基于风格记忆的跨域小样本异常检测

杨太海[1)], 顾智浩[2)], 马利庄[1)*]

[1)] (华东师范大学计算机科学与技术学院 上海 200062)
[2)] (上海交通大学电子信息与电气工程学院 上海 200240)
(lzma@cs.ecnu.edu.cn)

摘 要: 小样本异常检测仅利用少量正常样本进行检测和异常定位, 现有的小样本异常检测方法的训练和测试都在相同的域中进行, 对于跨域场景的研究尚处于空白. 针对跨域小样本异常检测问题, 为了将源域学到的知识迁移到目标域中进行小样本异常检测, 提出基于风格记忆的跨域小样本异常检测方法. 该方法包含 2 个模块, 首先将域差异定义为风格特征差异, 设计风格记忆模块存储来自源域的域内统计特征, 并将学习到的特征集成到目标域中, 减少域差异带来的影响, 提升性能; 然后引入对比学习策略, 区分不同类别的增强查询特征, 进一步加强模型对源域中不同类别的表征能力. 在 MVTec AD, MPDD 和 MVTec LOCO-AD 数据集间进行跨域小样本异常检测任务实验, 结果表明, 与代表方法相比, 所提方法取得了最佳性能, 其中, 图像 AUROC 指标提升 3%~5%; 在本域内小样本异常检测任务上, 该方法仍然有优异的表现.

关键词: 异常检测; 小样本学习; 域差异
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023-00458

Style-Aware Cross Domain Few-Shot Anomaly Detection

Yang Taihai[1)], Gu Zhihao[2)], and Ma Lizhuang[1)*]

[1)] (*School of Computer Science and Technology, East China Normal University, Shanghai* 200062)
[2)] (*School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai* 200240)

Abstract: Few-shot anomaly detection involves using a limited number of normal samples to detect and localize anomalies. While existing methods for few-shot anomaly detection are primarily applied within the same domain, research on cross-domain scenarios remains limited. To address the problem of cross-domain few-shot anomaly detection, a cross-domain few-shot anomaly detection method based on style memory is proposed. This method includes two modules. The first module defines domain differences in terms of differences in style features and employs a style memory module to store intra-domain statistical features from the source domain. These features are then integrated into the target domain to mitigate the impact of domain differences and enhance few-shot detection performance. Then, a contrastive learning strategy is introduced to distinguish enhanced query features of different categories, further enhancing the model's ability to represent different categories in the source domain. Experiments on cross-domain few-shot anomaly detection tasks between the MVTec AD, MPDD and MVTec LOCO-AD datasets show that the proposed method achieves the best performance compared to the represntative methods, with an image AUROC indicator increase of 3% to 5%. Moreover, the proposed model still has the best performance on the few-shot anomaly detection task under the intra-domain setting.

收稿日期: 2023-07-20; 修回日期: 2023-12-07. **基金项目**: 国家自然科学基金(61972157, 72192821); 上海市科技创新行动计划人工智能科技支撑项目(21511101200). **杨太海**(1994—), 男, 博士研究生, 主要研究方向为视觉检测; **顾智浩**(1994—), 男, 博士研究生, 主要研究方向为视觉检测、人脸安全; **马利庄**(1963—), 男, 博士, 教授, 博士生导师, CCF 会员, 论文通信作者, 主要研究方向为计算机图形学、计算机视觉、智能信息处理.

Key words: anomaly detection; few-shot learning; domain gap

随着深度神经网络的快速发展，工业产品的表面异常/缺陷检测受到了前所未有的关注[1-2]. 由于异常样本获取难度较大，通常使用无监督方法进行异常检测(anomaly detection, AD)，即训练过程仅使用正常样本. 在过去几年中，无监督 AD 取得了巨大的进展，包括基于归一化流的[3-4]、基于记忆库的[1,5]、基于重建的[6-7]和基于知识蒸馏的[2,8]这 4 类方法，它们在 MVTec AD[9]和 MPDD[10]等数据集中都取得了优异的性能. 然而，这些方法在测试时需要针对每个类别单独训练一个模型，对于未知的类别直接测试通常性能表现很差[11]. 为了解决这个问题，近年来，一些研究倾向于探索小样本异常检测(few-shot AD, FSAD)问题.

AD方法在训练时通常需要一定数量的正常样本支撑，而 FSAD 方法[11-12]只需少量样本就能建模，减少了模型对训练样本数量的依赖，能够将模型快速地适配到新类别. 现有的FSAD方法假设训练集和测试集属于同一分布，利用生成或者变换等方式[12]扩充包含少量正常样本的支持集，提取不同类别间的共性[11]特征进行建模计算分布，在查询样本上检测异常，但这一假设并不总是成立的. 在工业场景中，随着越来越多的新产线投入使用，产线之间的域差异也越来越明显. 为每条产线的算法或者机器单独进行微调是十分耗时的，同时新产线的数据收集和标注在初期往往是困难且不足的，使得将利用以往产线(训练集)学习到的知识快速地适配到新产线(测试集)的需求大大提高.

本文引入跨域 FSAD(cross domain FSAD, CD-FSAD)任务：在测试时只有少量样本可以利用，在关注域内类别间特征差异的同时还需考虑域间的巨大差距. 本文将域间差异定义为风格特征差异，建立一个轻量级的风格记忆模块，用于存储每个训练实例中的风格信息，实例的均值和方差被聚合到记忆库中，用于表示训练域的风格特征. 在训练阶段，使用具有最低相似性的内存块增强特征表示，使模型强制忽略风格差异进行特征匹配；在推理中，选择具有最高相似性的内存块增强特征表示，减少域差异，使模型的泛化性更好. 此外，还使用对比学习策略增强不同类别间的特征表示，进一步提高模型小样本检测能力.

本文提出的 CD-FSAD 方法，通过学习不同风格特征来缓解域差异；利用类别间对比损失提高跨场景下小样本检测性能. 在 3 个数据集间进行跨域实验，结果表明了本文方法的有效性，在 CD-FSAD 场景几乎取得了最优性能；且在本域上测试同样有优异的性能表现.

