基于深度度量学习的稳健装配进度估计

预印本

Kazuma Miura中央大学理工研究生院精密工程课程，日本东京
miura@sensor.mech.chuo-u.ac.jp

Sarthak Pathak中央大学理工学部精密机械系，日本东京

Kazunori Umeda中央大学理工学部精密机械系，日本东京

arXiv:2601.00422v1 [cs.CV] 1 Jan 2026

2026年1月5日

摘要

近年来，人工智能技术的进步加速了智能工厂的发展。特别是，对产品装配进度进行自动监控，对于提高运营效率、最大限度减少废弃零件的成本以及提升工厂生产力至关重要。然而，对于需要跨多天手动执行装配任务的情况，实施智能工厂系统仍面临挑战。以往的研究提出了 Anomaly Triplet-Net，通过对产品的视觉特征应用深度度量学习来估计装配进度。然而，当连续任务之间的视觉变化较为细微时，往往会发生误分类。为了解决这一问题，本文提出了一种稳健的装配进度估计系统，即使在存在遮挡或视觉变化极小的情况下，也能利用小规模数据集进行估计。我们的方法采用基于四元组损失的学习方法处理异常图像，并引入自定义数据加载器，有策略地选择训练样本，以提高估计准确率。我们使用一组在台式电脑装配过程中采集的图像数据集评估了该方法。所提出的异常四元组网络在该数据集上的表现优于现有方法。具体而言，在台式电脑数据集上，该方法将估计准确率提高了 1.3%，并将相邻任务之间的误分类率降低了 1.9%，证明了所提方法的有效性。

关键词 第一个关键词 ·第二个关键词 ·更多

1 引言

1.1 研究背景

近年来，人工智能技术的进步加速了制造工厂的数字化转型，这一趋势被称为智能工厂倡议 [1, 2]。这些努力对于提高盈利能力发挥着至关重要的作用。尤其是，了解工厂内部的运营速度有助于提升工作流程的效率。对于

由多个组件组装而成的产品而言，管理组件库存并监控装配进度对于尽量降低多余零件的处置成本至关重要。因此，智能工厂系统正越来越多地应用于生产线。

此类生产线中使用的工业机器人包括 SCARA机器人（图1）和大型焊接机器人（图2）。这些机器人可以全天候运行，速度也比人类更快，从而显著提高生产效率。然而，在流水线生产系统中制造的产品相对容易部署传感技术，但在非标准环境下进行人工装配的场景中，传感技术的应用则面临挑战。单元式生产系统就是此类场景的典型例子。
图3展示了人类与机器人协作式单元生产系统的概况，图4则描绘了人类与机器人协同工作的场景。这类系统中的传感之所以困难，原因有以下几点。首先，人工装配往往没有严格统一的标准，因此不同工人的操作流程可能各不相同。其次，对于持续数天的任务，要将操作流程标准化本身就很有挑战性；即便实现了标准化，工人也可能难以始终如一地遵循这些流程。
因此，智能工厂实践尚未在许多现实工厂中广泛采用。进度跟踪仍常由人工完成，例如将进度写在纸上并贴在产品上。为解决这一问题，一些研究尝试通过人体动作识别来估计进度 [3, 4, 5]。然而，这类方法通常仅适用于可在较短时间内完成的相对简单任务，未必适用于现实工厂环境。

为克服这些局限，一些研究着重于根据产品的视觉特征而非人体动作来估计进度。其中一项研究采用实例分割识别产品部件，并利用这些信息估计装配进度 [6]。该方法通过对各个部件的训练图像进行遮罩处理，力求在数据集较小的情况下仍达到较高的准确率，从而有助于降低实际应用成本。

然而，在经常发生遮挡的实际工厂环境中，小部件可能会被完全遮住，导致部件检测的准确率降低，进而降低进度估计的准确率。
相比之下，另一类研究着重于捕捉产品的整体外观，而非检测各个部件 [7]。这种方法利用深度度量学习来估计装配进度。通过将代表不同装配阶段的图像作为训练数据，进度估计任务被重新表述为分类问题。此外，模型还使用合成遮挡图像进行训练，使其能够应对实际场景中的遮挡。实验表明，即使在存在遮挡的情况下，也能成功估计进度。
然而，仍然存在一项重大挑战：当连续的任务在外观上差异很小时，由于各阶段之间的视觉变化微乎其微，模型往往会将它们错误分类。
为解决这一问题，本研究提出了一种稳健的进度估计系统，重点关注已装配产品的外观特征。具体而言，我们旨在构建一种系统，即使在存在遮挡且装配步骤之间的视觉变化极小的情况下，也能通过小规模数据集训练准确估计进度。

1.2 相关工作

1.2.1 基于骨骼关键点信息的方法

日立工业控制解决方案提出了一种方法，通过将工人的骨骼关键点信息作为时间序列数据进行捕捉，来估计任务进度 [11]。在该方法中，使用网络摄像头或监控摄像机等通用摄像机作为固定位置摄像机，以捕捉工人的动作。系统从录制的视频中提取骨骼关键点，并根据这些关键点的动作和姿势估计任务完成时间。

该方法的一个优点是不依赖可穿戴设备，因此不会干扰工人的动作。然而，该方法容易受到不同工人体格和工作速度差异的影响，可能导致进度估计不准确。此外，在产品装配持续数天的场景中，每个短时子任务往往没有严格标准化，而是以个性化、临时性的方式完成。因此，仅凭骨骼关键点信息可能不足以在此类情况下准确估计进度。



图 1：装配线中使用的 SCARA机器人 [8]

1.2.2 聚焦于产品状态的方法

Oshima等人提出了一种基于深度学习的进度估计方法 [12]。该方法的概述如图5所示。该方法使用与预定义装配进度阶段对应的裁剪后产品图像训练模型。这些带标签的图像被输入ResNet [13]，（一种卷积神经网络（CNN）），模型则输出一个预定义的进度阶段。

然而，许多装配过程的一个共同特点是，直到装配接近完成时，产品的视觉外观才会发生显著变化。各阶段之间缺乏视觉差异可能导致误分类，这给该方法带来了挑战。

1.2.3 聚焦于产品组件的方法

Yumoto等人提出了一种基于实例分割 [6]的进度估计方法。该系统首先训练一个 YOLACT [14]模型，以检测产品装配所需的组件。然后，通过识别产品中存在的组件类型，估计当前的装配阶段。

检测到的组件组合与进度阶段之间的关系是预先定义的。为了降低在实际工厂中的部署成本，使用随机擦除 [15]，进行数据增强，使模型能够使用更小的数据集进行训练。这种增强技术既提高了组件检测效果，也提升了进度估计效果

然而，当组件较小或距离相机较远时，这种方法会面临挑战。在这种情况下，遮挡可能会使组件完全不可见，从而导致检测失败，进而造成进度估计不准确。



图 2：装配线上使用的大型焊接机器人 [9]

1.3 研究目标

本研究是与一家工业合作伙伴共同开展的合作研究项目。因此，最终目标是建立一种可应用于实际工业环境的方法。目前，产品装配的进度由现场工人手动记录，并通过在产品上贴附手写标签进行管理。

然而，从工厂的角度来看，人们迫切需要仅使用固定位置摄像机远程监控装配进度，而无需直接前往现场。此外，由于现场工人通常不熟悉深度学习，因此必须设计一种不会设置障碍的系统，例如不要求手动创建数据集。

由于知识产权和保密限制，无法直接使用真实工厂产品的数据，因此本研究的实验采用了一个模拟工厂产品的数据集。尽管实验使用的是模拟数据集，但所提出方法的预期应用对象是真实工厂产品。

本研究所考虑的目标产品是半导体检测设备，通常需要三到四天完成装配。图 6 展示了这类设备的一个示例。

由于知识产权方面的限制，本文无法纳入合作公司的实际工厂图像。因此，我们在图 7 中提供了一张工厂环境示意图。此外，图 8 和图 9 展示了大型装配任务的示例，例如服务器装配。

摄像机布置受到物理条件的限制，拍摄的产品视频往往会因其他产品、工人或工具而受到遮挡。此外，由于相机距离较远，装配产品的外观可能只发生细微变化，若缺乏领域知识，便难以察觉这些变化。

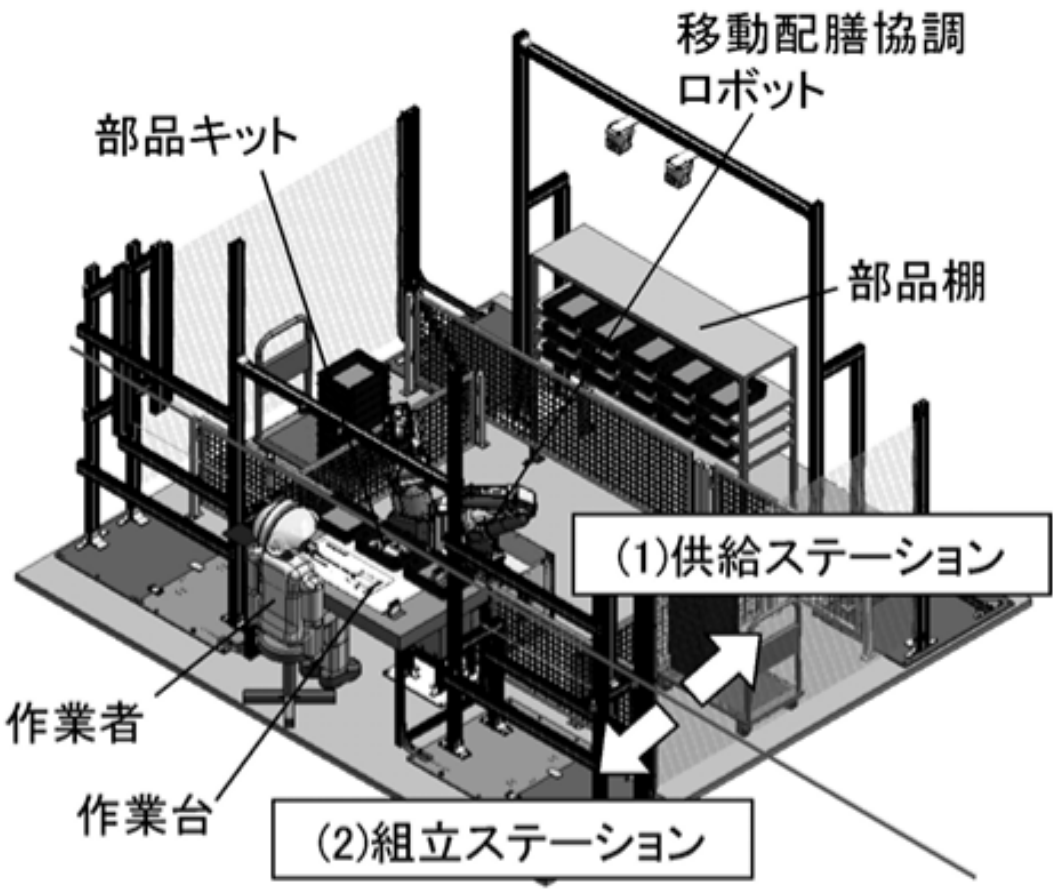


图3：人机协作单元生产系统概览 [10]

因此，本研究旨在开发一种稳健的系统，使其能够在小规模数据集上运行，并聚焦于装配目标的视觉特征，同时能够适应装配过程中的遮挡和细微外观变化。



图4：人类与机器人实际协作工作的示例 [10]

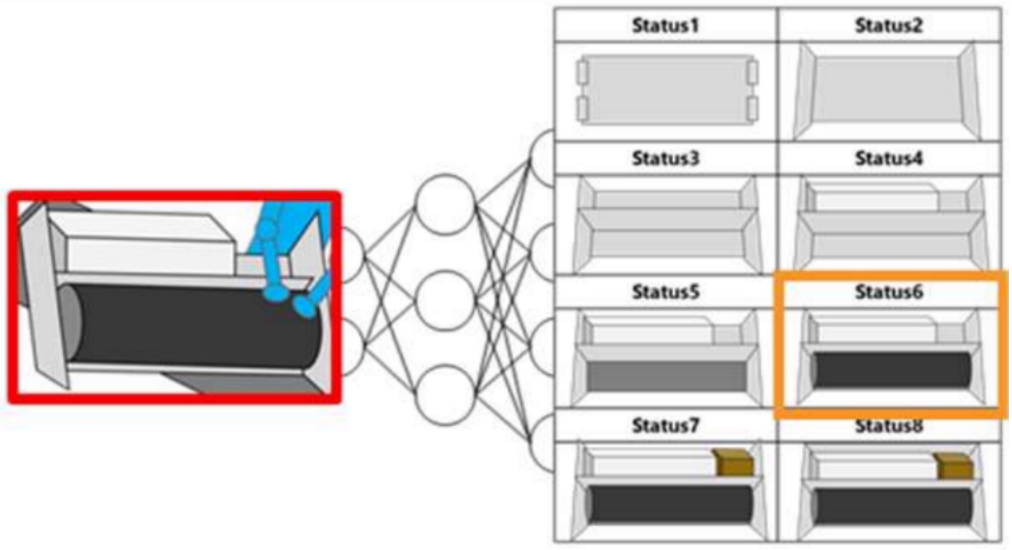


图5：基于产品状态的进度估计 [12]



图 6：高加速度扫描电子显微镜测量系统 [16]

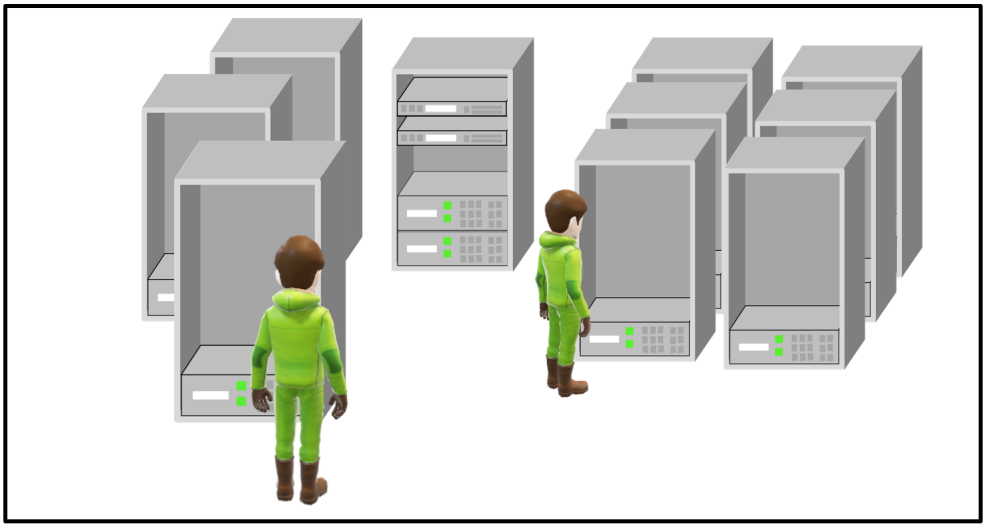


图 7：假设的工厂环境示例



图8：大规模服务器组装示例（1） [17]



