AirRoom: Objects Matter in Room Reidentification

Runmao Yao   Yi Du   Zhuoqun Chen   Haoze Zheng   Chen Wang

Spatial AI & Robotics (SAIR) Lab, University at Buffalo

{yaorunmao, zhz19231211}@gmail.com, {yid, chenw}@sairlab.org, zhc057@ucsd.edu

Abstract

Room reidentification (ReID) is a challenging yet essential task with numerous applications in fields such as augmented reality (AR) and homecare robotics. Existing visual place recognition (VPR) methods, which typically rely on global descriptors or aggregate local features, often struggle in cluttered indoor environments densely populated with man-made objects. These methods tend to overlook the crucial role of object-oriented information. To address this, we propose AirRoom, an object-aware pipeline that integrates multi-level object-oriented information—from global context to object patches, object segmentation, and keypoints—utilizing a coarse-to-fine retrieval approach. Extensive experiments on four newly constructed datasets—MPReID, HMReID, GibsonReID, and ReplicaReID—demonstrate that AirRoom outperforms state-of-the-art (SOTA) models across nearly all evaluation metrics, with improvements ranging from 6% to 80%. Moreover, AirRoom exhibits significant flexibility, allowing various modules within the pipeline to be substituted with different alternatives without compromising overall performance. It also shows robust and consistent performance under diverse viewpoint variations. Project website: https://sairlab.org/airroom/.

1. Introduction

With the rapid development of spatial computing, room reidentification (ReID) has become a key area of interest, enabling advancements in applications like augmented reality (AR) [37] and homecare robotics [33]. It plays a crucial role in enhancing user experiences across various scenarios. For instance, on devices like the Apple Vision Pro, accurate room ReID enables smooth transitions between virtual and real-world elements. Similarly, in AR-guided museum tours, precisely identifying a user's position within specific rooms is essential for delivering location-sensitive content.

Unlike outdoor environments, where visual place recognition (VPR) methods have matured and perform reliably [2, 13, 16], indoor room ReID remains a challenging prob-

Figure 1. AirRoom leverages multi-level, object-oriented features, including global context, object patches, object segmentation, and keypoints, to perform coarse-to-fine room reidentification.

lem. A primary reason for this difficulty is the cluttered nature of indoor scenes, which are often densely packed with man-made objects [45]. These densely distributed objects often pose significant challenges to existing methods, which were originally designed for city-style and distinct structures [23]. Consequently, these methods struggle to fully capture the intricate details and varied spatial layouts of indoor environments. For instance, foundation models like DINO [9] and DINOv2 [25] can generate global descriptors that capture broad scene-level features. However, these descriptors may struggle in semantically similar environments, such as adjacent rooms with similar layouts or decorations, where distinguishable features are minimal [7]. In contrast, methods like Patch-NetVLAD [13], AirLoc [3] and AnyLoc [16] create a global descriptor by aggregating local features, which can enhance discriminative power. Yet, in indoor settings densely populated with similar and repetitive objects, these approaches may still face difficulties in distinguishing between highly similar features, reducing their effectiveness in such contexts [35].

Additionally, different from room categorization [18], which relies on identifying object types to classify spaces into semantic categories, room ReID requires accurately retrieving the same room instance from a reference database

based on a given query image. For instance, reidentifying a particular kitchen demands a combination of global functional contexts and fine-grained matching of specific object attributes. Moreover, room ReID must handle viewpoint variations, which necessitates tolerance for partial mismatches in object arrangement and appearance. These requirements often result in the failure of algorithms based solely on object categorization, as they lack the precision needed to reidentify unique room instances accurately [39].

This raises an important question: *"What kinds of object attributes are truly essential for room ReID?"* To address this, we conduct the first comprehensive study exploring multi-level object-oriented information and its impact on room ReID. As shown in Figure 1, our experiments show that all four levels of object-oriented information, *i.e.*, global context, object patches, object segmentation, and keypoints, are essential. Specifically, we find that each level plays a unique role in room ReID. Global context, such as the combination of objects like a couch and television, conveys essential semantic information for categorizing a room as a living room. Object patches provide finer details, enabling differentiation within a room, such as distinguishing a bedside table in a bedroom from a desk in a workspace. Object segmentation offers further granularity by isolating individual items, like separating a dining table from surrounding chairs to clarify the room layout. Finally, keypoints on objects, such as handles on a dresser, enhance room ReID by filtering out visually similar furniture in other rooms. Moreover, integrating multi-level object-oriented information adds robustness to viewpoint variations.

