

# AirRoom: 物体在房间重识别中的重要性

Runmao Yao Yi Du Zhuoqun Chen Haoze Zheng Chen Wang  
Spatial AI & Robotics (SAIR) Lab, University at Buffalo  
{yaorunmao, zhzh19231211}@gmail.com, {yid, chenw}@sairlab.org, zhcz057@ucsd.edu

## Abstract

房间重识别 (ReID) 是一项具有挑战性但至关重要的任务, 广泛应用于增强现实 (AR) 和家庭护理机器人等领域。现有的视觉位置识别 (VPR) 方法通常依赖于全局描述符或聚合局部特征, 但在拥挤的室内环境中, 尤其是那些充满人造物体的环境中, 常常难以有效工作。这些方法往往忽视了物体相关信息的关键作用。为了解决这一问题, 我们提出了 AirRoom, 这是一种物体感知管道, 整合了多层次的物体相关信息——从全局上下文到物体补丁、物体分割和关键点——并采用粗到细的检索方法。在四个新构建的数据集 (MPReID、HMReID、GibsonReID 和 ReplicaReID) 上的大量实验表明, AirRoom 在几乎所有评估指标上都超越了最先进的 (SOTA) 模型, 提升幅度从 6% 到 80% 不等。此外, AirRoom 展现出显著的灵活性, 允许管道中的各种模块被不同的替代方案替换, 而不会影响整体性能。它还在不同视角变化下表现出强大且稳定的性能。项目网站: <https://sairlab.org/airroom/>。

## 1. 引言

随着空间计算的迅速发展, 房间重识别 (Room ReID) 已成为一个关键研究领域, 推动了诸如增强现实 (AR) [37] 和居家护理机器人 [33] 等应用的进步。它在多种场景中提升用户体验方面发挥着至关重要的作用。例如, 在 Apple Vision Pro 等设备上, 精确的房间重识别能够实现虚拟与现实元素之间的平滑过渡。同样, 在 AR 引导的博物馆导览中, 准确识别用户在特定房间内的位置对于提供基于位置的内容至关重要。

与室外环境中已经成熟且表现可靠的视觉位置识别 (VPR) 方法不同 [2, 13, 16], 室内房间重识别仍然是一个具有挑战性的问题。造成这一困难的主要原因在于室内场景的杂乱特性, 这些场景通常密集布置着大量人造物体 [45]。这些密集分布的物体对现有方法构成了重大挑战, 而这些方法最初是为城市风格和结构清晰的环境设计的 [23]。因此, 这些方法难以充分捕捉室内环境中复杂的细节和多样的空间布局。例如, 像 DINO [9] 和 DINOv2 [25] 等基础模型能够生成捕捉整体场景特征的全局描述符。然而, 在语义相似的环境中, 例如布局或装饰风格相近的相邻房间, 这些描述符可能难



Figure 1. AirRoom 利用多层次、面向对象的特征, 包括全局上下文、目标区域、目标分割和关键点, 执行由粗到细的房间重新识别。

以区分细微差别 [7]。相比之下, Patch-NetVLAD [13]、AirLoc [3] 和 AnyLoc [16] 等方法通过聚合局部特征来构建全局描述符, 从而提升区分能力。尽管如此, 在物体高度相似且重复出现的室内环境中, 这些方法仍可能难以区分相似特征, 从而降低在此类场景中的效果 [35]。

此外, 与依赖物体类型识别以将空间分类为语义类别的房间分类方法不同 [18], 房间重识别的目标是在给定查询图像的基础上, 从参考数据库中精确检索出同一个房间实例。例如, 重新识别某个特定厨房需要结合全局功能上下文以及对特定物体属性的细粒度匹配。此外, 房间重识别还需应对视角变化, 因此必须具备容忍物体排列和外观部分不匹配的能力。这些需求常常导致仅基于物体分类的算法失效, 因为它们缺乏准确识别唯一房间实例所需的精度 [39]。

这引出了一个重要问题: “哪些物体属性对于房间重识别是真正关键的?” 为了解决这个问题, 我们开展了首个全面研究, 探索多层次面向物体的信息及其对房间重识别的影响。如 Figure 1 所示, 我们的实验表明, 四种层次的面向物体信息, 即全局上下文、物体图块、物体分割以及关键点, 都是必不可少的。具体而言, 我们发现每个层次在房间重识别中扮演着独特的角色。全局上下文 (例如沙发与电视的组合) 传达了用于将房间分类为客厅的关键语义信息。物体图块提供更精细的

细节,使得可以在房间内部进行区分,例如将卧室中的床头柜与工作区的书桌区分开来。物体分割进一步细化,通过分离餐桌与周围椅子等个体物体,有助于澄清房间布局。最后,物体上的关键点(如衣柜上的把手)可通过过滤其他房间中外观相似的家具来增强房间重识别的能力。此外,集成多层次的面向物体信息还能增强对视角变化的鲁棒性。

基于上述观察,我们提出了 AirRoom——一个简单却高效的房间重识别系统(ReID),该系统由三个阶段组成:全局、局部和细粒度阶段。在全局阶段,使用全局特征提取器捕捉全局上下文特征,进而粗略筛选出五个功能相似的候选房间。在局部阶段,首先应用实例分割识别出单个物体,然后通过感受野扩展器提取物体图块。接着使用物体特征提取器提取物体及图块特征,并通过面向物体的评分机制将候选范围缩小到两个房间。最后,在细粒度阶段,利用特征匹配精确地识别出最终房间。

总之,我们的贡献包括:

- 我们介绍了 AirRoom,一种基于物体感知的房间重识别管道,具有两个新颖的模块:感受野扩展器和物体感知评分,充分利用多层次的面向物体的信息,克服了以往方法中的局限性。
- 我们精心整理了四个全面的房间重识别数据集——MPReID、HMReID、GibsonReID 和 ReplicaReID——为评估房间重识别方法提供了多样化的基准。
- 大量实验表明, AirRoom 超越了当前的最先进技术,在显著的视角变化下仍能保持强大的可靠性和稳定的性能。

## 2. 相关工作

在本节中,我们回顾了与我们工作最相关的研究方向, i.e., 图像检索和视觉位置识别。

### 2.1. 图像检索

图像检索是计算机视觉中的一项基础且成熟的任务,其目标是在一个大型数据库中搜索与给定查询图像相似的图像。图像检索过程通常包括两个阶段:全局检索和重排序。在第一个阶段,使用聚合局部特征的全局描述符从大型数据库中检索出  $k$  个候选图像。随后通过局部特征匹配进行空间验证,以对这  $k$  个候选图像进行重排序。早期研究依赖于手工设计的特征 [5, 22],而当前的方法则利用深度网络学习具有区分力的表示 [8, 28]。

大多数图像检索方法关注于选择多样且相关的图像,以帮助用户在真实应用中发现符合其兴趣或需求的选项 [43]。尽管这些方法在检索相似图像方面表现出色,但它们通常缺乏对类别区分或精确 ReID 的重视 [11]。与此相比,我们的方法优先考虑实现精确的 ReID。我们遵循“全局检索和重排序”的流程,首先利用全局上下文特征识别排名前五的房间候选项。随后,我们的面向物体的机制以由粗到细的方式细化搜索,逐步区分

候选项,直至识别出最相似的房间,从而获得准确的结果。

### 2.2. 视觉位置识别

视觉位置识别(VPR)通常被框架化为一个特殊的图像检索问题,旨在将某一位置的视图与在不同条件下拍摄的同一地点的图像进行匹配。先前的方法分为两类:直接使用全局描述符的方法和将局部特征聚合为全局描述符的方法。早期依赖全局描述符的方法主要使用基于卷积神经网络(CNN)的骨干网络,如 ResNet [14],来生成这些描述符。然而,最近的方法则利用 DINOv2 等基础模型 [25] 来增强特征表示。在聚合类别中,早期技术采用了手工制作的特征,如 SIFT [22]、SURF [4] 和 ORB [31]。后来的进展,包括 NetVLAD 系列 [2, 13] 和 AnyLoc [16],采用了基于学习的模型来提取特征图,并将局部特征结合成全面的全局描述符。

