

利用子集群 ADACOS 进行异常声音检测 域转移条件

技术报告

凯文·威尔金霍夫

弗劳恩霍夫通信、信息处理和人体工程学研究所 FKIE
Fraunhoferstraße 20, 53343 Wachtberg, 德国
kevin.wilkinghoff@fkie.fraunhofer.de

抽象的

基于子集群 AdaCos 的异常声音检测系统在 DCASE 2020 数据集上产生了最先进的异常声音检测性能。与前一年相比，属于 DCASE 挑战 2021 的任务 2 “在域移位条件下进行机器状态监测的无监督异常声音检测”的数据集不仅包含具有 1000 每台机器的正常训练样本，以及具有不同声学条件的所谓目标域，其中只有 3 正常训练样本可用。为了解决这个额外的问题，提出了一种基于子集群 AdaCos 的新型异常声音检测系统，用于 DCASE 挑战 2021。该系统经过训练以提取嵌入，其分布以不同方式为源域和目标域估计，并利用它们的负对数似然作为异常分数。在实验评估中，表明所提出的系统在开发集的源域和目标域上都显著优于基线系统。

索引词— 异常声音检测、机器监听、表征学习、角余量损失、域偏移

一、介绍

半监督异常声音检测的目标是确定给定的音频样本是否与训练数据相似，即 *普通的* 或与训练数据有很大不同，因此是 *异常的*。基本上，可以区分异常声音检测的两种主要策略：第一种方法是基于训练自动编码器将正常数据编码到低维空间，然后再次重建[1, 2]。基本假设是训练后可以很好地重建正常数据，而异常数据不会导致更高的重建误差。因此，重建误差可以用作异常分数。第二种方法是训练神经网络区分类别，例如机器类型，并利用训练过的神经网络提取数据的表示，即所谓的嵌入，作为特征 [3, 4, 5, 6, 7]。这里，假设是区分类别所需的信息因此包含在嵌入中也足以区分正常数据和异常数据。在这种情况下，诸如 ArcFace [8] 或 AdaCos [9] 之类的 Angular margin loss 已被证明优于标准的 softmax 损失，这些损失确保了类别之间的边界。据我们所知，属于 DCASE 挑战 2020 [10] 任务 2 的异常声音检测数据集上性能最好的系统使用

AdaCos 的扩展，称为子集群 AdaCos [11]。这种损失为每个类学习了多个平均值，以估计比标准 AdaCos 限制更少的嵌入分布，并利用高斯混合模型 (GMM) 来估计正态数据的这些分布，而不是通过使用将嵌入与学习的平均值进行比较余弦相似度。这种卓越的性能是这项工作完全专注于基于子集群 AdaCos 损失的系统的原因。

本文中介绍的系统是为 DCASE 挑战 2021 [12] 的任务 2 “在域移位条件下进行机器状态监测的无监督异常声音检测”而设计并提交的。此任务的数据集由长度为 10 秒和采样率 16 kHz 属于来自 ToyADMOS2 [13] 的机器类型 “ToyCar” 和 “ToyTrain” 以及来自 MIMII DUE [14] 的机器类型 “风扇”、“齿轮箱”、“泵”、“滑轨”和“阀门”。挑战的组织者还提供了两个基线系统：一个自动编码器，与前一版任务的基线系统相同，以及一个经过区分训练的基于 MobileNetV2 的基线，用于预测给定音频样本所属的部分。在检测异常数据时，两个基线系统具有相似的整体性能。

与 DCASE 挑战 2020 相比，今年的任务有几个不同之处：首先，数据集被拆分为源域，其中大约 1000 为每个训练样本提供正常的训练样本 6 每个机器类型的部分和具有不同声学条件的相同部分的所谓目标域，而不是源域 3 正常训练样本可用。对于这两个域，提供相同数量的测试样本，大约 100 正常样本和 100 异常样本。此外，数据集分为由一半部分组成的开发集和由另一半部分组成的评估集。数据集之间的另一个区别是，部分不直接对应机器类型的特定产品，但相同的产品可以出现在不同的部分中，或者不同的产品可以出现在相同的部分中。这两个变化使任务比以前更具挑战性。最后但并非最不重要的是，DCASE 2020 数据集由稍微不同的机器类型组成，即来自 ToyADMOS 数据集 [15] 的 “ToyCar” 和 “ToyConveyor” 以及机器类型 “风扇”、“泵”、“滑轨”和“阀门”来自 MIMII 数据集 [16]。