1　相关工作

1.1　FSAD 方法

近年来, FSAD开始受到关注，它通过支持集中的少量正常样本进行检测和定位异常. 一些 AD 工作[1,5,13]也在小样本设定下评估了它们在 MVTec AD 数据集上的性能表现. 已有一些工作利用元学习研究工业场景下 FSAD. Sheynin 等[12]设计了一个分层生成(transformation discriminating generative, TDG)模型，用于捕获每个支持样本的多尺度块特征分布，对图像块应用旋转变换，并优化块鉴别器区分异常块，最后投票图像块的正确变换获得评价指标—异常分数；配准异常检测(registration-based anomaly detection, RegAD)[11]则利用匹配网络在查询集和支持集之间进行对齐，获得一个可泛化的AD模型，最后通过计算支持集和查询集的分布差异来识别异常.

1.2　基于记忆库的方法

记忆库[14]最初是为了捕获序列数据中的时序相关信息设计的. 文献[15]通过比较特征相似性同时利用线性组合操作，以端到端的方式训练记忆网络；记忆引导学习(learning memory-guided normality, LMN)[16]基于端到端的方式，通过聚合实例信息扩展其能力，进而对内存项更新和学习；之后，分割重构异常检测(divide-and-assemble anomaly detection, DAAD)[7]使用块状记忆模块，通过最大化正常和异常数据检索误差之间的差异提升检测性能. 最近，PatchCore[1]通过存储基于预训练模型的分块特征，并选取最具代表性的特征，用于推理过程中的特征检索比较. 与上述学习正常特征的工作不同，本文方法扩展了记忆库以存储域内特定统计信息，并风格化特征来模拟域迁移，提高模型的泛化能力.

1.3　域泛化方法

域泛化(domain generalization, DG)[17]的目标是从一个或多个不同但相关的域(即多种训练数据集)中学习一个模型，它可以在未知的测试域中良

好运行. 尽管训练数据来自多个域, 但在训练期间无法访问测试域使 DG 具有挑战性. 为了提高模型的泛化能力, 基于表征学习、数据增强、域对齐等策略的模型不断被提出. 元学习也在 DG 中得到了广泛应用, 通过分布鲁棒优化[18-19]和新的元学习任务[15,20]来模拟域迁移, 并取得了良好的结果. 由于本文的训练数据来自单一域, 且测试域中的类别未知, 因此难以直接利用大多数 DG 方法模拟域迁移. 本文利用存储在记忆库中的风格信息多样化特征, 加强模型的泛化性.

1.4 特征匹配网络框架

特征匹配网络(feature registration network, FRN)[11]是 RegAD 中的一种用于 FSAD 的方法, 具有优异的性能表现; 特别地, 它包含 1 个 ResNet-18[21]的骨干网络 F, 3 个空间变换网络[22], 编码器网络 E 和预测网络 P. 记 p 表示经过编码和预测网络后的特征 $p_{\mathrm{q}}=P\left(E\left(F\left(Q^{\mathrm{train}}\right)\right)\right)$; z 表示只经过编码网络后的特征 $z_{\mathrm{s}}=E\left(\frac{1}{K}\sum_{k}F(S_k^{\mathrm{train}})\right)$. 其中, Q^{train} 和 S^{train} 分别表示训练集中的查询样本和支持样本. 利用 SimSiam 范式[23]对模型进行训练

$$\mathcal{L}_{\mathrm{reg}}=-\frac{1}{2}\left(D_{\mathrm{CS}}\left(p_{\mathrm{s}},\mathrm{SG}\left(z_{\mathrm{q}}\right)\right)+D_{\mathrm{CS}}\left(p_{\mathrm{q}},\mathrm{SG}\left(z_{\mathrm{s}}\right)\right)\right) \tag{1}$$

其中, $D(\cdot)$ 表示距离; 下标 CS 表示余弦相似性(cosine similarity, CS)度量; $\mathrm{SG}(\cdot)$ 表示梯度停止(stop gradient, SG)操作.

在推理中, 首先对测试样本进行多次图像增强, 连接 FRN 前 3 层的输出; 然后使用基于统计的估计器[5]计算每个特征点位置 (i,j) 的协方差 $\boldsymbol{\Sigma}_{ij}$. 假设 $\boldsymbol{f}$ 是查询样本的特征表示, 则分布距离使用 Mahalanobis 距离计算, 公式为

$$\boldsymbol{M}_{ij}=\sqrt{(\boldsymbol{f}_{ij}-\boldsymbol{\mu}_{ij})^{\mathrm{T}}\boldsymbol{\Sigma}_{ij}^{-1}(\boldsymbol{f}_{ij}-\boldsymbol{\mu}_{ij})} \tag{2}$$

其中, $\boldsymbol{\mu}_{ij}$ 表示 $\boldsymbol{f}_{ij}$ 的样本均值. 查询样本与支持样本分布的 Mahalanobis 距离作为最终分数图, 较高的分数表示相应位置存在异常, 更多细节见文献[11].

FRN 最初是为相同分布下的 FSAD 场景设计的, 而 CD-FSAD 场景更具有挑战性. 如图 1 展示了 CD-FSAD 与 FSAD 的区别, 在 FSAD 任务中, 训练集和测试集在同一领域内(MVTec AD), 而在 CD-FSAD 设定中两者来自不同的领域(MVTec AD 和 MVTec LOCO-AD). 域差异将会导致式(2)对测试域中的 $\boldsymbol{\Sigma}_{ij}$ 的错误估计. 为此, 本文提出风格记忆模块和类间对比损失 2 个模块来解决这个问题.

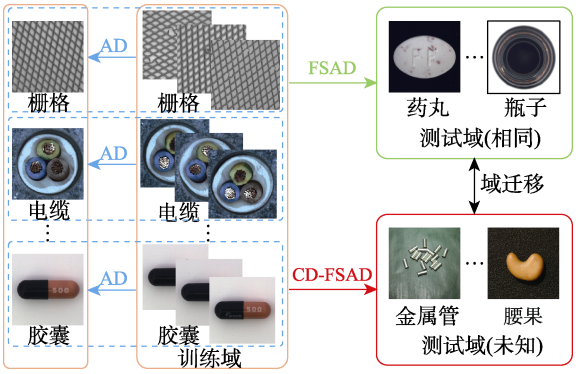


图 1 CD-FSAD 与传统设定对比

2 本文方法

2.1 概 述

在跨域设定下, FSAD 方法主要面临源域和目标域之间存在巨大的域差异, 以及目标域内类别差异造成的内在 AD 困难. 本文的目标是生成多样化的域特征, 通过学习域特征减小域差异, 并学习各个特征之间的区分性表示, 从而提高小样本检测性能. 为此, 提出风格记忆模块和类间对比损失解决跨域设定下的 FSAD 问题.