然而,大多数 VPR 方法的高性能主要归功于在专门的 VPR 数据集上进行的大规模训练 [16]。由于日光、天气和季节的自然变化,收集户外场景的广泛数据相对简单。然而,在室内房间中进行此类数据收集则更加困难,这使得在室内数据集上进行大规模训练变得困难,并可能限制其有效性。我们的方法通过专注于面向物体的特征表示有效地解决了这一挑战,允许我们利用成熟的预训练模型进行物体特征学习。这一设计使得 AirRoom 能够在无需对特定数据集进行额外训练或微调的情况下,提供强大的性能。

## 3. 提出的方法

我们提出了一种简单而高效的管道, AirRoom,用于房间重新识别,利用多层次的面向对象信息,如 Figure 2 所示。接下来,我们将按照执行阶段的顺序系统地介绍管道的每个模块。

### 3.1. 全局阶段

在此阶段,我们利用全局特征提取器捕获全局上下文特征,这些特征来源于房间内物体的集体存在。这些特征随后用于全局检索,从数据库中粗略地选择语义相似的候选房间。

#### 3.1.1. 全局特征提取器

与户外环境相比,室内房间的变化较少。它们缺乏多样的地形特征,如空中、地下或水下特征,也不经历昼夜或季节性变化这样的时间性变化。因此,为每个室内房间收集大规模数据集是具有挑战性的,这使得许多视觉定位和重识别(VPR)方法的规模化训练变得复杂 [1, 2, 13]。

然而,室内房间本身在物体上具有丰富的多样性,每个物体都对房间的整体语义上下文做出贡献。通过利用这种全局上下文信息,我们可以将参考搜索专门集中在与查询图像具有相似语义特征的房间上。为此,我们倾向于选择在大规模图像数据集上预训练的骨干网络,因为它们提供了较强的泛化能力,能够有效地捕捉有价值的全局上下文特征 [17]。因此,我们的模型选



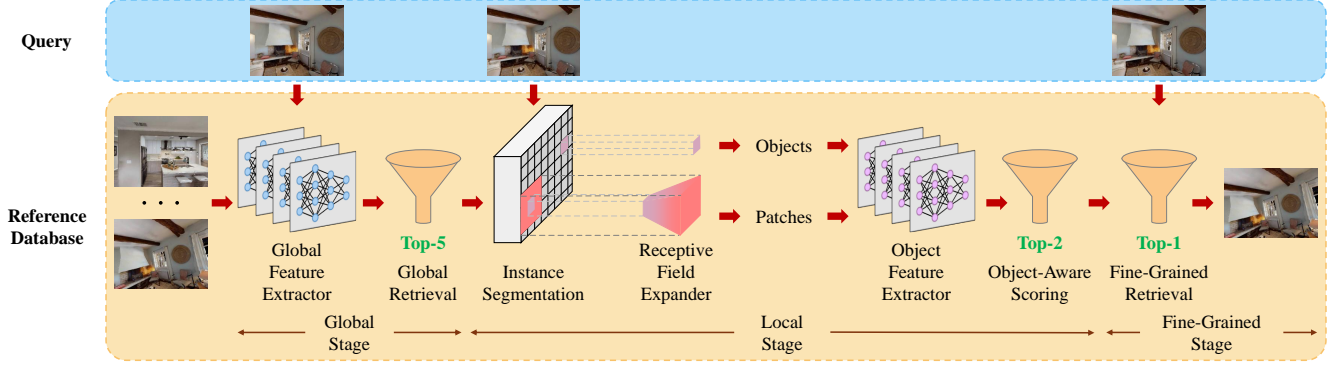


Figure 2. AirRoom 粗到细的处理流程。该流程首先由 Global Feature Extractor 开始，捕获全局上下文特征以检索前 5 张参考图像。然后通过 Instance segmentation 生成目标掩码，接着 Receptive Field Expander 提取目标图像块。Object Feature Extractor 对目标和图像块特征进行处理。Object-Aware Scoring 模块将候选图像缩小到前 2 张，最后 Fine-Grained Retrieval 识别出最合适的参考图像。

择包括基于卷积神经网络（CNN）的预训练模型，如 ResNet [14]，以及基于变换器的自监督学习模型，如 DINOv2 [25]。

### 3.1.2. 全局检索

使用全局特征提取器，我们为  $M$  个查询图像和  $N$  个参考图像提取全局上下文特征。令  $Q \in \mathbb{R}^{M \times D_g}$  和  $R \in \mathbb{R}^{N \times D_g}$  分别表示查询特征和参考特征，其中  $D_g$  是特征维度。然后计算余弦相似度矩阵  $S$  为：

$$S_{ij} = \frac{Q_i \cdot R_j}{\|Q_i\| \|R_j\|}. \quad (1)$$

对于每个查询，我们使用以下公式选择前 5 个最相似的参考候选：

$$\text{Top}_5(S_{i,:}) = \text{argsort}(-S_{i,:})[:5], \quad (2)$$

其中  $S_{i,:}$  表示第  $i$  个查询的余弦相似度。

### 3.2. 局部阶段

全局上下文特征提供了有价值的语义信息，有助于缩小候选列表的范围。然而，当面对许多语义相似的房间时，仅依赖全局上下文是不够的，局部特征变得越来越重要。在此阶段，我们采用局部视角，首先应用实例分割和感受野扩展器来识别物体和图像块。随后，我们使用物体特征提取器从物体和图像块中提取特征，接着通过面向物体的评分进一步优化候选列表。

#### 3.2.1. 实例分割

对于每张查询图像及其对应的五个候选图像，我们采用实例分割方法，如 Mask R-CNN [15] 和 Semantic-SAM [20]，来识别并描绘出各个独立的物体。该过程会生成每个物体的掩码和边界框。接下来，我们利用边界框计算每个物体的中心点  $c$ ，如下所示：

$$c = \left( \frac{x+W}{2}, \frac{y+H}{2} \right). \quad (3)$$

在此公式中， $x$  和  $y$  表示边界框左上角的像素坐标，而  $W$  和  $H$  分别表示边界框的宽度和高度。

#### 3.2.2. 感受野扩展器

单个物体的信息本身并不足以具有良好的判别性。例如，尽管不同的书桌可能具有不同的外观，它们既可能出现在食堂中，也可能出现在办公室中。然而，当一个物体与其邻近物体（如与计算机、键盘或笔记本并列的书桌）相关联时，就暗示该房间更有可能是办公室而不是食堂。这一见解促使我们将感受野从单一物体扩展到包含多个物体的图像区域。

给定图像中所有物体的中心点，我们采用 Delaunay 三角剖分法 [6] 来生成物体关系的三角形图。具体而言，Delaunay 三角剖分作用于物体中心点集合，确保没有任何物体中心位于任意三角形的外接圆内部。该方法最大化三角形的最小角度，避免出现狭长的三角形，从而确保物体邻接关系更加均匀。通过分析所得三角形之间的邻接关系，我们可以构建物体邻接矩阵，该矩阵编码了房间内物体之间的空间与关系接近度。

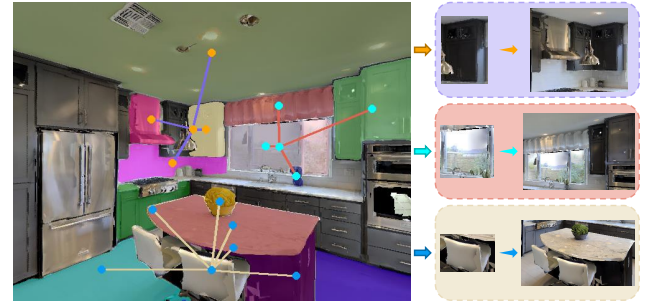


Figure 3. 感受野扩展器将感受野从单个物体扩展到富含上下文信息的区域。通过利用物体邻接矩阵和每个物体的边界框，它将单个物体如橱柜、窗户玻璃和椅子扩展为物体区域，如模块化厨房、多窗玻璃和餐桌套件，分别。

