以在不损失空间分辨率或特征图可用性的前提下，增大感受野尺寸并提升对小空间偏移的鲁棒性。为此，我们将上述针对 $\phi_{i,j}(h, w)$ 的表示方法扩展至考虑不均匀的补丁尺寸 p （这些尺寸对应所考察的邻域范围），从而整合来自邻域的特征向量。

$$\mathcal{N}_p^{(h,w)} = \{(a, b) | a \in [h - \lfloor p/2 \rfloor, \dots, h + \lfloor p/2 \rfloor], b \in [w - \lfloor p/2 \rfloor, \dots, w + \lfloor p/2 \rfloor]\}, \quad (1)$$

并在位置 (h, w) 处具有局部感知特征为

$$\phi_{i,j}(\mathcal{N}_p^{(h,w)}) = f_{\text{agg}} \left(\{\phi_{i,j}(a, b) | (a, b) \in \mathcal{N}_p^{(h,w)}\} \right), \quad (2)$$

其中 f_{agg} 是邻域 $\mathcal{N}_p^{(h,w)}$ 内特征向量的某种聚合函数。对于 *PatchCore*，我们采用自适应平均池化。这类似于对每个独立特征图进行局部平滑处理，并在预定义维度 d 的 (h, w) 处生成单一表征，该操作针对所有满足 $h \in \{1, \dots, h^*\}$ 和 $w \in \{1, \dots, w^*\}$ 条件的配对 (h, w) 执行，从而保持特征图分辨率。对于特征图张量 $\phi_{i,j}$ ，其局部感知的块特征集合 $\mathcal{P}_{s,p}(\phi_{i,j})$ 可表示为

$$\mathcal{P}_{s,p}(\phi_{i,j}) = \{\phi_{i,j}(\mathcal{N}_p^{(h,w)}) | h, w \bmod s = 0, h < h^*, w < w^*, h, w \in \mathbb{N}\}, \quad (3)$$

通过可选地使用步长参数 s ，我们将其设为 1，除非在 §4.4.2 节进行的消融实验中。根据经验，并与[10] 和 [14] 的研究相似，我们发现聚合的

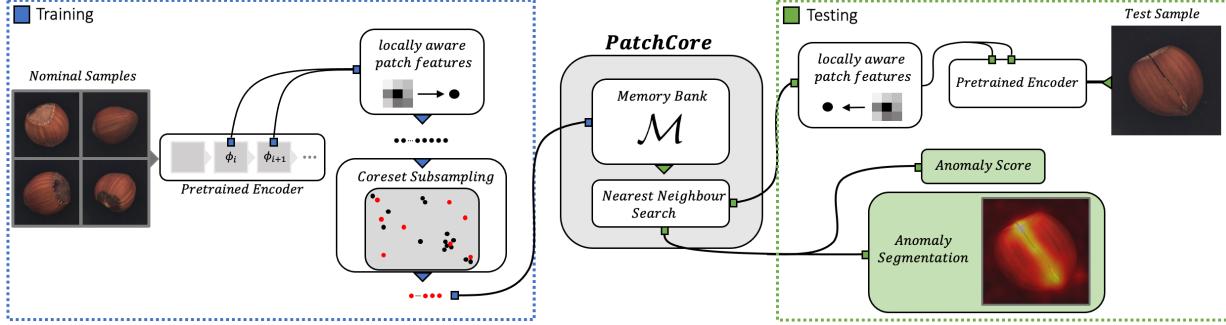


图2. *PatchCore*概述。正常样本被分解为邻域感知的块级特征记忆库。为降低冗余度和推理时间，该记忆库通过贪心核心集下采样进行降采样。测试时，若图像中至少存在一个异常块，则判定为异常图像；同时通过逐块特征评分生成像素级异常分割图。

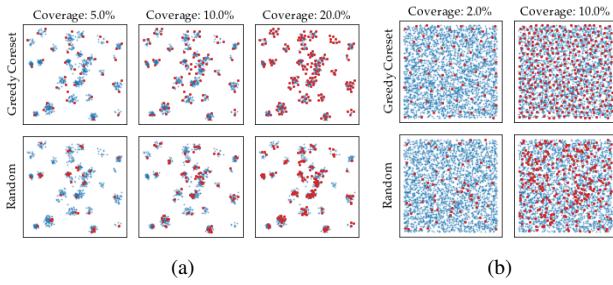


图3. 对比：从(a)多模态分布与(b)均匀分布中采样的二维数据（蓝色）的核集采样（上图）与随机子采样（下图）（红色）效果。直观可见，核集子采样能更好地逼近空间支撑集，随机子采样在多模态情况下会遗漏聚类簇，且在(b)中均匀性较差。

多个特征层次结构确实能带来一些好处。然而，为了保持所用特征的通用性以及空间分辨率，*PatchCore*仅采用两个中间特征层次结构 j 和 $j +$ 。这一设计通过计算 $\mathcal{P}_{s,p}(\phi_{i,j+1})$ ，并将每个元素与所用最低层次（即最高分辨率）的对应图像块特征进行聚合实现。具体而言，我们通过双线性重缩放 $\mathcal{P}_{s,p}(\phi_{i,j+1})$ 使 $|\mathcal{P}_{s,p}(\phi_{i,j+1})|$ 与 $|\mathcal{P}_{s,p}(\phi_{i,j})|$ 相匹配来完成这一过程。

最后，对于所有名义训练样本 $x_i \in \mathcal{X}_N$ ，*PatchCore*记忆库 \mathcal{M} 被简单地定义为

$$\mathcal{M} = \bigcup_{x_i \in \mathcal{X}_N} \mathcal{P}_{s,p}(\phi_j(x_i)). \quad (4)$$

3.2. 核心集缩减的补丁特征记忆库

随着 \mathcal{X}_N 尺寸的增加， \mathcal{M} 变得极其庞大，随之而来的是评估新测试数据所需的推理时间和存储需求。这一问题在SPADE[10]中已被指出，该方法同时利用了低层和高层特征图进行异常分割。由于计算限制，SPADE需要预选

基于较弱的图像级异常检测机制——即依赖于全图像深度特征表示（如最后一层特征图的全局平均池化）——在特征图的融合阶段进行像素级异常检测。这导致生成低分辨率、偏向ImageNet特征的全图像表示，可能对检测与定位性能产生负面影响。

这些问题可以通过使 \mathcal{M} 对更大尺寸和数量的图像具有有意义的可搜索性来解决，允许基于补丁的比较，这对异常检测和分割都有益。这要求编码在 \mathcal{M} 中的名义特征覆盖得以保留。不幸的是，随机子采样，特别是几个数量级的采样，会丢失 \mathcal{M} 中编码的名义特征覆盖所包含的重要信息（另见§4.4.2中的实验）。在这项工作中，我们使用核心集采样机制来减少 \mathcal{M} ，我们发现这能在保持性能的同时减少推理时间。

从概念上讲，核心集选择旨在找到一个子集 $\mathcal{S} \subset \mathcal{A}$ ，使得在 \mathcal{A} 上的问题解能够最接近且尤其更快速地通过 \mathcal{S} 上计算出的解来近似[1]。根据具体问题的不同，所关注的核心集也会有所变化。由于*PatchCore*使用了最近邻计算（下一节），我们采用 *minimax facility location* 核心集选择方法（参见例如[48]和[49]），以确保在补丁级特征空间中， \mathcal{M} -核心集 \mathcal{M}_C 的覆盖范围与原始记忆库 \mathcal{M} 大致相似。

$$\mathcal{M}_C^* = \arg \min_{\mathcal{M}_C \subseteq \mathcal{M}} \max_{m \in \mathcal{M}} \min_{n \in \mathcal{M}_C} \|m - n\|_2. \quad (5)$$

精确计算 \mathcal{M}_C^* 是NP难题[54]，我们采用[48]提出的迭代贪心近似算法。为进一步缩短核心集选择时间，我们遵循[49]的方法，利用Johnson-Lindenstrauss定理[11]通过随机线性投影 $\psi: \mathbb{R}^d \rightarrow \mathbb{R}^{d^*}$ （其中 $d^* < d$ ）来降低元素 $m \in \mathcal{M}$ 的维度。内存库缩减流程总结于算法1。在符号表示上，我们

使用 $PatchCore - n\%$ 表示原始记忆库被二次采样到的百分比 n ，例如 $PatchCore - 1\%$ 表示 \mathcal{M} 的 100 倍缩减。图 3 直观对比了贪婪核心集二次采样与随机选择的空间覆盖范围。

Algorithm 1: *PatchCore* memory bank.

Input: Pretrained ϕ , hierarchies j , nominal data \mathcal{X}_N , stride s , patchsize p , coresset target l , random linear projection ψ .

Output: Patch-level Memory bank \mathcal{M} .