给定训练域 D_{train}, 其中, 每个类别都有足够的无异常样本, 本文目标是学习一个 AD 器, 该检测器可以利用少量正常样本有效地检测测试域 D_{test} 中新类别的异常; 还假设 D_{train} 和 D_{test} 之间存在巨大的域差异, 且 D_{test} 中的类别不包含在 D_{train} 中.

本文遵循已有工作[11-12]中的设定, 对于 D_{train} 中的每个类别, 采样大量的查询-支持对, 每个对都包括一个正常的查询样本 Q^{train} 和 K 个正常的支持样本 $\left\{S_k^{\mathrm{train}}\right\}_{k=1}^{K}$. D_{test} 中的每个查询样本 Q^{test} 也有 K 个正常的支持样本 $\left\{S_k^{\mathrm{test}}\right\}_{k=1}^{K}$. 在训练过程中, FRN 在 Q^{train} 和 $\left\{S_k^{\mathrm{train}}\right\}_{k=1}^{K}$ 之间对齐特征, 并且将 Q^{train} 计算出的域内统计信息不断聚合到风格记忆模块, 用于丰富样本风格, 强制模型学习风格差异, 实现鲁棒检测. 采用类间对比损失对不同类别的风格化特征进行约束, 获得特征的区分性表示, 提高小样本检测性能. 在推理中, 利用存储的统计信息从 $\left\{S_k^{\mathrm{test}}\right\}_{k=1}^{K}$ 估计逐像素正态分布, 在 Q^{test} 上超出分布的位置被视为异常. 本文方法架构如图 2 所示, 包括基础的 FRN, 风格记忆模块以及类间对比损失模块.

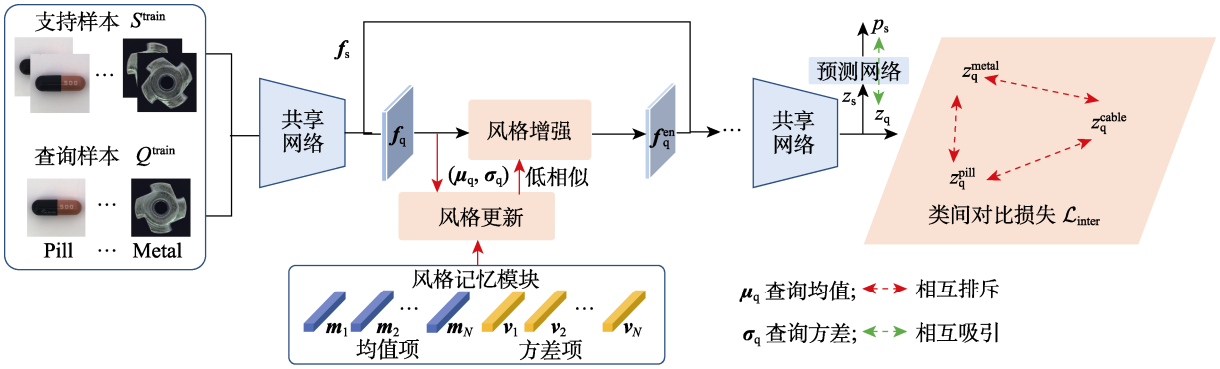


图 2　本文方法架构

2.2　风格记忆模块

将不同域之间的差异定义为风格差异，并选择均值和方差反映域特定的统计数据，相关研究表明[24]，这样可以在一定程度上衡量域分布信息. 本文设计可以存储 N 个均值-方差内存项的风格记忆模块 M，$M=\left\{\left(\boldsymbol{m}_j,\boldsymbol{v}_j\right)\middle|\boldsymbol{m}\in\mathbb{R}^C,\boldsymbol{v}\in\mathbb{R}^C\right\}_{j=1}^{N}$；其中，$C$ 表示特征通道的维度. 如图 3 所示，风格记忆模块包含 2 个操作:

(1) 风格记忆库中内存项更新操作. 图 3a 中，举例展示了均值项的更新流程，方差项更新操作类似.

(2) 风格特征增强操作. 图 3b 所示为训练时特征风格化的流程，推理时则选取了相似度最高的内存项进行风格特征增强.

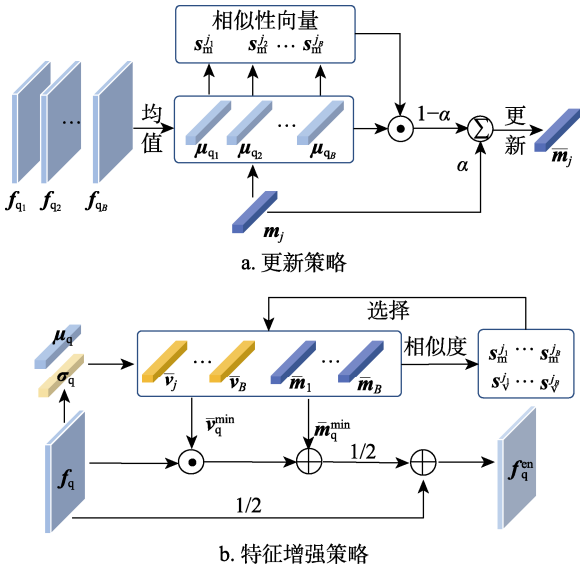


图 3　风格记忆模块更新和增强策略

2.2.1　风格特征更新

在训练阶段，从查询 Q 中提取特征平均值和方差，并以迭代的方式更新每个内存项 $(\boldsymbol{m}_j,\boldsymbol{v}_j)$. 给出 Q 的特征向量 $\boldsymbol{f}_q\in\mathbb{R}^{C\times H\times W}$，其中，$H$ 和 W 是特征向量的高度和宽度，均值 $\boldsymbol{\mu}_q\in\mathbb{R}^{C\times1\times1}$ 和方差 $\boldsymbol{\sigma}_q\in\mathbb{R}^{C\times1\times1}$ 的计算公式为

$$\begin{cases}\boldsymbol{\mu}_q=\dfrac{1}{HW}\displaystyle\sum_{H,W}\boldsymbol{f}_q\\ \boldsymbol{\sigma}_q=\sqrt{\dfrac{1}{HW}\displaystyle\sum_{H,W}\left(\boldsymbol{f}_q-\boldsymbol{\mu}_q\right)^2}\end{cases}\tag{3}$$