图9：大规模服务器组装示例（2） [18]

2 提出的方法

2.1 装配进度估计方法

图 10 展示了所提出系统的概览。本研究以 Anomaly Triplet-Net 框架为基础，引入了多项改进，以实现更高的进度估计准确率。其主要改进在于，将原有的进度估计模型替换为一个新提出的模型，用于处理检测到的产品图像。此外，还实施了多项修改，包括重新设计训练过程中使用的数据加载器、调整特征空间的维度，以及改进损失函数。

所提出的方法尤其着重于减少相邻任务之间的误分类——这些任务在视觉外观上仅有细微变化，且连续执行。通过解决这一问题，该方法有望提高其应用于经过时序处理的视频数据时的进度估计准确率。具体而言，在以往的研究中，时序建模未能提高涉及此类相邻任务场景下的准确率。所提出的改进旨在克服这一局限，并提升估计性能。

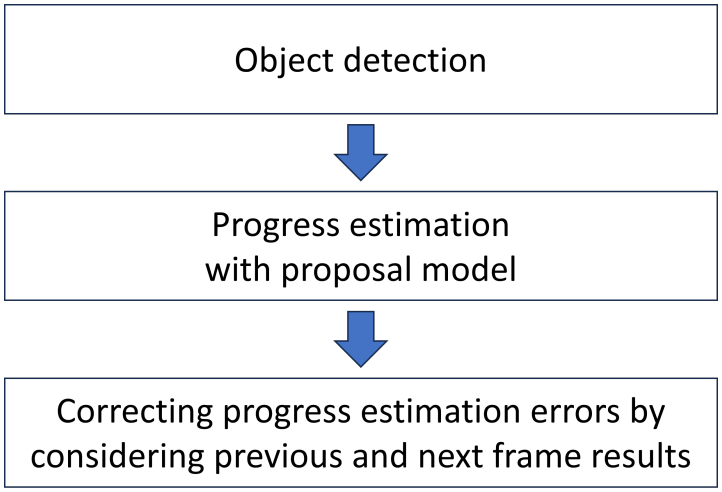


图10：所提出系统的示意图

2.2 装配产品检测

使用目标检测算法 YOLOv8[19] 检测用于进度估计的目标对象。随后创建自定义数据集并进行微调。微调是指从预先初始化的权重开始，对预训练网络的权重重新训练。

本研究考虑到，创建数据集的工作可能由人工智能知识有限的工厂工人负责，因此应尽量降低数据集创建成本。因此，同时构建用于 YOLOv8 的自定义数据集和用于所提出的异常四元组网络的数据集。

首先创建少量 YOLO 格式标注样本。使用这些少量数据集训练模型，并在推理期间用其裁剪产品图像。随后，通过目视确认正确检测的样本，并将其归入相应的进度阶段（步骤）。

按步骤划分的数据集用于异常四元组网络，而未按步骤分类的统一数据集则用于 YOLO 训练。推理期间，在自定义数据集上训练的 YOLO 会检测产品区域，随后将这些区域裁剪出来并传入深度度量学习模型，以进行进度估计。

2.3 考虑遮挡的异常四元组网络模型

2.3.1 异常四元组网络的整体结构

所提出的进度估计模型——异常四元组网络——的整体架构如图11所示。该模型包含两个阶段：训练阶段和推理阶段。在训练阶段，卷积神经网络的权重会得到优化。在推理阶段，利用学习到的权重将未见数据映射到特征空间中。

为处理视觉差异细微的类别，我们设计了一种自定义数据加载器，用于策略性地选取训练样本，以计算四元组损失[21]。

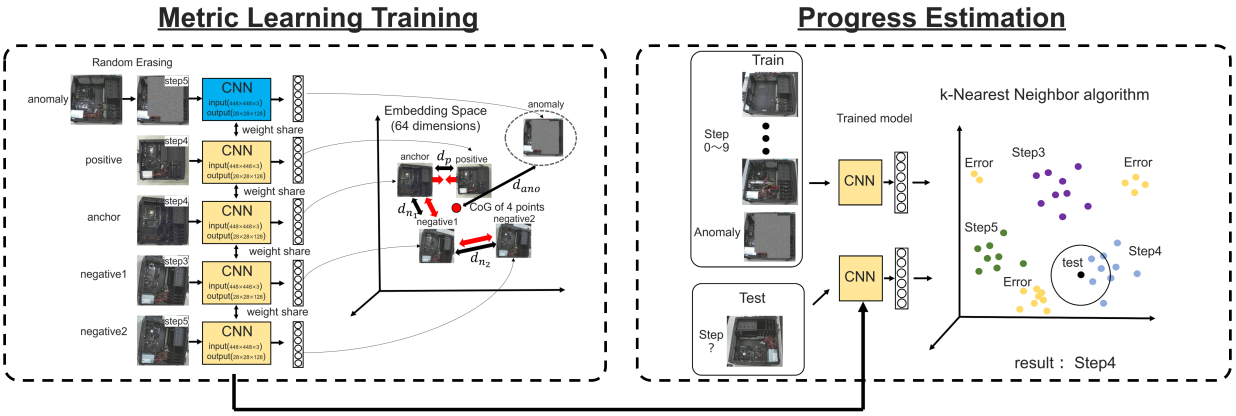


图 11：异常四元组网络总体结构

2.3.2 训练阶段

训练阶段的模型架构如图 12 所示。首先，根据装配流程定义进度步骤，并据此准备训练。从装配视频中裁剪出用于进度估计的物体，并按对应步骤保存。

利用保存的步骤图像，准备五种类型的训练样本：锚点样本、正样本、负样本1、负样本2和异常样本。随后使用这些样本进行训练。每个样本都会经过 CNN特征提取器。该 CNN 由四个卷积层组成，输入为一张 448×448×3 图像（高度 448、宽度 448、3 个通道），输出为一张 28×28×128 特征图。CNN 的权重在所有输入之间共享。

经过卷积神经网络层后，特征会传入全连接层，并被投影到任意维度的特征空间中。随后，使用所提出的损失函数在该特征空间中进行距离度量学习。

异常四元组网络通过将卷积神经网络提取的特征嵌入任意维度的特征空间来进行度量学习。因此，需要确定该特征空间的维度。合适的维度取决于输入图像的模式、训练轮次等因素，通常根据经验知识和实验结果来确定。然而，当维度达到两位数以上时，特征空间中的距离可能会变得几乎无法区分。为避免这种被称为“维数灾难”的现象，通常认为有必要将维度限制在两位数或更少。图 13 [23]展示了维数灾难。

异常样本的生成方法使用随机擦除 [15]，以随机像素值遮盖图像中随机选取的矩形区域。图 14 展示了应用随机擦除后带有噪声的图像示例。通过遮挡产品图像的部分区域，我们生成了伪遮挡图像，即使人类也无法据此估计装配进度。

此外，本方法中使用的随机擦除设置如下所述。在训练期间，随机擦除仅应用于从异常样本所属类别以外的类别中选取的输入图像。训练期间的样本选择过程经过专门设计，将非异常类别的图像用作随机擦除的目标，从而避免异常特定特征退化。

矩形区域面积与输入图像面积之比在 0.02 至 0.4 的范围内随机选取。矩形区域的宽高比在 0.3 至 3.3 之间随机选取。对每张图像执行两次随机擦除操作。

在损失函数的设计中，我们考虑了遮挡和外观特征变化较小的任务。我们通过改进传统方法 Anomaly Triplet-Net 中各样本的位置关系，并引入四元组损失，对该方法进行了改进。所提出的损失函数如公式（1）所示。

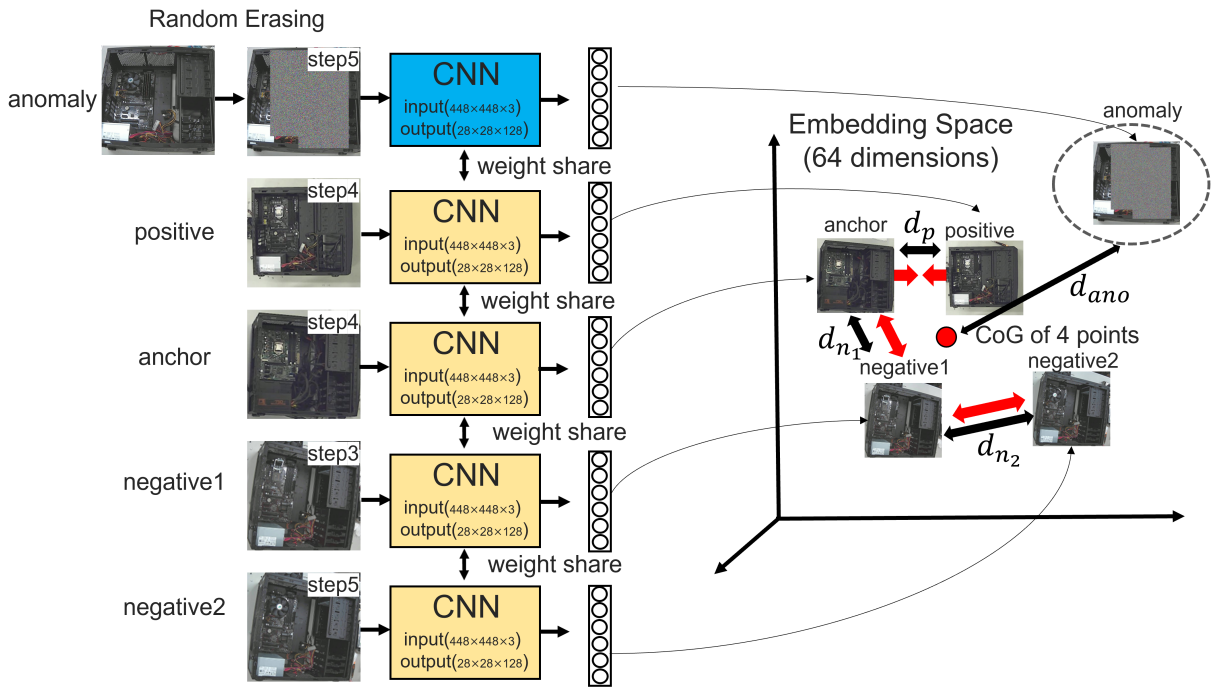


图12：训练阶段的模型架构

$$L_{\text{AnomalyQuadruplet}} = \max(d_p - d_{n1} + m_\alpha, 0) + \max(d_p - d_{n2} + m_\beta, 0) + \lambda \max(d_p - d_{ano} + m_c, 0) \quad (1)$$

d_p、 d_{n1}和 d_{n2} 分别表示锚样本与正样本、负样本1和负样本2之间在特征空间中的距离。 d_{ano} 表示异常样本与四个样本（即锚点样本、正样本、负样本1和负样本2）的质心之间在特征空间中的距离。

m_α、 m_β和 m_c 是间隔参数，用于定义训练期间样本之间的目标距离。 λ 是一个加权系数，用于调整涉及 d_p 和 d_{ano} 的项相对于涉及 d_p和 d_{n1}以及 d_p 和 d_{n2}的项的贡献。换言之， λ 控制从哪个训练轮次开始使用异常样本进行训练。

采用这种设计，模型首先专注于充分学习四元组损失部分。在充分学习四元组损失后，逐步引入使用异常样本进行的训练，从而更有效地学习异常样本。

例如，考虑一个包含100个训练轮次的训练过程，其中从第50个训练轮次开始使用异常样本进行学习。在这种情况下， λ 在第49个训练轮次之前设为0，然后在第50至第100个训练轮次之间逐渐从0增加到1。

为确保在外观特征变化较小时仍能识别，并尽可能降低计算成本，我们精心设计了数据加载器。通过合理构建数据集，可以减少相邻装配步骤之间的误分类，同时降低训练所需的PC配置要求和处理时间。

当数据集包含 n个类别，且每个类别由 m 张图像组成时，TripletLoss[20] 的数据组合可能数量由式（2）表示，而四元组损失的数据组合可能数量由公式（3）表示。

$$ {}_nC_1 \times {}_mP_2 \times {}_{n-1}C_1 \times {}_mP_1 \quad (2)$$

$$ {}_nC_1 \times {}_mP_2 \times {}_{n-1}C_1 \times {}_mP_1 \times {}_{n-2}C_1 \times {}_mP_1 \quad (3)$$

例如，当数据集由八个类别组成，且每个类别包含四十张图像时，三元组损失的组合数量约为350万，而四元组损失的组合数量约为8亿。如此巨大的计算成本给处理时间和计算资源带来了严重问题。

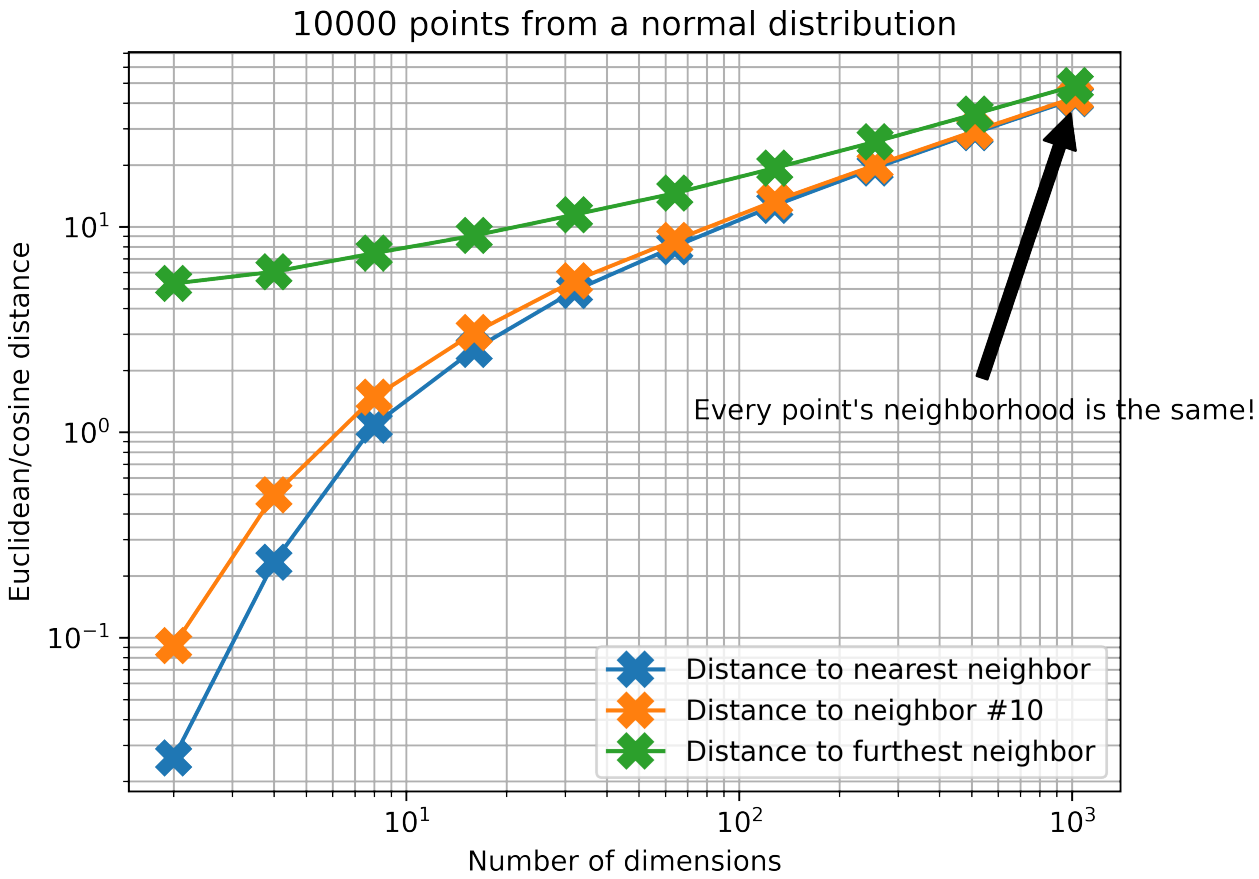


图 13：维数灾难示意图 [24]