这项工作的目标是研究如何将子集群 AdaCos 损失用于 DCASE 2021 异常声音检测数据集及其新颖的挑战。为此，一个基于系统的

表 1: 用于所有实验的修改后的 ResNet 架构。

图层名称	结构体	输出尺寸
输入	—	313×128
二维卷积	7×7 、步幅=2	$157 \times 64 \times 16$
残差块	3×3 (3×3) $\times 2$ 、步幅=1	$78 \times 31 \times 16$
残差块	3×3 (3×3) $\times 2$ 、步幅=1	$39 \times 16 \times 32$
残差块	3×3 (3×3) $\times 2$ 、步幅=1	$20 \times 8 \times 64$
残差块	3×3 (3×3) $\times 2$ 、步幅=1	$10 \times 4 \times 128$
最大池化	10×1 、步幅=1	4×128
压平	—	512
密集 (表示)	线性	128
子集群 AdaCos	—	42
子集群 AdaCos	—	199

在子集群上呈现 AdaCos 损失。作为第二个贡献，提出了计算源域和目标域的异常分数的不同方法。此外，还展示了如何仅根据来自正常数据的这些分数来确定样本是正常还是异常。在实验评估中，表明所提出的系统在开发集的源域和目标域上都显著优于基线系统。

2. 建议的方法

2.1. 数据预处理

为了计算神经网络的输入特征，log-Mel 谱图 128 Mel-bins，窗口大小为 1024 和一个啤酒花大小 512 以采样率从所有原始波形中抽取 16 kHz 导致大小特征 313×128 。然后通过减去时间平均值并除以从所有训练文件估计的时间标准偏差来标准化这些特征。

2.2. 神经网络架构

整个工作中使用的网络架构与 [11] 中使用的相同，可以在表中找到。1. 它由几个残差块 [17] 组成，其输出随着时间的推移通过最大池化、展平和线性密集层进一步处理以获得大小的嵌入 128。在每个残差块中，应用批量归一化 [18] 和 LeakyRelu [19] $\alpha = 0.1$ 用作非线性传递函数。

为了训练神经网络，使用 Adam [20] 最小化两个具有相同权重的子集群 AdaCos 损失 [11]。一种用于在段和机器类型之间进行联合分类，另一种用于在文件名中给出的不同属性信息之间进行分类。训练时，训练集和附加训练集中包含的所有正常数据都被使用了，总共 42 部分和 199 不同的属性信息。此外，在训练期间使用混合 [21] 以避免模型过度拟合训练数据。该网络在 Tensorflow [22] 中实现并针对 400 批次大小为 64。

2.3. 计算异常分数

在整个工作中，所有异常分数都是通过嵌入上训练高斯混合模型 (GMM) 并利用负加权对数似然作为分数来计算的。在 [11] 中，已经表明使用 GMM 来估计嵌入的底层分布优于其他后端，例如使用与类均值的余弦相似度。除非另有说明，所有 GMM 都是使用 scikit-learn [23] 实现的，使用子集群 AdaCos 损失的学习平均值进行初始化，并通过添加正则化协方差矩阵 10^{-3} 到对角线。为了计算异常分数，使用了源域和目标域的两种不同策略。

对于源域，对源域中属于一个 section 的每个法线数据训练一个 GMM，对于源域中属于不同属性信息的每个法线数据训练另一个 GMM。让 $X \in \text{电阻}_{128}$ 表示嵌入， $\text{秒}(X) \in S$ 表示它的部分和一种($\text{秒}(X)$) $\subset A$ 表示本节中出现的所有属性信息。然后，异常分数 $Z_{\text{来源}}(X)$ 为了 X 是由

$$Z_{\text{来源}}(X) := - \underset{\text{克}}{\text{最大日志 磷}}(X/s(X), k) - \underset{\text{克} \text{ 一种} \in \text{一种}(\text{秒}(X))}{\text{最大限最大日志 磷}}(X/a, k) \quad (1)$$

在哪里 $\text{磷}(. / s, k), P(. / - , k)$ 表示分量的加权似然 克 为部分训练的 GMM 秒 $\in S$ 和目标信息 一种 $\in A$ ，分别。