为了使记忆模块存储的风格更加多样化，首先计算每个内存项 $(\boldsymbol{m}_j,\boldsymbol{v}_j)$ 和 $(\boldsymbol{\mu}_q,\boldsymbol{\sigma}_q)$ 之间的相似性，公式为

$$\begin{cases}\boldsymbol{s}_{\text{m}}^{j_b}=D_{\text{CS}}\left(\boldsymbol{m}_j,\boldsymbol{\mu}_{q_b}\right)\Big/\displaystyle\sum_b D_{\text{CS}}\left(\boldsymbol{m}_j,\boldsymbol{\mu}_{q_b}\right)\\ \boldsymbol{s}_{\text{v}}^{j_b}=D_{\text{CS}}\left(\boldsymbol{v}_j,\boldsymbol{\sigma}_{q_b}\right)\Big/\displaystyle\sum_b D_{\text{CS}}\left(\boldsymbol{v}_j,\boldsymbol{\sigma}_{q_b}\right)\end{cases}.$$

其中，$b\in(1,2,\cdots,B)$ 表示批次样本的索引，B 表示批处理大小. 然后根据得出的相似度 $\left(\boldsymbol{s}_{\text{m}}^{j_b},\boldsymbol{s}_{\text{v}}^{j_b}\right)$ 加权更新内存项 $\left(\boldsymbol{m}_j,\boldsymbol{v}_j\right)$，即内存项在学习固定风格信息 $\left(\boldsymbol{\mu}_q,\boldsymbol{\sigma}_q\right)$ 的同时也生成了更多相似的源域风格. 更新公式为

$$\begin{cases}\bar{\boldsymbol{m}}_j\leftarrow\alpha\cdot\boldsymbol{m}_j+(1-\alpha)\cdot\displaystyle\sum_b\boldsymbol{s}_{\text{m}}^{j_b}\boldsymbol{\mu}_{q_b}\\ \bar{\boldsymbol{v}}_j\leftarrow\alpha\cdot\boldsymbol{v}_j+(1-\alpha)\cdot\displaystyle\sum_b\boldsymbol{s}_{\text{v}}^{j_b}\boldsymbol{\sigma}_{q_b}\end{cases}.$$

其中，α 表示聚合权重，用来控制更新速率，即新内存项为原始内存项和新批次特征利用相似度加权结果的线性组合. 因为记忆模块存储了源域的风格信息，所以便于进一步调用，从而对原始特征进行风格变换. 本文中每个批次后所有内存模块都会立即进行风格更新，以获得更丰富的风格信息.

2.2.2　风格特征增强

由于 $\left\{\left(\bar{\boldsymbol{m}}_j,\bar{\boldsymbol{v}}_j\right)\right\}_{j=1}^{N}$ 包含训练域中不同类的风格

信息, 因此可以利用它们间接生成不同风格的特征. 受到 AdaIN[25]的启发, 本文采用一种类似方式将风格信息整合到查询特征中.

给定查询特征 $\boldsymbol{f}_{\mathrm{q}} \in \mathbb{R}^{C\times H\times W}$, 其均值和方差信息可以通过式(3)计算得到, 则归一化后的查询特征表示为 $\boldsymbol{f}_{\mathrm{q}}^{\mathrm{norm}}=\left(\boldsymbol{f}_{\mathrm{q}}-\boldsymbol{\mu}_{\mathrm{q}}\right)/\boldsymbol{\sigma}_{\mathrm{q}}$.

为了有效地利用记忆库中的风格特征, 首先计算 $\left(\boldsymbol{\mu}_{\mathrm{q}},\boldsymbol{\sigma}_{\mathrm{q}}\right)$ 与记忆库 M 中每个内存项之间的相似度; 然后选择差异化最大, 即具有最低相似度的内存项 $\left(\bar{\boldsymbol{m}}_{\mathrm{q}}^{\min},\bar{\boldsymbol{v}}_{\mathrm{q}}^{\min}\right)$ 进行特征增强. 则变换后的增强特征表示为 $\boldsymbol{f}_{\mathrm{q}}^{\mathrm{en}}=\frac{1}{2}\left((\boldsymbol{f}_{\mathrm{q}}^{\mathrm{norm}}\cdot\bar{\boldsymbol{v}}_{\mathrm{q}}^{\min}+\bar{\boldsymbol{m}}_{\mathrm{q}}^{\min})+\boldsymbol{f}_{\mathrm{q}}\right)$. $\boldsymbol{f}_{\mathrm{q}}^{\mathrm{en}}$ 将通过网络的剩余部分并使用式(1)与未增强的支持样本进行特征匹配, 拉近两者的距离, 使模型强制忽略风格差异, 达到更好的泛化效果.

在推理中, 本文在测试查询的特征增强时选择具有最大相似度值的内存项, 进一步减少域间差异提升模型性能.

2.3 类间对比损失

经过 SimSiam 范式[23]后, 模型实现了查询和支持样本的特征一致性. 为了进一步加强模型的表征能力, 引入对比学习策略[26], 使模型能够利用同类之间的相似性以及类间的差异性进行鉴别性的表征. 考虑在式(1)中同类距离已经被拉近, 本文只需使不同类之间的距离互相排斥, 模型就可以获得更好的检测能力.