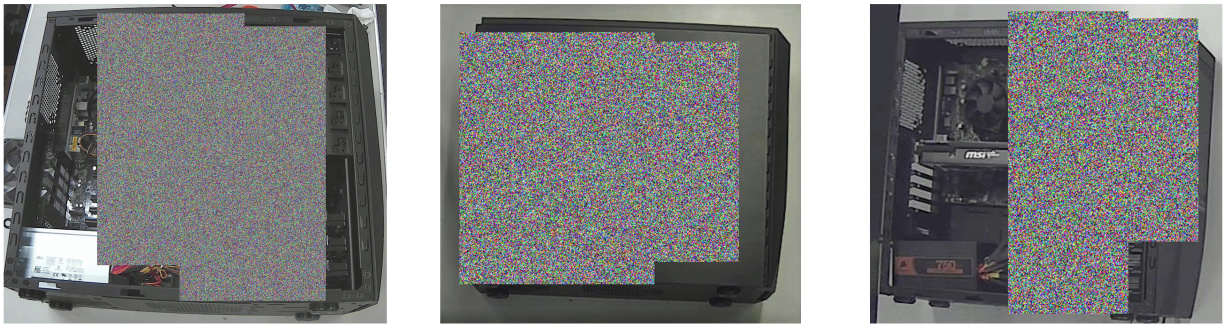


图 14：使用随机擦除添加噪声生成的异常样本

为解决这一问题，所提出的方法关注这样一个事实：连续的类别对应于连续的装配步骤，因此具有相似的外观特征。通过仅选择三个连续类别之间的组合，训练数据组合的数量得以显著减少。

在异常四元组网络中，从每个 n 类别中选取一个锚样本。正样本选自与锚样本相同但来自不同图像的类别。负样本1和负样本2分别选自紧邻锚点样本类别之前和之后的类别。异常样本则通过对从锚点样本类别以外的类别中选取的图像应用随机擦除 [15] 生成。

为了在特征空间中充分分离相似类别，将负样本1和负样本2的类别进行交换，并重新生成组合。因此，组合总数减少至

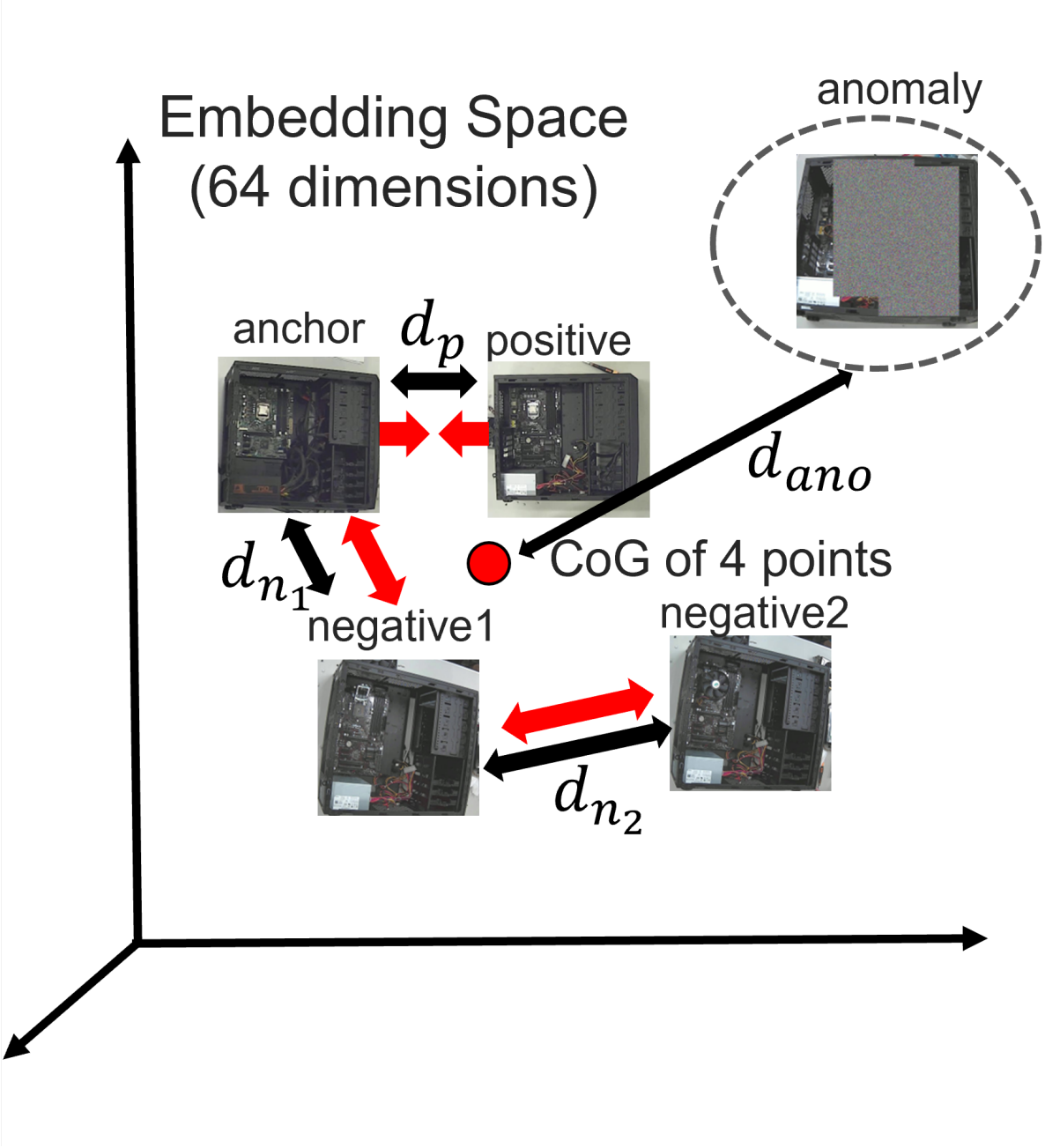


图15：各样本在特征空间中的相对位置

$2 \times n \times m$。这种设计保留了所有类别的聚类结构，并能够正确学习特征空间的全局结构。

图16展示了训练数据组合策略。

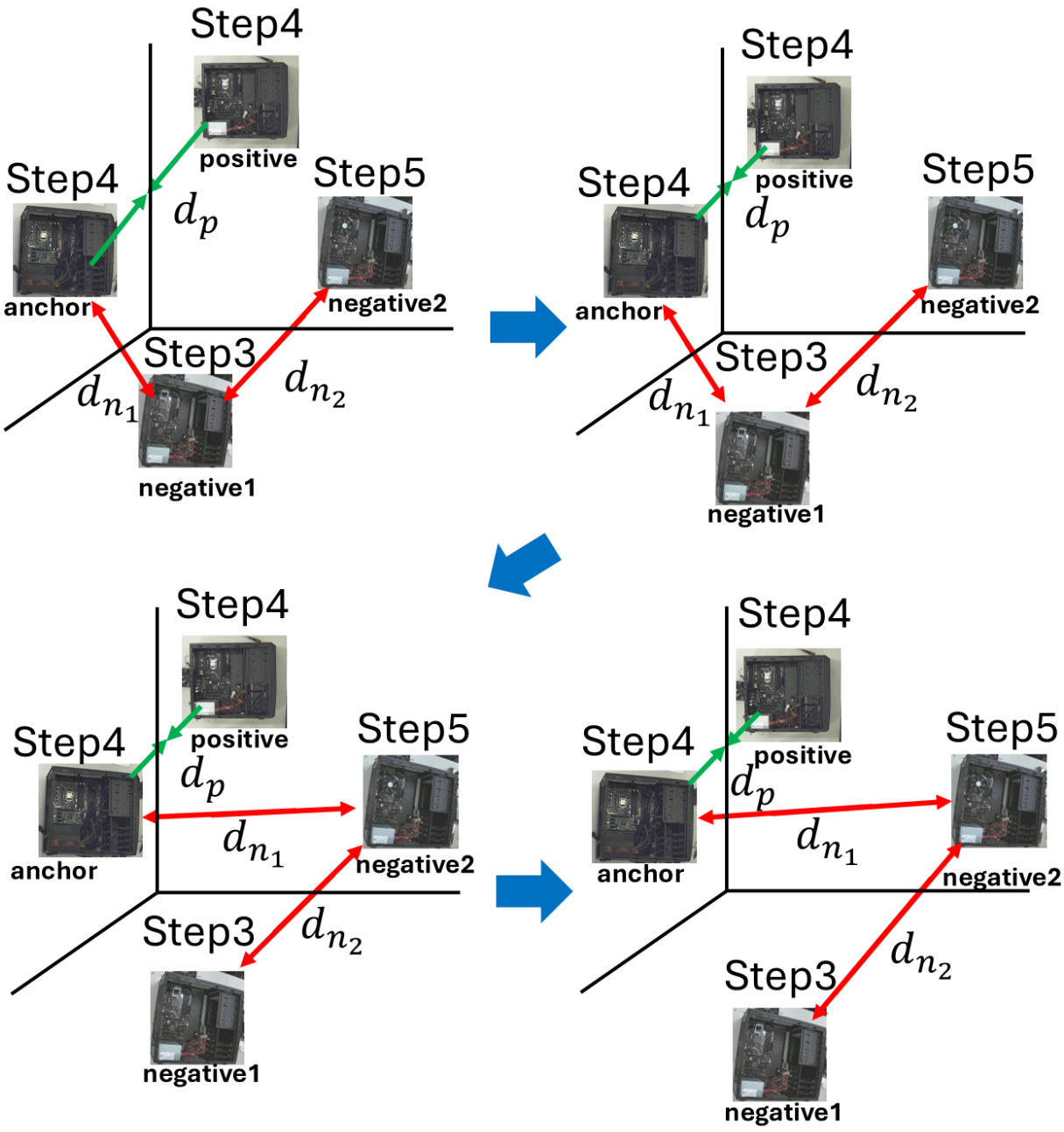


图16：训练期间使用的样本组合

3 推理阶段

在推理阶段，使用训练阶段得到的已训练模型。对于未知输入数据，通过在训练数据的特征空间中搜索邻近样本来进行估计。通过这种方式，该方法从训练数据集包含的类别以及训练数据中不包含的异常图像中识别出最接近的类别，并将结果作为进度估计输出。

3.1 使用k近邻方法进行步骤估计

推理模型如图17所示。与先前的工作 Anomaly Triplet-Net 类似，首先将训练期间使用的每个步骤的图像分别输入已训练模型，并嵌入特征空间，转换为特征向量。随后，使用已训练模型将待估计的输入图像嵌入同一特征空间。通过对这些嵌入数据应用k近邻（kNN）算法 [22]，估计未知输入数据所属的步骤。

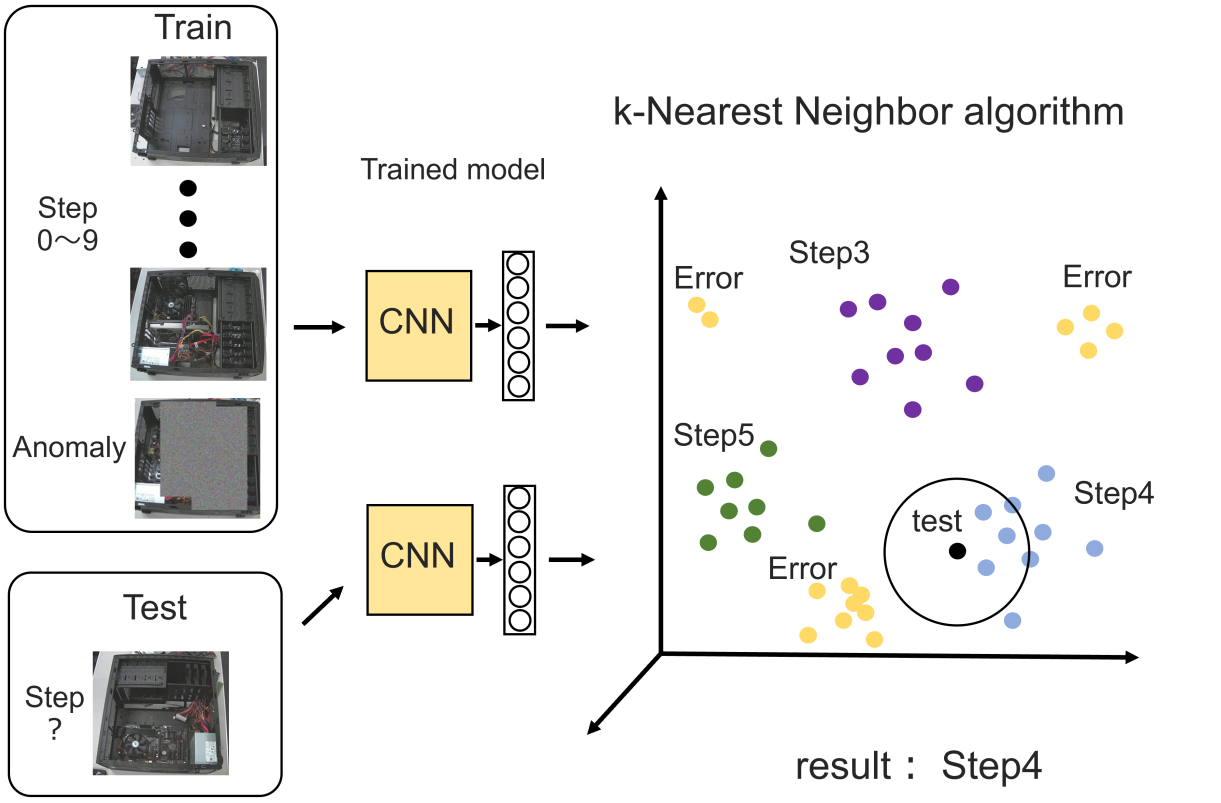


图17：推理阶段的模型结构

在实际环境中，由于产品或工具排列紧密，或者人工操作员正在执行装配任务，产品图像中可能会出现遮挡。尤其是在工业应用中，提前将此类情况收集为数据集并进行显式学习十分困难。因此，在所提出的方法中，训练期间将生成的伪遮挡图像作为异常样本，并嵌入特征空间中。