Based on these observations, we propose AirRoom, a simple yet highly effective room reidentification (ReID) system consisting of three stages: Global, Local, and Fine-Grained. In the Global stage, a Global Feature Extractor is used to capture global context features, which are then employed to coarsely select five functionally similar candidate rooms. In the Local stage, instance segmentation is applied to identify individual objects, followed by the Receptive Field Expander to extract object patches. An Object Feature Extractor is then used to obtain both object and patch features, which are utilized in Object-Aware Scoring to narrow the selection down to two candidate rooms. Finally, in the Fine-Grained stage, feature matching is employed to precisely identify the final room.

In summary, our contributions include:

- We introduce AirRoom, an object-aware room ReID pipeline with two novel modules: the Receptive Field Expander and Object-Aware Scoring, effectively leveraging multi-level object-oriented information to overcome the limitations observed in previous methods.
- We have curated four comprehensive room reidentification datasets—MPReID, HMReID, GibsonReID, and ReplicaReID—providing diverse benchmarks for

evaluating room reidentification methods.
- Extensive experiments demonstrate that AirRoom outperforms SOTAs, maintaining robust and reliable performance even under significant viewpoint variations.

2. Related Work

In this section, we review areas mostly related to our work, *i.e.*, image retrieval and visual place recognition.

2.1. Image Retrieval

Image retrieval is a fundamental and well-established task in computer vision that involves searching for images similar to a given query within a large database. The process of image retrieval typically consists of two stages: global retrieval and re-ranking. In the first stage, a global descriptor that aggregates local features is used to retrieve k candidates from a large database. This is followed by spatial verification through local feature matching to re-rank these k candidates. Early research relied on handcrafted features [5, 22], while current methods utilize deep networks to learn informative representations [8, 28].

Most image retrieval methods focus on selecting diverse relevant images to help users discover options that align with their interests or needs in real-world applications [43]. Although these methods are effective in retrieving similar images, they often lack the emphasis on distinguishing between categories or achieving precise ReID [11]. In contrast, our approach prioritizes achieving accurate ReID. Following a "global retrieval and re-ranking" pipeline, we first use global context features to identify the top five room candidates. Our object-aware mechanism then refines the search in a coarse-to-fine manner, progressively distinguishing among candidates until the most similar room is identified, yielding accurate results.

2.2. Visual Place Recognition

Visual place recognition (VPR) is often framed as a special image retrieval problem, aiming to match a view of a location with an image of the same place taken under different conditions. Previous methods fall into two categories: those that directly use global descriptors and those that aggregate local features into a global descriptor. Earlier approaches that relied on global descriptors primarily used CNN-based backbones, such as ResNet [14], to generate these descriptors. More recent methods, however, leverage foundation models like DINOv2 [25] for enhanced feature representation. In the aggregation category, early techniques employed handcrafted features like SIFT [22], SURF [4], and ORB [31]. Later advancements, including the NetVLAD series [2, 13] and AnyLoc [16], adopted learning-based models to extract feature maps and combine local features into comprehensive global descriptors.
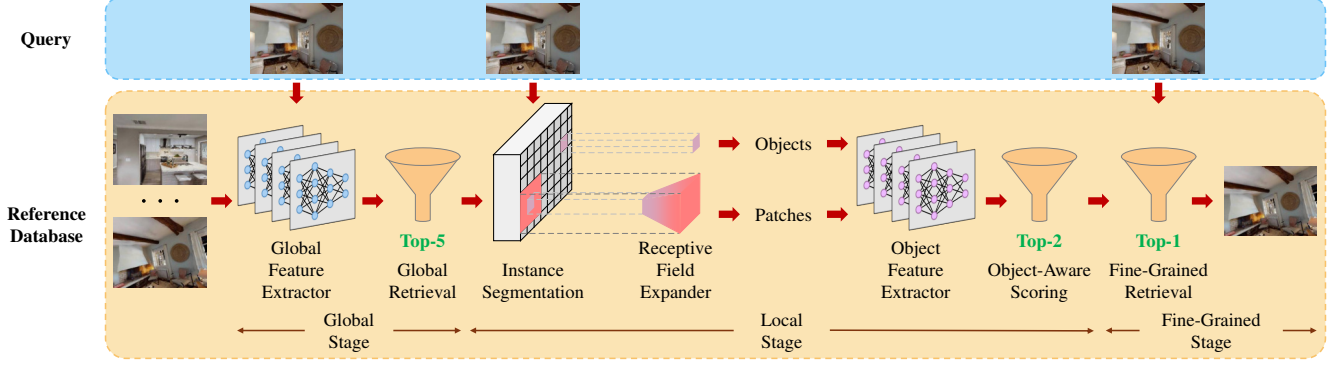
Figure 2. **The AirRoom coarse-to-fine pipeline**. The pipeline begins with the Global Feature Extractor, which captures global context features to retrieve the top-5 reference images. Instance segmentation then generates object masks, followed by the Receptive Field Expander, which extracts object patches. The Object Feature Extractor processes both object and patch features. The Object-Aware Scoring module narrows the selection to the top-2 candidates, and Fine-Grained Retrieval identifies the most suitable reference image.