给定一个批次内由 $\boldsymbol{f}_{\mathrm{q}_i}^{\mathrm{en}}$ 生成的查询特征 z_{q_i}, 同一批次中其他类的特征被认为是负样本. 则类间的对比损失被表述为

$$\mathcal{L}_{\mathrm{inter}}=\frac{1}{B}\sum_{i=1}^{B}\ln\left(\frac{1}{|P_i|}\sum_{z_p\in P_i}\mathrm{e}^{D_{\mathrm{CS}}\left(z_{\mathrm{q}_i},z_p\right)/\tau}\right).$$

其中, τ 表示温度系数; P_i 表示由与 z_{q_i} 不同类别特征 z_p 组成的集合.

由于原始匹配损失 $\mathcal{L}_{\mathrm{reg}}$ 始终为负数, 为了使其与类间损失 $\mathcal{L}_{\mathrm{inter}}$ 正负性一致, 加上常数项后使其保持为正, 因此模型总损失函数定义为

$$\mathcal{L}=(1+\mathcal{L}_{\mathrm{reg}})+\lambda\mathcal{L}_{\mathrm{inter}}.$$

其中, λ 表示平衡系数.

风格记忆模块是一种简洁的即插即用的模块, 可以放置在 FRN 的任何层之间. 本文中将其放置在 FRN 的前 2 层之后, 可获得更通用的域特征表达.

3 实验及结果分析

3.1 AD 数据集

为了证明本文方法在 CD-FSAD 场景的有效性, 在 MVTec AD[9], MPDD[10]和 MVTec LOCO-AD[27]数据集上与已有表现优异的方法进行比较.

MVTec AD 是一个常用的 AD 数据集, 包含 15 个类别的共计 5 000 多幅图像, 分别是 5 类纹理类数据和 10 类物体类数据. 该数据集中图像的异常主要表现在各种结构变化上.

MPDD 数据集是专门针对金属零件涂装过程中产生的缺陷而收集的, 拥有 6 类约 1 300 幅图像. 该数据集中的图像具有不同的空间方向、光强度和非均质背景.

MVTec LOCO-AD 数据集包括 5 个类别, 约 3 700 幅图像, 其中不仅包括局部结构异常, 如划痕和凹痕, 还包括逻辑异常, 如允许的物体出现在无效的位置.

本文实验中, 使用 MVTec AD 和 MPDD 数据集作为训练域.

3.2 实验设置

所有实验都是基于 Pytorch 实现的. 在训练期间, 使用动量随机梯度下降(stochastic gradient descent, SGD)更新参数, 设置学习率为 0.000 1, 共进行 30 个批次训练, 批量大小为 32; 除非特别说明, 使用经过 ImageNet 预训练的 ResNet18 网络作为骨干网络. 设更新参数 α =0.9, 损失函数平衡系数 λ =0.1, 温度系数 τ =0.1; 记忆库内存项数量 N =10, 同时所有内存项被随机初始化. 采用图像级别曲线下面积(area under the receiver operating characteristic curve, AUROC)和像素级别曲线下面积作为衡量指标, 即图像 AUC 和像素 AUC.

3.3 对比实验

实验中, 所有方法都将测试 10 次, 取平均图像 AUC 或像素 AUC 作为结果.

(1) 与表现优异的 FSAD 方法进行比较, 包括 TDG[12], DifferNet[13]和 RegAD[11]. 由于本文主要测试跨域性能表现, 因此这些方法需要首先在训练域所有类上进行训练, 然后在测试域的每个类上进行测试. 遵循 RegAD 数据集中的设定, 对于 TDG 和 DifferNet 方法, 分别利用训练域对变换分类器和基于归一化流的估计器进行初始化; 再到测试域中进行微调测试. 对于 RegAD 方法, 同样使用训练域中的所有类进行 FRN 预训练, 但是直接在测试域的各类上进行推理.

表 1 所示为在 MVTec LOCO 测试域上, MVTec AD 和 MPDD 作为训练域时 CD-FSAD 的图像 AUC 结果. 可以看出: a. 在大多数场景下, 本文方法都获得了最佳性能. b. 与基线框架 RegAD[11]相比, 本文方法在样本数量分别为 2, 4, 8 的设定下, 训练域为 MVTec AD 时, 图像 AUC 指标分别提升 2.9 个百分点, 3.4 个百分点和 4.0 个百分点; 在训练域为 MPDD 时, 分别提升 3.4 个百分点, 2.5 个百分点和 3.8 个百分点. 说明本文方法在跨域场景下性能表现提升明显. c. 虽然在实验设置中, DifferNet 方法在测试时会先利用 k 个支持样本进行微调训练, 但本文方法仍然在大部分情况下获得了更好的性能, 同时, 该方法在测试时不需要训练微调, 拥有更快的适配速度. d. 不同域进行训练时, 性能表现也有差异. 大多数场景下, 在 MVTec AD 上的训练结果好于在 MPDD, 有 2 点原因: 一是源域训练中更多的类别和图像数量会增强模型性能表现, 而 MVTec AD 恰恰拥有更多图像类别和数量; 二是 MVTec AD 本身场景复杂程度就低于 MPDD, 模型在简单场景下更容易进行图像对齐训练.

表 1　MVTec LOCO 测试域上 CD-FSAD 的图像 AUC %

训练域	方法	k		
		2	4	8
MVTec AD	TDG[12]	51.6	52.3	51.3
	DifferNet[13]	60.9	67.1	70.6
	RegAD[11]	58.9	64.5	67.8
	本文	**61.8**	**67.9**	**71.8**
MPDD	TDG[12]	45.9	48.5	53.9
	DifferNet[13]	59.4	**66.8**	69.3
	RegAD[11]	57.7	62.7	66.7
	本文	**61.1**	65.2	**70.5**

注. 粗体表示最优值.

(2) 为了更好地探究本文方法在不同数据集上的鲁棒性, 对 MVTec AD 和 MPDD 数据集间的跨域性能表现进行实验, 结果如表 2 所示. 可以看出: a. 在所有场景下, 本文方法都获得了最佳性能表现; b. 与 RegAD 方法相比, 本文方法在样本数量为 2, 4 和 8 的情况下, 从 MPDD 到 MVTec AD, 性能最高提升 3.5 个百分点; 从 MVTec AD 到 MPDD, 最高性能提升更是达到 5.1 个百分点. 证明本文方法在不同数据集上的跨域性能表现都有不小提升, 有较好的应用价值.