然而，在估计未知错误图像时，其嵌入可能位于特征空间中远离训练伪遮挡图像嵌入的位置。为处理此类情况，除基于 k近邻 方法的进度估计外，还引入了距离阈值。如果未知样本与训练样本之间的距离超过预定义阈值，则不应用基于 k近邻 的估计。

图18展示了错误判定机制。根据基于 k近邻 的估计，test1 和 test2 分别被分类为属于步骤4和步骤3。然而，由于 test2 在特征空间中远离步骤3，因此通过应用距离阈值，将其更正为错误判定。用于此判定的距离计算为测试样本与其 k 个最近邻之间的平均距离。

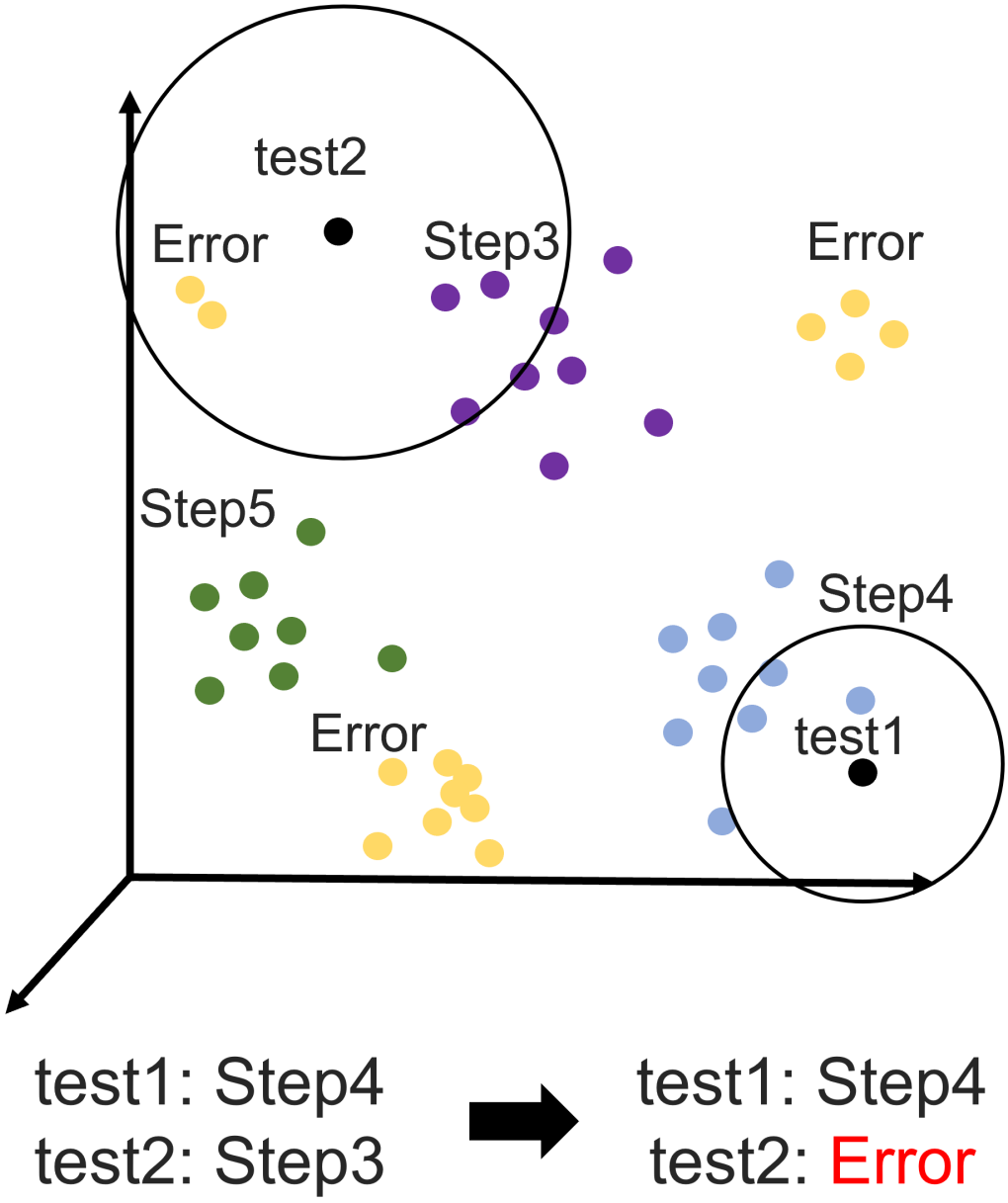


图18：在特征空间中使用距离阈值进行错误判定

3.2 实验

本小节介绍对所提出的异常四元组网络进行的评估实验。通过将所提出的异常四元组网络与参数经过调整的既有方法进行比较，开展了评估。为验证所提出的异常四元组网络的有效性，我们将其与既有方法 Anomaly Triplet-Net 进行了对比实验。为此，我们对 Anomaly Triplet-Net 的训练参数进行了实验。具体而言，我们通过改变训练轮次总数、异常学习的起始训练轮次、模型特征空间的维度、k近邻中的 k 值，以及推理期间的错误判定阈值来开展实验。本小节讨论了适用于既有方法的有效训练参数，并利用这些参数得出进度估计结果，将其用于与所提出的异常四元组网络进行对比实验。实验使用了台式电脑组装图像。

3.2.1 实验环境

首先，实验采用了与先前研究 Anomaly Triplet-Net 相同的数据集。因此，使用了台式电脑数据集。图19展示了台式电脑组装图像中的各个步骤。这些图像是通过相机将台式电脑组装过程录制成视频，再从视频中提取产品图像构建数据集而得到的。相机采用了 I-O DATA DEVICE株式会社生产的网络摄像机 Qwatch TS-WRLP[25] ，图像分辨率设为 1980×1080。

3.2.2 实验条件

待分类的步骤包括从步骤1到步骤8的八个步骤，以及对应异常检测的错误类别，共计九个类别。各步骤的判定标准如下所述。步骤1表示仅有电脑机箱的状态。步骤2表示已安装电源的状态。步骤3表示已安装主板的状态。步骤4表示CPU已安装在主板上的状态。步骤5表示CPU散热器已安装在主板上的CPU上的状态。步骤6表示GPU已安装在主板上的状态。步骤7表示HDD已安装在电脑机箱中的状态。步骤8表示已装上电脑机箱盖的状态。作为异常数据，数据集中仅添加了发生人类造成的遮挡的状态。需要注意的是，异常数据不包含在训练数据集和验证数据集中，也不会在训练期间使用。也就是说，异常数据仅用于测试训练好的模型。

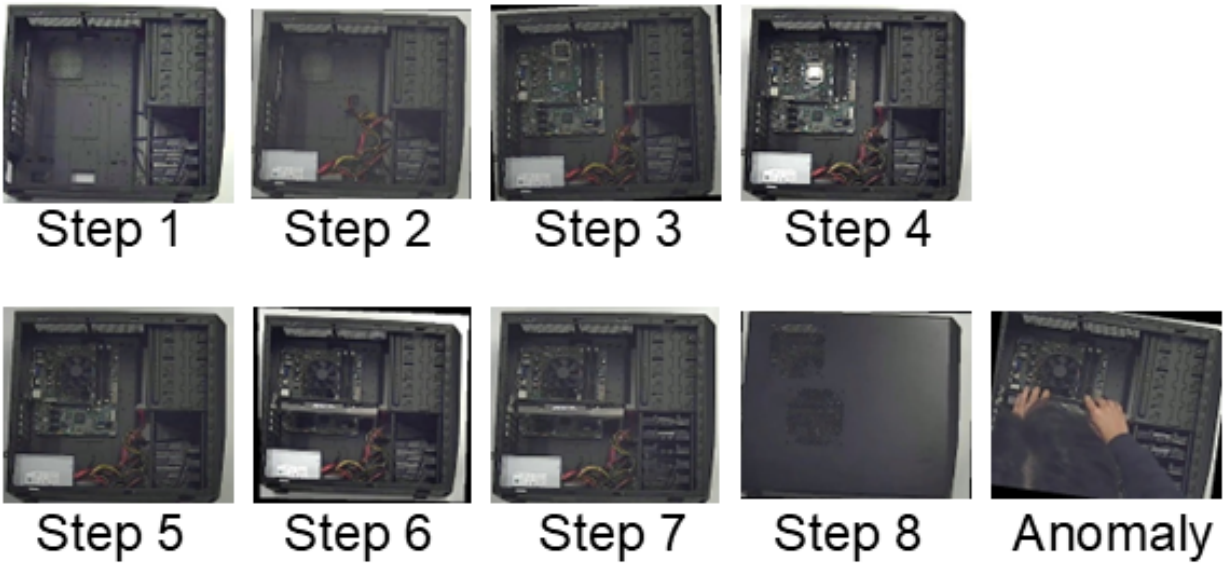


图19：台式电脑数据集

构成该数据集的台式个人电脑图像与 Anomaly Triplet-Net 论文中所述数据集使用的图像相同。训练数据集共包含320张图像，步骤1至步骤8各有40张。验证数据集共包含1,264张图像，每个步骤各有150张。测试数据集共包含1,600张图像，包括步骤1至步骤8的图像以及错误图像。

3.2.3 实验结果与讨论

首先，在接近原始 Anomaly Triplet-Net 论文所述的条件下进行了实验。训练条件设置如下：总训练轮次为 100，从第 50 轮开始进行异常学习， $k = 10$ 对于 kNN，特征空间的维度为 128，学习率设为 $lr = 0.0001$。该条件称为*AnomalyTriplet* 条件 *1*。训练结果包括准确率、损失、特征空间中的距离，以及使用 t-SNE[26], 对特征空间进行的二维可视化，分别如图 20、图 21、图 22 和图 23 所示。此外，错误判定距离阈值设为 1,425,000 时，进度估计结果的混淆矩阵如图 24 所示。测试数据集的准确率为 74.3%。

从图 20 和图 21 可以观察到，验证数据集的估计准确率下降，而损失增加，表明发生了过拟合。此外，图 24 显示步骤3与步骤4之间出现了误分类。另一方面，图 22 和图 23 表明，特征空间中的距离学习本身已成功完成。