给定图像中的物体邻接矩阵和边界框，对于每一个物体，我们考虑其邻接物体的边界框，并将当前物体的边界框扩展，以包含所有邻接物体。这种扩展增加了感受野，使我们能够捕捉更丰富的上下文信息，如 Figure 3 所示。随后我们应用非极大值抑制（Non-Maximum Suppression, NMS），选取置信度最高的边界框，并基于其交并比（IoU）分数移除重叠边界框，从而获得一

组干净且信息丰富的物体图像块。

### 3.2.3. 面向对象的细化

面向对象的细化模块由三个关键子模块组成：对象特征提取器、互近邻算法和面向对象的评分。

**对象特征提取器** 为了有效地利用对象块和对象分割信息，我们优先考虑全局特征，而不是局部特征聚合。后者方法可能无法有效捕捉对象特征，并且可能显著增加计算复杂度和存储需求 [49]。如第 3.1.1 节所讨论的，我们继续依赖于在大规模图像数据集上预训练的模型。使用对象特征提取器，我们可以获得查询块和参考块及对象的特征。设  $Q_p = \{p_i^q\}_{i=1}^{n_{qp}}$  和  $Q_o = \{o_i^q\}_{i=1}^{n_{qo}}$  分别表示查询块和对象的特征集。对于查询的五个候选参考图像，我们将参考块和对象的特征集定义为  $R_p = \{p_i^r\}_{i=1}^{n_{rp}}$  和  $R_o = \{o_i^r\}_{i=1}^{n_{ro}}$ 。

**互近邻算法** 给定一组查询特征  $\{f_i^q\}_{i=1}^{n_q}$  和参考特征  $\{f_i^r\}_{i=1}^{n_r}$ ，通过对两个特征集进行穷举比较，识别出互近邻匹配对。设  $P$  表示这些互近邻匹配对的余弦相似度得分集，则有

$$P = \{\cos(f_i^q, f_j^r) \mid i = \text{NN}_r(f_i^q), j = \text{NN}_q(f_j^r)\} \quad (4)$$

其中

$$\text{NN}_q(f_i^q) = \arg \max_j \left( \frac{f_i^q \cdot f_j^r}{\|f_i^q\| \|f_j^r\|} \right), \quad (5)$$

$$\text{NN}_r(f_j^r) = \arg \max_i \left( \frac{f_i^q \cdot f_j^r}{\|f_i^q\| \|f_j^r\|} \right), \quad (6)$$

$$\cos(f_i^q, f_j^r) = \frac{f_i^q \cdot f_j^r}{\|f_i^q\| \|f_j^r\|}. \quad (7)$$

通过利用互近邻算法，我们可以显著提高检索准确性，同时缩小搜索空间并提高整体检索效率 [50]。

**面向对象的评分** 面向对象的得分  $s$  是全局得分  $s_{\text{global}}$ （在方程 1 中计算）、块得分  $s_{\text{patch}}$  和对象得分  $s_{\text{object}}$  的和：

$$s = s_{\text{global}} + s_{\text{patch}}(Q_p, R_p) + s_{\text{object}}(Q_o, R_o). \quad (8)$$

其中， $s_{\text{patch}}$  和  $s_{\text{object}}$  可以是  $s_{\text{mean}}$  或  $s_{\text{max}}$ ，其中

$$s_{\text{mean}}(Q_t, R_t) = \frac{1}{|P(Q_t, R_t)|} \sum_{x \in P(Q_t, R_t)} x, \quad (9a)$$

$$s_{\text{max}}(Q_t, R_t) = \max_{x \in P(Q_t, R_t)} x. \quad (9b)$$

在这些方程中， $P$  表示互近邻匹配对的余弦相似度得分集， $Q_t$  表示  $Q_p$  或  $Q_o$ ，而  $R_t$  表示  $R_p$  或  $R_o$ 。全局得分  $s_{\text{global}}$  作为先验，表明初始的五个候选具有不同的相关性。因此，我们保留这一项以考虑它们的不同相关性。

**面向对象的细化** 对于每个查询，我们使用面向对象的评分从初始的五个候选中选择最相似的前两个参考候选：

$$\text{Top}_2(s_i) = \text{argsort}(-s_i)[2], \quad (10)$$

其中， $s_i$  是第  $i$  个查询的面向对象的得分。

### 3.3. 细粒度阶段

补丁和物体特征为理解房间布局提供了有价值的信息；然而，在区分高度视觉相似的房间时，尤其是在视角变化和遮挡存在的情况下，它们可能不足以提供足够的区分度。与此相比，物体上的关键点表现出对纹理和外观变化的强大鲁棒性，使其能够有效地处理部分遮挡并排除异常值 [24]。这使得关键点能够提供一种更精细的方法，捕捉更细致的细节，从而实现更精确的房间识别。在此阶段，我们使用细粒度检索来选择最终的 top-1 结果。

#### 3.3.1. 细粒度检索

深度匹配器，如 SuperGlue [34]，在室内外的挑战性条件下，在视觉定位任务中表现良好。然而，它们通常面临效率问题。相比之下，LightGlue [21] 提供了高效性，并且没有牺牲匹配准确性，使其成为我们细粒度检索的理想选择。

对于每个查询图像及其两个候选参考图像，我们将查询图像与每个候选图像进行匹配，并记录匹配的关键点对数量。更多的匹配通常意味着两张图像的特征之间有更大的重叠和一致性，表明它们内容的相似度较高 [22]。具有更多匹配的候选图像被选为最终结果。

## 4. 实验结果

### 4.1. 数据集

目前没有现有的室内场景数据集完全适用于房间再识别任务，因为没有数据集能完全满足要求。像 ScanNet++ [46] 和 MIT Indoor Scenes [27] 这样的数据集缺乏房间级别的分割，导致多个房间共享一个场景标签。17 Places [32] 数据集包含了唯一标签的房间，但视角变化有限，而且图像往往较为模糊。尽管该数据集也包含昼夜变化，但这些变化对于大多数室内场景并不特别相关。Reloc110 [3] 数据集可能是最合适的选择；然而，它的质量不够理想，许多图像仅包含纯色的墙壁或地板，由于随机采样，导致上下文信息非常少。

一些高质量的室内 3D 数据集——如 Matterport3D [10]、Habitat-Matterport3D [30]、Gibson Database of 3D Spaces [44] 和 Replica [38]——提供了真实世界的室内场景。基于这些资源，并利用交互式 Habitat Simulator [26, 36, 40]，我们创建了四个新的数据集：MPReID、HMReID、GibsonReID 和 ReplicaReID，如 Figure 4 所示。

使用 Habitat Simulator，我们为每个房间配置了一个代理，并手动选择了 5 到 10 个关键姿势来引导其探索。代理从不同角度捕捉了 640×480 的 RGB-D 图像，每个房间的图像数量为 300 到 800 张，具体取决于关