Algorithm:

```

 $\mathcal{M} \leftarrow \{\}$ 
for  $x_i \in \mathcal{X}_N$  do
     $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{P}_{s,p}(\phi_j(x_i))$ 
end
/* Apply greedy coresset selection. */
 $\mathcal{M}_C \leftarrow \{\}$ 
for  $i \in [0, \dots, l - 1]$  do
     $m_i \leftarrow \arg \max_{m \in \mathcal{M} - \mathcal{M}_C} \min_{n \in \mathcal{M}_C} \|\psi(m) - \psi(n)\|_2$ 
     $\mathcal{M}_C \leftarrow \mathcal{M}_C \cup \{m_i\}$ 
end
 $\mathcal{M} \leftarrow \mathcal{M}_C$ 

```

3.3. 使用 *PatchCore* 进行异常检测

利用名义补丁特征记忆库 \mathcal{M} ，我们通过测试图像 x^{test} 的补丁集合 $\mathcal{P}(x^{\text{test}}) = \mathcal{P}_{s,p}(\phi_j(x^{\text{test}}))$ 中测试补丁特征与 \mathcal{M} 中各自最近邻 m^* 之间的最大距离分数 s^* 来估算其图像级异常分数 $s \in \mathbb{R}$:

$$m^{\text{test},*}, m^* = \arg \max_{m^{\text{test}} \in \mathcal{P}(x^{\text{test}})} \arg \min_{m \in \mathcal{M}} \|m^{\text{test}} - m\|_2 \quad (6)$$

$$s^* = \|m^{\text{test},*} - m^*\|_2.$$

为了获得 s ，我们使用缩放 w 对 s^* 进行处理，以考虑相邻图像块的行为：如果最接近异常候选 $m^{\text{test},*}$ 、 m^* 的记忆库特征本身远离邻近样本，因此已经是罕见的正常情况，我们会提高异常分数。

$$s = \left(1 - \frac{\exp \|m^{\text{test},*} - m^*\|_2}{\sum_{m \in \mathcal{N}_b(m^*)} \exp \|m^{\text{test},*} - m\|_2} \right) \cdot s^*, \quad (7)$$

其中 $\mathcal{N}_b(m^*)$ 是测试补丁特征 m^* 在 \mathcal{M} 中的 b 个最近邻补丁特征。我们发现这种重新加权比仅使用最大补丁距离更为稳健。给定 s 后，分割结果可直接得出。公式 7（第一行）中的图像级异常分数需要通过 $\arg \max$ 运算计算每个补丁的异常分数。类似 [14] 的方法，可通过基于计算出的补丁异常分数进行重新对齐，在同一步骤中生成分割图。

在各自的空间位置上。为了匹配原始输入分辨率（我们可能希望使用中间网络特征），我们通过双线性插值对结果进行上采样。此外，我们使用核宽度为 $\sigma = 4$ 的高斯核对结果进行了平滑处理，但未对此参数进行优化。

4. 实验

4.1. 实验细节

数据集。为了研究工业异常检测性能，我们的大部分实验都在 MVTec 异常检测基准 [5] 上进行。

MVTec AD 包含 15 个子数据集，总计 5354 张图像，其中 1725 张属于测试集。每个子数据集均划分为仅包含正常样本的训练数据，以及针对特定产品，涵盖多种缺陷类型的测试集——测试集同时包含正常与异常样本，并提供相应的异常真实标注掩码。遵循 [10, 14, 56] 的设置，图像分别被调整尺寸并中心裁剪至 256×256 和 224×224 的规格。未采用数据增强技术，因为这需要关于保持类别特性的增强方法的先验知识。

我们还研究了更专业化任务上的工业异常检测。为此，我们利用 [42] 中使用的 *Magnetic Tile Defects (MTD)* [26] 数据集，该数据集包含 925 张无缺陷和 392 张有缺陷的磁砖图像，这些图像具有不同的光照水平和尺寸。与 [42] 相同，我们在测试时评估 20% 的无缺陷图像，其余用于冷启动训练。

最后，我们还强调了 *PatchCore* 在非工业图像数据上的潜在适用性，参照 [52] 和 [14] 等研究的方法，在 *Mini Shanghai Tech Campus (mSTC)* 上进行了冷启动异常定位的基准测试。*mSTC* 是原始 *STC* 数据集 [32] 的降采样版本，仅使用每五帧的训练和测试视频帧。该数据集包含来自 12 个不同场景的行人视频。训练视频仅包含正常行人行为，而测试视频可能包含打架或骑自行车等不同行为。为确保冷启动实验的可比性，我们遵循既定的 *mSTC* 协议 [14, 52]，不使用任何异常监督信息，并将图像尺寸调整为 256×256 。

评估指标。图像级异常检测性能通过使用生成的异常分数计算接收者操作特征曲线下面积 (AUROC) 来衡量。根据先前的工作，我们在 MVTec 上计算类别平均 AUROC [2, 10, 14]。为评估分割性能，我们首先同时使用像素级 AUROC 和 PRO 指标，两者均遵循 [6] 的方法。PRO 分数考虑了连通异常区域的重叠与恢复情况，以更好地适应 MVTec AD 数据集中不同尺寸的异常，详见 [6]。

表1. MVTec AD [5]上的异常检测性能 (AUROC)。PaDiM*表示来自[14]的采用问题特定主干网络选择的结果。误分类总数是在给定F1最优阈值下，假阳性与假阴性预测之和。由于缺乏竞争方法的个体异常分数，我们仅能为PatchCore计算该数值。

Method	SPADE [10]	PatchSVDD [56]	DifferNet [42]	PaDiM [14]	Mah.AD [40]	PaDiM* [14]	PatchCore-25%	PatchCore-10%	PatchCore-1%
AUROC \uparrow	85.5	92.1	94.9	95.3	95.8	97.9	99.1	99.0	99.0
Error \downarrow	14.5	7.9	5.1	4.7	4.2	2.1	0.9	1.0	1.0
Misclassifications \downarrow	-	-	-	-	-	-	42	47	49

表2. MVTec AD [5]上的异常分割性能 (像素级AUROC)。

Method	AE _{SSIM} [5]	γ -VAE + grad. [15]	CAVGA-R _w [52]	PatchSVDD [56]	SPADE [10]	PaDiM [14]	PatchCore-25%	PatchCore-10%	PatchCore-1%
AUROC \uparrow	87	88.8	89	95.7	96.0	97.5	98.1	98.1	98.0
Error \downarrow	13	11.2	11	4.3	4.0	2.5	1.9	1.9	2.0

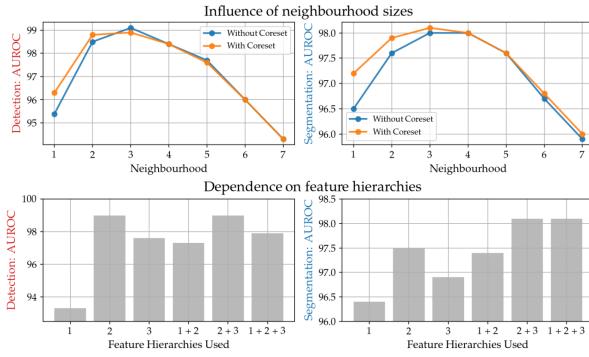


图4. 局部感知与网络特征深度对检测性能的影响。补充材料中提供了PRO评分结果。

4.2. MVTec AD数据集上的异常检测

MVTec数据集上图像级异常检测的结果如表1所示。对于PatchCore，我们报告了内存库不同子采样比例（2%、10%和1%）下的性能。在所有情况下，PatchCore均实现了显著更高的平均图像异常检测性能，且在所有子数据集上均保持稳定优异的表现（详细对比见补充材料B）。需注意的是，将误差从2.1%（PaDiM）降低至PatchCore-25%的0.9%，意味着误差减少了57%。在工业检测场景中，这是具有实际意义的显著改进。在MVTec数据集最佳F1阈值下，1725张图像中仅有42张被错误分类，且三分之一的类别达到完全正确分类。补充材料B进一步表明，无论在F1最优工作点还是全召回率下，其分类误差均低于SPADE和PaDiM。采用PatchCore时，误分类图像不足50张。此外，PatchCore在异常分割任务中达到最先进水平：像素级AUROC指标（表2，98.1对比PaDiM的97.5）与PRO指标（表3，93.5对比92.1）均表现优异。图1中的分割样本直观展示了其精确的异常定位能力。

此外，得益于我们核心集记忆子采样的高效性，我们能够将PatchCore-1%应用于更高分辨率的图像（例如280/320而非224），并提

在保持推理时间低于PatchCore-默认分辨率10%的同时，我们进一步提升了图像级和像素级异常检测性能，如表4所示（详细结果见附录）。部分指标甚至再次将误差减半以上（例如图像级AUROC从1% \rightarrow 降至0.4%）。

4.3. 推理时间

我们关注的另一个维度是推理时间。我们在表5中报告了结果（实现细节见附录A），并与使用WideResNet50并在可能情况下在GPU上运行的SPADE[10]和PaDiM[14]复现版本进行了比较。这些推理时间包括骨干网络的前向传播。可以看出，PatchCore-100%（未进行子采样）的联合图像级与像素级异常检测推理时间低于SPADE[10]，但性能更高。通过核心集子采样，PatchCore可以进一步加速，其推理时间甚至低于PaDiM，同时保持最先进的图像级异常检测与分割性能。最后，我们研究了采用近似最近邻搜索（IVFPQ[27]）的PatchCore-100%作为另一种降低推理时间的正交方法（该方法也可应用于SPADE，但其性能甚至已明显低于PatchCore-1%）。我们发现性能有所下降，尤其是在图像级异常检测方面，而推理时间仍高于PatchCore-1%。但即使性能下降，PatchCore-100%的近似最近邻搜索仍优于其他方法。结合核心集与近似最近邻搜索将进一步减少推理时间，从而能够扩展到更大的数据集。

