

表 3. PRO [%] [5、10] で測定された MVTec AD [5] における異常検出性能。

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

表4. PatchCore-1%（高解像度／大規模なバックボーン／アンサンブル）。コアセットのサブサンプリングにより、計算コストの高い設定でも高速な推論を維持できます。

Metric→	AUROC	pwAUROC	PRO
DenseN-201 および RNext-101 および WRN-101 (2+3)、画像サイズ 320			
Score ↑	99.6	98.2	94.9
Error ↓	0.4	1.8	5.6
WRN-101 (2+3)、画像サイズ 280			
Score ↑	99.4	98.2	94.4
Error ↓	0.6	1.8	5.6
WRN-101 (1+2+3)、画像サイズ 280			
Score ↑	99.2	98.4	95.0
Error ↓	0.8	1.6	5.0

表5. MVTec ADにおける画像ごとの平均推論時間。スコアは（画像AUROC、ピクセルAUROC、PROメトリクス）です。

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

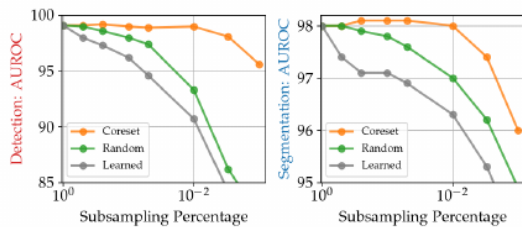


図5. 異なるサブサンプリング手法における性能維持。PROスコアの結果は補足資料に記載されています。

4.4.1 局所的に認識可能なパッチ特徴と階層

局所的に意識したパッチ特徴 (§3.3) の重要性を、式1における異なる近傍サイズでの異常検出性能の変化を評価することで調査します。図4の上半分の結果は、パッチベースの異常予測において局所性とグローバルな文脈の間の明確な最適解を示しており、これにより近傍サイズ $p = 3$ が動機付けられます。よりグローバルな文脈は、ネットワーク階層を下することで実現可能ですが（例：[10, 14]）、解像度の低下とImageNetクラスバイアスの増加を伴います (§3.1)。WideResNet50の最初の3つのブロックを13でインデックス付けすると、図4（下段）は再び、高度に局所的な予測、よりグローバルな文脈、およびImageNetバイアス間の最適点を強調しています。

は既に最先端の性能を達成できますが、後続の階層レベル（2+3、デフォルト設定として選択）から追加のフィーチャーマップを取得することでさらに性能が向上します。

4.4.2 コアセットサブサンプリングの重要性

図5は、異なるメモリバンクMサブサンプリング手法を比較しています：貪欲なコアセット選択、ランダムサブサンプリング、およびサブサンプリング目標パーセンテージ p_{target} に対応するベースプロキシのセットを学習する手法です。後者の場合、 $|P| = p_{target} \cdot |M|$ となる $P \subset M$ からプロキシ p_i をサンプリングし、その後、ベース再構築目的関数を最小化するタスクを課します。

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

図5では、3つの設定を比較し、コアセットに基づくサブサンプリングが他の選択肢よりも優れた性能を示すことが確認されました。サブサンプリングを行わない場合の性能は、サイズが2桁小さいコアセット削減メモリバンクとほぼ同等です。また、サブサンプリングされたメモリバンクにははるかに少ない冗長性が含まれていることも確認しました。非サブサンプリングとコアセットサブサンプリングされたメモリバンクにおいて、テスト時に使用されるメモリバンクサンプルの割合を記録しました。当初、メモリバンクサンプルの30%未満しか使用されていませんでしたが、コアセットサブサンプリング（1%まで）により、この割合は95%近くまで増加しました。特定のサブサンプリング間隔（50%から10%程度）では、サブサンプリングを行わないPatchCoreと比較して、異常検出と位置特定に関する総合的なパフォーマンスが部分的に向上することも確認されました。最後に、ストライドの増加（式3を参照）によってメモリバンクサイズMを縮小すると、解像度コンテキストの低下によりパフォーマンスが低下し、ストライド $s = 2$ では画像異常検出のAUROCが97.6%、ストライド $s = 3$ ではAUROCが96.8%となります。

4.5. 低ショット異常検出

限定された名義データへのアクセスは、現実世界の検査における重要な設定です。そのため、フルMVTec ADでの結果報告に加え、訓練例が少ない場合の性能も検討しました。トレーニングサンプルの数を1（総名義トレーニングデータの0.4%に相当）から50（21%）まで変化させ、同じバックボーン（WideResNet50）を使用したSPADE [10]とPaDiM [14]の再実装と比較しました。結果は図6に要約されており、詳細な結果は補足資料に記載されています。

Table 3. Anomaly Detection Performance on MVTec AD [5] as measured in PRO [%] [5, 10].