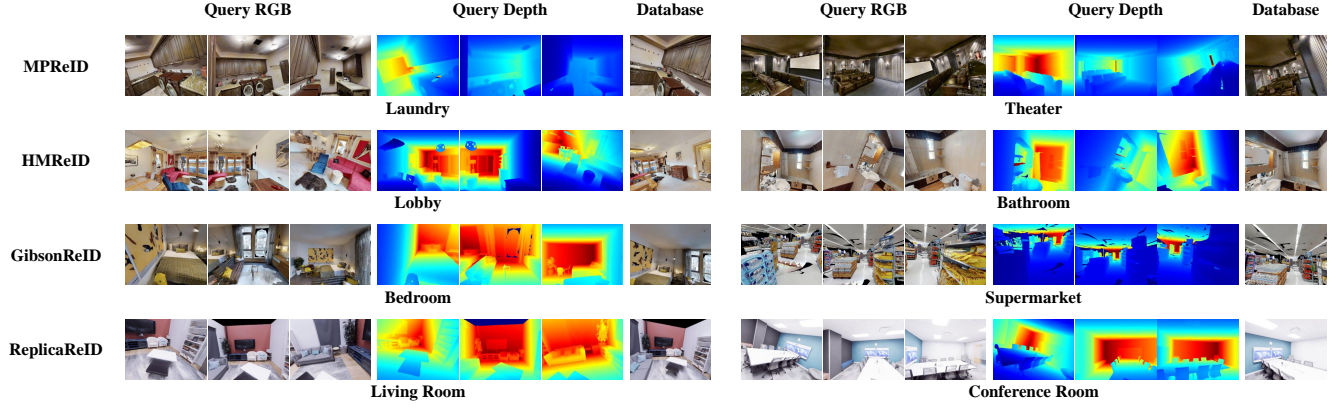


Figure 4. 四个新构建的房间重新识别数据集示意图：MPReID、HMReID、GibsonReID 和 ReplicaReID。每个房间在数据库中仅提供一张参考图像，而查询图像则从不同视角捕捉每个房间。

Methods	MPReID				HMReID				GibsonReID				ReplicaReID			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
CVNet	17.45	29.52	17.45	19.34	11.71	25.42	11.95	13.86	12.04	24.06	12.07	14.27	15.93	20.53	15.74	16.64
DINov2	59.36	64.68	59.36	58.91	53.91	60.52	53.73	54.69	61.01	65.88	61.78	61.71	78.06	79.68	77.97	77.44
Patch-NetVLAD	64.32	70.47	64.36	65.53	64.86	68.78	64.32	65.16	61.47	66.90	62.04	62.51	63.77	64.97	63.86	63.87
AnyLoc	92.34	93.23	92.36	92.32	89.69	90.25	89.53	89.62	85.85	87.42	86.15	86.21	88.57	89.89	88.46	88.42
AirRoom	93.96	94.52	93.98	93.91	93.80	94.01	93.55	93.62	91.68	92.41	91.79	91.63	87.18	89.39	87.08	87.24

Table 1. AirRoom 与基线模型在四个新构建的房间重识别数据集上的整体性能对比。

键姿势的数量。然而，许多随机采样的图像质量较差，通常只包含墙壁或地板，缺乏足够的上下文信息。为了解决这一问题，我们仔细筛选了每个房间的图片，保留了那些准确代表空间并为房间 ReID 提供有价值信息的图像。

总的来说，这些数据集如下：MPReID 包括 15 个场景、105 个房间和 16,231 张 RGB-D 图像；HMReID 包含 21 个场景、105 个房间和 15,781 张 RGB-D 图像；GibsonReID 包含 24 个场景、45 个房间和 6,743 张 RGB-D 图像；ReplicaReID 包括 12 个场景、19 个房间和 2,862 张 RGB-D 图像。

#### 4.2. 数据库预处理

在房间重识别设置中，我们有多个查询图像和一个参考数据库。对于每个数据集，我们仅选择每个房间的一张图像来构建数据库。具体来说，对于每个房间的所有图像，我们首先使用 CLIP [29] 提取特征嵌入。然后，我们应用 K-means 聚类，设定聚类数为 1。距离聚类中心最近的图像被选择为参考图像，因为它最能代表房间的视觉特征 [42]。

在构建参考数据库后，我们对特征进行预处理。首先，我们使用全局特征提取器来获取并保存全局上下文特征。接着，我们应用实例分割模块对物体进行分割。然后，我们使用我们的感受野扩展器来获取物体补丁，并使用物体特征提取器来提取并保存物体和补丁的特征。

#### 4.3. 实验概述

我们进行了五个主要实验：整体性能比较、分组性能比较、管道灵活性评估、消融研究和运行时分析。在评估过程中，我们使用了准确率、精确度、召回率和 F1 分数作为评价指标。每个类别的精确度、召回率和 F1 分数是通过多类混淆矩阵计算得出的，随后进行了宏平

均。准确率是通过正确匹配的查询与总查询数之比来衡量的。详细的运行时分析和其他实验结果在附录中提供。

#### 4.4. 整体性能比较

在本节中，我们展示了我们方法的最佳版本与多种最先进方法之间的性能对比，从而使我们能够在不同的特征提取和检索策略下，将我们的管道与已有的房间重识别模型进行基准测试。

我们选择了三类基线方法：图像检索方法（CVNet [19]）、基于全局描述子的视觉位置识别（VPR）（DINov2 [25]），以及使用局部特征聚合的 VPR 方法（Patch-NetVLAD [13] 和 AnyLoc [16]）。具体来说，我们使用了 DINov2 的 Base 版本，将 CVNet 配置为 ResNet50 [14] 主干网络，并将降维维度设为 2048，选择了 Patch-NetVLAD 的性能优化版本，并配置 AnyLoc 为 AnyLoc-VLAD-DINov2，使用 32 个 VLAD 聚类。

Table 1 展示了 AirRoom 与基线方法之间的定量比较，结果表明 AirRoom 在几乎所有指标和数据集上均优于所有基线方法。在房间重识别任务中，图像检索方法由于其并非专注于 top-1 精度，通常在分类指标上表现较差，而 VPR 方法则取得了更好的结果。基于全局描述子的 VPR 方法仅捕捉高层语义信息，常常检索到语义相似但缺乏细节的房间；相比之下，使用局部特征聚合的 VPR 方法（如 Patch-NetVLAD）强调低层编码，但可能忽视全局上下文，从而导致检索准确性下降。Figure 5 展示了 CVNet、DINov2、Patch-NetVLAD 和 AnyLoc 的失败案例，突显了这些方法的局限性。

尽管 AnyLoc 因其在“任何位置、任何时间、任何视角”VPR 中表现稳健而著称，并具有良好的性能，但 AirRoom 进一步提升了表现，在可用提升空间内相较



Methods	MPReID				HMReID				GibsonReID				ReplicaReID			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
ResNet50	76.14	79.21	76.20	76.58	69.03	73.21	68.61	69.07	68.84	72.30	69.50	69.00	75.05	78.61	75.30	74.88
CVNet	17.45	29.52	17.45	19.34	11.71	25.42	11.95	13.86	12.04	24.06	12.07	14.27	15.93	20.53	15.74	16.64
AirRoom-ResNet50	86.16	87.69	86.19	86.16	81.23	83.90	80.76	81.23	82.53	84.91	82.86	82.54	83.51	84.85	83.54	83.17
NetVLAD	82.22	86.77	82.24	82.92	72.04	80.79	71.83	73.05	68.86	81.00	69.24	71.01	77.04	81.31	77.28	77.63
Patch-NetVLAD(4096)	64.32	70.47	64.36	65.53	64.86	68.78	64.32	65.16	61.47	66.90	62.04	62.51	63.77	64.97	63.86	63.87
Patch-NetVLAD(512)	66.62	71.85	66.67	67.62	65.63	69.28	65.01	65.57	60.95	69.16	61.43	62.46	66.00	68.75	66.25	66.22
Patch-NetVLAD(128)	65.04	70.84	65.09	66.15	61.17	66.71	60.69	61.42	58.31	66.15	58.69	59.66	61.88	66.29	62.12	62.05
AirRoom-NetVLAD	89.38	90.99	89.40	89.50	83.47	86.91	83.08	83.66	82.29	87.27	82.61	82.98	83.58	84.42	83.60	83.37