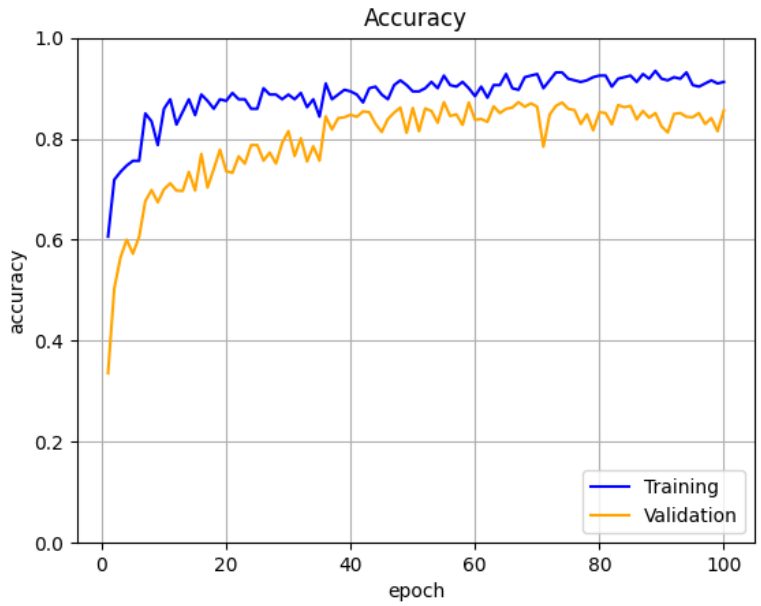


图 20：AnomalyTriplet 条件 1：准确率随训练轮次的变化

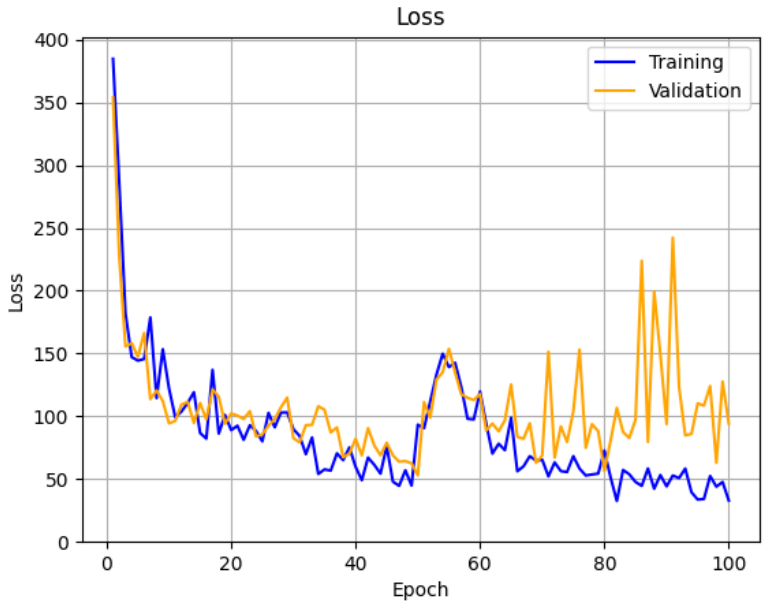


图 21：AnomalyTriplet 条件 1：损失随训练轮次变化的情况

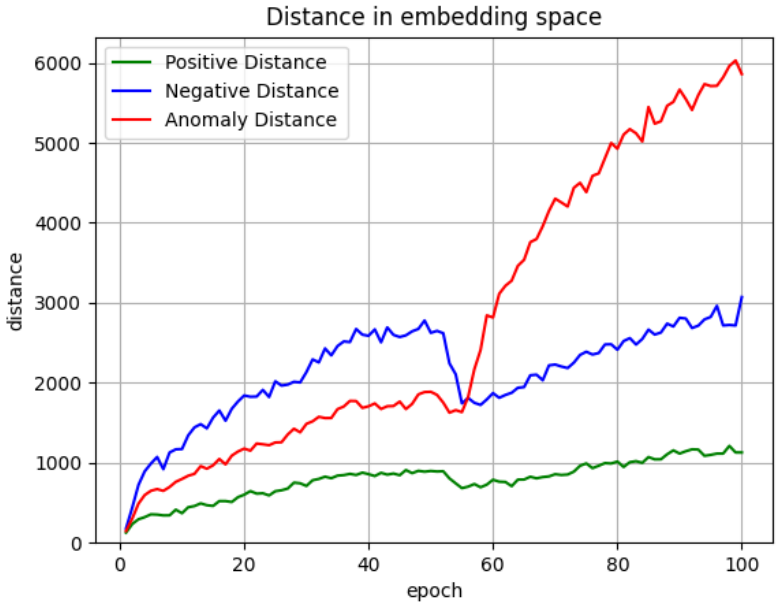


图 22：AnomalyTriplet 条件 1：特征空间中样本之间的距离

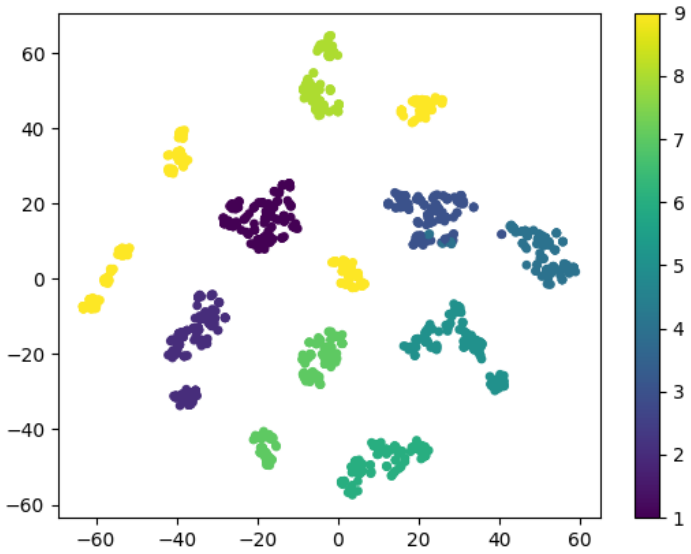


图 23 ：AnomalyTriplet 条件 1：使用 t-SNE 对特征空间进行二维可视化

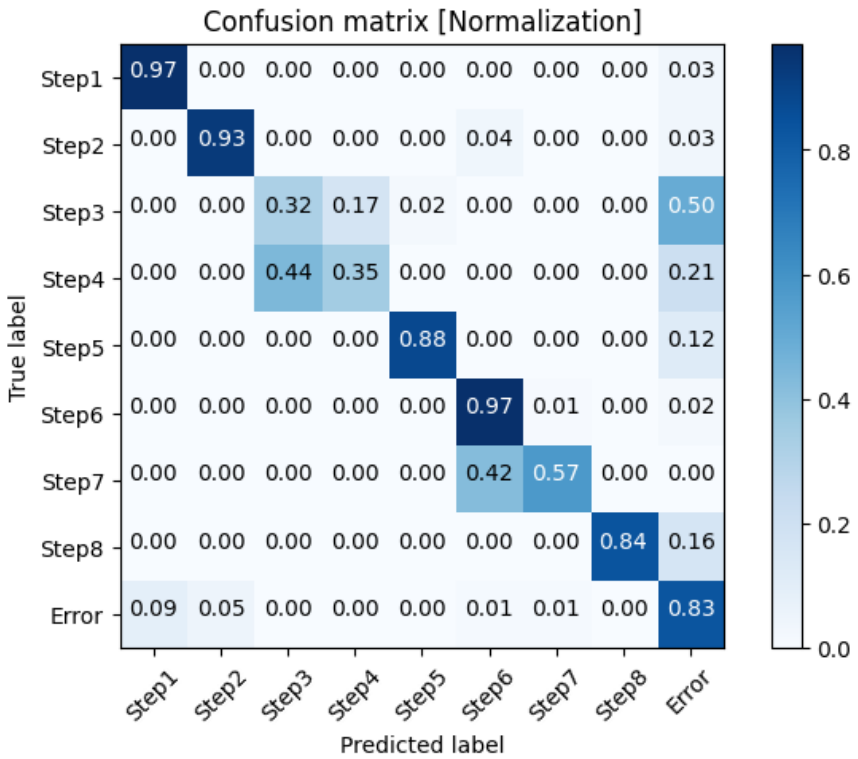


图 24：AnomalyTriplet 条件 1：使用混淆矩阵进行评估

尽管已经学习了特征空间中样本之间的距离，但特征空间本身是一个维度达到三位数的高维空间，这可能会导致维数灾难。因此，我们降低了特征空间的维度，并再次进行了实验。训练条件设置为将特征空间维度降至 64，其余条件均与 AnomalyTriplet 条件 1 相同。该条件称为*AnomalyTriplet 条件2*。训练结果包括准确率、损失、特征空间中的距离以及使用 t-SNE 进行的二维可视化，分别如图 25、图 26、图 27 和图 28 所示。与条件 1 一样，错误判定距离阈值设为 1,425,000 时的进度估计结果混淆矩阵如图 29 所示。测试数据集的准确率显著提高至 95.4%。

从图 27 和图 28 可以看出，与条件 1 类似，成功地在特征空间中进行了距离学习。通过在两位数维度的特征空间中正确学习距离，避免了维数灾难，从而提高了估计准确率。此外，图 25 和图 26 表明，将特征空间维度降至 64 后，无需改变训练轮次即可避免过拟合。这可能是因为，降低特征空间维度减少了深度学习所学习的特征数量，因此需要更多训练轮次才能实现适当的学习。

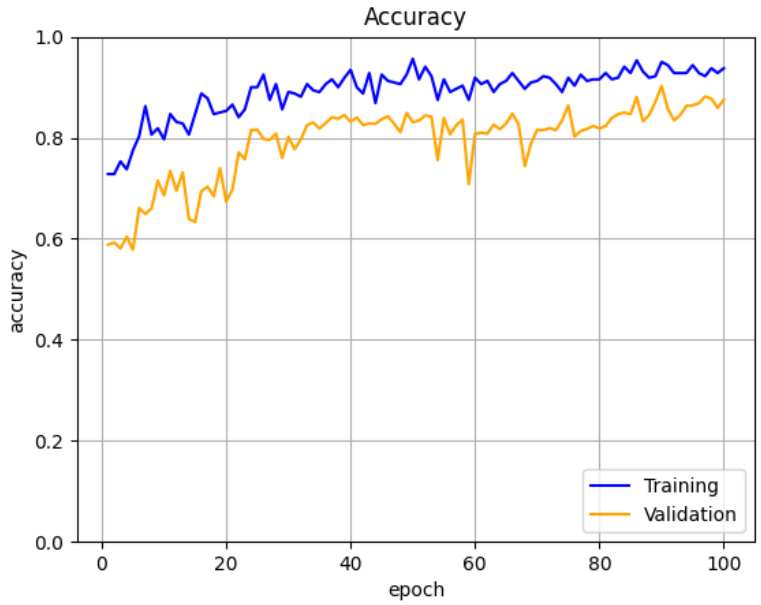


图 25：异常　Triplet 异常条件2：准确率随训练轮次的变化

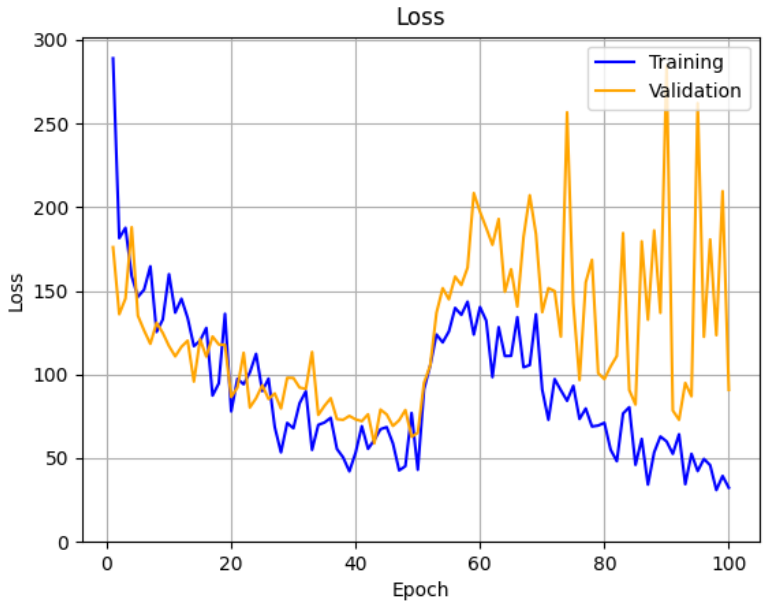


图 26：AnomalyTriplet 条件 2：损失随训练轮次的变化

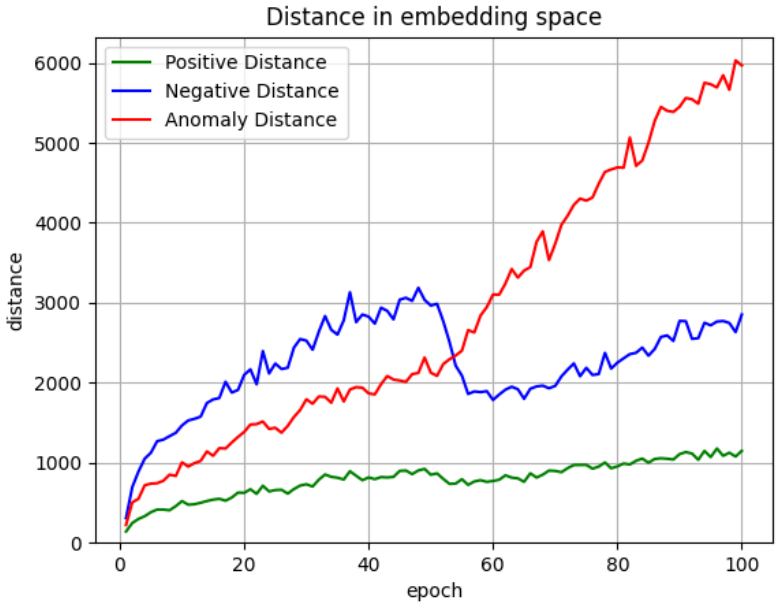


图 27：AnomalyTriplet 条件 2：特征空间中样本之间的距离

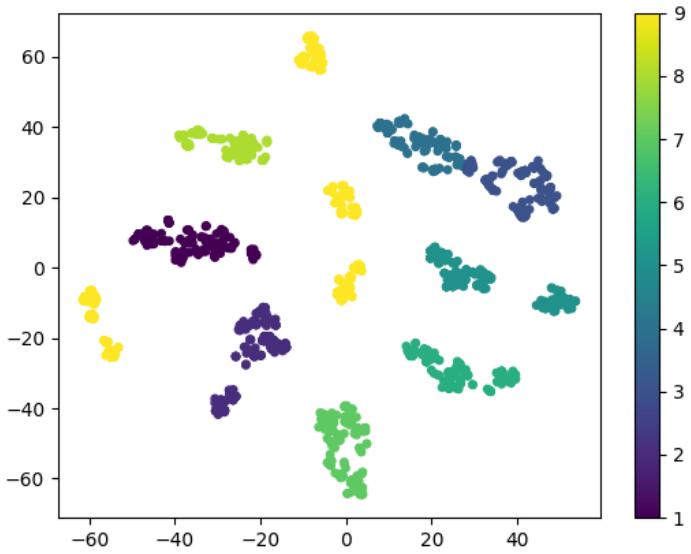


图28 ：AnomalyTriplet 条件 2：使用 t-SNE 对特征空间进行二维可视化

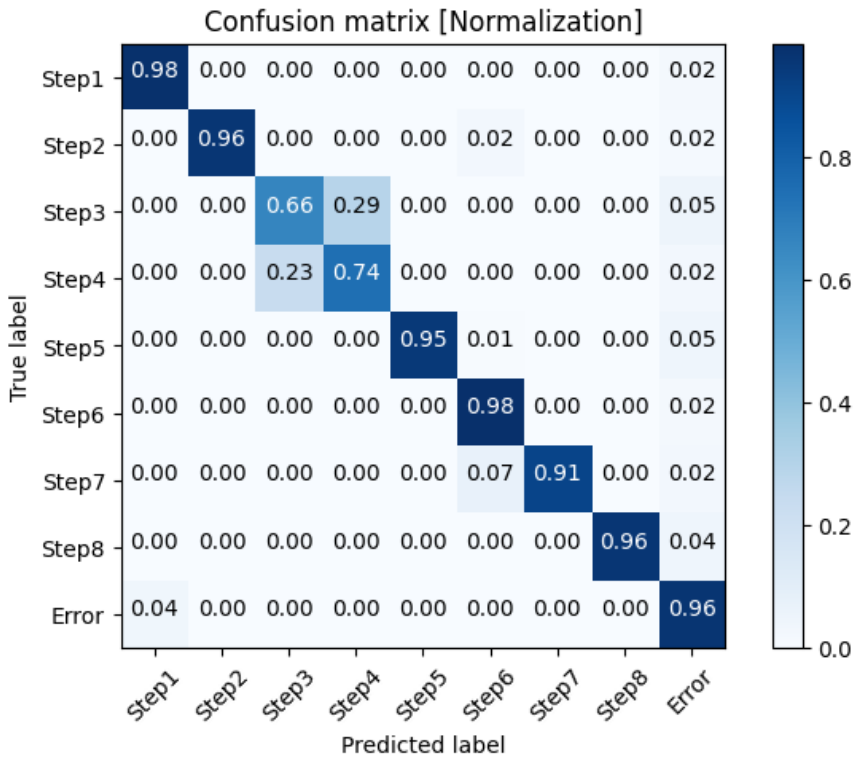


图28：AnomalyTriplet 条件 2：使用混淆矩阵进行评估

由于降低特征空间维度被发现有助于提高估计准确率，因此通过进一步降低维度开展了更多实验。此外，当维度降低至 64 时，由于特征数量减少，过拟合更早发生。因此，训练轮次总数也相应增加。

训练条件设置如下：训练轮次总数为 200，异常学习从第 100 个训练轮次开始，$k = 10$ 对于 k近邻，将特征空间维度降低至 32，并将学习率设置为$lr = 0.0001$。该条件称为*AnomalyTriplet* 条件 *3*。训练结果包括准确率、损失、特征空间中的距离，以及使用 t-SNE 对特征空间进行的二维可视化，分别如图 30、图 31、图 32 和图 33 所示。进度估计结果的混淆矩阵如图 34 所示。测试数据集的准确率为 90.6%。

尽管图 30、图 32、图 33 和图 34 表明学习过程较为顺利，但图 31 显示，验证损失在约 50 个训练轮次后迅速恶化。我们认为，这种性能下降是由异常学习从第 100 个训练轮次才开始这一事实造成的。在异常学习开始之前，模型仅使用传统的三元组损失进行训练。因此，在异常学习开始之前，三元组损失学习部分便已过拟合，导致整体估计准确率下降。