4.4. 消融研究

我们报告了局部感知补丁特征和核心集缩减方法的消融实验。补充实验显示了不同骨干网络之间的一致性（§C.2）、随图像分辨率提升的可扩展性（§C.3）以及对剩余错误的定性分析（§C.4）。

表3. 在MVTec AD [5]数据集上以PRO [%] [5, 10]衡量的异常检测性能。

Method	AE_{SSIM} [5]	Student [6]	SPADE [10]	PaDiM [14]	PatchCore-25%	PatchCore-10%	PatchCore-1%
PRO \uparrow	69.4	85.7	91.7	92.1	93.4	93.5	93.1
Error \downarrow	30.6	14.3	8.3	7.9	6.6	6.5	6.9

表4. 采用更高分辨率/更大骨干网络/集成模型的PatchCore-1%。核心集子采样技术使得计算成本高昂的配置成为可能，同时仍保持快速推理能力。

Metric \rightarrow	AUROC	pwAUROC	PRO
DenseN-201 & RNext-101 & WRN-101 (2+3), Imagesize 320			
Score \uparrow	99.6	98.2	94.9
Error \downarrow	0.4	1.8	5.6
WRN-101 (2+3), Imagesize 280			
Score \uparrow	99.4	98.2	94.4
Error \downarrow	0.6	1.8	5.6
WRN-101 (1+2+3), Imagesize 280			
Score \uparrow	99.2	98.4	95.0
Error \downarrow	0.8	1.6	5.0

表5. MVTec AD上每张图像的平均推理时间。分数为（图像AUROC、像素AUROC、PRO指标）。

Method	PatchCore-100%	PatchCore-10%	PatchCore-1%
Scores	(99.1, 98.0, 93.3)	(99.0, 98.1, 93.5)	(99.0, 98.0, 93.1)
Time (s)	0.6	0.22	0.17
Method	PatchCore-100% + IVFPQ	SPADE	PaDiM
Scores	(98.0, 97.9, 93.0)	(85.3, 96.6, 91.5)	(95.4, 97.3, 91.8)
Time (s)	0.2	0.66	0.19

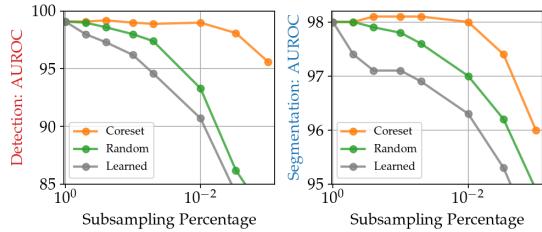


图5. 不同子采样器的性能保持情况，补充材料中PRO分数的结果。

4.4.1 局部感知的补丁特征与层次结构

我们通过评估公式1中不同邻域大小下异常检测性能的变化，研究了局部感知图像块特征的重要性（§3.3）。图4上半部分的结果显示，基于图像块的异常预测在局部性与全局上下文之间存在明确的最优点，因此确定了邻域大小 $p = 3$ 。通过向网络层级下游移动（参见例如[10, 14]）也可以获得更全局的上下文信息，但代价是分辨率降低和更强的ImageNet类别偏差（§3.1）。将前三个WideResNet50块索引为1-3，图4 (bottom) 再次强调了高度局部化预测、更全局的上下文与ImageNet偏差之间的最优平衡。可见，来自层级2的特征

已经能够达到最先进的性能，但受益于从后续层次级别提取的额外特征图 (2+3，此为默认设置选择)。

4.4.2 核心集子采样重要性

图5比较了不同记忆库 \mathcal{M} 子采样方法：贪婪核心集选择、随机子采样以及学习一组与子采样目标百分比 p_{target} 对应的基础代理。对于最后一种方法，我们使用 $|\mathcal{P}| = p_{target} \cdot |\mathcal{M}|$ 采样代理 $p_i \in \mathcal{P} \subset \mathbb{R}^d$ ，这些代理随后被用于最小化基础重构目标。

$$\mathcal{L}_{rec}(m_i) = \left\| m_i - \sum_{p_k \in \mathcal{P}} \frac{e^{\|m_i - p_k\|_2}}{\sum_{p_j \in \mathcal{P}} e^{\|m_i - p_j\|_2}} p_k \right\|_2^2, \quad (8)$$

寻找最能描述记忆库数据 \mathcal{M} 的 N 代理。在图5中，我们比较了三种设置，发现基于核心集的子采样性能优于其他可能的选择。未进行子采样的性能与缩小两个数量级的核心集减记记忆库相当。我们还发现子采样后的记忆库冗余度大幅降低。我们记录了非子采样与核心集子采样记忆库在测试阶段实际使用的样本比例：初始状态下仅有不到30%的记忆库样本被调用，而经过核心集子采样（压缩至1%）后，该比例提升至近95%。在特定子采样区间（约50%至10%之间），我们甚至观察到异常检测与定位的联合性能相较于未子采样的PatchCore出现部分提升。最后，通过增加步幅来缩减记忆库大小 \mathcal{M} （见公式3）会因分辨率上下文减少导致性能下降——步长 $s = 2$ 时图像异常检测AUROC为97.6%，步长 $s = 3$ 时AUROC为96.8%。

4.5. 低样本异常检测

仅能获取有限的正常数据是现实世界检测中的一个相关场景。因此，除了报告完整MVTec AD数据集上的结果外，我们还研究了在训练样本较少时的性能表现。我们将训练样本数量从1个（相当于总正常训练数据的0.4%）变化到50个（21%），并与使用相同骨干网络（WideResNet50）重新实现的SPADE[10]和PaDiM[14]进行比较。结果总结于图6中，详细结果可在补充材料中查阅。

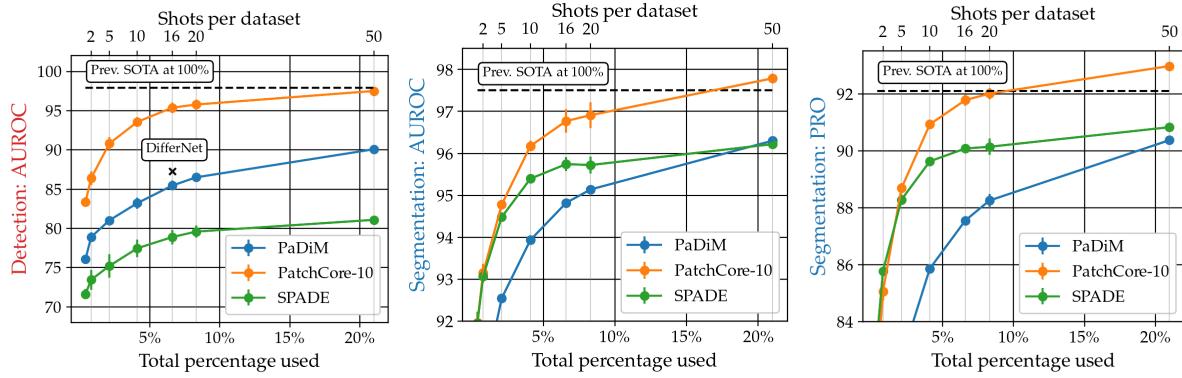


图6. *PatchCore* 在样本效率上显著优于其他方法，仅用少量名义训练数据就达到了先前最先进技术的水平。请注意，为保持可比性，*PaDiM*和*SPADE*均采用WideResNet50进行了重新实现。

表6. 在mSTC [32, 52]上的异常分割及在MTD [26]上的异常检测，与[42]中报告结果的对比。

mSTC	CAVGA-R _u [52]	SPADE [10]	PaDiM [14]	<i>PatchCore</i> -10
Pixelwise AUROC [%]	85	89.9	91.2	91.8
MTD	GANomaly [2]	1-NN [35]	DifferNet [42]	<i>PatchCore</i> -10
AUROC [%]	76.6	80.0	97.7	97.9

§C.1. 如图所示，仅使用五分之一的标称训练数据，*PatchCore* 仍能达到以往最先进方法的性能。此外，与文献[42]中进行的16样本实验相比，我们发现 *PatchCore* 的表现优于他们的方法——该方法在已预训练的特征之上适配了一个标准化流模型。与文献[10]中的图像级记忆方法相比，我们仅用5/1的标称样本数就实现了相当的定位与检测性能。

4.6. 在其他基准测试上的评估

我们在两个额外的异常检测性能基准上对*PatchCore* 进行了评估：上海科技大学校园数据集（STC）[32]和磁砖缺陷数据集（MTD）[26]。STC的评估如§4.1所述，遵循[52]、[14]和[10]的方法。我们在STC视频数据的子采样版本（mSTC）上报告了无监督异常定位性能，图像尺寸调整为 256×256 [14]。由于检测环境更接近ImageNet中可用的自然图像数据，我们利用了层次级别3和4的更深层网络特征图，但除此之外未对*PatchCore* 进行任何超参数调整。表6 (top) 中的结果显示，最先进的异常定位性能表明*PatchCore*在此类领域具有良好的可迁移性。最后，我们研究了MTD数据集，其中包含不同尺寸的磁砖缺陷图像，空间刚性方法（如*PaDiM*）无法直接应用。在此，正常数据已表现出与异常样本中类似的高变异性[42]。我们遵循[42]提出的协议来测量图像级异常检测性能，并发现性能达到（甚至超越）