对于目标域，使用在属于单个部分的目标域的正常数据上训练的相同 GMM。此外，在三个目标样本上训练另一个具有三个分量的 GMM，因此还利用到最近的正常目标样本的余弦距离。使用与之前相同的符号，异常分数 $Z_{\text{目标}}(X)$ 用于嵌入 X 是 (谁) 给的

$$Z_{\text{目标}}(X) := - \underset{\text{克}}{\text{最大日志 磷}}(X/s(X), k) - \underset{\text{克}=1,2,3}{\text{最大限最大日志 磷}}(X/X_{\text{目标}}(\text{秒}(X)), k) \quad (2)$$

在哪里 $X_{\text{目标}}(\text{秒}(X)) \subset \text{电阻}_{128}$ 表示属于部分的目标域的正常训练样本 X 。

在 [11] 中，已经表明，从输入特征导出的简单表示导致 DCASE 2020 数据集上机器类型“阀门”的出色性能。这就是为什么基于 log-Mel 谱图的时间最大值的附加项的原因，表示为 $\text{吨最大限最大}(X) \in \text{电阻}_{128}$ 用于嵌入 X 在计算机器类型“阀门”的源域的异常分数时引入。为此，训练具有单个高斯分量的 GMM 并改变

异常分数 $Z_{\text{来源}}(X)$ 用于嵌入 X 属于机器类型“阀门”由下式给出

$$Z_{\text{来源}}(X) := Z_{\text{来源}}(X) - \underset{\text{一种} \in \text{一种}(\text{秒}(X))}{\text{最大限最大日志 磷}}(\text{吨最大限最大}(X)/a) \quad (3)$$

在哪里 $\text{磷}(\text{吨最大限最大}(. / \text{一种}))$ 表示在属于目标信息的 log-Mel 谱图的时间最大值上训练的单个高斯的加权似然 一种 $\in A$ 。

2.4. 集成策略

如 [11] 中所做的那样，所提出的用于提取嵌入的神经网络是用不同数量的子集群进行训练的

从 2₀ 到 2₄. 两个子集群 AdaCos 损失使用相同的值。因此，有 5 每个网络的不同训练版本以提取嵌入。此外，每次之后 100 在训练的 epochs 中，提取嵌入并训练 GMM 以计算异常检测分数。然后，将所有这些分数相加，得到 4 每个网络的子系统具有指定数量的子集群，因此包含总共 $4 \times 5 = 20$ 个模型。

除此之外，通过仅使用在节和机器类型之间进行分类的单个子集群 AdaCos 损失来重复所描述的集成过程，从而去除第二个子集群 AdaCos 损失。这导致某些机器类型的性能略好，而其他机器类型的性能略差。为了获得每种机器类型的异常分数，使用单个系统可以为给定的机器类型带来更好的性能。更具体地说，对于机器类型“ToyCar”、“ToyTrain”、“pump”和“slide rail”，使用对两种损失训练的模型获得的异常分数以及“fan”、“gearbox”和“valve”使用仅在单个损失上训练的模型获得的异常分数。

2.5. 设置决策阈值

接下来，描述如何获得用于仅基于异常分数来判定给定测试样本是正常还是异常的判定阈值。为此，该 90 计算属于给定部分和给定域的所有正常训练样本的异常分数的第 th 个百分位数。然后，所有高于该阈值的属于同一部分和域的测试样本的异常分数都标记为异常的。对于源域， $Z_{\text{来源}}(X)$ 如方程式中所定义。(1) 被使用，但对于目标域，只有第一项 $Z_{\text{目标}}(X)$ 用来。原因是属于训练数据的第二项的似然高得不合适，因为相应 GMM 的三个均值被初始化为三个训练样本。因此，当还使用方程的第二项时。(2) 决策阈值也会被估计得太高，因此属于目标域的所有测试数据样本都将被视为异常。

3. 结果

与两个基线系统相比，使用所提出的系统获得的结果可以在表中找到。2. 可以看出，所提出的系统在源域和目标域上明显优于两个基线系统，两者的整体性能大致相同。然而，AUC 方面的改进远大于 pAUC。对于几乎所有数据集拆分，所提出的系统具有比两个基线系统更高的 AUC。但是对于某些数据集拆分，基于 MobileNetV2 的基线系统比提议的系统具有更高的 pAUC。对于机器类型“齿轮箱”，属于所提出系统的所有 pAUC 的调和平均值甚至比基于 MobileNetV2 的基线系统的调和平均值略差。

4. 提交

总共有四个系统获得的结果已提交给挑战。更具体地说，已经提交了使用先前描述的建议系统及其三个轻微变化获得的结果。第一个变体仅包含使用两个子集群 AdaCos 训练的集成子系统

损失。第二个变体是使用两个子系统的得分平均值，而不是对每种机器类型使用性能最佳的子系统。第三个变体是所提出的系统，也没有使用机器类型“阀门”的简单时间最大表示，即不使用等式中给出的改变的异常分数。(3) 但方程中给出的那个。(1).