Table 2. 与基准模型的组别性能比较，以评估面向对象机制的有效性。

Methods	MPReID				HMReID				GibsonReID				ReplicaReID			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
ViT	81.90	85.27	81.96	81.71	76.47	79.37	76.04	75.91	76.46	78.51	77.00	76.88	77.99	81.41	78.15	77.46
AirRoom-ViT	89.70	90.97	89.72	89.35	86.58	88.13	86.12	86.23	87.08	88.24	87.33	87.19	84.84	86.85	84.79	84.45
DINO	80.66	84.32	80.73	81.14	73.54	77.73	73.13	73.79	72.28	74.92	72.92	72.89	86.58	87.77	86.60	86.49
AirRoom-DINO	88.00	89.59	88.05	88.09	83.62	85.43	83.14	83.40	84.62	86.23	84.95	84.83	87.49	88.56	87.41	87.25
DINOv2	59.36	64.68	59.36	58.91	53.91	60.52	53.73	54.69	61.01	65.88	61.78	61.71	78.06	79.68	77.97	77.44
AirRoom-DINOv2	76.10	79.03	76.11	75.80	70.95	73.86	70.66	70.78	78.63	80.44	79.00	78.45	85.57	86.58	85.45	85.19
AnyLoc(16)	90.22	91.18	90.25	90.17	84.63	86.40	84.56	84.91	82.20	83.77	82.59	82.74	85.64	87.52	85.59	85.67
AirRoom-AnyLoc(16)	93.05	93.66	93.08	92.99	91.55	92.12	91.32	91.47	89.04	89.97	89.21	89.13	86.83	89.03	86.76	86.90
AnyLoc(8)	88.03	89.33	88.08	88.01	81.93	83.89	81.94	82.25	79.27	81.29	79.72	79.71	84.98	86.19	85.03	84.88
AirRoom-AnyLoc(8)	92.37	93.14	92.40	92.32	90.24	90.85	90.01	90.13	88.37	89.38	88.56	88.52	85.81	87.67	85.77	85.80

Table 3. 全局特征提取器的灵活性。

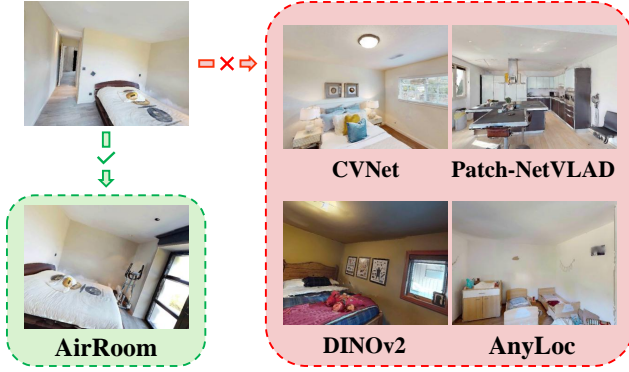


Figure 5. 给定一个卧室查询，AirRoom 通过利用物体相关性进行房间重识别，准确地检索目标图像。相比之下，CVNet 检索视觉上相似的图像，但没有保持场景的准确性，DINOv2 捕捉语义内容但忽略了颜色细节，Patch-NetVLAD 使用聚合的局部特征形成全局描述符，检索到的图像语义信息不匹配，而 AnyLoc 考虑了语义和颜色属性，但忽略了房间内物体的重要性。

AnyLoc 提高了 20% 到 40%。例如，AnyLoc 在 HMReID 上取得了 89.69% 的准确率，留下了约 10% 的提升空间；而 AirRoom 凭借 93.80% 的准确率，在剩余空间内实现了高达 40% 的提升。这些结果突显了 AirRoom 在房间重识别任务中卓越的精度与优化表现。

#### 4.5. 按组别的性能比较

许多基准方法采用“骨干网络 + 增强机制”的范式，我们的方法也遵循这一范式。在本节中，我们将使用与每个组基准相同的骨干网络，将我们的面向对象增强机制与几种最先进的方法进行性能比较。此设置使我们能够直接评估我们的面向对象增强机制的有效性。

对于 ResNet50 骨干网络组，我们使用 CVNet [19] 作为基准。在 NetVLAD 骨干网络组中，我们采用 Patch-NetVLAD [13] 作为基准，并在三种降维下进行测试：4096、512 和 128。

Table 2 显示，在每个组别中，单一的骨干网络优于那些通过各种机制尝试增强性能的基准方法，这表明这些机制未能有效地捕捉到室内环境中的关键信息。相比之下，我们的面向对象增强机制通过强调室内环境中对象的重要性，显著提升了骨干网络的性能。

#### 4.6. 管道灵活性评估

在本节中，我们通过测试其关键模块的不同配置，系统地评估了 AirRoom 的灵活性和适应性。结果清楚地表明，AirRoom 并不依赖于任何特定模型，能够有效集成各种不同类型的模型。

##### 4.6.1. 全局特征提取器

我们测试了多种全局特征提取器，包括 ViT [12]、DINO [9]、DINOv2 [25] 和 AnyLoc-VLAD-DINOv2 [16]，VLAD 簇大小分别为 16 和 8。

如 Table 3 所示，AirRoom 在几乎所有情况下，在所有度量标准和数据集上始终能够超过 85%，无论使用的全局特征提取器的能力如何。即使是在 DINOv2 的唯一例外情况下，AirRoom 的表现仍然提高了近 15%。这表明我们的管道的有效性并不依赖于任何特定的全局特征提取器，突显了 AirRoom 在各种主干配置下的适应性，并强调了其强大的灵活性。

##### 4.6.2. 实例分割

我们将传统的实例分割方法（如 Mask R-CNN [15]）与更近期的 approaches，包括 Semantic-SAM [20] 进行比较，后者利用先进技术实现更细粒度的分割。

Table 4 显示，无论使用何种实例分割模块，AirRoom 始终比基准方法高出超过 15%。这证明了我们的管道不依赖于任何特定的实例分割方法，强调了其在此组件中的适应性。

Methods	HMReID			
	Accuracy	Precision	Recall	F1
DINOv2	53.91	60.52	53.73	54.69
AirRoom-MaskRCNN	69.44	72.23	69.08	69.07
AirRoom-SSAM	70.95	73.86	70.66	70.78

Table 4. 实例分割的灵活性。

#### 4.6.3. 目标特征提取器

我们实验了传统的骨干网络，如 ResNet50 [14]，以及更现代的骨干网络，如 DINOv2 [25]，作为目标特征提取器。

如 Table 5 所示，AirRoom 在基准模型上实现了显著的性能提升，不同目标特征提取器之间的性能变化很小。这支持了我们管道在适应各种特征提取方法方面的灵活性。

Methods	HMReID			
	Accuracy	Precision	Recall	F1
DINOv2	53.91	60.52	53.73	54.69
AirRoom-ResNet50	70.95	73.86	70.66	70.78
AirRoom-DINOv2	68.67	71.81	68.33	68.59

Table 5. 目标特征提取器的灵活性。

#### 4.6.4. 面向对象评分

我们评估了均值 ( $s_{\text{mean}}$ ) 和最大值 ( $s_{\text{max}}$ ) 两种策略，用于计算补丁得分 ( $s_{\text{patch}}$ ) 和对象得分 ( $s_{\text{object}}$ )，并评估它们对整体性能的影响。

Table 6 显示，无论使用何种面向对象的评分方法，AirRoom 的性能保持稳定。这突显了面向对象信息在房间重新识别中的稳健性，并展示了 AirRoom 在适应不同评分策略方面的灵活性。

Methods	HMReID			
	Accuracy	Precision	Recall	F1
DINOv2	53.91	60.52	53.73	54.69
AirRoom-Max(patch)-Mean(object)	70.95	73.86	70.66	70.78
AirRoom-Max(patch)-Max(object)	71.02	74.02	70.72	70.85
AirRoom-Mean(patch)-Max(object)	70.85	73.85	70.55	70.70
AirRoom-Mean(patch)-Mean(object)	70.90	73.78	70.62	70.73

Table 6. 面向对象的评分灵活性。

#### 4.7. 消融研究

在本节中，我们从我们的管道中移除某些模块——包括全局得分  $s_{\text{global}}$ 、局部得分  $s_{\text{patch}}$ 、目标得分  $s_{\text{object}}$ （在目标感知评分中）以及整个细粒度检索（FGR）——以评估每个组件的重要性和有效性。