However, the high performance of most VPR approaches is largely attributed to large-scale training on VPR-specific datasets [16]. Collecting extensive data for outdoor scenes is relatively straightforward due to natural variations in daylight, weather, and seasons. However, such data collection is more challenging in indoor rooms, making large-scale training on indoor datasets difficult and potentially limiting their effectiveness. Our approach effectively tackles this challenge by focusing on object-oriented feature representations, allowing us to leverage mature, pre-trained models for object feature learning. This design enables AirRoom to deliver robust performance without requiring any additional training or fine-tuning on specific datasets.

3. Proposed Approach

We propose a simple yet highly effective pipeline, Air-Room, for room reidentification that leverages multi-level object-oriented information, as shown in Figure 2. We will now systematically introduce each module of the pipeline, following the sequence of stages in which they are executed.

3.1. Global Stage

In this stage, we utilize the Global Feature Extractor to capture global context features, which are derived from the collective presence of objects within the room. These features are then used for Global Retrieval, coarsely selecting semantically similar candidate rooms from the database.

3.1.1. Global Feature Extractor

Indoor rooms exhibit fewer variations compared to outdoor environments. They lack diverse topographies, such as aerial, subterranean, or underwater features, and do not experience temporal changes like day-night or seasonal variations. Consequently, collecting large datasets for each indoor room is challenging, complicating large-scale training as seen in many VPR methods [1, 2, 13].

However, indoor rooms are inherently rich in objects,

each contributing to the room's overall semantic context. By leveraging this global context information, we can refine the reference search to specifically focus on rooms with similar semantic features to those in the query image. For this purpose, we prefer backbones pretrained on large image datasets, as they provide strong generalizability and effectively capture informative global context features [17]. Our model selections, therefore, include pretrained CNN-based models such as ResNet [14] and transformer-based self-supervised models like DINOv2 [25].

3.1.2. Global Retrieval

Using the Global Feature Extractor, we extract global context features for M query and N reference images. Let $\mathbf{Q} \in \mathbb{R}^{M \times D_g}$ and $\mathbf{R} \in \mathbb{R}^{N \times D_g}$ denote the query and reference features, respectively, where D_g is the feature dimension. The cosine similarity matrix $\mathbf{S}$ is then computed as:

$$\mathbf{S}_{ij} = \frac{\mathbf{Q}_i \cdot \mathbf{R}_j}{\|\mathbf{Q}_i\|\|\mathbf{R}_j\|}. \tag{1}$$

For each query, we select the top-5 most similar reference candidates using the following formula:

$$\text{Top}_5(\mathbf{S}_{i,:}) = \text{argsort}(-\mathbf{S}_{i,:})[:5], \tag{2}$$

where $\mathbf{S}_{i,:}$ represents the cosine similarity for the i-th query.

3.2. Local Stage

Global context features provide valuable semantic information that helps narrow down the candidate list. However, when faced with many semantically similar rooms, relying solely on global context is insufficient, and local features become increasingly essential. In this stage, we adopt a local perspective by first applying instance segmentation and the Receptive Field Expander to identify objects and patches. We then use the Object Feature Extractor to extract features from both objects and patches, followed by Object-Aware Scoring to further refine the candidate list.

3.2.1. Instance Segmentation

For each query image and its corresponding five candidates, we employ instance segmentation methods, such as Mask R-CNN [15] and Semantic-SAM [20], to identify and delineate individual objects. This process generates each object's mask and bounding box. Next, we calculate the center point c of each object using its bounding box, as shown below:

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

In this equation, x and y represent the pixel coordinates of the top-left corner of the bounding box, while W and H denote the width and height of the bounding box, respectively.