略微优于）[42]的结果（表6, bottom）。

5. 结论

本文介绍了用于冷启动异常检测的*PatchCore*算法，该算法仅需利用正常样本的知识，即可在测试时检测并分割异常数据。*PatchCore*通过使用包含局部感知的正常图像块级特征表示的记忆库（这些特征提取自ImageNet预训练网络），在测试时保留最大程度的正常上下文信息，同时通过核心集子采样实现最小化运行时开销，从而在两者间取得平衡。最终形成了一种在工业异常检测基准上具有低计算成本的先进冷启动图像异常检测与定位系统。在MVTec数据集上，我们在相关小规模训练集场景中以最高样本效率实现了超过99%的图像异常检测AUROC。

更广泛的影响。由于自动化工业异常检测是计算机视觉最成功的应用之一，通过*PatchCore*获得的改进对该领域的从业者具有显著意义。由于我们的工作专门聚焦于工业异常检测，其负面影响有限。尽管这一基础方法可能被应用于更具争议性领域的检测系统，但我们认为，我们的改进尚不足以显著改变此类系统的社会应用方式。

局限性。尽管*PatchCore*在工业异常检测中表现出高效性，无需针对特定问题领域进行专门适配，但其适用性通常受限于所利用预训练特征的可迁移性。这一问题可以通过结合*PatchCore*的有效性与对所用特征的适配来解决。我们将这一有趣的扩展留待未来工作。

致谢

我们感谢Yasser Jadidi和Alex Smola对我们计算基础设施的搭建支持。K.R.感谢国际马克斯·普朗克智能系统研究院（IMPRS-IS）以及欧洲学习与智能系统实验室（ELLIS）博士项目的支持。

参考文献

- [1] Pankaj Agarwal、Sariel Har、Peled Kasturi 和 R Varadarajan。通过核心集进行几何近似。*Combinatorial and Computational Geometry*, 第52卷, 2004年11月。第2、4页[2] Samet Akcay、Amir Atapour-Abarghouei 和 Toby P Breckon。Ganomaly：通过对抗训练进行半监督异常检测。收录于*Asian Conference on Computer Vision*, 第622–637页。Springer, 2018年。第1、2、5、8、4页[3] Jerome Andrews、Thomas Tanay、Edward Morton 和 Lewis Griffin。用于异常检测的迁移表示学习。2016年7月。第4页[4] Liron Bergman、Niv Cohen 和 Yedid Hoshen。深度最近邻异常检测。*CoRR*, abs/2002.10445, 2020年。第1、2、3页[5] Paul Bergmann、Michael Fauser、David Sattlegger 和 Carsten Steger。Mvtect ad – 一个用于无监督异常检测的综合真实世界数据集。收录于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019年6月。第1、2、5、6、7、4页[6] Paul Bergmann、Michael Fauser、David Sattlegger 和 Carsten Steger。不知情的学生：具有判别性潜在嵌入的师生异常检测。收录于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020年6月。第2、5、7、4页[7] Paul Bergmann、Sindy Löwe、Michael Fauser、David Sattlegger 和 Carsten Steger。通过将结构相似性应用于自编码器来改进无监督缺陷分割。*Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019年。第2页[8] Wieland Brendel 和 Matthias Bethge。在 ImageNet 上，用局部特征袋模型近似 CNN 效果出奇地好。收录于*International Conference on Learning Representations*, 2019年。第2页[9] Kenneth L. Clarkson。核心集、稀疏贪婪近似和 Frank-Wolfe 算法。*ACM Trans. Algorithms*, 第6卷, 第4期, 2010年9月。第2页[10] Niv Cohen 和 Yedid Hoshen。基于深度金字塔对应关系的子图像异常检测。*CoRR*, abs/2005.02357, 2020年。第1、2、3、4、5、6、7、8页[11] Sanjoy Dasgupta 和 Anupam Gupta。Johnson-Lindenstrauss 定理的一个初等证明。*Random Structures & Algorithms*, 第22卷, 第1期, 第60–65页, 2003年。第4页[12] Diana Davletshina、Valentyn Melnychuk、Viet Tran、Hitansh Singla、Max Berrendorf、Evgeniy Ferman、Michael Fromm 和 Matthias Schubert。X射线图像的无监督异常检测, 2020年。第1页[13] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft。使用生成对抗网络进行图像异常检测。载于 Michele Berlingerio, Francesco Bonchi, Thomas Gärtner, Neil Hurley, and Georgiana Ifrim 编辑的 *Machine Learning and Knowledge Discovery in Databases*, 第3–17页, Cham, 2019年。Springer International Publishing。[14] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim：一种用于异常检测与定位的块分布建模框架。载于 Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani 编辑的 *Pattern Recognition. ICPR International Workshops and Challenges*, 第475–489页, Cham, 2021年。Springer International Publishing。2, 3, 5, 6, 7, 8, 1, 4[15] David Dehaene, Oriel Frigo, Sébastien Combexelle, and Pierre Eline. 基于迭代能量投影的正常数据流形异常定位。载于 *International Conference on Learning Representations*, 2020年。6, 4[16] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: 一个大规模分层图像数据库。载于 *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 第248–255页, 2009年。2[17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio。使用 Real NVP 进行密度估计。载于 *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*。OpenReview.net, 2017年。2[18] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo。A Geometric Framework for Un-supervised Anomaly Detection, 第77–101页。Springer US, Boston, MA, 2002年。2[19] Dan Feldman, Matthew Faulkner, and Andreas Krause。通过核心集进行混合模型的可扩展训练。载于 J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger 编辑的 *Advances in Neural Information Processing Systems*, 第24卷, 第2142–2150页。Curran Associates, Inc., 2011年。2[20] Izhak Golan and Ran El-Yaniv。使用几何变换进行深度异常检测。载于 S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett 编辑的 *Advances in Neural Information Processing Systems*, 第31卷, 第9758–9769页。Curran Associates, Inc., 2018年。2, 4[21] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel。记忆常态以检测异常：用于无监督异常检测的记忆增强深度自编码器。载于 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019年10月。2[22] Sariel Har-Peled and Akash Kushal。用于 k-中值和 k-均值聚类的更小核心集。*Discrete and Computational Geometry*, 37:3–19, 2007年12月。2[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun。用于图像识别的深度残差学习。载于 *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016年6月。3, 6 [24] Geoffrey Hinton, Oriol Vinyals, 和 Jeffrey Dean。蒸馏神经网络中的知识。于 *NIPS Deep Learning and Representation Learning Workshop*, 2015年。2 [25] Chaoqing Huang, Jinkun Cao, Fei Ye, Maosen Li, Ya Zhang, 和 Cewu Lu。用于异常检测的逆变换自编码器。于 *CoRR*, abs/1911.10676, 2019年。2, 4 [26] Yibin Huang, C. Qiu, 和 K. Yuan。磁砖表面缺陷的显著性。

The Visual Computer, 36:85–96, 2018年。2, 5, 8 [27] Jeff Johnson, Matthijs Douze, 和 Hervé Jégou。使用GPU进行十亿级相似性搜索。于 *IEEE Transactions on Big Data*, 页码 1–1, 2019年。6, 1 [28] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, 和 Michael Terry。XRAI: 通过区域实现更好的归因。

于 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019年10月。2 [29] Ki Hyun Kim, Sangwoo Shim, Yongsu Lim, Jongseob Jeon, Jeongwoo Choi, Byungchan Kim, 和 Andre S. Yoon。RAPP: 沿投影路径重建的新颖性检测。于 *International Conference on Learning Representations*, 2020年。2 [30] Durk P Kingma 和 Prafulla Dhariwal。Glow: 使用可逆 1×1 卷积的生成流。于 S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, 和 R. Garnett, 编辑, *Advances in Neural Information Processing Systems*, 卷 31。Curran Associates, Inc., 2018年。2 [31] Wenyuan Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyang Wu, Bir Bhanu, Richard J. Radke, 和 Octavia Camps。面向视觉解释变分自编码器。于 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020年6月。4 [32] W. Liu, D. Lian, W. Luo, 和 S. Gao。用于异常检测的未来帧预测——一个新的基线。于 2018 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018年。5, 8 [33] Prasanta Chandra Mahalanobis。论统计学中的广义距离。