5. 结论

在这项工作中，提出了一种基于域转移条件下子集群 AdaCos 损失函数的异常声音检测系统。所提出的系统由多个经过判别训练的神经网络组成，用于从对数梅尔谱图提取嵌入，并利用多个 GMM 来估计正态嵌入的分布。然后使用这些估计的分布来计算测试数据的对数似然，并将它们组合成实际的异常分数。为了确定给定的测试样本是异常还是正常，每个部分的单独决策阈值是通过取 90 来自相应正常训练样本的对数似然的第 th 个百分位数。在对 DCASE 挑战 2021 的任务 2 的数据集进行的实验评估中，表明所提出的系统在开发集的源域和目标域的 AUC 和 pAUC 方面显著优于挑战的基线系统。

6. 参考资料

- [1] S. Kapka, “用于无监督异常检测的 ID 条件自动编码器”，在 *声学场景和事件检测与分类研讨会 (DCASE)*，2020 年，第 71-75 页。
- [2] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda 和 K. Takeda, “采用半监督学习和数据增强的基于 Conformer 的声音事件检测”，载于 *声学场景和事件检测与分类研讨会 (DCASE)*，2020 年，第 100-104 页。
- [3] T. Inoue, P. Vinayavekhin, S. Morikuni, S. Wang, T. Hoang Trong, D. Wood, M. Tatsubori 和 R. Tachibana, “使用分类置信度检测机器状态监测的异常声音”，在 *声学场景和事件检测与分类研讨会 (DCASE)*，2020 年，第 66-70 页。
- [4] R. Giri, SV Tenneti, F. Cheng, K. Helwani, U. Isik 和 A. Krishnaswamy, “用于检测异常声音的自监督分类”，在 *声学场景和事件检测与分类研讨会 (DCASE)*，2020 年，第 46-50 页。
- [5] Q. Zhou, “用于 DCASE 2020 任务 2 的基于 ArcFace 的声音移动网络”，DCASE2020 挑战赛，技术。众议院，2020 年。
- [6] JA Lopez, H. Lu, P. Lopez-Meyer, L. Nachman, G. Stemmer 和 J. Huang, “异常检测的说话人识别方法”，在 *声学场景和事件检测与分类研讨会 (DCASE)*，2020 年，第 96-99 页。
- [7] K. Wilkinghoff, “使用看、听和学习嵌入来检测机器状态监测中的异常声音”，在 *声学场景和事件检测与分类研讨会 (DCASE)*，2020，第 215-219 页。