3.2.2. Receptive Field Expander

Single object information alone is not sufficiently discriminative. For example, although different desks may have distinct appearances, they can be found in both dining halls and offices. However, when an object is connected with its neighboring items—such as a desk alongside a computer, keyboard, or notebook—it suggests that the room is more likely to be an office rather than a dining hall. This insight motivates us to expand the receptive field from a single object to a patch containing multiple objects.

Given the center points of all objects in an image, we employ Delaunay triangulation [6] to generate a triangulated graph of object relationships. Specifically, Delaunay triangulation is applied to the set of object centers, ensuring that no object centers are inside the circumcircle of any triangle. This method maximizes the minimum angle of the triangles, preventing narrow, elongated triangles and ensuring more uniform object adjacency. By analyzing the adjacency relationships among the resulting triangles, we can construct the object adjacency matrix, which encodes the spatial and relational proximity of objects within the room.
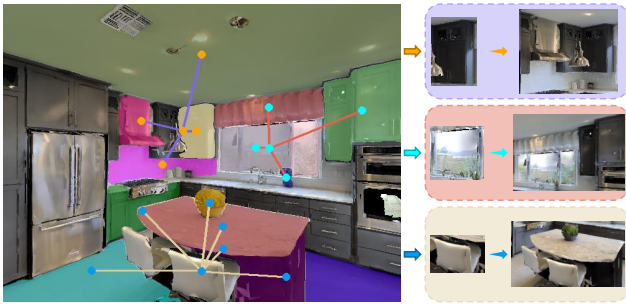

Figure 3. The Receptive Field Expander broadens the receptive field from individual objects to patches rich in contextual information. Leveraging the object adjacency matrix and each object's bounding box, it expands single objects such as a cupboard, window pane, and chair into object patches like a modular kitchen, multi-pane window, and dining set, respectively.

Given the object adjacency matrix and bounding boxes in an image, for each object, we consider the bounding boxes of its neighboring objects and enlarge the current object's bounding box to encompass all adjacent objects. This expansion increases the receptive field, enabling us to capture richer contextual information, as illustrated in Figure 3. We then apply Non-Maximum Suppression (NMS) to select the highest confidence bounding boxes, removing overlapping ones based on their Intersection over Union (IoU) scores. This results in a set of clean, informative object patches.

3.2.3. Object-Aware Refinement

The Object-Aware Refinement module is composed of three key submodules: Object Feature Extractor, Mutual Nearest Neighbors, and Object-Aware Scoring.

Object Feature Extractor To effectively leverage object patches and object segmentation information, we prioritize global features over local feature aggregation. The latter approach may fail to capture object characteristics effectively and can significantly increase computational complexity and storage demands [49]. As discussed in Section 3.1.1, we continue to rely on models pre-trained on large image datasets. Using the Object Feature Extractor, we obtain features for both query and reference patches and objects. Let $Q_p = \{\mathbf{p_i^q}\}_{i=1}^{n_{qp}}$ and $Q_o = \{\mathbf{o_i^q}\}_{i=1}^{n_{qo}}$ represent the query patch and object feature sets, respectively. For each reference image among the query's five candidates, we define the reference patch and object feature sets as $R_p = \{\mathbf{p_i^r}\}_{i=1}^{n_{rp}}$ and $R_o = \{\mathbf{o_i^r}\}_{i=1}^{n_{ro}}$.

Mutual Nearest Neighbors Given a set of query features $\{\mathbf{f_i^q}\}_{i=1}^{n_q}$ and reference features $\{\mathbf{f_i^r}\}_{i=1}^{n_r}$, we obtain feature pairs by identifying mutual nearest neighbor matches through exhaustive comparison of the two sets. Let P denote the set of cosine similarity scores for these mutual nearest neighbor matches, then we have

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

where

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

$$\mathrm{NN}_r(\mathbf{f_i^r}) = \arg\max_j \left(\frac{\mathbf{f_i^r} \cdot \mathbf{f_j^q}}{\|\mathbf{f_i^r}\|\|\mathbf{f_j^q}\|} \right), \tag{6}$$

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

By utilizing mutual nearest neighbors, we can significantly improve retrieval accuracy, simultaneously narrowing the search space and enhancing overall retrieval efficiency [50].