Proceedings of the National Institute of Sciences (Calcutta), 2:49–55, 1936年。2 [34] Ben Mussay, Margarita Osadchy, Vladimir Braverman, Samson Zhou, 和 Dan Feldman。通过核心集进行数据无关的神经剪枝。于 *International Conference on Learning Representations*, 2020年。2 [35] Tiago S. Nazaré, Rodrigo Fernandes de Mello, 和 Moacir A. Ponti。预训练的CNN是监控视频异常检测的良好特征提取器吗? 于 *CoRR*, abs/1811.08495, 2018年。8 [36] Duc Tam Nguyen, Zhongyu Lou, Michael Klar, 和 Thomas Brox。使用多假设预测的异常检测。于 Kamalika Chaudhuri 和 Ruslan Salakhutdinov, 编辑, *Proceedings of the 36th International Conference on Machine Learning*, 卷 97, *Proceedings of Machine Learning Research*, 页码 4800–4809。PMLR, 2019年6月9–15日。1 [37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, 纳塔利娅·吉梅尔申、卢卡·安蒂加、阿尔班·德梅松、安德烈亚斯·K·普夫、爱德华·杨、扎克·德维托、马丁·雷森、阿里汗·特贾尼、萨桑克·奇拉姆库尔西、伯努瓦·施泰纳、方璐、白俊杰和苏米斯·钦塔拉。PyTorch: 一种命令式风格的高性能深度学习库, 2019年。1 [38] 普拉穆迪塔·佩雷拉、拉梅什·纳拉帕蒂和宾·项。OC-GAN: 使用具有约束潜在表示的生成对抗网络进行单类新颖性检测。于 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019年6月。2 [39] 斯坦尼斯拉夫·皮德霍斯基、拉尼娅·阿尔莫森、唐纳德·A·阿杰罗和詹弗兰科·多雷托。使用对抗性自编码器的生成概率新颖性检测。于 *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS18, 第6823–6834页, 美国纽约州红钩, 2018年。Curran Associates Inc. 1, 2 [40] 奥利弗·里佩尔、帕特里克·默滕斯和多里特·梅尔霍夫。在预训练深度特征中对正常数据分布进行建模以进行异常检测。于 *2020 25th International Conference on Pattern Recognition (ICPR)*, 第6726–6733页, 2021年。2, 6, 4 [41] 卡斯滕·罗特、蒂莫·米尔比奇、萨马思·辛哈、普拉蒂克·古普塔、比约恩·奥默和约瑟夫·保罗·科恩。重新审视深度度量学习中的训练策略和泛化性能。于哈尔·多姆é三世和阿蒂·辛格编辑的 *Proceedings of the 37th International Conference on Machine Learning*, 第119卷 *Proceedings of Machine Learning Research*, 第8242–8252页。PMLR, 2020年7月13–18日。3 [42] 马尔科·鲁道夫、巴斯蒂安·万特和博多·罗森哈恩。相同但又不同: 使用标准化流的半监督缺陷检测。于 *Winter Conference on Applications of Computer Vision (WACV)*, 2021年1月。1, 2, 5, 6, 8, 4 [43] 穆罕默德·萨博克鲁、穆罕默德·哈洛伊、马哈茂德·法西和埃桑·阿德利。对抗性学习的单类分类器用于新颖性检测。于 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018年6月。1, 2 [44] 樱田麻由和矢入武久。使用非线性降维自编码器进行异常检测。于 *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, MLSDA'14, 第4–11页, 美国纽约州纽约市, 2014年。计算机协会。1, 2 [45] 穆罕默德·礼萨·萨利希、尼乌沙·萨贾迪、索罗什·巴塞利扎德、穆罕默德·侯赛因·罗班和哈米德·R·拉比埃。用于异常检测的多分辨率知识蒸馏, 2020年。2 [46] 伯恩哈德·舍尔科普夫、罗伯特·C·威廉姆森、亚历克斯·J·斯莫拉、约翰·肖·泰勒和约翰·C·普拉特。用于新颖性检测的支持向量方法。于 *Advances in Neural Information Processing Systems 12*, 第582–588页, 美国马萨诸塞州剑桥, 2000年6月。马克斯·普朗克学会, 麻省理工学院出版社。2 [47] R. R. 塞尔瓦拉朱、M. 科格斯韦尔、A. 达斯、R. 韦丹塔姆、D. 帕里克和D. 巴特拉。Grad-CAM: 通过基于梯度的定位从深度网络进行可视化解释。于 2017 *IEEE*

International Conference on Computer Vision (ICCV), 第618–626页, 2017年。2 [48] Ozan Sener 和 Silvio Savarese。卷积神经网络主动学习的核心集方法。于 *International Conference on Learning Representations*, 2018年。2, 4 [49] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle 和 Augustus Odena。Small-GAN: 使用核心集加速GAN训练。于 Hal Daumé III 和 Aarti Singh 编辑, *Proceedings of the 37th International Conference on Machine Learning*, 第119卷 *Proceedings of Machine Learning Research*, 第9005–9015页。PM LR, 2020年7月13–18日。3, 4 [50] David M. J. Tax 和 Robert P. W. Duin。支持向量数据描述。*Machine Learning*, 54:45–66, 2004年。2 [51] Guido Van Rossum 和 Fred L. Drake。*Python 3 Reference Manual*。CreateSpace, 斯科茨谷, 加利福尼亚州, 2009年。1 [52] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh 和 Abhijit Mahalanobis。图像中注意力引导的异常定位。于 Andrea Vedaldi, Horst Bischof, Thomas Brox 和 Jan-Michael Frahm 编辑, *Computer Vision – ECCV 2020*, 第485–503页, Cham, 2020年。Springer International Publishing。2, 5, 6, 8, 4 [53] Ross Wightman。PyTorch图像模型。<https://github.com/rwightman/pytorch-image-models>, 2019年。1 [54] Laurence A. Wolsey 和 George L. Nemhauser。*Integer and Combinatorial Optimization*。离散数学与优化Wiley系列。Wiley, 2014年。4 [55] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu 和 Kaiming He。深度神经网络的聚合残差变换。于 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017年7月。6 [56] Jihun Yi 和 Sungroh Yoon。Patch SVDD: 用于异常检测与分割的块级SVDD。于 *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020年11月。1, 2, 5, 6, 4 [57] Sergey Zagoruyko 和 Nikos Komodakis。宽残差网络。于 Edwin R. Hancock, Richard C. Wilson 和 William A. P. Smith 编辑, *Proceedings of the British Machine Vision Conference (BMVC)*, 第87.1–87.12页。BMVA Press, 2016年9月。3, 1, 6 [58] Shuangfei Zhai, Yu Cheng, Weineng Lu 和 Zhongfei Zhang。用于异常检测的深度结构化能量模型。于 Maria Florina Balcan 和 Kilian Q. Weinberger 编辑, *Proceedings of The 33rd International Conference on Machine Learning*, 第48卷 *Proceedings of Machine Learning Research*, 第1100–1109页, 纽约, 纽约, 美国, 2016年6月20–22日。PMLR。4 [59] Zhou Wang, A. C. Bovik, H. R. Sheikh 和 E. P. Simoncelli。图像质量评估: 从误差可见性到结构相似性。*IEEE Transactions on Image Processing*, 13(4):600–612, 2004年。2 [60] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho 和 Haifeng Chen。用于无监督异常检测的深度自编码高斯混合模型。于 *International Conference on Learning Representations*, 2018年。2

补充：实现工业异常检测的全面召回

A. 实现细节

我们在Python 3.7 [51]和PyTorch [37]中实现了我们的模型。实验在Nvidia Tesla V4 GPU上运行。我们使用了来自torchvision和PyTorch Image Models仓库[53]的torchvision ImageNet预训练模型。默认情况下，遵循[10]和[14]的方法，*PatchCore*使用WideResNet50主干网络[57]以确保直接可比性。块2和块3的最终输出通过特征图聚合提取补丁级特征。对于所有最近邻检索和距离计算，我们使用faiss [27]。

B. 完整MVTec AD对比

本节包含对MVTec AD更详细的比较。我们纳入了更多模型，并在所有可用的MVTec AD子数据集上进行了更细粒度的性能比较。在论文主体部分，这已在§4.2中提及。相应的结果表格为S1、S2和S3。我们观察到 $\{v^*\}$ 在15个MVTec数据集中解决了六个，并在大多数数据集及平均表现上取得了最高的AUROC性能。

图S3展示了*PatchCore*变体以及使用WideResNet50骨干网络重新实现的、可比较的方法SPADEF[10]和PaDiM[14]的精确率-召回率曲线和ROC曲线。我们还绘制了在100%召回率下以及F1最优阈值下的分类误差，以提供一个可比较的工作点。可以看出，与SPADEF和PaDiM相比，*PatchCore*在定义的工作点上同样实现了持续较低的分类误差，并在各数据集上获得了接近最优的精确率-召回率曲线和ROC曲线。

最后，表S4展示了在所有MVTec AD子数据集上，针对更大图像尺寸（280 $\{v^*\}$ 280）和WideResNet-101骨干网络，使用PatchCore $\{v^*\}$ 1%以进一步提升性能的详细结果，该方法即使在处理较大图像时也能在推理阶段实现高效异常检测。

C. 额外消融实验与细节说明

C.1. 详细的小样本实验

本节为本文主体部分（§4.5）提供的低样本学习方法研究提供了详细的数值数据。相关结果已包含在表S5中，我们发现在检测与异常定位指标上均呈现出一致更高的数值。

C.2. 对预训练网络的依赖性

我们使用不同的骨干网络测试了*PatchCore*，结果如S6所示。我们发现，在不同骨干网络的选择下，结果基本保持稳定。选择WideResNet50是为了与SPADEF和PaDiM保持可比性。

C.3. 图像分辨率的影响

接下来我们研究图像尺寸对性能的影响。在主论文中，我们使用了 224×224 的尺寸以便与先前工作进行比较。在图S4中，我们将图像尺寸从 288×288 、 360×360 调整至 448×448 ，并将邻域尺寸（P）在3、5、7、9之间变化。我们观察到检测性能略有提升，且在*PatchCore*时性能趋于饱和。对于异常分割任务，我们观察到性能持续提升，因此若精确定位至关重要，这是需要验证的要素之一。

