PatchCore-n% は、元のメモリバンクがサブサンプリングされた割合 n を表します。例えば、PatchCore-1% は M を 100 倍に削減したことを意味します。図 3 は、貪欲なコアセットサブサンプリングとランダム選択の空間的カバー範囲の視覚的な比較を示しています。

アルゴリズム 1: PatchCore メモリバンク。

入力：事前学習済み ϕ、階層 j、公称データ X_N 、ストライド s、パッチサイズ p、コアセットターゲット l、ランダム線形投影 ϕ。
出力：パッチレベルのメモリバンク 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 coreset selection. */
$\mathcal{M}_C \leftarrow \{\}$
for $i \in [0, ..., l-1]$ **do**
  $m_i \leftarrow \underset{m \in \mathcal{M}-\mathcal{M}_C}{\arg\max} \min_{n \in \mathcal{M}_C} \|\psi(m) - \psi(n)\|_2$
  $\mathcal{M}_C \leftarrow \mathcal{M}_C \cup \{m_i\}$
end
$\mathcal{M} \leftarrow \mathcal{M}_C$

3.3. PatchCore による異常検出

標準パッチ特徴メモリバンク M を使用し、テスト画像 x^{test} の画像レベル異常スコア s $\in_R$ を、そのパッチ集合 $P(x^{test}) = P_{s,p}$ 内のテストパッチ特徴間の最大距離スコア s* により推定します。（$\phi_j(x^{test})$）とM内の各最寄りの近傍m*との最大距離スコアs*を計算します：

$$m^{\text{test},*}, m^* = \underset{m^{\text{test}} \in \mathcal{P}(x^{\text{test}})}{\arg\max} \; \underset{m \in \mathcal{M}}{\arg\min} \left\| m^{\text{test}} - m \right\|_2$$
$$s^* = \left\| m^{\text{test},*} - m^* \right\|_2 . \tag{6}$$

s を取得するために、s* にスケーリング係数 w を適用し、隣接パッチの挙動を考慮します：異常候補 $m^{test,*}$ に最も近いメモリバンクの特徴が、隣接サンプルから遠く離れており、既に稀な標準的な発生である場合、異常スコアを増加させます。

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

N_b（m* ）は、テストパッチ特徴量m*に対するM内のb番目に近いパッチ特徴量です。この再加重は、単なる最大パッチ距離よりも頑健であることが判明しました。sが与えられると、セグメンテーションは直接導出されます。式7（最初の行）の画像レベル異常スコアは、各パッチの異常スコアをarg max操作で計算する必要があります。セグメンテーションマップは、[14]と同様に、計算されたパッチ異常スコアをそれぞれの空間的位置に基づいて再配置することで、同じステップで計算できます。

元の入力解像度と一致させるため（中間ネットワーク特徴量を使用する場合）、バイリニア補間により結果をアップスケールします。さらに、カーネル幅 $\sigma = 4$ のガウス関数で結果を平滑化しましたが、このパラメーターは最適化していません。

4. Experiments

4.1. 実験の詳細

データセット。産業用異常検出性能を研究するため、実験のほとんどはMVTec異常検出ベンチマーク[5]上で実施されています。MVTec ADには、合計5,354枚の画像を含む15のサブデータセットがあり、そのうち1,725枚がテストセットです。各サブデータセットは、特定の製品における多様な欠陥タイプに対応するノーマルサンプルのみを含むトレーニングデータと、ノーマルと異常サンプルを含むテストセットに分割されています。また、それぞれに対応する異常の真値マスクも含まれています。[10, 14, 56]と同様に、画像はそれぞれ256×256と224×224にリサイズされ、中央部分のみが切り出されます。データ拡張は適用されていません。これは、クラスを維持する拡張方法に関する事前知識が必要だからです。また、より専門的なタスクにおける産業用異常検出についても研究しています。そのため、[42] で使用された Magnetic Tile Defects (MTD) [26] データセットを活用します。このデータセットには、多様な照明レベルと画像サイズを有する欠陥なし画像 925 枚と異常磁気タイル画像 392 枚が含まれます。[42] と同様、欠陥なし画像の 20% はテスト時に評価され、残りはコールドスタートトレーニングに使用されます。最後に、PatchCoreの非産業用画像データへの潜在的な適用可能性を強調するため、[52]や[14]で実施されたように、Mini Shanghai Tech Campus (mSTC) でのコールドスタート異常局所化をベンチマークしています。mSTCは、元のSTCデータセット[32]のサブサンプリング版で、トレーニングとテストの動画フレームの5つに1つを使用しています。12の異なるシーンからの歩行者動画を含んでいます。トレーニングビデオには通常の歩行者の行動が含まれていますが、テストビデオには、喧嘩や自転車に乗るなどのさまざまな行動が含まれている場合があります。コールドスタート実験の比較可能性を確保するため、確立された mSTC プロトコル [14、52] に従い、異常の監視は使用せず、画像は 256 × 256 にリサイズしています。

評価指標。画像レベルの異常検出性能は、生成された異常スコアを用いて受信者動作特性曲線（ROC曲線）下の面積（AUROC）で測定されます。先行研究に従い、MVTecにおいてクラス平均AUROCを計算します[2, 10, 14]。セグメンテーション性能を測定するために、まずピクセル単位のAUROCとPROメトリクスの両方を採用します。これらは[6]に従っています。PROスコアは、MVTec ADにおける異常サイズの変動を適切に考慮するため、接続された異常成分の重なりと回復を考慮しています。詳細は[6]を参照してください。

use *PatchCore*$-n\%$ to denote the percentage n to which the original memory bank has been subsampled to, e.g., *PatchCore*-1% a 100x times reduction of $\mathcal{M}$. Figure 3 gives a visual impression of the spatial coverage of greedy coreset subsampling compared to random selection.