(3) 除了跨域场景外, 本文使用 RegAD 原始设定留一法在域内设定中进行实验; 并在实验中加入 SPADE[28], STPM[29]和 RD4AD[2]这 3 种表现优异的 AD 方法进行比较. 为了公平比较, AD 方法训练中只使用 k 个支持样本.

表 2　MVTec AD 和 MPDD 间 CD-FSAD 的图像 AUC %

训练域	测试域	方法	k		
			2	4	8
MPDD	MVTec AD	TDG[12]	66.2	67.2	68.6
		DifferNet[13]	76.0	78.9	82.8
		RegAD[11]	82.1	85.1	87.7
		本文	**84.8**	**88.3**	**91.2**
MVTec AD	MPDD	TDG[12]	54.3	59.1	60.9
		DifferNet[13]	57.6	62.2	66.4
		RegAD[11]	60.1	63.4	67.6
		本文	**65.1**	**68.5**	**72.1**

注. 粗体表示最优值.

实验结果如表 3 所示, 其中, 在给出图像 AUC 结果的同时也展示了像素 AUC 结果, TDG[12]和 DifferNet[13]方法由于其自身的原因没有像素 AUC 结果. 结合表 2 比较 4 种小样本方法在 2 种场景下的性能表现可以看出: a. 跨域环境中, 4 种方法的结果都有所下降, 其中, 在 MVTec AD 测试集上, TDG[12]方法的图像 AUC 性能损失最高达 8.0 个百分点; 基线方法 RegAD[11]的性能在 $k=2$ 时有最大损失 3.6 个百分点; 而本文方法虽然性能也有所下降, 但最多只下降 1.6 个百分点. 说明本文方法的鲁棒性更强, 相应模块在遇到域差异或者域内类别差异时都能表现很好. b. 与图像 AUC 结果相比, 像素 AUC 结果下降得不太明显, 但与基线方法相比, 大部分设定下都有所提升. c. 在上述小样本实验设定下, 3 种对比的 AD 方法表现都不太好, 尤其是与本文方法或者基线方法相比性能差距较大. 主要原因是这 3 种方法不是专门针对小样本问题而提出的, 当训练样本较少时, 模型很难收敛到较好的权重, 因此它们在小样本问题上表现较差.

实验结果表明, 无论在 CD-FSAD 场景或者域内 FSAD 场景, 与基线方法和其他先进方法相比, 本文方法都有更好的性能表现; 在提升 FSAD 场景性能的同时, 更能针对性地提升跨域场景表现.

3.4　消融实验

对本文提出的主要模块、风格记忆模块在 FRN 中的位置、内存块量和超参数 α 对模型性能的影响进行消融实验.

(1) 从 MVTec AD 到 MPDD 的 CD-FSAD 任务中, 对每个模块的有效性, 即训练中特征增强、测

表 3 MVTec AD 和 MPDD 间 CD-FSAD 及 FSAD 结果比较 %

场景	训练域	测试域	方法	*k*(图像 AUC)			*k*(像素 AUC)		
				2	4	8	2	4	8
CD-FSAD	MPDD	MVTec AD	RegAD[11]	82.1	85.1	87.7	94.0	94.9	95.9
			本文	**84.8**	**88.3**	**91.2**	**94.5**	**95.1**	**96.3**
	MVTec AD	MPDD	RegAD[11]	60.1	63.4	67.6	92.9	93.1	94.2
			本文	**65.1**	**68.5**	**72.1**	**93.1**	**93.9**	**95.2**
FSAD	MVTec AD	MVTec AD	TDG[12]	73.2	74.4	76.6			
			DifferNet[13]	80.6	81.3	83.2			
			RegAD[11]	85.7	88.2	91.2	**94.6**	**95.8**	96.7
			SPADE[28]	70.7	71.6	75.3	79.8	80.2	80.5
			STPM[29]	74.2	74.8	77.6	59.8	60.8	61.6
			RD4AD[2]	75.5	76.9	78.5	79.3	81.4	83.9
			本文	**85.8**	**89.9**	**92.5**	**94.6**	95.7	**96.9**
	MPDD	MPDD	TDG[12]	60.3	63.5	68.2			
			DifferNet[13]	60.2	63.3	68.5			
			RegAD[11]	63.4	68.3	71.9	93.2	93.9	95.1
			SPADE[28]	58.2	58.3	58.5	75.4	75.9	76.2
			STPM[29]	62.4	62.6	63.1	75.8	76.2	76.6
			RD4AD[2]	61.8	62.1	62.4	74.5	75.5	75.7
			本文	**65.9**	**69.7**	**73.2**	**93.9**	**94.4**	**95.3**

注. 粗体表示最优值.

试中特征增强和类别间对比损失进行消融实验, 结果如表 4 所示. 其中, 性能优异的 FSAD 方法 RegAD[11]被选为基线. 可以看出, 由于域差异较大, 没有学习域特征的基线方法表现最差; 在训练和推理中对特征进行风格化处理, 在不同样本数量下图像 AUC 指标分别提高 2.7 个百分点, 3.6 个百分点和 2.5 个百分点, 高于只使用类间损失的性能提升(即 0.9 个百分点, 2.3 个百分点和 1.2 个百分点), 表明学习到的风格特征起到了更重要的作用; 在类间对比损失的基础上, 进一步整合训练特征增强, 分别得到 1.5 个百分点, 0.6 个百分点和 0.9 个百分点的提高; 结合所有模块进一步缩小域间差异, CD-FSAD 性能相比基线方法得到显著提升(分别为 5.0 个百分点, 5.1 个百分点和 4.5 个百分点).

表 4 MVTec AD→MPDD 主要模块消融实验结果 %

训练增强	测试增强	类间损失	*k*		
			2	4	8
			60.1	63.4	67.6
✓	✓		62.8	67.0	70.1
		✓	61.0	65.7	68.8
✓		✓	62.5	66.3	69.7
✓	✓	✓	**65.1**	**68.5**	**72.1**

注. 粗体表示最优值.

