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:

```
M ← {}
for x_i ∈ X_N do
  | M ← M ∪ P_{s,p}(φ_j(x_i))
end
/* Apply greedy coreset selection.      */
M_C ← {}
for i ∈ [0, ..., l − 1] do
  | m_i ← argmax_{m ∈ M − M_C} min_{n ∈ M_C} ||ψ(m) − ψ(n)||_2
  | M_C ← M_C ∪ {m_i}
end
M ← 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 $\mathcal{M}$:

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

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

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

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.