Object-Aware Scoring The object-aware score s is the sum of the global score s_{global} (calculated in Equation 1), the patch score s_{patch}, and the object score s_{object}:

$$s = s_{\mathrm{global}} + s_{\mathrm{patch}}(Q_p, R_p) + s_{\mathrm{object}}(Q_o, R_o). \tag{8}$$
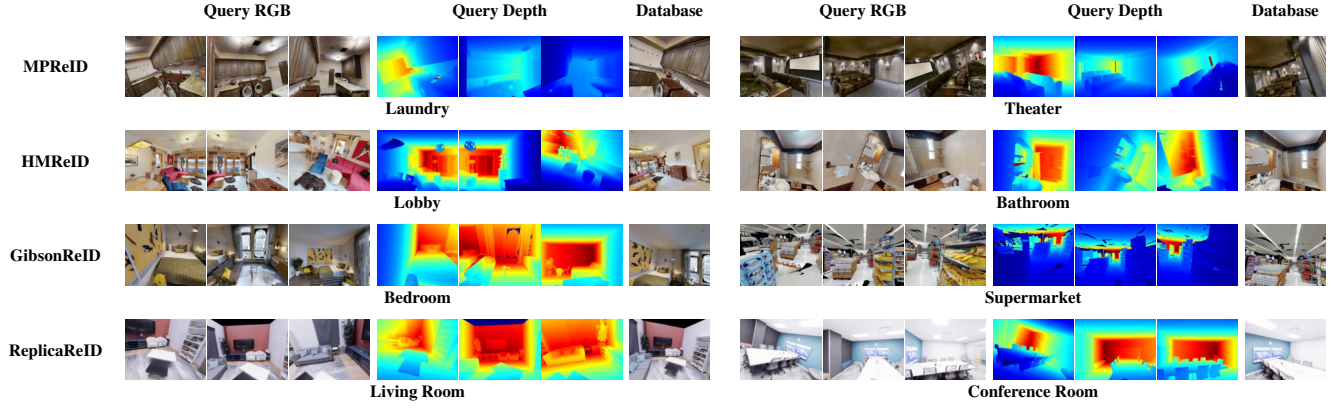
Figure 4. Illustration of four newly constructed room reidentification datasets: MPReID, HMReID, GibsonReID, and ReplicaReID. Each room provides only one reference image in the database, while query images for each room capture varied viewpoints.

Here, s_{patch} and s_{object} can either be s_{mean} or s_{max}, where

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

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

In these equations, P denotes the set of cosine similarity scores for mutual nearest neighbor matches, with Q_t representing either Q_p or Q_o, and R_t representing either R_p or R_o. The global score s_{global} serves as a prior, indicating that the initial five candidates vary in relevance. Thus, we retain this term to account for their differing levels of relevance.

Object-Aware Refinement For each query, we select the top-2 most similar reference candidates from the initial five using the Object-Aware Scoring:

$$\text{Top}_2(\mathbf{s}_i) = \text{argsort}(-\mathbf{s}_i)[: 2], \qquad (10)$$

where $\mathbf{s}_i$ is the object-aware scores for the i-th query.

3.3. Fine-Grained Stage

Patch and object features provide valuable information for understanding the room layout; however, they may be insufficient when distinguishing highly visually similar rooms, particularly in the presence of viewpoint variations and occlusions. Keypoints on objects, by contrast, exhibit strong robustness to texture and appearance variations, enabling them to effectively handle partial occlusions and reject outliers [24]. This allows keypoints to offer a more refined approach, capturing finer details for more accurate room identification. In this stage, we use Fine-Grained Retrieval to select the final top-1 result.

3.3.1. Fine-Grained Retrieval

Deep matchers, such as SuperGlue [34], perform well in visual localization tasks under challenging conditions, both

indoors and outdoors. However, they tend to face efficiency issues. In contrast, LightGlue [21] offers high efficiency without compromising matching accuracy, making it an ideal choice for our Fine-Grained Retrieval.

For each query image and its two candidate reference images, we match the query to each candidate and record the number of matching keypoint pairs. A higher number of matches typically indicates greater overlap and consistency between the features of the two images, suggesting a higher degree of similarity in their content [22]. The candidate with more matches is selected as the final result.

4. Experimental Results

4.1. Datasets

No existing indoor scene datasets are ideally suited for room reidentification tasks, as none fully satisfy the requirements. Datasets like ScanNet++ [46] and MIT Indoor Scenes [27] lack room-level segmentation, resulting in multiple rooms sharing a single scene label. The 17 Places [32] dataset includes uniquely labeled rooms but offers limited viewpoint variations, and the images are often vague. While this dataset also includes day-night changes, these are not particularly relevant for most indoor scenarios. The Reloc110 [3] dataset is likely the most suitable option; however, its quality is insufficient, with many images containing only solid-colored walls or floors due to random sampling, resulting in minimal contextual information.