Algorithm 1: *PatchCore* memory bank.

Input: Pretrained ϕ, hierarchies j, nominal data $\mathcal{X}_N$, stride s, patchsize p, coreset 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 coreset selection. */`
$\mathcal{M}_C \leftarrow \{\}$
for $i \in [0, ..., l-1]$ **do**
  $m_i \leftarrow \underset{m \in \mathcal{M}-\mathcal{M}_C}{\arg\max} \min_{n \in \mathcal{M}_C} \|\psi(m) - \psi(n)\|_2$
  $\mathcal{M}_C \leftarrow \mathcal{M}_C \cup \{m_i\}$
end
$\mathcal{M} \leftarrow \mathcal{M}_C$

3.3. Anomaly Detection with *PatchCore*

With the nominal patch-feature memory bank $\mathcal{M}$, we estimate the image-level anomaly score $s \in \mathbb{R}$ for a test image x^{test} by the maximum distance score s^* between test patch-features in its patch collection $\mathcal{P}(x^{\text{test}}) = \mathcal{P}_{s,p}(\phi_j(x^{\text{test}}))$ to each respective nearest neighbour m^* in $\mathcal{M}$:

$$m^{\text{test},*}, m^* = \underset{m^{\text{test}} \in \mathcal{P}(x^{\text{test}})}{\arg\max} \underset{m \in \mathcal{M}}{\arg\min} \left\| m^{\text{test}} - m \right\|_2$$
$$s^* = \left\| m^{\text{test},*} - m^* \right\|_2 . \tag{6}$$

To obtain s, we use scaling w on s^* to account for the behaviour of neighbour patches: If memory bank features closest to anomaly candidate $m^{\text{test},*}$, m^*, are themselves far from neighbouring samples and thereby an already rare nominal occurrence, we increase the anomaly score

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

with $\mathcal{N}_b(m^*)$ the b nearest patch-features in $\mathcal{M}$ for test patch-feature m^*. We found this re-weighting to be more robust than just the maximum patch distance. Given s, segmentations follow directly. The image-level anomaly score in Eq. 7 (first line) requires the computation of the anomaly score for each patch through the $\arg\max$-operation. A segmentation map can be computed in the same step, similar to [14], by realigning computed patch anomaly scores based on their respective spatial location. To match the original input resolution, (we may want to use intermediate network features), we upscale the result by bi-linear interpolation. Additionally, we smoothed the result with a Gaussian of kernel width $\sigma = 4$, but did not optimize this parameter.

4. Experiments

4.1. Experimental Details

Datasets. To study industrial anomaly detection performance, the majority of our experiments are performed on the MVTec Anomaly Detection benchmark [5].
MVTec AD contains 15 sub-datasets with a total of 5354 images, 1725 of which are in the test set. Each sub-dataset is divided into nominal-only training data and test sets containing both nominal and anomalous samples for a specific product with various defect types as well as respective anomaly ground truth masks. As in [10, 14, 56], images are resized and center cropped to 256×256 and 224×224, respectively. No data augmentation is applied, as this requires prior knowledge about class-retaining augmentations.
We also study industrial anomaly detection on more specialized tasks. For that, we leverage the *Magnetic Tile Defects (MTD) [26]* dataset as used in [42], which contains 925 defect-free and 392 anomalous magnetic tile images with varied illumination levels and image sizes. Same as in [42], 20% of defect-free images are evaluated against at test time, with the rest used for cold-start training.
Finally, we also highlight potential applicability of *PatchCore* to non-industrial image data, benchmarking cold-start anomaly localization on *Mini Shanghai Tech Campus (mSTC)* as done in e.g. [52] and [14]. *mSTC* is a subsampled version of the original *STC* dataset [32], only using every fifth training and test video frame. It contains pedestrian videos from 12 different scenes. Training videos include normal pedestrian behaviour while test videos can contain different behaviours such as fighting or cycling. For comparability of our cold-start experiments, we follow established *mSTC* protocols [14, 52], not making use of any anomaly supervision and images resized to 256×256.

Evaluation Metrics. Image-level anomaly detection performance is measured via the area under the receiver-operator curve (AUROC) using produced anomaly scores. In accordance with prior work we compute on MVTec the class-average AUROC [2, 10, 14]. To measure segmentation performance, we use both pixel-wise AUROC and the PRO metric first, both following [6]. The PRO score takes into account the overlap and recovery of connected anomaly components to better account for varying anomaly sizes in MVTec AD, see [6] for details.