Table 7 显示，移除任何模块都会导致性能下降。然而，只要至少保留一个模块，我们的管道仍然优于基线。Table 8 证明，当全局特征提取器（ViT）表现良好时，全局得分  $s_{\text{global}}$  显著提升性能。另一方面，当全局特征提取器（DINOv2）效果较差时，全局得分  $s_{\text{global}}$  会产生轻微的负面影响，导致性能略微下降。这个结果与我们在第 3.2.3 节中的假设一致，其中全局得分充当优先级排序的先验，用于排名五个候选者的优先级。总体而言，这些消融研究确认了我们管道中的每个模块都是重要且必要的。

#### 4.8. 局限性

尽管 AirRoom 在不同视角变化下的房间重识别任务中达到了最先进的性能，但我们工作的一个局限性是无法验证其对由可移动物体引起的室内物品重排的鲁棒性。尽管我们基于互最近邻的物体感知评分方法在一定程度上对这种重排具有鲁棒性，但我们实验中使用的数据集缺乏这些情况。相比之下，最近在动态场景理

Methods	HMReID			
	Accuracy	Precision	Recall	F1
DINOv2 (AirRoom-w/o all)	53.91	60.52	53.73	54.69
AirRoom-w/o $s_{\text{patch}}$	66.68	70.04	66.42	66.68
AirRoom-w/o $s_{\text{object}}$	69.77	72.84	69.48	69.64
AirRoom-w/o FGR	66.11	70.85	65.80	66.41
AirRoom-w/o $s_{\text{patch}}$ & $s_{\text{object}}$	62.26	66.43	62.03	62.46
AirRoom-w/o $s_{\text{patch}}$ & FGR	59.39	65.25	59.14	59.97
AirRoom-w/o $s_{\text{object}}$ & FGR	63.44	68.68	63.14	63.84
AirRoom	70.95	73.86	70.66	70.78

Table 7. 消融研究（不包括全局评分实验）。

Methods	HMReID			
	Accuracy	Precision	Recall	F1
ViT	76.47	79.37	76.04	75.91
AirRoom-ViT-w/o $s_{\text{global}}$	84.86	86.82	84.34	84.61
AirRoom-ViT	86.58	88.13	86.12	86.23
DINOv2	53.91	60.52	53.73	54.69
AirRoom-DINOv2-w/o $s_{\text{global}}$	71.73	74.97	71.44	71.64
AirRoom-DINOv2	70.95	73.86	70.66	70.78

Table 8. 关于全局得分的消融研究。

解方面的进展 [47] 专注于在存在移动物体的情况下识别场景，可能比我们的方法提供更强的鲁棒性。未来的工作应考虑构建包含物体重排的数据集，并集成新技术以增强对可移动物体的鲁棒性，从而提高房间重识别的性能。

#### 5. 结论

房间重识别是一个具有挑战性但至关重要的研究领域，在增强现实和居家护理机器人等领域的应用不断增长。在本文中，我们提出了 AirRoom，这是一种无训练、面向对象的房间重识别方法。AirRoom 利用多层次的面向对象特征来捕捉室内房间的空间和上下文信息。为了评估 AirRoom，我们专门构建了四个新的数据集用于房间重识别。实验结果证明，AirRoom 在视角变化下具有较强的鲁棒性，并且在几乎所有度量和数据集上都优于现有的最先进方法。此外，该管道高度灵活，在不依赖于特定模型配置的情况下仍能保持高性能。总的来说，我们的工作确立了 AirRoom 作为一种强大且多功能的房间重识别解决方案，具有广泛的现实应用潜力。

致谢

本研究得到了 DARPA 资助（奖项编号：HR00112490426）。本文中所表达的任何观点、发现、结论或建议均属于作者本人观点，并不一定代表 DARPA 的立场。

#### References

- [1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Mixvpr: Feature mixing for visual place recognition, 2023. 2
- [2] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition, 2016. 1, 2
- [3] Aryan, Bowen Li, Sebastian Scherer, Yun-Jou Lin, and Chen Wang. Airlloc: Object-based indoor relocation, 2023. 1, 4

- [4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 2
- [5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. Similarity Matching in Computer Vision and Multimedia. 2
- [6] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008. 3
- [7] Yingfeng Cai, Junqiao Zhao, Jiafeng Cui, Fenglin Zhang, Chen Ye, and Tiantian Feng. Patch-netvlad+: Learned patch descriptor and weighted matching strategy for place recognition, 2022. 1
- [8] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search, 2020. 2
- [9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 1, 6
- [10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 4
- [11] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2), 2008. 2
- [12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 6
- [13] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition, 2021. 1, 2, 5, 6
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2, 3, 5, 7
- [15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 3, 6
- [16] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition, 2023. 1, 2, 5, 6
- [17] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better?, 2019. 2
- [18] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation, 2017. 1
- [19] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. Correlation verification for image retrieval, 2022. 5, 6
- [20] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity, 2023. 3, 6
- [21] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed, 2023. 4
- [22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 2, 4
- [23] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J. Leonard, David Cox, Peter Corke, and Michael J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016. 1
- [24] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 4
- [25] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2024. 1, 2, 3, 5, 6, 7
- [26] Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlaváč, Tiffany Min, Theo Gervet, Vladimír Vondruš, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023. 4
- [27] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420, 2009. 4
- [28] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation, 2018. 2
- [29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 5
- [30] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech



- Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 4
- [31] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 2
- [32] Raghavender Sahdev and John K. Tsotsos. Indoor place recognition system for localization of mobile robots. In *2016 13th Conference on Computer and Robot Vision (CRV)*, pages 53–60, 2016. 4
- [33] Gabriel Sarch, Zhaoyuan Fang, Adam W. Harley, Paul Schydlow, Michael J. Tarr, Saurabh Gupta, and Katerina Fragkiadaki. Tidee: Tidying up novel rooms using visuo-semantic commonsense priors, 2022. 1
- [34] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks, 2020. 4
- [35] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression, 2019. 1
- [36] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 4
- [37] Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, Peizhao Zhang, Bastian Leibe, Peter Vajda, and Ji Hou. Controlroom3d: Room generation using semantic proxy rooms, 2023. 1
- [38] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Giese, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 4
- [39] Niko Sünderhauf, Sareh Abolahrari Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems*, 2015. 1
- [40] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 4
- [41] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis, 2018. 2
- [42] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005. 5
- [43] Ji Wan, Dayong Wang, Steven C. H. Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. *Proceedings of the 22nd ACM international conference on Multimedia*, 2014. 2
- [44] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR)*, 2018 IEEE Conference on. IEEE, 2018. 4
- [45] Yifan Xu, Pourya Shamsolmoali, and Jie Yang. Clusvpr: Efficient visual place recognition with clustering-based weighted transformer, 2023. 1
- [46] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes, 2023. 4
- [47] Yanpeng Zhao, Yiwei Hao, Siyu Gao, Yunbo Wang, and Xiaokang Yang. Dynamic scene understanding through object-centric voxelization and neural rendering, 2024. 7
- [48] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020. 2
- [49] Lingxi Zheng, Yi Zheng, and Yi Yang. Sift meets cnn: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1224–1244, 2018. 4
- [50] Zhun Zhong, Liang Zheng, Dengpan Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1318–1327, 2017. 4

# AirRoom: 物体在房间重识别中的重要性

## Supplementary Material

### 6. 数据集

表 9 展示了 MPreID 的组成, 而表 10、表 11 和表 12 分别展示了 HMReID、GibsonReID 和 ReplicaReID 的组成。表 13 报告了每个房间 ReID 数据集中语义上不同房间的数量。

Scene	Rooms	Images	Scene	Rooms	Images
8WUmhLawc2A	8	1232	EDJbREhghzL	7	1078
RPmz2sHmrrY	5	770	S9hNv5qa7GM	9	1423
ULsKaCPVFJR	5	780	VzqfbhrpDEA	7	1078
WYY7iVyf5p8	4	616	X7HyMhZNos0	7	1078
YFuZgdQ5vWj	7	1078	i5noydfURQK	7	1078
jh4fc5c5qoQ	5	770	mJXqzFtmKg4	9	1386
w7QyJ3H9Bp	8	1232	wc2JMjhGNzB	11	1708
yqstnuAEVhm	6	924	Total	105	16231