C.4. 剩余误分类

高图像级异常检测性能使我们能够详细审视所有剩余的误分类情况。我们采用F1最优值确定工作点（高于此阈值的分数被视为异常）。在此阈值下，共存在19个假阳性与23个假阴性误差，所有误差均可视化展示于图S1与图S2中。每个分割图均按阈值进行归一化处理，因此在某些情况下背景分数会呈现不成比例的突出显示。

观察图S1，我们发现大多数假阳性错误要么源于a)（蓝色部分）标注的模糊性，即图像变化也可能被标记为异常；要么源于b)（橙色部分）极高的名义方差，

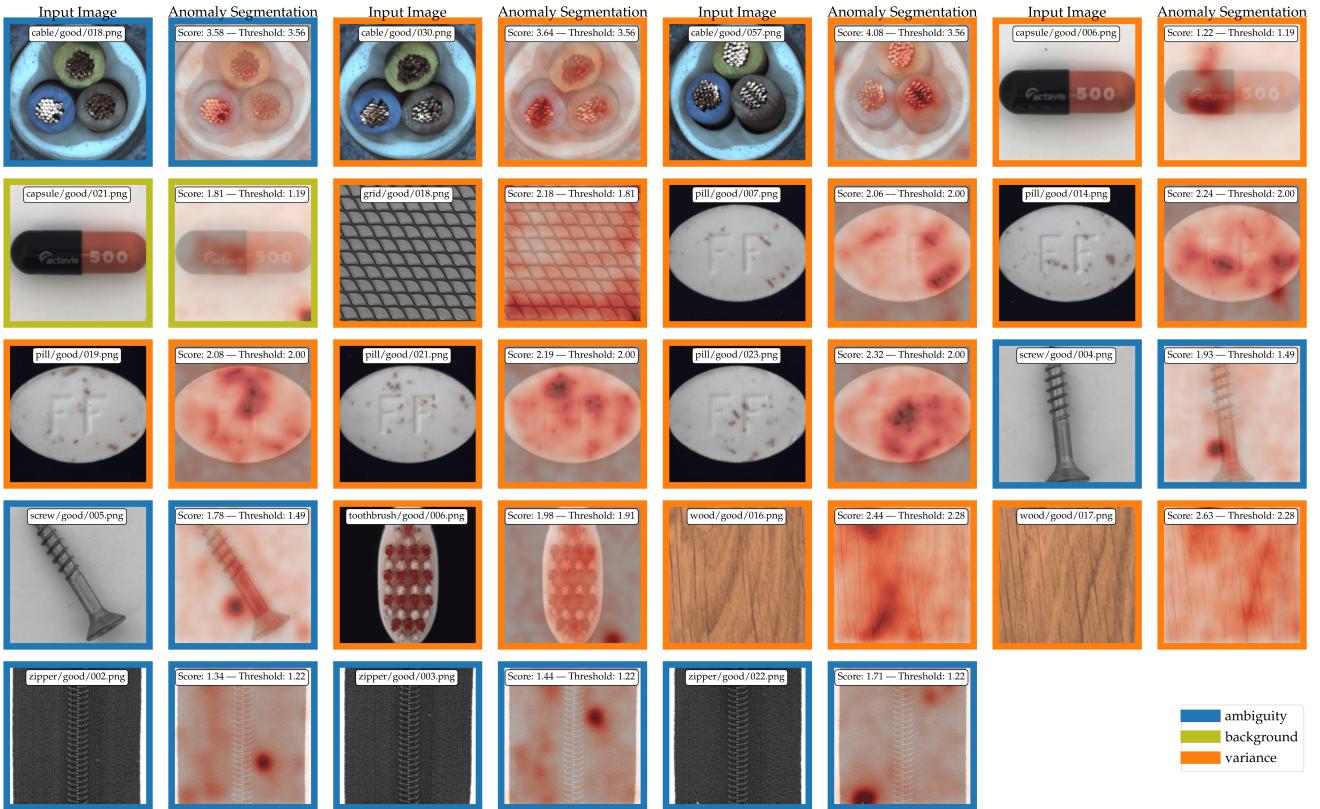


图 S1. 剩余误报分类的可视化（基于 F1 最优阈值）。颜色表示不同的错误来源。橙色表示将高程度的正常方差误判为异常，蓝色表示因标注上下文中的异常导致的误分类，橄榄色表示将背景变化误判为异常内容。

类似于潜在的异常。虽然前者很难通过现有方法解决，但后者可以通过对名义数据进行某种形式的适应来应对。然而，由于 *PatchCore* 的表现优于自适应方法，这种适应在与基于预训练的方法（如 *PatchCore*）结合使用时，将展现出最大的潜力。

为了理解产生的假阴性错误，我们在图 S2 中包含了生成的语义分割图和真实标注掩码。如图所示，大部分异常区域被较好地定位，但异常区域分配的权重不足，可能通过某些后处理手段得到改善。其他误分类主要由两种情况导致：一是被误判为异常场景的高度正常性变异，二是提升图像分辨率后可能捕捉到的细粒度异常。相比之下，完全漏检的异常数量较少，其中一例是由于图像预处理裁剪掉了实际异常区域所致。

C.5. 局部感知与子采样

为了完整性，我们在 S5 和 S6 中重复了主论文中的图 4 和图 5，并包含了 PRO 评分结果。

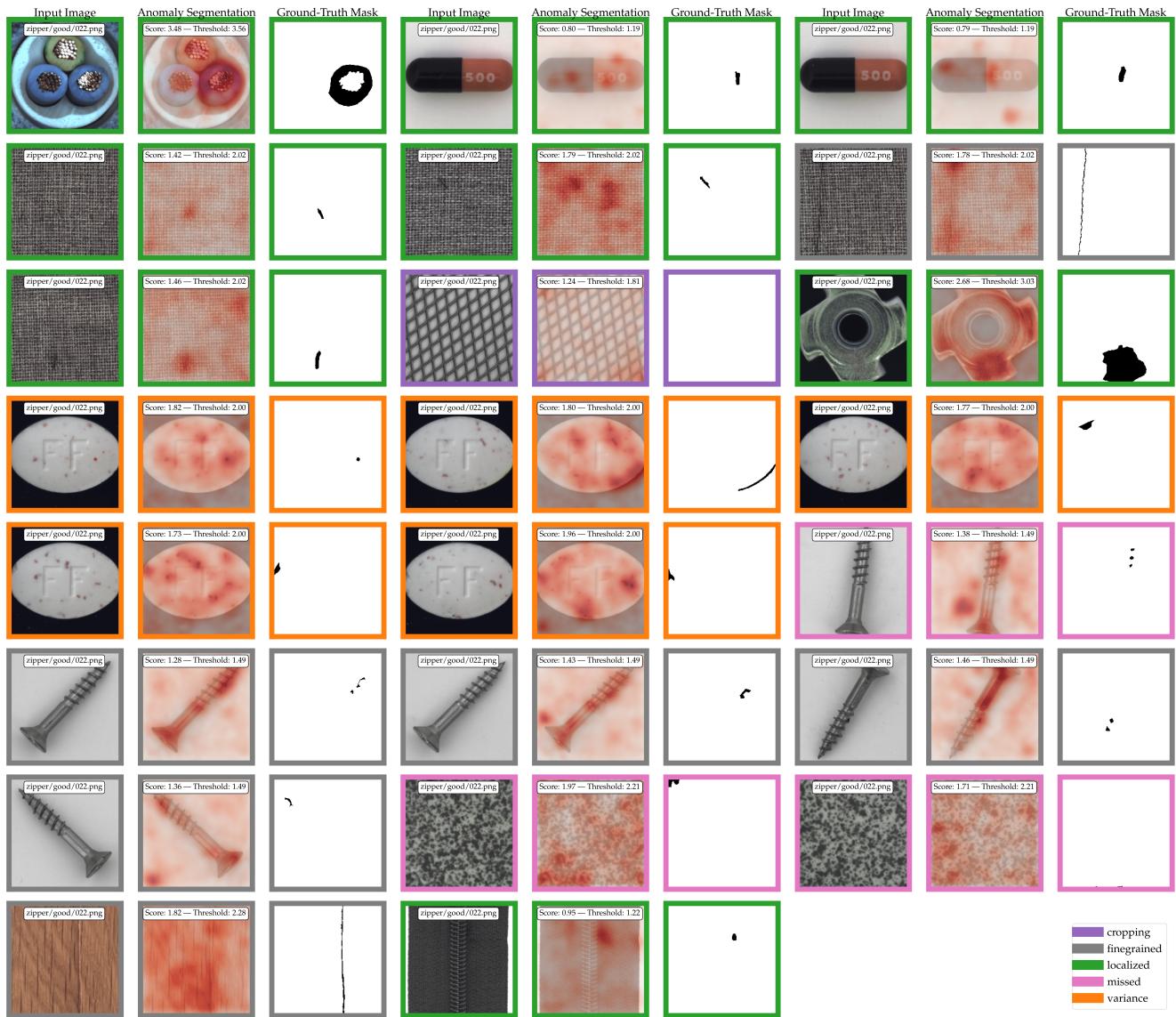


表 S1. MVTec AD [5] 上的异常检测性能 (AUROC)。PaDiM* 表示来自 [14] 的结果，其主干网络专门为图像级异常检测任务选取，我们未能复现该结果。