(2) 从 MVTec AD 到 MPDD, 对风格记忆模块的位置进行消融实验, 结果如表 5 所示. 可以看出, 采用记忆机制存储风格信息和风格化特征对 CD-FSAD 任务是有利的; 此外, 将其放置在前 2 层后的效果最好, 而在第 3 层添加风格记忆模块反而会有性能损失. 可能的原因是在浅层的特征更具有通用性, 而高层的语义则不适合跨域任务.

表 5 风格记忆模块的位置消融实验结果 %

第 1 层	第 2 层	第 3 层	*k*		
			2	4	8
			60.1	63.4	67.6
✓			62.5	66.8	69.4
✓	✓		**65.1**	**68.5**	**72.1**
✓	✓	✓	63.4	67.4	70.9

注. 粗体表示最优值.

(3) 从 MVTec AD 到 MPDD, 对风格记忆模块中的内存块数量进行消融实验, 结果如表 6 所示. 可以看出, 随着内存块的数量 N 增多, 即存储的风格特征越多, 性能表现越好, 但在 $N=10$ 之前性能提升较大, 之后性能提升较小. 这是由于 10 个内存块已经足够模拟大量的风格信息, 因此实验

表 6　$k=2$ 时内存块数量 N 消融实验结果　　%

N	图像 AUC
0	60.1
5	62.8
10	65.1
15	65.3
20	**65.8**

注. 粗体表示最优值.

表 7　$k=2$ 时 α 的消融实验结果　　%

α	图像 AUC
1.0	61.9
0.9	**65.1**
0.7	64.9
0.4	63.3
0.0	62.1

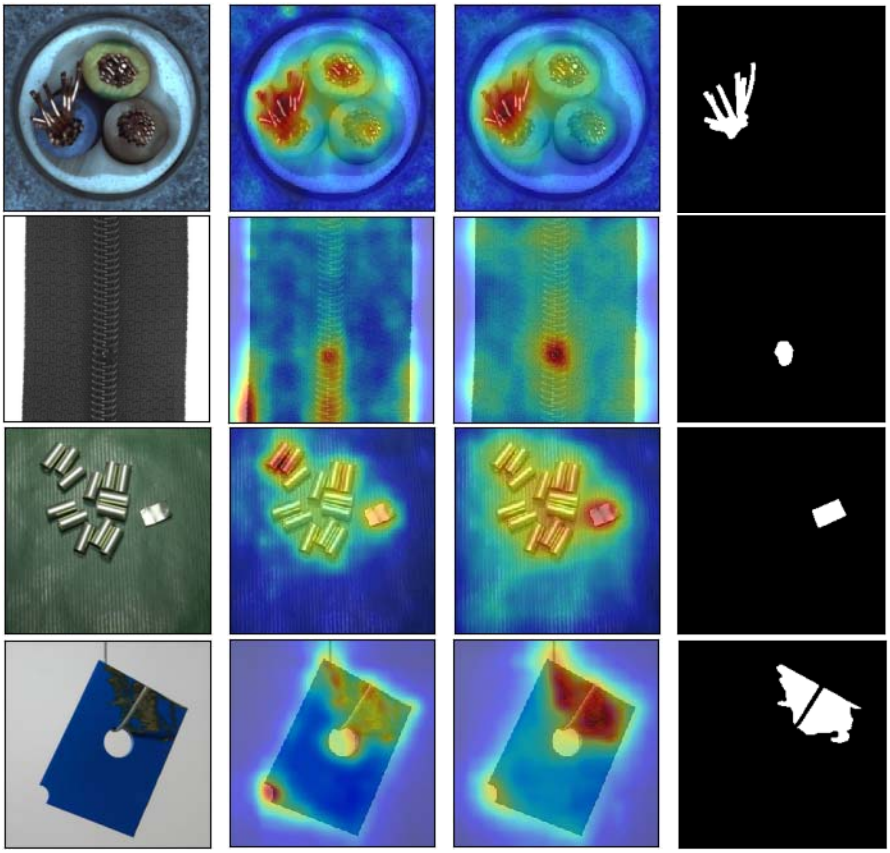
注. 粗体表示最优值.

中综合考虑内存消耗及速度，选择 $N=10$.

(4) 从 MVTec AD 到 MPDD，对超参数 α 进行消融实验，结果如表 7 所示. 可以看出，随着 α 的减小，性能先提升后下降，但无论 α 设置为多少，风格增强后的性能表现总是优于未使用内存模块前；当 $\alpha=1$ 时，内存模块只是初始化并不更新，因此有限的风格特征对性能提升影响较小；随着 α 减小，风格特征的多样性给模型带来较大的性能提升，且在 $\alpha=0.9$ 时 AUC 得到最大值；当 α 进一步减小时，虽然模拟了更多风格特征，但也对源域风格造成了破坏，使模型难以进行风格特征匹配，因此性能也逐渐下降. 当然，对 α 更细粒度的消融实验可能会取得更优异的性能表现，但从表 6 中可以看出，更多样性的内存块数量性能提升有限，因此本实验中选择 $\alpha=0.9$.

3.5　可视化分析

$k=2$ 时，对本文方法的异常定位性能进行定性分析，结果如图 4 所示. 其中，前 2 行是 MPDD 到 MVTec AD 的异常定位结果，后 2 行是 MVTec AD 到 MPDD 的定位结果；图 4b 所示为基线方法 RegAD 的热力图结果. 可以看出，本文方法比基线方法的定位结果更接近真实值，如第 2 行的皮革图像中，本文方法更关注于缺陷附近，而基线方法可能会额外关注到属于正常区域的边缘，说明本文方法对性能提升是有效的，可以更好地解决跨域小样本任务.



a. 异常图　　b. 原始热力图　　c. 本文方法热力图　　d. 真实值

图 4　$k=2$ 时 CD-FSAD 可视化结果

4 结 语

本文提出的 CD-FSAD 方法, 目的是将训练域的知识迁移到测试域的新类上. 首先提出一个风格记忆模块来连接源域和目标域, 包括减少域间差异加强语义特征表示, 在训练期间从训练数据中存储域的特征信息, 在学习风格化的特征同时将其转移到测试域中, 提升模型跨域检测能力; 然后提出类间对比损失, 进一步加强模型的表征能力. 充分的对比实验和消融实验结果表明, 本文方法在 CD-FSAD 场景下性能提升明显, 并且在域内 FSAD 设定下仍有优异表现.