- [8] J. Deng、J. Guo、N. Xue 和 S. Zafeiriou, “ArcFace: 用于深度人脸识别的附加角边缘损失”, 在 *计算机视觉与模式识别 (CVPR)* 会议. IEEE, 2019 年, 第 4690-4699 页。
- [9] X. Zhang、R. Zhao、Y. Qiao、X. Wang 和 H. Li, “AdaCos: 自适应缩放余弦对数以有效学习深度人脸表征”, 载于 *计算机视觉与模式识别 (CVPR)* 会议. IEEE, 2019 年, 第 10 823-10 832 页。
- [10] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido、R. Tanabe、H. Purohit、K. Suefusa、T. Endo、M. Yasuda 和 N. Harada, “DCASE2020 挑战任务 2 的描述和讨论: 用于机器状态监测的无监督异常声音检测”, 载于 *声学场景和事件检测与分类研讨会 (DCASE)*, 2020 年, 第 81-85 页。
- [11] K. Wilkinghoff, “子集群 AdaCos: 异常声音检测的学习表示”, 在 *国际神经网络联合会议 (IJCNN)*, 2021。
- [12] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi、R. Tanabe、H. Purohit 和 T. Endo, “DCASE 2021 挑战任务 2 的描述和讨论: 域转移条件下机器状态监测的无监督异常声音检测”, 在 *arXiv 电子版中: 2106.04492, 1-5*, 2021。
- [13] N. Harada、D. Niizumi、D. Takeuchi、Y. Ohishi、M. Yasuda 和 S. Saito, “ToyADMOS2: 用于域转移条件下异常声音检测的微型机器操作声音的另一个数据集”, *arXiv 预印本 arXiv:2106.02369*, 2021。
- [14] R. Tanabe、H. Purohit、K. Dohi、T. Endo、Y. Nikaido、T. Nakamura 和 Y. Kawaguchi, “MIMII DUE: 用于工业机器故障调查和检查的声音数据集操作和环境条件的变化”, 在 *arXiv 电子版: 2006.05822, 1-4*, 2021。
- [15] Y. Koizumi、S. Saito、H. Uematsu、N. Harada 和 K. Imoto, “ToyADMOS: 用于异常声音检测的微型机器操作声音数据集”, 在 *信号处理在音频和声学中的应用研讨会 (WAS-PAA)*. IEEE, 2019 年, 第 313-317 页。
- [16] H. Purohit、R. Tanabe、T. Ichige、T. Endo、Y. Nikaido、K. Suefusa 和 Y. Kawaguchi, “MIMII 数据集: 用于故障工业机器调查和检查的声音数据集”, 载于 *声学场景和事件检测与分类研讨会 (DCASE)*. 纽约大学, 2019 年, 第 209-213 页。
- [17] K. He、X. Zhang、S. Ren 和 J. Sun, “图像识别的深度残差学习”, 在 *计算机视觉与模式识别 (CVPR)* 会议. IEEE, 2016 年, 第 770-778 页。
- [18] S. Ioffe 和 C. Szegedy, “批量归一化: 通过减少内部协变量偏移来加速深度网络训练”, 在 *第 32 届机器学习国际会议 (ICML)*, 卷。37, 2015 年, 第 448-456 页。
- [19] AL Maas、AY Hannun 和 AY Ng, “整流器非线性改善神经网络声学模型”, 在 *第 30 届机器学习国际会议 (ICML)*, 2013。
- [20] DP Kingma 和 J. Ba, “Adam: 一种随机优化方法”, 在 *第三届国际学习代表会议 (ICLR)*, Y. Bengio 和 Y. LeCun, 编辑, 2015 年。
- [21] H. Zhang、M. Cisse、YN Dauphin 和 D. Lopez-Paz, “混合: 超越经验风险最小化”, 在 *国际学习代表会议 (ICLR)*, 2018。
- [22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin、S. Ghemawat、G. Irving、M. Isard、等。 , “Tensorflow: 一种用于大规模机器学习的系统”, 在 *第 12 届 USENIX 操作系统设计与实现 (OSDI)* 研讨会, 2016 年, 第 265-283 页。
- [23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher、M. Perrot 和 E. Duchesnay, “Scikit-learn: Python 中的机器学习”, *机器学习研究杂志*, 卷。12, 第 2825-2830 页, 2011 年。