\downarrow Method \Dataset	Avg	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
GeoTrans [20]	67.2	74.4	78.3	67.0	43.7	61.9	35.9	84.1	81.3	63.0	50.0	41.7	97.2	86.9	61.1	82.0
GANomaly [2]	76.2	89.2	75.7	73.2	69.9	70.8	78.5	84.2	70.0	74.3	74.6	79.4	65.3	79.2	83.4	74.5
DSEBM [58]	70.9	81.8	68.5	59.4	41.3	71.7	76.2	41.6	67.9	80.6	99.9	69.0	78.1	74.1	95.2	58.4
OCSVM [3]	71.9	99.0	80.3	54.4	62.7	41.0	91.1	88.0	61.1	72.9	74.7	87.6	61.9	56.7	95.3	51.7
ITAE [25]	83.9	94.1	83.2	68.1	70.6	88.3	85.5	86.2	66.7	78.6	100	73.5	100	84.3	92.3	87.6
SPADE [10]	85.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CAVGA-R _w [52]	90	96	92	93	88	84	97	89	82	86	81	97	89	99	79	96
PatchSVDD [56]	92.1	98.6	90.3	76.7	92.9	94.6	92.0	90.9	94.0	86.1	81.3	97.8	100	91.5	96.5	97.9
DifferNet [42]	94.9	99.0	95.9	86.9	92.9	84.0	99.3	97.1	96.1	88.8	96.3	99.4	98.6	91.1	99.8	95.1
PaDiM [14]	95.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MahalanobisAD [40]	95.8	100	95.0	95.1	100	89.7	99.1	100	94.7	88.7	85.2	99.8	96.9	95.5	99.6	97.9
PaDiM* [14]	97.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PatchCore-25	99.1	100	99.5	98.1	98.7	98.2	100	100	100	96.6	98.1	98.7	100	100	99.2	99.4
PatchCore-10	99.0	100	99.4	97.8	98.7	97.9	100	100	100	96.0	97.0	98.9	99.7	100	99.0	99.5
PatchCore-1	99.0	100	99.3	98.0	98.0	98.6	100	100	99.7	97.0	96.4	99.4	100	99.9	99.2	99.2

表S2. 在MVTec [5]上以像素级AUROC衡量的异常分割性能。

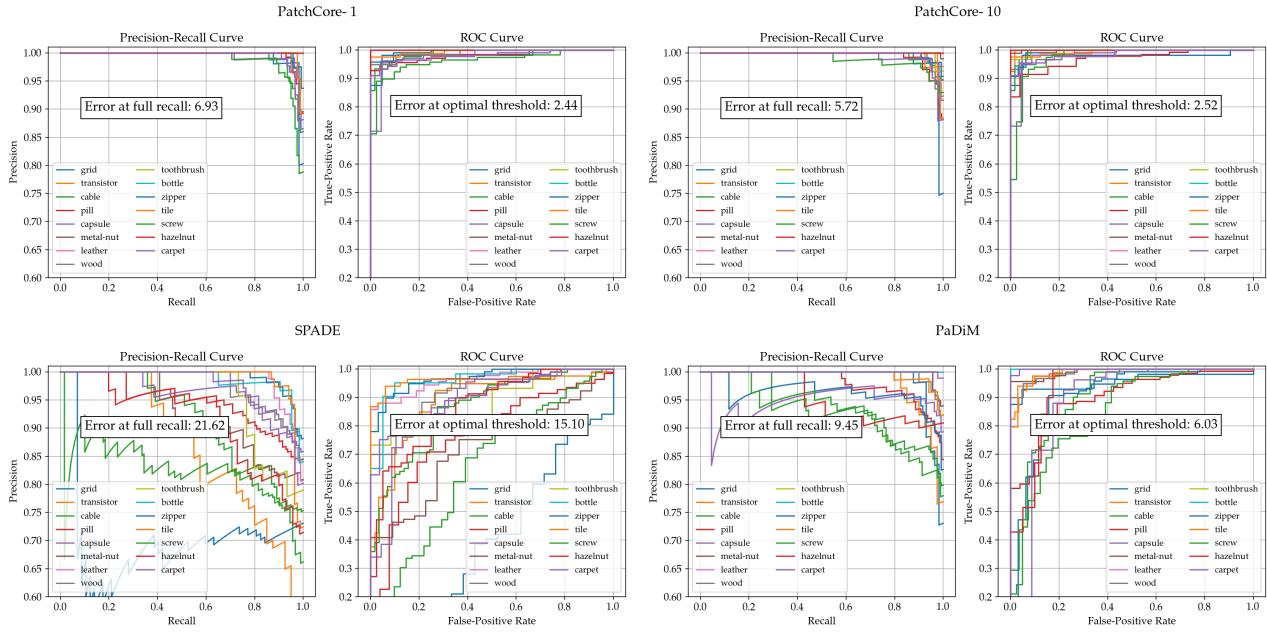
\downarrow Method \Dataset	Avg	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
vis. expl. VAE [31]	86	87	90	74	78	73	98	95	94	83	97	80	94	93	77	78
AE _{SSIM} [5]	87	93	82	94	87	94	97	78	89	91	96	59	92	90	73	88
γ -VAE + grad. [15]	88.8	93.1	88.0	91.7	72.7	97.9	98.8	89.7	91.4	93.5	97.2	58.1	98.3	93.1	80.9	87.1
CAVGA-R _w [52]	89	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PatchSVDD [56]	95.7	98.1	96.8	95.8	92.6	96.2	97.5	97.4	98.0	95.1	95.7	91.4	98.1	97.0	90.8	95.1
SPADE [10]	96.0	98.4	97.2	99.0	97.5	93.7	99.1	97.6	98.1	96.5	98.9	87.4	97.9	94.1	88.5	96.5
PaDiM [14]	97.5	98.3	96.7	98.5	99.1	97.3	98.2	99.2	97.2	95.7	98.5	94.1	98.8	94.9	94.9	98.5
PatchCore-25	98.1	98.6	98.4	98.8	99.0	98.7	98.7	99.3	98.4	97.4	99.4	95.6	98.7	96.3	95.0	98.8
PatchCore-10	98.1	98.6	98.5	98.9	99.1	98.7	98.7	99.3	98.4	97.6	99.4	95.9	98.7	96.4	95.1	98.9
PatchCore-1	98.0	98.5	98.2	98.8	98.9	98.6	98.6	99.3	98.4	97.1	99.2	96.1	98.5	94.9	95.1	98.8

表 S3. 在 MVTec [5] 数据集上的异常分割性能，以 PRO [%] [5, 10] 衡量。

\downarrow Method \Dataset	Avg	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
AE _{SSIM} [5]	69.4	83.4	47.8	86.0	64.7	84.9	91.6	56.1	60.3	83.0	88.7	17.5	78.4	72.5	60.5	66.5
Student [6]	85.7	91.8	86.5	91.6	69.5	81.9	93.7	81.9	89.5	93.5	92.8	91.2	86.3	70.1	72.5	93.3
SPADE [10]	91.7	95.5	90.9	93.7	94.7	86.7	95.4	97.2	94.4	94.6	96.0	75.6	93.5	87.4	87.4	92.6
PaDiM [14]	92.1	94.8	88.8	93.5	96.2	94.6	92.6	97.8	85.6	92.7	94.4	86.0	93.1	84.5	91.1	95.9
PatchCore-25	93.4	96.2	92.5	95.5	96.6	96.0	93.8	98.9	91.4	93.2	97.9	87.3	91.5	83.7	89.4	97.1
PatchCore-10	93.5	96.1	92.6	95.5	96.6	95.9	93.9	98.9	91.3	94.1	97.9	87.4	91.4	83.5	89.6	97.1
PatchCore-1	93.1	95.9	91.6	95.5	96.5	96.1	93.8	98.9	91.2	92.9	97.1	88.3	90.2	81.2	89.5	97.0

表 S4。使用更大图像 (280 × 280) 和 WideResNet101 骨干网络的 PatchCore-1 在 MVTec AD [5] 上的异常检测与定位性能 (AUROC)。

\downarrow Metric \Dataset	Avg	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
PatchCore-1, Hierarchies (2, 3), Imagesize 280																
AUROC	99.4	100	99.6	98.2	98.4	99.8	100	100	100	97.2	98.9	98.9	100	100	99.5	99.9
pwAUROC	98.2	98.6	98.4	99.1	98.7	98.7	98.8	99.3	98.8	97.8	99.3	96.1	98.8	96.4	95.1	98.9
PRO	94.4	96.6	93.8	96.0	97.4	96.8	91.2	99.1	94.8	94.0	97.5	89.5	95.5	84.8	91.7	97.8
PatchCore-1, Hierarchies (1, 2, 3), Imagesize 280																
AUROC	99.2	100	99.7	98.1	98.2	98.3	100	100	100	97.1	99.0	98.9	98.9	99.7	99.9	99.7
pwAUROC	98.4	98.6	98.7	99.1	98.7	98.8	98.8	99.3	99.0	98.6	99.5	96.3	98.9	97.1	95.2	99.0
PRO	95.0	96.6	94.6	96.3	97.5	97.0	91.5	99.1	95.4	96.0	98.1	90.0	95.8	85.9	92.0	98.0