当然在实际检测场景中, 对模型的推理速度有着极高的要求. 而当前的 CD-FSAD 方法由于风格记忆模块需要在训练期间存储大量的域特征信息, 并在测试阶段进行特征转移和融合, 无疑增加了模型的计算复杂度, 导致推理过程较为耗时, 难以满足实时性检测的应用场景.

因此, 在未来研究中主要从 2 个方面展开研究: (1) 考虑引入轻量级的神经网络架构来替代部分复杂的模块, 以提高模型的推理速度, 使得模型在减少参数数量的和计算量的同时, 仍能保持较好的特征提取能力. (2) 对风格记忆模块进行优化, 一方面, 可以研究如何对存储的域特征信息进行压缩和编码, 减少存储量和计算量; 另一方面, 可以探索在特征转移过程中如何采用更轻量级的计算方式, 避免不必要的计算开销.

参考文献(References):

[1] Roth K, Pemula L, Zepeda J, *et al*. Towards total recall in industrial anomaly detection[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 14298-14308

[2] Deng H Q, Li X Y. Anomaly detection via reverse distillation from one-class embedding[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 9727-9736

[3] Yu J W, Zheng Y, Wang X, *et al*. FastFlow: unsupervised anomaly detection and localization via 2D normalizing flows[OL]. [2023-07-20]. https://arxiv.org/abs/2111.07677

[4] Gudovskiy D, Ishizaka S, Kozuka K. CFLOW-AD: real-time unsupervised anomaly detection with localization via conditional normalizing flows[C] //Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Los Alamitos: IEEE Computer Society Press, 2022: 1819-1828

[5] Defard T, Setkov A, Loesch A, *et al*. PaDiM: a patch distribution modeling framework for anomaly detection and localization[C] //Proceedings of the Pattern Recognition. ICPR International Workshops and Challenges. Heidelberg: Springer, 2021: 475-489

[6] Zavrtanik V, Kristan M, Skočaj D. Reconstruction by inpainting for visual anomaly detection[J]. Pattern Recognition, 2021, 112: Article No.107706

[7] Hou J L, Zhang Y Y, Zhong Q Y, *et al*. Divide-and-assemble: learning block-wise memory for unsupervised anomaly detection[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2021: 8771-8780

[8] Salehi M, Sadjadi N, Baselizadeh S, *et al*. Multiresolution knowledge distillation for anomaly detection[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 14897-14907

[9] Bergmann P, Fauser M, Sattlegger D, *et al*. MVTec AD-A comprehensive real-world dataset for unsupervised anomaly detection[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 9584-9592

[10] Jezek S, Jonak M, Burget R, *et al*. Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions[C] //Proceedings of the 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops. Los Alamitos: IEEE Computer Society Press, 2021: 66-71

[11] Huang C Q, Guan H Y, Jiang A F, *et al*. Registration based few-shot anomaly detection[C] //Proceedings of the 17th European Conference on Computer Vision. Heidelberg: Springer, 2022: 303-319

[12] Sheynin S, Benaim S, Wolf L. A hierarchical transformation-discriminating generative model for few shot anomaly detection[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2021: 8475-8484

[13] Izmailov P, Kirichenko P, Finzi M, *et al*. Semi-Supervised learning with normalizing flows[C] //Proceedings of the 37th International Conference on Machine Learning, Lille: PMLR Press, 2020: 4615-4630

[14] Weston J, Chopra S, Bordes A. Memory networks[OL]. [2023-07-20]. https://arxiv.org/abs/1410.3916

[15] Jeon S, Hong K, Lee P, *et al*. Feature stylization and domain-aware contrastive learning for domain generalization[C] //Proceedings of the 29th ACM International Conference on Multimedia. New York: ACM Press, 2021: 22-31

[16] Park H, Noh J, Ham B. Learning memory-guided normality for anomaly detection[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2020: 14360-14369

[17] Wang J D, Lan C L, Liu C, *et al*. Generalizing to unseen domains: a survey on domain generalization[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(8): 8052-8072

[18] Koh P W, Sagawa S, Marklund H, *et al*. WILDS: a benchmark of in-the-wild distribution shifts[C] //Proceedings of the 38th International Conference on Machine Learning. Lille: PMLR Press, 2021: 5637-5664

[19] Du Y T, Wang J D, Feng W J, *et al*. AdaRNN: adaptive learning

and forecasting of time series[C] //Proceedings of the 30th ACM International Conference on Information & Knowledge Management. New York: ACM Press, 2021: 402-411

[20] Kim D, Yoo Y, Park S, *et al*. SelfReg: self-supervised contrastive regularization for domain generalization[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2021: 9599-9608

[21] He K M, Zhang X Y, Ren S Q, *et al*. Deep residual learning for image recognition[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 770-778

[22] Jaderberg M, Simonyan K, Zisserman A, *et al*. Spatial transformer networks[C] //Proceedings of the 28th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2015: 2017-2025

[23] Chen X L, He K M. Exploring simple siamese representation learning[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 15745-15753

[24] Zhou K Y, Yang Y X, Qiao Y, *et al*. Domain generalization with MiStyle[OL]. [2023-07-20]. https://iclr.cc/media/iclr-2021/Slides/2738.pdf

[25] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 1510-1519

[26] Chen T, Kornblith S, Norouzi M, *et al*. A simple framework for contrastive learning of visual representations[C] //Proceedings of the 37th International Conference on Machine Learning. Lille: PMLR Press, 2020: 1597-1607

[27] Bergmann P, Batzner K, Fauser M, *et al*. Beyond dents and scratches: logical constraints in unsupervised anomaly detection and localization[J]. International Journal of Computer Vision, 2022, 130(4): 947-969

[28] Wang G D, Han S M, Ding E R, *et al*. Student-teacher feature pyramid matching for anomaly detection[C] //Proceedings of the 32nd British Machine Vision Conference. Durham: BMVA Press, 2021: 1-14

[29] Reiss T, Cohen N, Bergman L, *et al*. PANDA: adapting pretrained features for anomaly detection and segmentation[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 2805-2813