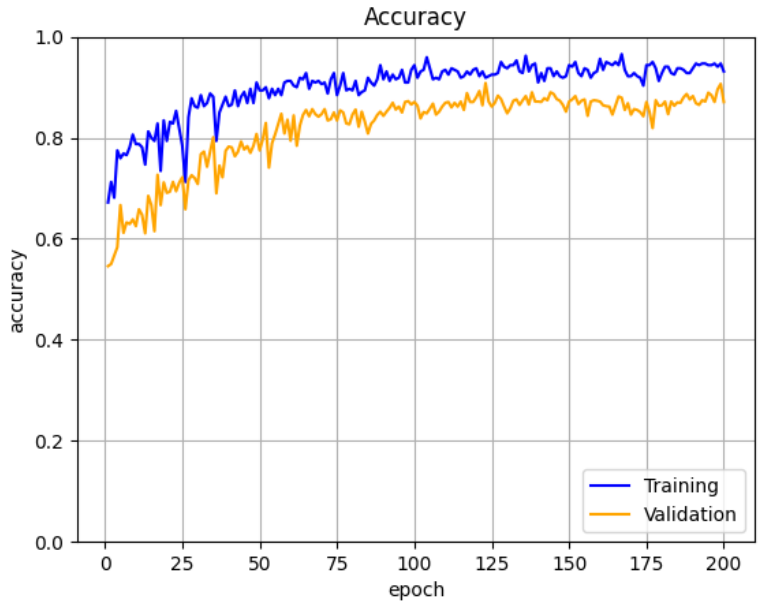


图 30：异　　常三元组条件 3：准确率随训练轮次数量的变化

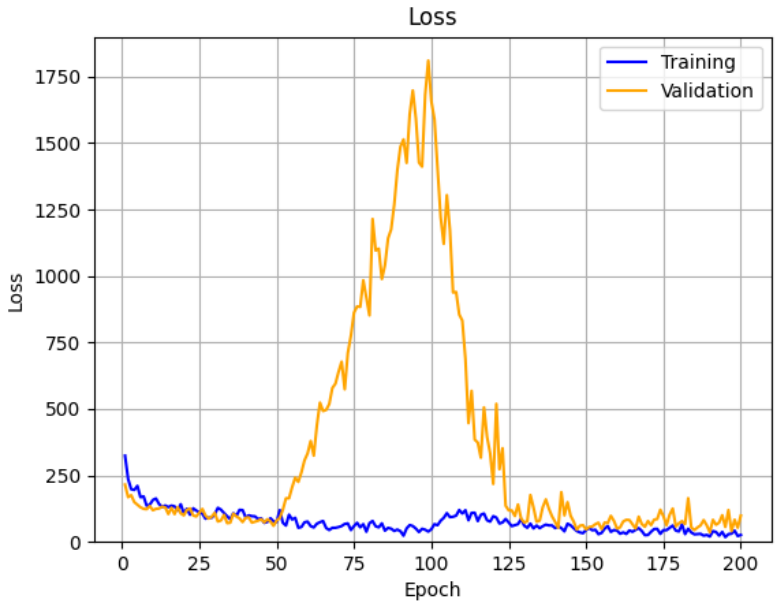


图31：AnomalyTriplet 条件 3：损失随训练轮次变化的情况

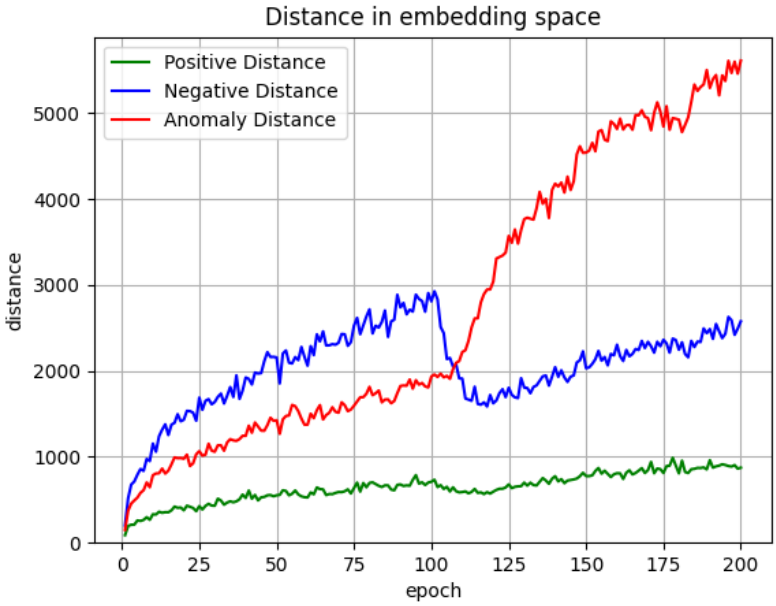


图32：AnomalyTriplet 条件 3：特征空间中的样本间距离

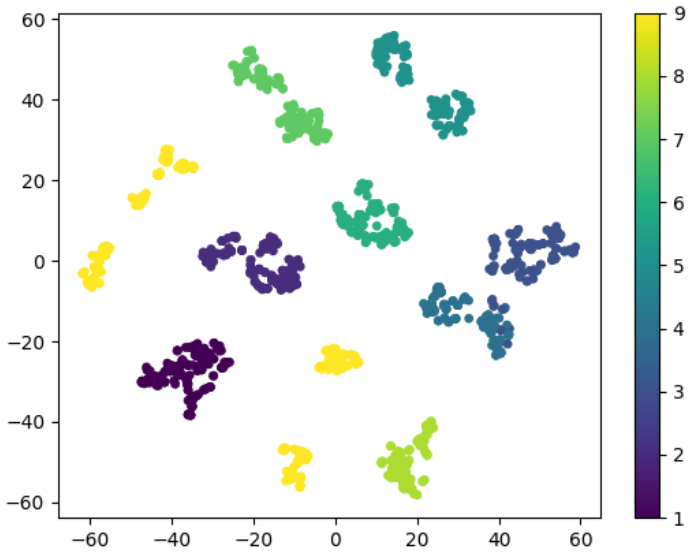


图33 ：AnomalyTriplet 条件 3：使用 t-SNE 对特征空间进行二维可视化

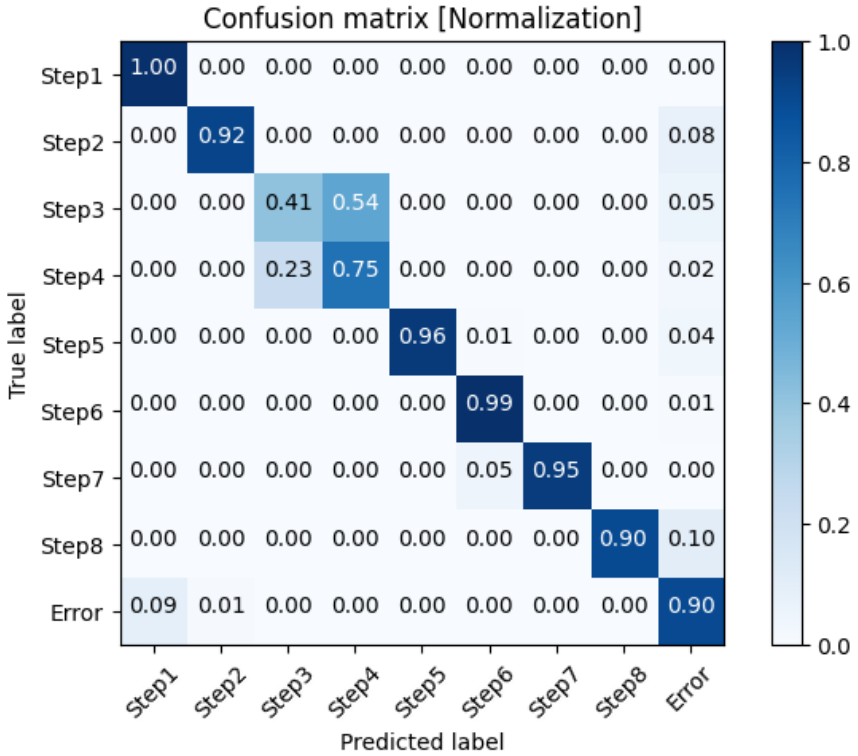


图34：AnomalyTriplet 条件 3：使用混淆矩阵进行评估

为避免过拟合，同时保持特征空间维度为 32，我们在修改后的训练条件下重新进行了实验。为防止过拟合，与 AnomalyTriplet 条件 3 相比，我们减少了训练轮次总数。

训练条件设置如下：训练轮次总数为 100，异常学习从第 50 轮开始，$k = 10$ 对于 k近邻，特征空间维度为 32，学习率设为 $lr = 0.0001$。此条件称为 ***AnomalyTriplet*** 条件 ***4***。训练结果（包括准确率、损失、特征空间中的距离以及使用 t-SNE 进行的二维可视化）分别如图 35、图 36、图 37 和图 38 所示。进度估计结果的混淆矩阵如图 39 所示。测试数据集的准确率为 76.3%。

从图 35 和图 36 可以确认，在异常学习开始的第 50 训练轮次之前不会发生过拟合。此外，第 50 训练轮次之后，准确率持续提高，直到第 100 训练轮次，这表明训练 100 轮次仍不充分，还需要进一步学习。