Several high-quality indoor 3D datasets—such as Matterport3D [10], Habitat-Matterport3D [30], the Gibson Database of 3D Spaces [44], and Replica [38]—offer real-world indoor scenes. Building on these resources and utilizing the interactive Habitat Simulator [26, 36, 40], we created four new datasets: MPReID, HMReID, GibsonReID, and ReplicaReID, as shown in Figure 4.

Using the Habitat Simulator, we configured an agent for each room and manually selected 5 to 10 key poses to guide its exploration. The agent captured 640×480 RGB-D images from various angles, resulting in 300 to 800 images

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. Overall performance comparison between AirRoom and baseline models on four newly constructed room ReID datasets.

per room, depending on the number of key poses. However, many randomly sampled images were of low quality, often containing only walls or floors with minimal context. To address this, we carefully filtered the images for each room, retaining those that accurately represented the space and provided valuable information for room ReID.

In total, the datasets are as follows: MPReID includes 15 scenes, 105 rooms, and 16,231 RGB-D images; HMReID consists of 21 scenes, 105 rooms, and 15,781 RGB-D images; GibsonReID contains 24 scenes, 45 rooms, and 6,743 RGB-D images; and ReplicaReID includes 12 scenes, 19 rooms, and 2,862 RGB-D images.

4.2. Database Preprocess

In the room reidentification setting, we have multiple query images and a reference database. For each dataset, we select only one image per room to build the database. Specifically, for all the images of each room, we first use CLIP [29] to extract feature embeddings. Then, we apply K-means clustering with the number of clusters set to 1. The image closest to the cluster center is chosen as the reference image, as it best represents the room's visual characteristics [42].

After building the reference database, we preprocess features. First, we use the Global Feature Extractor to obtain and save the global context features. Next, we apply the instance segmentation module to segment the objects. Then, we use our Receptive Field Expander to obtain object patches and the Object Feature Extractor to extract and save the features of both the objects and the patches.

4.3. Experimental Overview

We conducted five primary experiments: overall performance comparison, group-wise performance comparison, pipeline flexibility evaluation, ablation studies, and runtime analysis. For evaluation, we used accuracy, precision, recall, and the F1 score as metrics. Per-class precision, recall, and F1-score were computed using a multi-class confusion matrix, followed by macro averaging. Accuracy was measured as the ratio of correctly matched queries to the total number of queries. A detailed runtime analysis and additional experimental results are provided in the appendix.

4.4. Overall Performance Comparison

In this section, we present a performance comparison between the best-performing version of our approach and several state-of-the-art methods, allowing us to benchmark our pipeline against established room reidentification models across different feature extraction and retrieval strategies.

We selected three categories of baseline methods: image retrieval (CVNet [19]), global descriptor-based visual place recognition (VPR) (DINOv2 [25]), and VPR using aggregated local features (Patch-NetVLAD [13] and AnyLoc [16]). Specifically, we used the Base version of DINOv2, configured CVNet with a ResNet50 [14] backbone and a reduction dimension of 2048, selected the performance version of Patch-NetVLAD, and set up AnyLoc with AnyLoc-VLAD-DINOv2 using 32 VLAD clusters.
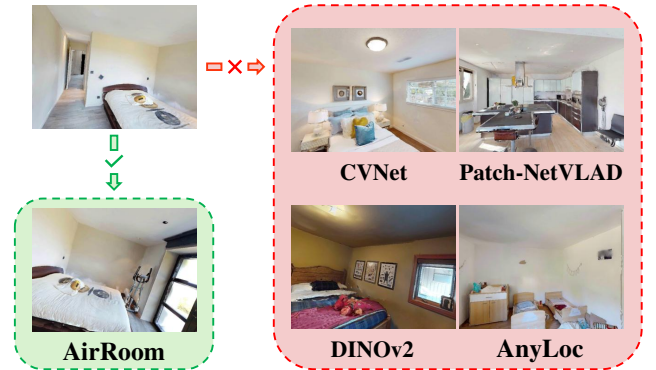

Figure 5. Given a bedroom query, AirRoom accurately retrieves the target image by leveraging object relevance for room reidentification. In contrast, CVNet retrieves visually similar images without preserving scene accuracy, DINOv2 captures semantic content but overlooks color details, Patch-NetVLAD, using aggregated local features to form global descriptors, retrieves images with mismatched semantic information, and AnyLoc considers semantic and color attributes but neglects object importance within rooms.