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

Table 4. PatchCore-1% with higher resolution/larger backbones/ensembles. The coreset subsampling allows for computationally expensive setups while still retaining fast inference.

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

Table 5. Mean inference time per image on MVTec AD. Scores are (image AUROC, pixel AUROC, PRO metric).

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

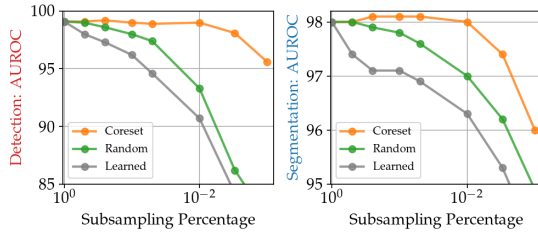


Figure 5. Performance retention for different subsamplers, results for PRO score in the supplementary.

4.4.1 Locally aware patch-features and hierarchies

We investigate the importance of locally aware patch-features (§3.3) by evaluating changes in anomaly detection performance over different neighbourhood sizes in Eq. 1. Results in the top half of Figure 4 show a clear optimum between locality and global context for patch-based anomaly predictions, thus motivating the neighbourhood size $p = 3$. More global context can also be achieved by moving down the network hierarchy (see e.g. [10, 14]), however at the cost of reduced resolution and heavier ImageNet class bias (§3.1). Indexing the first three WideResNet50-blocks with 1 - 3, Fig. 4 (bottom) again highlights an optimum between highly localized predictions, more global context and ImageNet bias. As can be seen, features from hierarchy level 2

can already achieve state-of-the-art performance, but benefit from additional feature maps taken from subsequent hierarchy levels (2+3, which is chosen as the default setting).

4.4.2 Importance of Coreset subsampling

Figure 5 compares different memory bank \mathcal{M} subsampling methods: Greedy coreset selection, random subsampling and learning of a set of basis proxies corresponding to the subsampling target percentage p_{target} . For the latter, we sample proxies $p_i \in \mathcal{P} \subset \mathbb{R}^d$ with $|\mathcal{P}| = p_{\text{target}} \cdot |\mathcal{M}|$, which are then tasked to minimize a basis reconstruction objective

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

to find N proxies that best describe the memory bank data \mathcal{M} . In Figure 5 we compare the three settings and find that coreset-based subsampling performs better than the other possible choices. The performance of no subsampling is comparable to a coreset-reduced memory bank that is two orders of magnitudes smaller in size. We also find subsampled memory banks to contain much less redundancy. We recorded the percentage of memory bank samples that are used at test time for non-subsampled and coreset-subsampled memory banks. While initially only less than 30% of memory bank samples are used, coreset subsampling (to 1%) increases this factor to nearly 95%. For certain subsampling intervals (between around 50% and 10%), we even find joint performance over anomaly detection and localization to partly increase as compared to non-subsampled PatchCore. Finally, reducing the memory bank size \mathcal{M} by means of increased striding (see Eq. 3) shows worse performance due to the decrease in resolution context, with stride $s = 2$ giving an image anomaly detection AUROC of 97.6%, and stride $s = 3$ an AUROC of 96.8%.

4.5. Low-shot Anomaly Detection

Having access to limited nominal data is a relevant setting for real-world inspection. Therefore in addition to reporting results on the full MVTec AD, we also study the performance with fewer training examples. We vary the amount of training samples from 1 (corresponding to 0.4% of the total nominal training data) to 50 (21%), and compare to reimplementations of SPADE [10] and PaDiM [14] using the same backbone (WideResNet50). Results are summarized in Figure 6, with detailed results available in Supp.