表 2：使用基线系统和提议系统获得的每种机器类型的 AUC 和 pAUC。每行中最高的 AUC 和 pAUC 都标有下划线。

数据集拆分			基线				提议的系统	
机器类型部分		领域	自编码器 AUC	吸光度 数	移动网络V2 AUC	pAUC	AUC	pAUC
玩具车	0	来源	67.63%	51.87%	66.56%	66.47%	79.15%	<u>61.63%</u>
玩具车	1	来源	61.97%	51.82%	71.58%	66.44%	91.44%	<u>71.26%</u>
玩具车	2	来源	74.36%	55.56%	40.37%	47.48%	96.89%	<u>85.05%</u>
玩具车	0	目标	54.50%	50.52%	61.32%	52.61%	92.14%	<u>80.89%</u>
玩具车	1	目标	64.12%	51.14%	72.48%	63.99%	77.10%	<u>63.26%</u>
玩具车	2	目标	56.57%	52.61%	45.17%	48.85%	62.63%	<u>57.95%</u>
玩具车		调和平均数	62.49%	52.36%	56.04%	56.37%	81.43%	<u>68.62%</u>
玩具火车	0	来源	72.67%	69.38%	69.84%	54.43%	96.51%	<u>91.11%</u>
玩具火车	1	来源	72.65%	62.52%	64.79%	54.09%	89.02%	<u>77.79%</u>
玩具火车	2	来源	69.91%	47.48%	69.28%	47.66%	87.91%	<u>47.37%</u>
玩具火车	0	目标	56.07%	50.62%	46.28%	51.27%	72.29%	<u>48.37%</u>
玩具火车	1	目标	51.13%	48.60%	53.38%	49.60%	50.86%	<u>49.89%</u>
玩具火车	2	目标	55.57%	50.79%	51.42%	53.40%	94.67%	<u>79.84%</u>
玩具火车		调和平均数	61.71%	53.81%	57.46%	51.61%	77.89%	<u>61.11%</u>
扇子	0	来源	66.69%	57.08%	43.62%	50.45%	73.41%	<u>62.26%</u>
扇子	1	来源	67.43%	50.72%	78.33%	78.37%	89.02%	<u>84.53%</u>
扇子	2	来源	64.21%	53.12%	74.21%	76.80%	84.01%	<u>76.32%</u>
扇子	0	目标	69.70%	55.13%	53.34%	56.01%	55.37%	<u>48.47%</u>
扇子	1	目标	49.99%	48.49%	78.12%	66.41%	87.94%	<u>75.53%</u>
扇子	2	目标	66.19%	56.93%	60.35%	60.97%	71.31%	<u>70.68%</u>
扇子		调和平均数	63.24%	53.38%	61.56%	63.02%	74.80%	<u>67.41%</u>
变速箱	0	来源	56.03%	51.59%	81.35%	70.46%	85.25%	<u>73.93%</u>
变速箱	1	来源	72.77%	52.30%	60.74%	53.88%	85.91%	<u>54.05%</u>
变速箱	2	来源	58.96%	51.82%	71.58%	62.23%	59.31%	<u>48.41%</u>
变速箱	0	目标	74.29%	55.67%	75.02%	64.77%	87.62%	<u>71.61%</u>
变速箱	1	目标	72.12%	51.78%	56.27%	53.30%	86.87%	<u>56.85%</u>
变速箱	2	目标	<u>66.41%</u>	53.66%	64.45%	55.58%	<u>65.41%</u>	<u>52.96%</u>
变速箱		调和平均数	65.97%	52.76%	66.70%	59.16%	<u>76.49%</u>	<u>58.19%</u>
泵	0	来源	67.48%	61.83%	64.09%	62.40%	77.15%	<u>63.53%</u>
泵	1	来源	82.38%	58.29%	86.27%	66.66%	98.14%	<u>90.47%</u>
泵	2	来源	63.93%	55.44%	53.70%	50.98%	79.15%	<u>65.68%</u>
泵	0	目标	58.01%	51.53%	59.09%	53.96%	58.54%	<u>51.21%</u>
泵	1	目标	47.35%	49.65%	71.86%	62.69%	87.89%	<u>61.37%</u>
泵	2	目标	62.78%	51.67%	50.16%	51.69%	73.57%	<u>57.74%</u>
泵		调和平均数	61.92%	54.41%	61.89%	57.37%	77.08%	<u>63.05%</u>
滑轨	0	来源	74.09%	52.45%	61.51%	53.97%	95.56%	<u>82.11%</u>
滑轨	1	来源	82.16%	60.29%	79.97%	55.62%	94.28%	<u>71.58%</u>
滑轨	2	来源	78.34%	65.16%	79.86%	71.88%	84.05%	<u>76.59%</u>
滑轨	0	目标	67.22%	57.32%	51.96%	51.96%	81.57%	<u>59.47%</u>
滑轨	1	目标	<u>66.94%</u>	53.08%	46.83%	52.02%	65.97%	<u>49.84%</u>
滑轨	2	目标	46.20%	50.10%	55.61%	55.71%	73.40%	<u>58.00%</u>
滑轨		调和平均数	66.74%	55.94%	59.26%	56.00%	81.07%	<u>64.29%</u>
阀门	0	来源	50.34%	50.82%	58.34%	54.97%	79.54%	<u>62.54%</u>
阀门	1	来源	53.52%	49.33%	53.57%	50.09%	91.02%	<u>67.53%</u>
阀门	2	来源	59.91%	51.96%	56.13%	51.69%	98.09%	<u>92.00%</u>
阀门	0	目标	47.12%	48.68%	52.19%	51.54%	68.91%	<u>63.79%</u>
阀门	1	目标	56.39%	53.88%	68.59%	57.83%	80.02%	<u>63.26%</u>
阀门	2	目标	55.16%	48.97%	53.58%	50.86%	78.30%	<u>57.05%</u>
阀门		调和平均数	53.41%	50.54%	56.51%	52.64%	81.60%	<u>66.16%</u>
全部		调和平均数	61.93%	53.27%	59.72%	56.37%	78.54%	<u>63.93%</u>