Table 1 presents a quantitative comparison between AirRoom and baseline methods, showing that AirRoom outperforms all baselines on nearly all metrics and datasets. In room reidentification tasks, image retrieval methods generally exhibit lower classification metrics due to their focus not being on top-1 precision, while VPR methods yield better results. Global descriptor-based VPR methods capture only high-level semantic information, often retrieving rooms with similar semantics but lacking detailed features. In contrast, VPR methods using aggregated local features, such as Patch-NetVLAD, emphasize low-level encodings but may overlook global context, resulting in less accurate retrievals. Figure 5 illustrates failure cases for CVNet, DINOv2, Patch-NetVLAD, and AnyLoc, highlighting these limitations. Although AnyLoc, known for its robust performance in "anywhere, anytime, anyview" VPR, performs

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. Group-wise performance comparison with baseline models to assess the effectiveness of the object-aware mechanism.

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. Global Feature Extractor Flexibility.

well, AirRoom further enhances performance, achieving a 20% to 40% improvement within the available margin compared to AnyLoc. For instance, AnyLoc achieves 89.69% accuracy on HMReID, leaving approximately 10% room for improvement. AirRoom, with an accuracy of 93.80%, demonstrates up to a 40% improvement within this remaining margin. These results highlight AirRoom's superior precision and refinement in room reidentification.

4.5. Group-Wise Performance Comparison

Many baseline methods adopt a "backbone + enhancement mechanism" paradigm, which our approach also follows. In this section, we compare the performance of our object-aware enhancement mechanism with that of several state-of-the-art methods, using the same backbone as each group's baseline. This setup allows us to directly assess the effectiveness of our object-aware enhancement mechanism.

For the ResNet50 backbone group, we use CVNet [19] as the baseline. In the NetVLAD backbone group, we employ Patch-NetVLAD [13] as the baseline, testing it at three reduction dimensions: 4096, 512, and 128.

Table 2 reveals that within each group, the single backbone outperforms the baseline methods that attempt to enhance performance through various mechanisms, indicating that these mechanisms do not effectively capture critical information in indoor rooms. In contrast, our object-aware enhancement mechanism significantly improves the backbone's performance by emphasizing the importance of objects in indoor environments.

4.6. Pipeline Flexibility Evaluation

In this section, we systematically evaluate the flexibility and adaptability of AirRoom by testing different configurations of its key modules. The results clearly demonstrate that AirRoom is not reliant on any specific model and can effectively integrate a diverse range of models.

4.6.1. Global Feature Extractor

We test various Global Feature Extractors, including ViT [12], DINO [9], DINOv2 [25], and AnyLoc-VLAD-DINOv2 [16] with VLAD cluster sizes of 16 and 8.

As shown in Table 3, AirRoom consistently achieves over 85% across all metrics and datasets in nearly every case, regardless of the capabilities of the Global Feature Extractor used. Even in the single exception with DINOv2, AirRoom still improves performance by nearly 15%. This demonstrates that the effectiveness of our pipeline is not reliant on any specific Global Feature Extractor, highlighting AirRoom's adaptability to various backbone configurations and underscoring its robust flexibility.

4.6.2. Instance Segmentation

We compare traditional instance segmentation methods, such as Mask R-CNN [15], with more recent approaches, including Semantic-SAM [20], which leverage advanced techniques for more granular segmentation.

Table 4 shows that AirRoom consistently outperforms the baseline by over 15%, regardless of the instance segmentation module used. This demonstrates that our pipeline is not dependent on any specific instance segmentation method, underscoring its adaptability in this component.

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. Instance Segmentation Flexibility.

4.6.3. Object Feature Extractor

We experiment with both traditional backbones, such as ResNet50 [14], and more modern backbones, like DINOv2 [25], as the Object Feature Extractor.

As shown in Table 5, AirRoom achieves substantial performance improvements over the baseline, with minimal performance variation between different Object Feature Extractors. This supports the flexibility of our pipeline in accommodating a range of feature extraction methods.

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. Object Feature Extractor Flexibility.

4.6.4. Object-Aware Scoring

We evaluate both the mean (s_{mean}) and max (s_{max}) strategies for computing the patch score (s_{patch}) and object score (s_{object}), assessing their impact on the overall performance.

Table 6 shows that AirRoom's performance remains stable regardless of the object-aware scoring method used. This underscores the robustness of object-oriented information in room reidentification and demonstrates AirRoom's flexibility in adapting to different scoring strategies.

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. Object-Aware Scoring Flexibility.