Table 9. MPreID 的组成。

Scene	Rooms	Images	Scene	Rooms	Images
7dmR22gwQpH	6	924	ACZZiU6BXLz	5	682
CETmJJqkhcK	5	813	CFVBbU9Rsyb	5	770
CoerRdivP7	3	462	DZsJKHoqEYg	5	793
EQSguCqe5Rk	5	819	Fgtk7tL8R9Y	5	822
GLAQ4DNUx5U	7	1156	GcfUJ79xCZc	5	572
NcK5aACg44h	5	754	P8L1328HrLi	5	819
VSxVP19Cdyw	5	769	b3CuYvwpzZv	5	690
ixTj1aTMup2	5	757	ochRmQAhtkF	5	641
qWb4MVxqCW7	6	879	rrijmoZhZCo	5	704
w7QyJ3H9Bp	5	692	zR6kPe1PsyS	5	803
zepmXAdrpjR	3	460	Total	105	15781

Table 10. HMReID 的组成。

Scene	Rooms	Images	Scene	Rooms	Images
Ackermanville	1	154	Angiola	1	154
Avonia	2	308	Beach	3	462
Branford	1	154	Brevort	1	154
Cason	2	262	Cooperstown	2	308
Corder	2	308	Creede	4	526
Elmira	2	308	Eudora	2	308
Fredericksburg	2	308	Greigsville	1	154
Idanha	1	154	Laytonsville	3	462
Lynxville	2	308	Mahtomedi	2	257
Mayesville	2	308	Northgate	1	154
Ogilvie	2	308	Ophir	3	462
Pablo	1	154	Sumas	2	308
-	-	-	Total	45	6743

Table 11. GibsonReID 的组成。

Scene	Rooms	Images	Scene	Rooms	Images
apartment_0	3	462	apartment_1	1	154
apartment_2	4	616	fri_apartment_0	3	426
hotel_0	1	154	office_0	1	154
office_2	1	140	office_3	1	140
office_4	1	154	room_0	1	154
room_1	1	154	room_2	1	154
-	-	-	Total	19	2862

Table 12. ReplicaReID 的组成。

	bathroom	kitchen	living	office	bedroom	theater	dining	wardrobe	gym	laundry	garage	storage	nursery	supermarket
MPreID	13	15	20	3	41	4	4	2	2	1	0	0	0	0
HMReID	10	18	29	8	31	0	3	1	0	1	0	2	2	0
GibsonReID	2	10	11	3	12	0	1	0	3	1	0	1	0	1
ReplicaReID	0	2	6	6	3	0	2	0	0	0	0	0	0	0

Table 13. 四个新构建的房间 ReID 数据集中语义不同房间的统计数据。

### 7. 实验细节

#### 7.1. 整体性能比较

**基准配置** 对于 CVNet, 我们使用 ResNet50 作为骨干网络, 并将降维维度设置为 2048。对于 DINOv2, 我们使用 DINOv2-Base 检查点。对于 Patch-NetVLAD, 我们加载在 Pittsburgh 数据集上优化的预训练权重, 应用 WPCA 来将特征嵌入维度降低到 4096, 设置 RANSAC 作为匹配器, 使用 0.45、0.15 和 0.4 的补丁权重, 配置补丁大小为 2、5 和 8, 步幅为 1。对于 AnyLoc, 我们采用 AnyLoc-VLAD-DINOv2, 使用 DINOv2 ViT-G/14 架构, 将描述符层设置为 31, 使用 VLAD 并设置 32 个聚类, 指定域为室内。

**基准适配** 对于 CVNet 和 Patch-NetVLAD, 我们通过选择前 5 个候选项进行全局检索, 然后进行重新排序。对于 CVNet, 选择具有最高 CVNet-Rerank 图像相似度得分的候选项作为最终结果; 而对于 Patch-NetVLAD, 则选择在成对局部匹配阶段具有最高 RANSAC 得分的参考项。对于 DINOv2 和 AnyLoc, 从查询和参考图像中提取全局特征, 并计算余弦相似度。选择具有最高余弦相似度得分的参考图像作为最终匹配项。

**AirRoom 配置** 对于全局特征提取器, 我们使用 AnyLoc-VLAD-DINOv2, 采用 DINOv2 ViT-G/14 架构, 将描述符层设置为 31, 应用 VLAD 并设置 32 个聚类, 指定域为室内。对于实例分割, 我们采用 Semantic-SAM, 并使用来自 SA-1B 的预训练权重和 SwinL 骨干网络。物体特征提取器使用在 ImageNet 数据集上预训练的 ResNet50 模型实现。对于细粒度检索, 我们使用 LightGlue, 并将最大关键点数设置为 2048。

#### 7.2. 按组性能比较

**基线配置** 对于 ResNet50 主干网络组, ResNet50 和 CVNet 的配置遵循 Section 7.1 中详细介绍的设置。对于 NetVLAD 主干网络组, 我们使用 NetVLAD 与 VGG-16 作为特征提取器, 配置为 64 个聚类和 512 维特征。对于 Patch-NetVLAD, 特征维度分别设置为 4096、512 和 128, 其余设置与 Section 7.1 一致。

**基线适配** 对于 ResNet50 主干网络组, ResNet50 从查询图像和参考图像中提取全局特征, 并使用余弦相

似度选择得分最高的参考图像作为最终匹配。CVNet 的适配过程详见 Section 7.1。对于 NetVLAD 主干网络组，NetVLAD 从查询图像和参考局部特征中聚合全局描述符，并选择余弦相似度得分最高的参考图像作为最终结果。Patch-NetVLAD 的适配过程同样遵循 Section 7.1。

**AirRoom 配置** 对于 ResNet50 主干网络组，ResNet50 作为全局特征提取器，配置与 Section 7.1 一致。对于 NetVLAD 主干网络组，NetVLAD 作为全局特征提取器，配置遵循本节中“基线配置”段落中的设置。两组中其余模块的配置也与 Section 7.1 一致。

### 7.3. 管道灵活性评估

#### 7.3.1. 全局特征提取器

**基准配置** 对于 ViT，我们使用 Base 变体，补丁大小为 16，输入图像大小为  $224 \times 224$ ，并加载来自 ImageNet 的预训练权重。对于 DINO，我们采用 DINO 预训练的 Vision Transformer Small (ViT-S/16) 变体。DINOv2 的配置遵循 Section 7.1。对于 AnyLoc，VLAD 聚类设置为 16 和 8，其他所有配置与 Section 7.1 一致。

**基准适配** 所有基准用于从查询图像和参考图像中提取特征，计算余弦相似度，以识别具有最高相似度得分的参考房间。

**AirRoom 配置** 为了与主干基准进行比较，主干被用作全局特征提取器。主干配置遵循本节“基准配置”段落中概述的内容，而 AirRoom 中其余模块的配置与 Section 7.1 一致。

#### 7.3.2. 实例分割

**AirRoom 配置** DINOv2 被用作全局特征提取器。对于 Mask R-CNN，我们使用带有 ResNet50 主干和 FPN 的 Mask R-CNN，并加载在 COCO 上训练的预训练权重。对于 Semantic-SAM，我们使用带有从 SA-1B 预训练的权重和 SwinL 主干的 Semantic-SAM。其余模块的配置与 Section 7.1 一致。

## 8. 大规模评估

由于四个房间 ReID 数据集采用了一致的格式，我们在它们的联合数据集上评估我们的方法，这样可以为每种房间类型提供更多的样本，并评估在数据扩展时所提出方法的可行性。为此，我们通过将所有四个数据集合并，构建了一个大规模数据集 UnionReID。表 14 展示了 AirRoom 与四种基线方法的性能比较，表明在大规模条件下，AirRoom 仍然优于它们。