图S3. PatchCore、变体及对比方法SPADE [10]和PaDiM [14]的精确率-召回率曲线（左）与ROC曲线（右）。曲线中不同颜色对应不同的MVTec类别。

表S5. 在MVTec [5]上测量的低样本异常检测性能，以AUROC为衡量指标。

\downarrow Method \ Shots \rightarrow	1	2	5	10	16	20	50
Retained %	0.4	0.8	2.1	4.1	6.6	8.3	21
IMAGE-LEVEL AUROC							
SPADE	71.6 ± 0.7	73.4 ± 1.3	75.2 ± 1.5	77.5 ± 1.1	78.9 ± 0.9	79.6 ± 0.8	81.1 ± 0.4
PaDiM	76.1 ± 0.4	78.9 ± 0.6	81.0 ± 0.2	83.2 ± 0.7	85.5 ± 0.6	86.5 ± 0.3	90.1 ± 0.3
DifferNet	-	-	-	-	87.3	-	-
PatchCore-10	83.4 ± 0.6	86.4 ± 0.9	90.8 ± 0.8	93.6 ± 0.6	95.4 ± 0.7	95.8 ± 0.6	97.5 ± 0.3
PatchCore-25	84.1 ± 0.7	87.2 ± 1.0	91.0 ± 0.9	93.8 ± 0.5	95.5 ± 0.6	95.9 ± 0.6	97.7 ± 0.4
PIXEL-LEVEL AUROC							
SPADE	91.9 ± 0.3	93.1 ± 0.2	94.5 ± 0.1	95.4 ± 0.1	95.7 ± 0.2	95.7 ± 0.2	96.2 ± 0.0
PaDiM	88.2 ± 0.3	90.5 ± 0.2	92.5 ± 0.1	93.9 ± 0.1	94.8 ± 0.1	95.1 ± 0.1	96.3 ± 0.0
PatchCore-10	92.0 ± 0.2	93.1 ± 0.2	94.8 ± 0.1	96.2 ± 0.1	96.8 ± 0.3	96.9 ± 0.3	97.8 ± 0.0
PatchCore-25	92.4 ± 0.3	93.3 ± 0.2	94.8 ± 0.1	96.1 ± 0.1	96.8 ± 0.3	96.9 ± 0.3	97.7 ± 0.0
PRO METRIC							
SPADE	83.5 ± 0.4	85.8 ± 0.1	88.3 ± 0.2	89.6 ± 0.1	90.1 ± 0.2	90.1 ± 0.3	90.8 ± 0.1
PaDiM	72.4 ± 1.2	77.8 ± 0.7	82.7 ± 0.2	85.9 ± 0.2	87.5 ± 0.2	88.2 ± 0.2	90.4 ± 0.1
PatchCore-10	82.4 ± 0.3	85.1 ± 0.3	88.7 ± 0.2	90.9 ± 0.1	91.8 ± 0.2	92.0 ± 0.2	93.0 ± 0.1
PatchCore-25	83.7 ± 0.5	86.0 ± 0.3	88.8 ± 0.2	90.9 ± 0.1	91.7 ± 0.1	91.9 ± 0.2	92.8 ± 0.0

表 e S6. 在MVTec [5]上测量的异常检测性能，以 AUROC。

\downarrow Backbone	% of \mathcal{M}	Img. AUROC	Pw. AUROC	PRO
ResNet50 [23]	10	99.0	98.1	93.3
	1	98.7	97.8	93.3
WideResNet50 [57]	10	98.9	98.1	93.5
	1	99.0	98.0	93.1
ResNet101 [23]	10	98.6	97.9	92.5
	1	98.7	97.8	92.2
WideResNet101 [57]	10	99.1	98.2	93.4
	1	99.0	98.1	93.0
ResNeXt101 [55]	10	98.9	98.0	92.8
	1	98.7	97.8	92.6

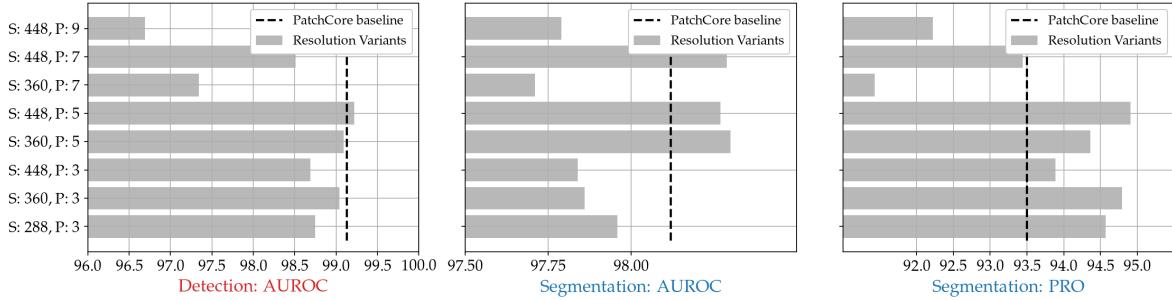


图 S4. 图像尺寸 (S) 和邻域尺寸 (P) 对 PatchCore 性能的影响。包含采用默认值的 PatchCore 基线以供参考。

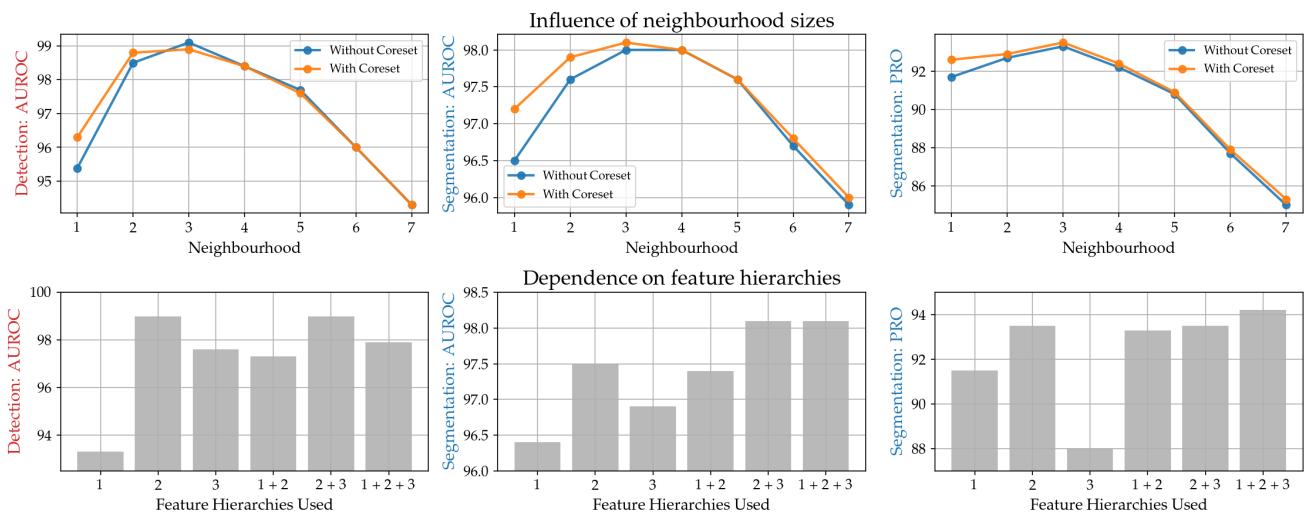


图 S5. 局部感知与网络特征深度对异常检测性能的影响。

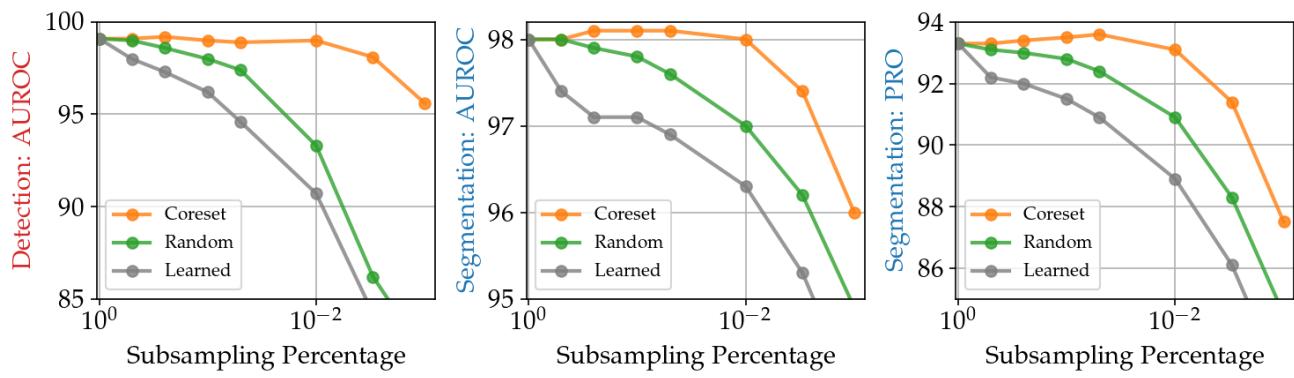


图 S6. 不同子采样器的性能保持情况。