4.7. Ablation Studies

In this section, we remove certain modules from our pipeline—including the global score s_{global}, the patch score s_{patch}, the object score s_{object}, within object-aware scoring, and the entire Fine-Grained Retrieval (FGR)—to assess the importance and effectiveness of each component.

Table 7 shows that removing any module from our pipeline leads to a performance drop. However, as long as at least one module remains, our pipeline still outperforms the baseline. Table 8 demonstrates that when the Global Feature Extractor (ViT) performs well, the global score s_{global} significantly enhances performance. On the other hand, when the Global Feature Extractor (DINOv2) is less effective, the global score s_{global} has a slight negative impact, causing a small drop in performance. This result aligns with our hypothesis in Section 3.2.3, where the global score acts as a prior to rank the priority of the five candidates. Overall, these ablation studies confirm that every module in our pipeline is both important and necessary.

4.8. Limitations

While AirRoom achieves state-of-the-art performance in room reidentification under various viewpoint variations, a limitation of our work is the inability to verify robustness to indoor object rearrangements caused by movable ob-

Methods	HMReID			
	Accuracy	Precision	Recall	F1
DINOv2 (AirRoom-w/o all)	53.91	60.52	53.73	54.69
AirRoom-w/o s_{patch}	66.68	70.04	66.42	66.68
AirRoom-w/o s_{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_{patch} & s_{object}	62.26	66.43	62.03	62.46
AirRoom-w/o s_{patch} & FGR	59.39	65.25	59.14	59.97
AirRoom-w/o s_{object} & FGR	63.44	68.68	63.14	63.84
AirRoom	**70.95**	**73.86**	**70.66**	**70.78**

Table 7. Ablation Studies (Excluding Global Score Experiments).

Methods	HMReID			
	Accuracy	Precision	Recall	F1
ViT	76.47	79.37	76.04	75.91
AirRoom-ViT-w/o s_{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_{global}	**71.73**	**74.97**	**71.44**	**71.64**
AirRoom-DINOv2	70.95	73.86	70.66	70.78

Table 8. Ablation Studies on Global Score.

jects. Although our mutual nearest neighbors-based Object-Aware Scoring method is somewhat robust to such rearrangements, the datasets used in our experiments lack these cases. In contrast, recent advances in dynamic scene understanding [47] focus on recognizing scenes in the presence of moving objects, potentially offering greater robustness than our approach. Future work should consider constructing datasets that include object rearrangements and integrating new techniques to enhance robustness to movable objects, thereby improving room reidentification.

5. Conclusion

Room reidentification is a challenging yet crucial research area, with growing applications in fields like augmented reality and homecare robotics. In this paper, we introduce AirRoom, a training-free, object-aware approach for room reidentification. AirRoom leverages multi-level object-oriented features to capture both spatial and contextual information of indoor rooms. To evaluate AirRoom, we constructed four novel datasets specifically for room reidentification. Experimental results demonstrate its robustness to viewpoint variations and superior performance over state-of-the-art methods across nearly all metrics and datasets. Furthermore, the pipeline is highly flexible, maintaining high performance without relying on specific model configurations. Collectively, our work establishes AirRoom as a powerful and versatile solution for precise room reidentification, with broad potential for real-world applications.

Acknowledgments

References

[1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Mixvpr: Feature mixing for visual place recognition, 2023. 3

[2] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition, 2016. 1, 2, 3

[3] Aryan, Bowen Li, Sebastian Scherer, Yun-Jou Lin, and Chen Wang. Airloc: Object-based indoor relocalization, 2023. 1, 5

[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. 4

[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, 7

[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. 5

[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. 7

[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, 3, 6, 7

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2, 3, 6, 7

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 4, 7

[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, 3, 6, 7

[17] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better?, 2019. 3

[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. 6, 7

[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. 4, 7

[21] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed, 2023. 5

[22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 2, 5

[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. 5

[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 Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 1, 2, 3, 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, Zsolt 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. 5

[27] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420, 2009. 5

[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. 6

[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. 5

[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. 5

[33] Gabriel Sarch, Zhaoyuan Fang, Adam W. Harley, Paul Schydlo, 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. 5

[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. 5

[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 Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 5

[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. 2

[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, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Ad-*

vances in Neural Information Processing Systems (NeurIPS), 2021. 5

[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. 6

[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. 5

[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. 5

[47] Yanpeng Zhao, Yiwei Hao, Siyu Gao, Yunbo Wang, and Xiaokang Yang. Dynamic scene understanding through object-centric voxelization and neural rendering, 2024. 8

[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