## 9. 室内定位数据集评估

严格来说，房间重识别 (Room ReID) 是一项新任务，尚无先前建立的数据集，并且与室内定位任务有本质区别。为了填补这一空白，我们引入了四个新数据集。

Methods	UnionReID			
	Accuracy	Precision	Recall	F1
CVNet	14.10	27.53	14.10	16.19
DINOv2	53.01	59.44	53.02	53.50
Patch-NetVLAD	61.15	67.53	61.04	62.31
AnyLoc	88.28	89.62	88.22	88.32
AirRoom	91.87	92.55	91.76	91.76

Table 14. 在 UnionReID 上与基线模型的对比，用于评估 AirRoom 在数据扩展下的性能。

然而，在审查现有的室内定位数据集后，我们识别出两个勉强可用的数据集：InLoc [41] 和 Structured3D [48]。InLoc [41] 采用基于区域的划分，而非基于房间的划分，其中一些图像仅捕捉走廊和角落。Structured3D [48] 包含数万个房间实例，但每个房间的视角少于六个。这些局限性减少了这两个数据集的适用性，尽管它们仍然在某种程度上可用。然而，在这些数据集上评估我们的方法仍能进一步加强其验证。

表 15 展示了在这两个室内定位数据集上的比较结果，结果表明，AirRoom 始终优于其他方法。此外，由于 InLoc 代表了更为现实的实际环境，结果进一步证明了 AirRoom 在实际环境中的鲁棒性。

Methods	InLoc				Structured3D			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
CVNet	8.41	12.49	8.41	8.99	12.60	21.39	12.60	14.22
DINOv2	11.13	19.93	11.13	11.85	53.00	63.60	53.00	54.04
Patch-NetVLAD	12.78	19.59	12.78	13.73	56.30	67.67	56.30	57.71
AnyLoc	15.78	26.11	15.78	17.04	73.40	79.75	73.40	73.90
AirRoom	16.80	26.36	16.80	18.05	76.20	82.88	76.20	76.70

Table 15. 在现有数据集上与基线模型的对比，以进一步验证我们的方法。

## 10. 运行时分析

在本节中，我们评估了每个模块的运行时，并将我们的管道的总运行时与几种最先进的方法进行比较，以评估我们方法的效率。

Modules	Runtime (ms)					
	t=0	t=0.1	t=0.2	t=0.3	t=0.4	t=0.5
Global Feature Extractor	48.8	44.1	43.2	44.0	43.0	43.8
Global Retrieval	0.1	0.1	0.1	0.1	0.1	0.1
Instance Segmentation	38.7	38.1	38.2	38.1	38.0	38.0
Receptive Field Expander	6.9	2.9	1.7	1.3	0.9	0.7
Object Feature Extractor	113.7	71.3	47.0	33.3	29.0	22.8
Object-Aware Scoring	2.9	2.2	1.7	1.5	1.4	1.2
Fine-Grained Retrieval	87.4	86.3	86.1	86.1	85.8	86.2
Total	299.9	246.5	219.4	205.7	199.5	194.2

Table 16. Mask R-CNN & ResNet 运行时间。

当使用 Mask R-CNN 进行实例分割时，Table 16 证明了当使用 ResNet 时，提高目标掩模分数阈值会显著减少对象特征提取器的运行时。这是由于需要处理的对象和补丁数量减少所致。使用 DINOv2 作为对象特



Modules	Runtime (ms)					
	t=0	t=0.1	t=0.2	t=0.3	t=0.4	t=0.5
Global Feature Extractor	65.0	58.6	52.7	50.2	48.6	47.9
Global Retrieval	0.1	0.1	0.1	0.1	0.1	0.1
Instance Segmentation	38.4	38.8	38.6	38.7	38.6	38.5
Receptive Field Expander	7.9	3.2	1.8	1.3	1.0	0.7
Object Feature Extractor	146.9	86.9	55.6	40.9	32.3	26.3
Object-Aware Scoring	2.9	2.2	1.7	1.5	1.4	1.2
Fine-Grained Retrieval	87.0	87.4	87.0	87.1	87.5	87.3
Total	349.5	278.6	238.8	221.1	210.9	203.4

Table 17. Mask R-CNN & DINOv2 运行时间。

Methods	Accuracy (%)					
	t=0	t=0.1	t=0.2	t=0.3	t=0.4	t=0.5
AirRoom-MaskRCNN-ResNet	92.70	92.68	92.58	92.59	92.22	92.15
AirRoom-MaskRCNN-DINOv2	87.67	87.62	87.10	87.20	87.24	87.09

Table 18. Mask R-CNN & ResNet / DINOv2 准确率。

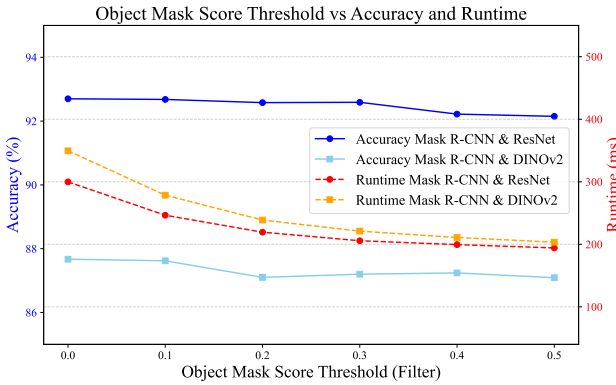


Figure 6. 随着目标掩码评分阈值的提高，AirRoom 的性能略有下降；然而，效率却显著提升。

Modules	Runtime (ms)	
	ResNet	DINOv2
Global Feature Extractor	42.5	56.2
Global Retrieval	0.1	0.1
Instance Segmentation	352.6	343.2
Receptive Field Expander	0.7	0.6
Object Feature Extractor	51.1	66.6
Object-Aware Scoring	2.2	2.1
Fine-Grained Retrieval	87.8	87.4
Total	538.5	557.6

Table 19. Semantic-SAM & ResNet / DINOv2 运行时间。

征提取器时，Table 17 中也观察到类似的趋势。此外，Table 18 显示，AirRoom 的性能在对象掩模分数阈值升高时保持基本不变，无论选择哪种对象特征提取器。这一观察在 Figure 6 中得到了进一步说明。然而，当使用 Semantic-SAM 进行实例分割时，由于 Semantic-SAM 显著较慢的性能，AirRoom 面临效率上的挑战，

Methods	Runtime (ms)	Accuracy (%)
CVNet	111.3	11.71
DINOv2	16.7	53.91
Patch-NetVLAD	100.5	64.86
AnyLoc	45.5	89.69
AirRoom	194.2	92.15

Table 20. 与当前最先进方法的运行时间比较。

具体内容见 Table 19。

Table 20 比较了不同方法的运行时。AirRoom 比 CVNet 多需要 80ms，但实现了超过 80% 的性能提升。与 Patch-NetVLAD 相比，AirRoom 的运行时大约是其两倍，性能提升超过 30%。虽然 DINOv2 完成任务需要 10-20ms，AirRoom 增加了 170ms 并提升了超过 40% 的性能。相较于 AnyLoc，AirRoom 增加了约 150ms 的运行时，但捕获了额外的 20% 性能潜力。这些结果表明，尽管在有限的改进空间内，AirRoom 仍能提供显著的性能提升，突显了其在运行时上的有效性。

目前，AirRoom 为细粒度检索分配了约 90ms，使用 LightGlue 进行特征匹配。探索更轻量 and 更快速的替代方案可以进一步提高效率。在如实时导航等实际应用中，房间重识别时间在 50-200ms 之间通常是可接受的，精度是主要考虑因素。虽然 AirRoom 比一些基线稍慢，但它实现了显著的精度提升，有效地平衡了运行时和性能。这使得 AirRoom 非常适合实际场景，能够满足现实世界的运行时要求，同时保持高可靠性和精确度。