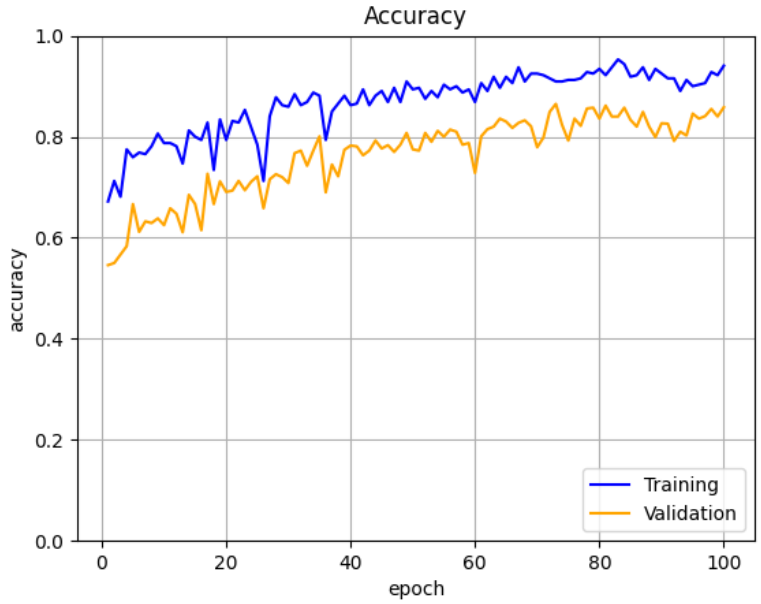


图 35：AnomalyTriplet 条件 4：准确率随训练轮次数量的变化

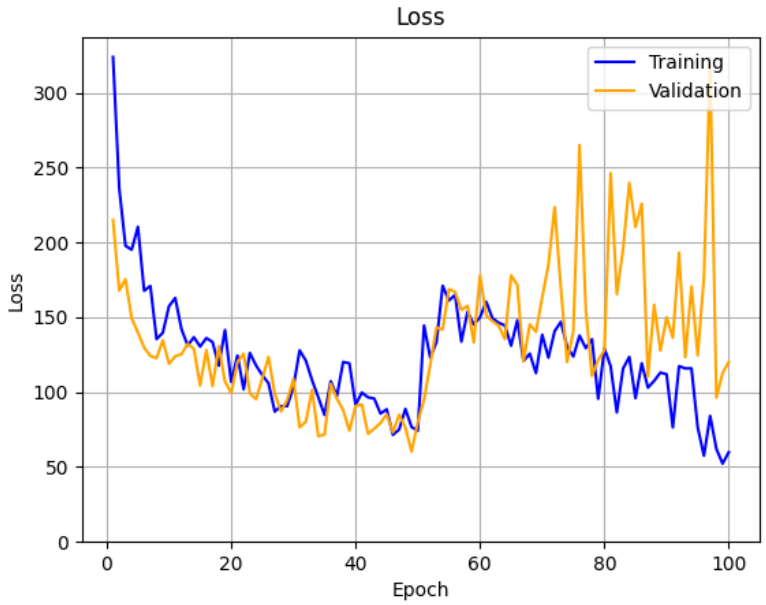


图36：AnomalyTriplet 条件 4：损失随训练轮次的变化

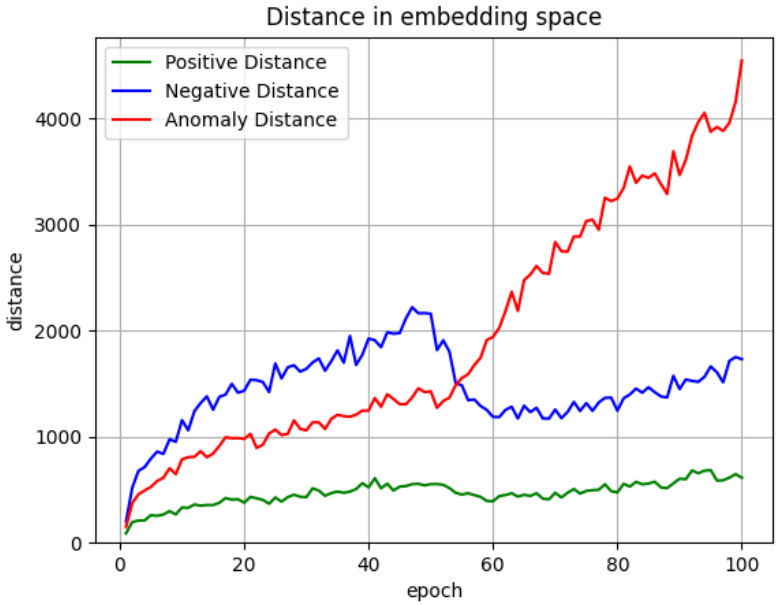


图37：AnomalyTriplet 条件 4：特征空间中样本之间的距离

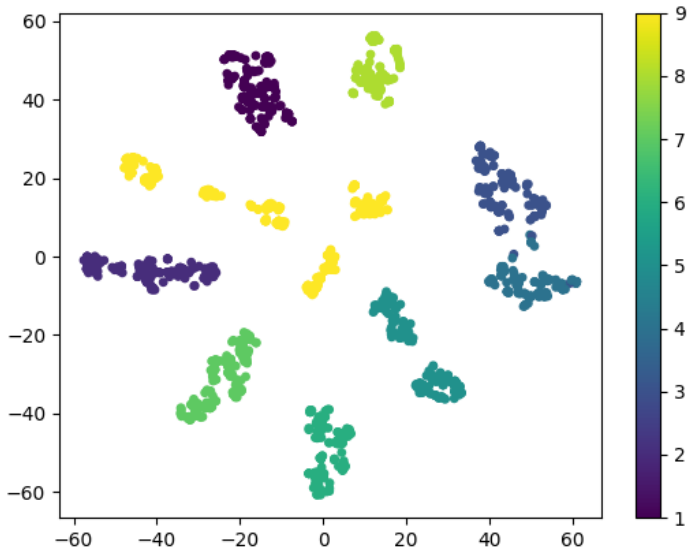


图38 ：AnomalyTriplet 条件 4：使用 t-SNE 对特征空间进行二维可视化

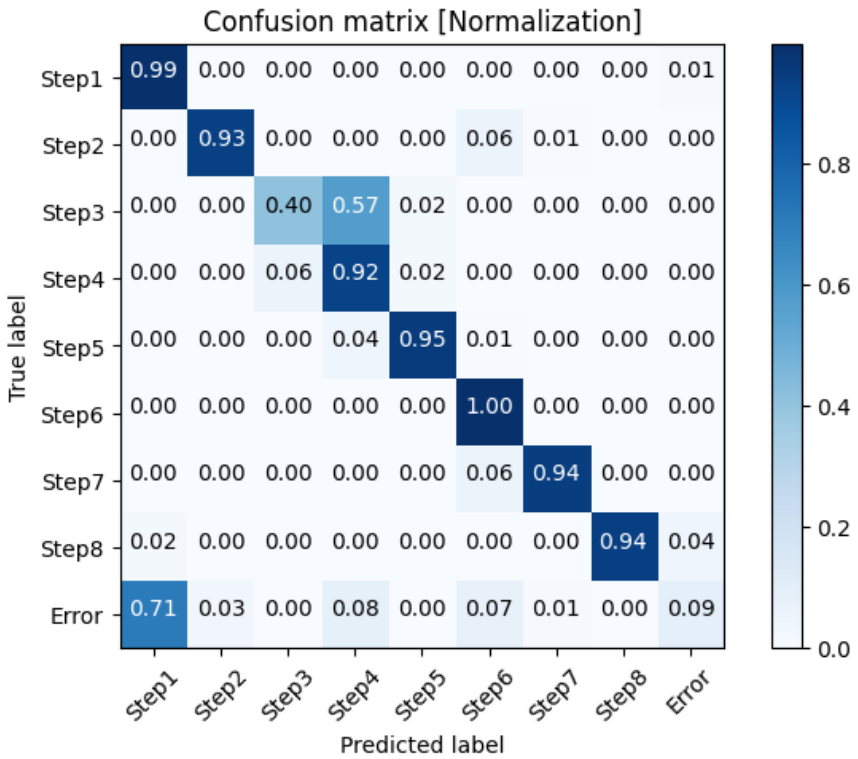


图39：AnomalyTriplet 条件 4：使用混淆矩阵进行评估

最后，通过调整训练轮次，在特征空间维度为 32 的条件下进行实验，以避免过拟合并确保充分学习。训练轮次总数增加至 200，而异常学习的起始训练轮次保持不变，为第 50 轮。

训练条件如下：训练轮次总数为 200，异常学习从第 50 轮开始，$k = 10$ 对于 k近邻，特征空间维度为 32，学习率设为 $lr = 0.0001$。该条件称为*AnomalyTriplet* 条件 *5*。训练结果包括准确率、损失、特征空间中的距离以及使用 t-SNE 进行的二维可视化，分别如图40、图41、图42和图43所示。错误判定距离阈值设为 1,025,000 时，进度估计结果的混淆矩阵如图44所示。测试数据集的准确率达到 96.5%。

从图40和图41可以确认，在此条件下没有发生过拟合。此外，图42和图43表明，特征空间中的距离度量学习得到了正确执行。
在本节评估的五种实验条件中，AnomalyTriplet 条件 5取得了最高的估计准确率。因此，在后续各节中，该条件将用作训练参数的基准设置，并以此条件进行对比评估。

图44中，先前条件下观察到的步骤3与步骤4之间的误分类有所减少。步骤3和步骤4分别对应于将 CPU 安装到个人电脑之前和之后的状态。由于这两个步骤之间的外观特征变化很小，因此容易发生误分类。在 AnomalyTriplet 条件 5下，相邻步骤之间的误分类率降至4.8%。

For 作为参考，在此条件下使用 RTX2080图形处理器进行训练的总耗时为12.1小时 .

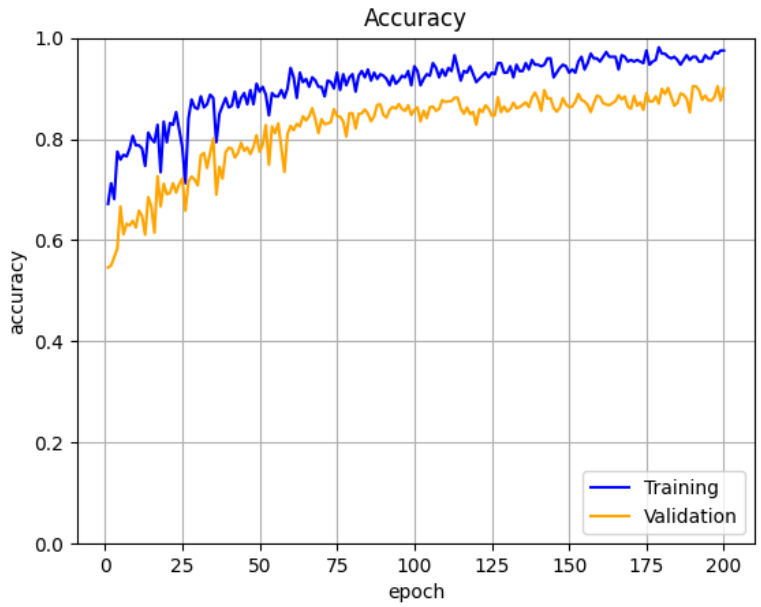


图40：AnomalyTriplet 条件 5：准确率随训练轮次的变化

为验证所提出的异常四元组网络的有效性，我们将其与现有方法 Anomaly Triplet-Net 进行了对比实验。实验使用台式个人电脑组装数据集进行。作为对比基准，我们采用了 Anomaly Triplet-Net 在 AnomalyTriplet 条件 5 下的结果。

我们选择台式个人电脑作为目标装配产品。数据集本身与 Anomaly Triplet-Net 对比实验所用的数据集相同。然而，由于数据加载器设计存在差异，用于训练和测试的图像数量有所调整。

训练数据集保持不变，从步骤1到步骤8每个步骤包含40张图像，共计320张图像。验证数据集同样保持不变，每个步骤包含150张图像，共计1,264张图像。测试数据集的图像数量有所调整；除步骤1至步骤8外，还加入了错误图像，因此每个步骤包含129张图像，共计1,161张图像。

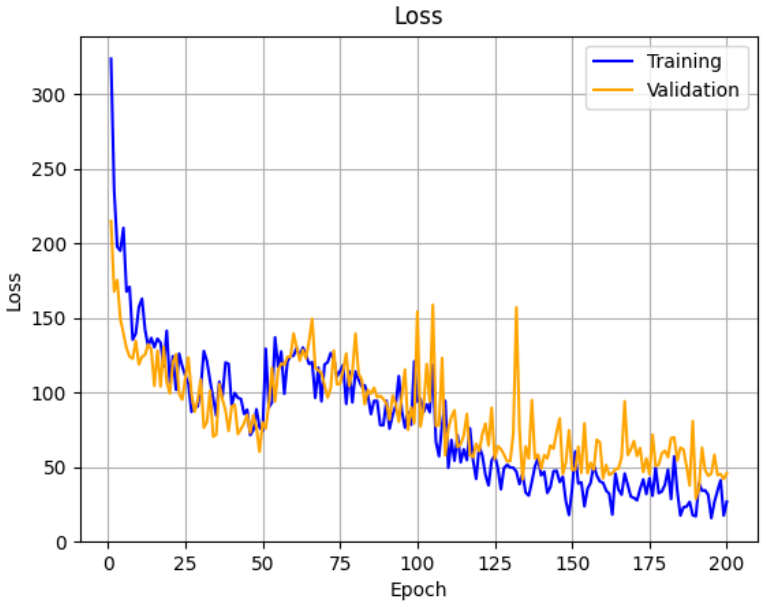


图 41：AnomalyTriplet 条件 5：损失随训练轮次变化的情况

通过改变训练轮次总数、启动异常学习的训练轮次以及特征空间维度，进行了多次实验。训练参数根据经验确定。

训练条件如下：总训练轮次为250，从第100轮开始进行异常学习，$k = 10$ 对于 kNN 算法，特征空间维度设为64，学习率为$lr = 0.0001$。此设置称为*AnomalyQuadruplet* 条件 *1*。

训练结果包括准确率、损失、特征空间中的距离的变化，以及使用 t-SNE 对特征空间进行的二维可视化，分别如图45、图46、图47和图48所示。
距离阈值设为9,000,000时，进度估计结果的混淆矩阵如图49所示。测试数据集的准确率达到97.8%。相邻步骤之间的误分类率为2.9%。

作为参考，在此条件下使用 RTX4090 GPU 的总训练时间为 18.6 小时。
实验结果表明，在台式电脑数据集上，所提出的异常四元组网络与传统的 Anomaly Triplet-Net 相比，准确率提高了 1.3%。此外，相邻装配步骤之间的误分类率降低了 1.9%。

通过比较图 44 和图 49 可以发现，在步骤3与步骤4之间将 CPU 装入个人电脑这一任务中的误分类显著减少。该任务仅涉及细微的外观特征变化，结果证实了所提出的异常四元组网络的有效性；该网络旨在减少此类外观变化较小的装配过程中的误分类。

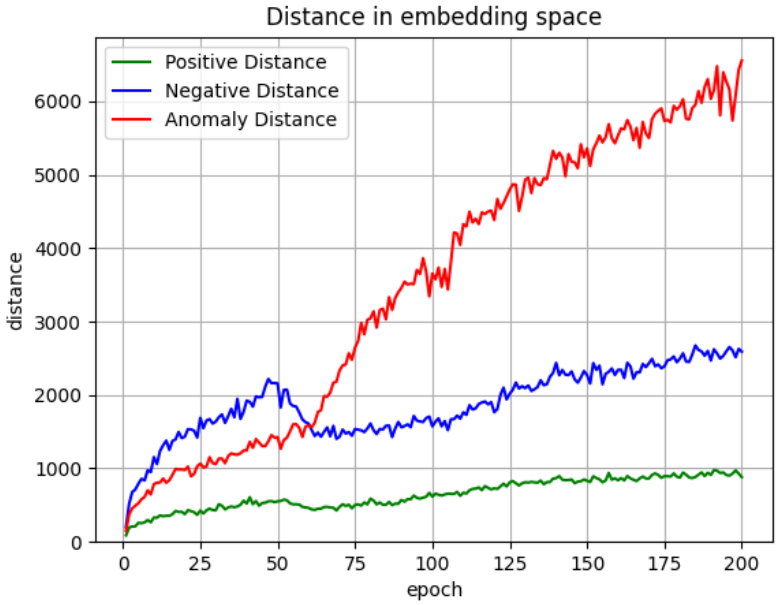


图42：AnomalyTriplet 条件 5：特征空间中样本之间的距离

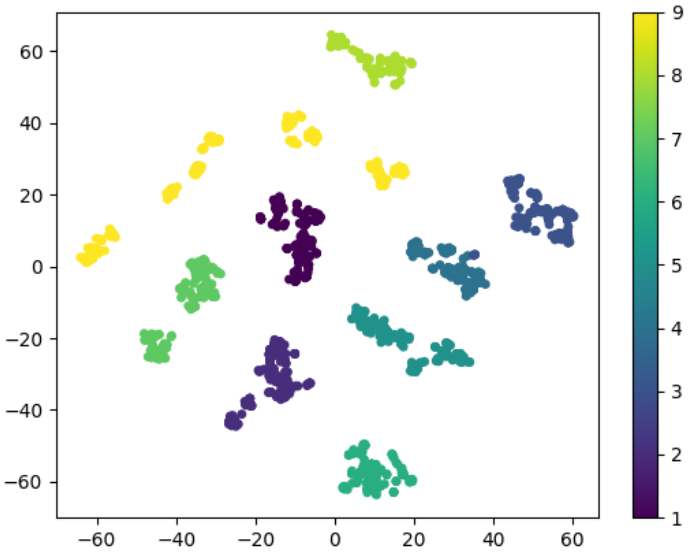


图43 ：AnomalyTriplet 条件 5：使用 t-SNE 对特征空间进行二维可视化

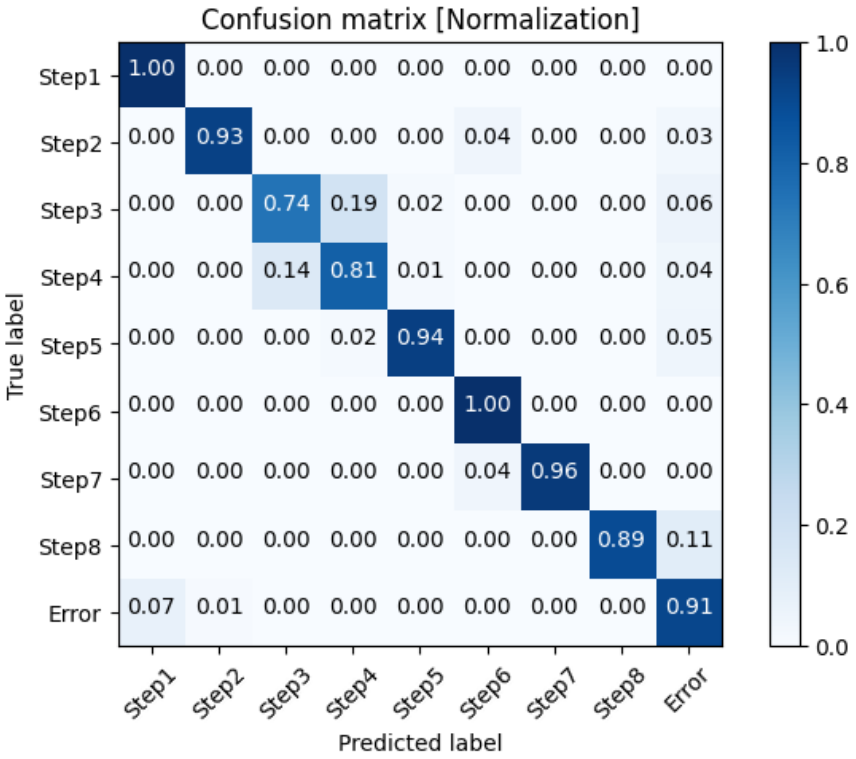


图44：AnomalyTriplet 条件 5：使用混淆矩阵进行评估

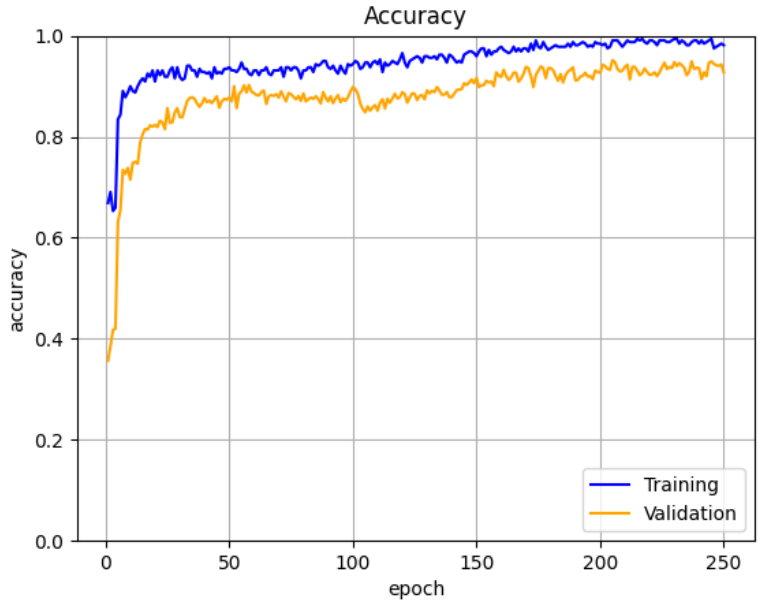


图45：AnomalyQuadruplet 条件 1：准确率随训练轮次变化的情况

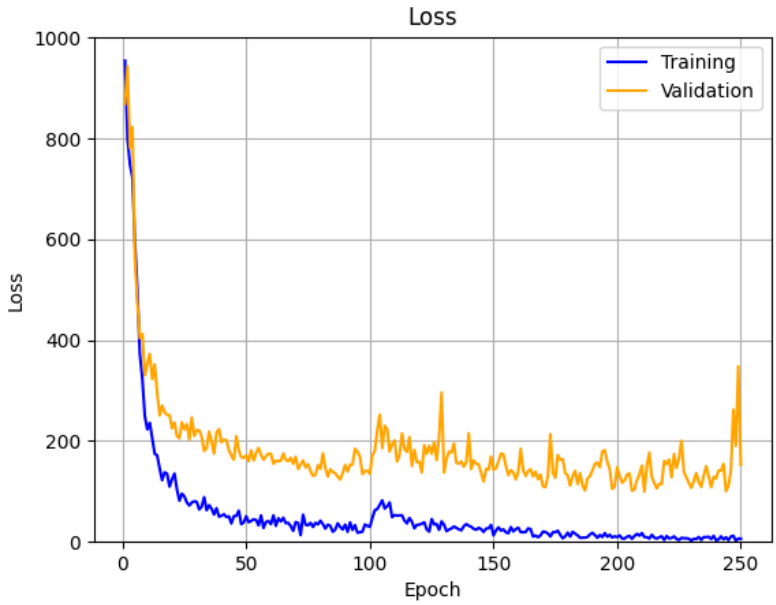


图46：AnomalyQuadruplet 条件 1：损失随训练轮次数量的变化

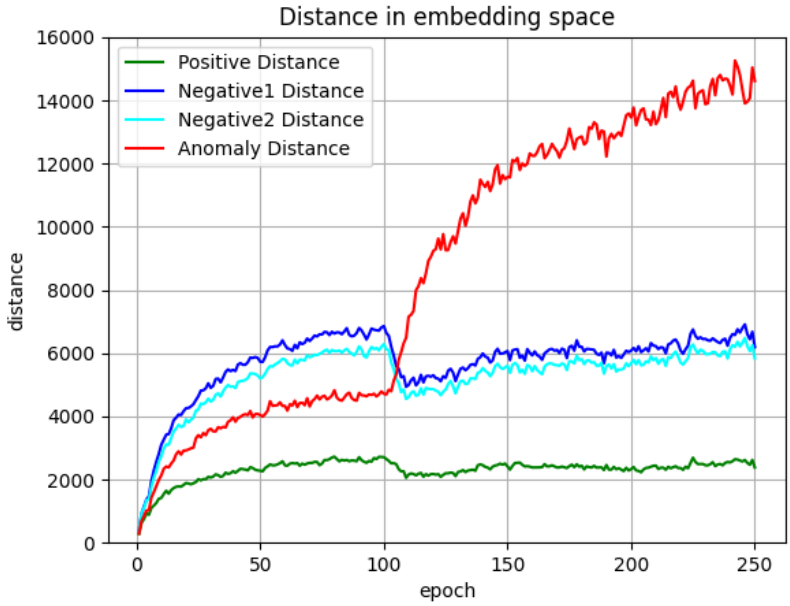


图47：AnomalyQuadruplet 条件 1：特征空间中样本之间的距离

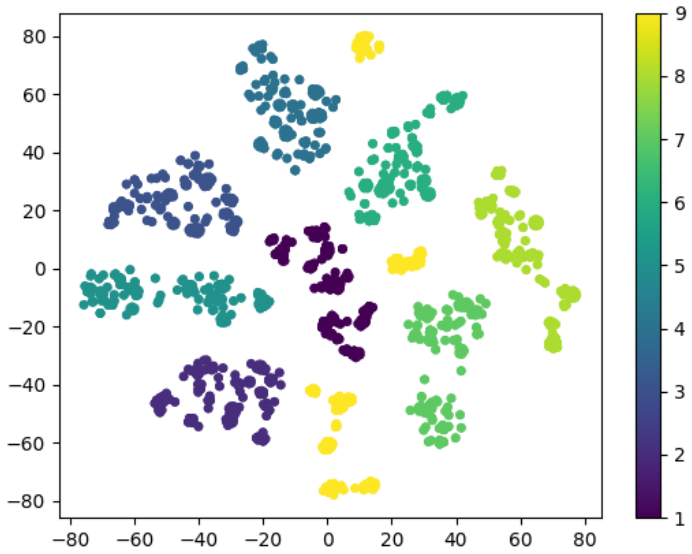


图 48：阿诺　malyQuadruplet 条件 1：使用 t-SNE 对特征空间进行二维可视化

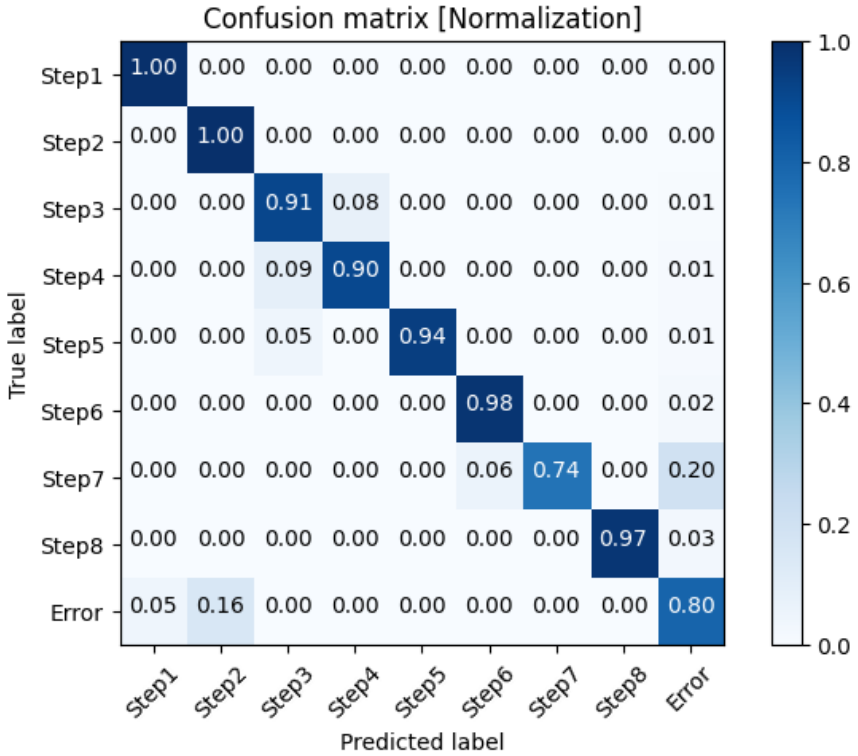


图 49：AnomalyQuadruplet 条件 1：使用混淆矩阵进行评估

4 结论

本研究开发了一种装配进度估计系统，重点关注装配对象外观特征的变化。针对进度估计，我们提出了一种基于深度度量学习的方法。此外，为了减少外观特征变化较小时出现的误分类，我们提出了利用四元组损失的异常四元组网络，并通过实验验证了所提方法的有效性。

具体而言，我们对所提出的异常四元组网络与先前的方法 Anomaly Triplet-Net 进行了比较实验。实验使用台式电脑数据集开展。结果证实，异常四元组网络的进度估计性能优于 Anomaly Triplet-Net。

在使用台式电脑数据集的实验中，进度估计的准确率达到 97.8%。此外，相邻装配步骤之间的误分类率被控制在 2.9%。此外，实验验证了所提方法对于外观特征变化较小、容易发生误分类的相邻装配步骤尤其有效。

未来的工作中，我们计划在存在多个物体的环境中，使用异常四元组网络开展基于视频的进度估计系统实验。此外，在真实环境中，目标跟踪过程中预计会发生切换。因此，我们旨在提出一种利用 ArUco标记的系统，并实现一种可实际应用于工业环境的方法。

参考文献

参考文献

[1] Wang, J., Ma, Y., Zhang, L., Gao, X. R. and Wu, D., “面向智能制造的深度学习：方法与应用”, 《制造系统杂志》, Vol.48, 2018. pp.144–156.[2] Kang, H. S., Lee, J. Y., Choi, S. S., Kim, H., Park, J. H., Son, J. Y., Kim, B. H. and Noh, S. D., “智能制造：既往研究、当前发现与未来方向”, 《精密工程与制造-绿色技术国际期刊》, Vol.3, 2016. pp.111–128.[3] Xia Qingxin, “用于识别和理解工厂作业活动的传感器数据处理方法研究”, 《泛在计算系统研究报告》 (UBI), Vol.31, 2022. pp.1–2.[4] Funk, M., Bachler, L., Kosch, T., Heidenreich, T. and Schmidt, A., “使用增强现实？装配工作场所现场指导的长期分析”, IEEE International Conference on Pervasive Technologies Related to Assistive Environments论文集, 2017. pp.222–229.[5] Takeda, Shingo, Paula Lago, Okita, Tsuyoshi, Inoue, Sozo and Ideno, Yoshinori, “以减少作业为目的，在作业识别中利用动作捕捉进行标记身体对应”, Joho Shori Gakkai Multimedia, Bunsan Kyodo to Mobile Symposium 2019 Ronbunshu, 2019. pp.1280–1286.[6] Hayami Shotsu, Shigeki Yumoto, Kazuma Miura, Alessandro Moro, Sarthak Pathak and Kazunori Umeda, “考虑标注数据的数据增强与实例分割相结合的装配进度估计”, 第17届人工视觉质量控制国际会议 (QCAV2025), 2025.6.[7] Kitsukawa, T., Miura, K., Yumoto, S., Pathak, S., Moro, A. and Umeda, K., “Anomaly Triplet-Net：采用深度度量学习并考虑遮挡的手工作业进度识别模型”, Advanced Robotics, Vol.39, No.2, 2024. pp.89–101.[8] FANUC：机器人款 FANUC SR系列机器人, https://www.fanuc.co.jp/ja/product/ robot/model/f_r_scara.html, 访问日期：2025-12-29.[9] FANUC：大型机器人 FANUC R-2000iD机器人, https://www.fanuc.co.jp/ja/product/new_product/2019/201904_r2000id210fh.html, 访问日期：2025-12-29.[10] Morioka, Masahiro, “人机协作型单元生产装配系统的开发”, Nihon Robotto Gakkai-shi, Vol.27, No.10, 2009. pp.1088–1091.[11] 日立产业控制解决方案, “基于人工智能的人体姿势动作识别解决方案”, https://info. hitachi-ics.co.jp/product/activity_evaluation/, 访问日期：2025-12-29.[12] Oshima, Hirotomo, Shirakawa, Yuta, Yoshii, Takanori, Kado, Takehiro, Maekawa, Takuya and Namioka, Yasuo, “基于深度学习的产品状态估计实现作业进度自动提取”, 2022年度生产系统部门讲演会论文集, 日本机械工程师学会, 2022. p.206.[13] Ren, S., He, K., Girshick, R. and Sun, J., “Faster R-CNN：基于区域提议网络实现实时目标检测”, 神经信息处理系统 (NIPS), 2015.[14] Bolya, D., Zhou, C., Xiao, F. and Lee, Y. J., “YOLACT：实时实例分割”, IEEE/CVF计算机视觉国际会议论文集, 2019. pp.9157–9166.[15] Zhong, Z., Zheng, L., Kang, G., Li, S. and Yang, Y., “随机擦除数据增强”, AAAI人工智能会议论文集, Vol.34, No.7, 2020. pp.13001–13008.[16] 日立高新技术：高速测量SEM设备CV6300系列, https://www.hitachi-hightech.com/jp/ja/products/semiconductor-manufacturing/cd-sem/metrology-solution/ semi-cv6300.html, 访问日期：2025-12-29.[17] IBM：Wiwynn公司为实现全球增长而强化生产体系, https://www.ibm.com/jp-ja/ case-studies/wiwynn-corporation, 访问日期：2025-12-29.[18] TECH+：PC服务器生产现场提升品质的关键是什么？——富士通的举措, https: // news.mynavi.jp/techplus/article/20080930-fit/2, 访问日期：2025-12-29.[19] Ultralytics：YOLOv8, https://docs.ultralytics.com/ja/models/yolov8/, 访问日期：2025-12-29.[20] Hoffer, E. and Ailon, N., “使用三元组网络进行深度度量学习”, International Workshop on Similarity-Based Pattern Recognition (SIMBAD), 2015. pp.84–92.[21] Chen, W., Chen, X., Zhang, J. and Huang, K., “超越三元组损失：用于行人重识别的深度四元组网络”, IEEE计算机视觉与模式识别会议, 2017. pp.1320–1329.[22] Cover, T. and Hart, P., “最近邻模式分类”, 《IEEE信息论汇刊》, Vol.13, 1967. pp.21–27.

[23] Bellman, R. E.，《自适应控制过程：导论》，普林斯顿大学出版社，1961 年。[24]erikbern：ann-presentation，https://github.com/erikbern/ann-presentation/tree/master，访问日期：2025-12-29。[25] IO DATA株式会社：网络摄像机 Qwatch TS-WRLP，https://www.iodata.jp/product/lancam/lancam/ts-wrlp/，访问日期：2025-12-29。[26] Maaten, L. van der 和 Hinton, G.，《使用 t-SNE 可视化数据》，《机器学习研究杂志》，第 9 卷，2008 年，第 2579–2605 页。