ARKitSceneRefer: Text-based Localization of Small Objects in Diverse Real-World 3D Indoor Scenes

Shunya Kato[1] **Shuhei Kurita**[2] **Chenhui Chu**[1] **Sadao Kurohashi**[1]
[1]Kyoto University [2]RIKEN
{s-kato, chu, kuro}@nlp.ist.i.kyoto-u.ac.jp shuhei.kurita@riken.jp

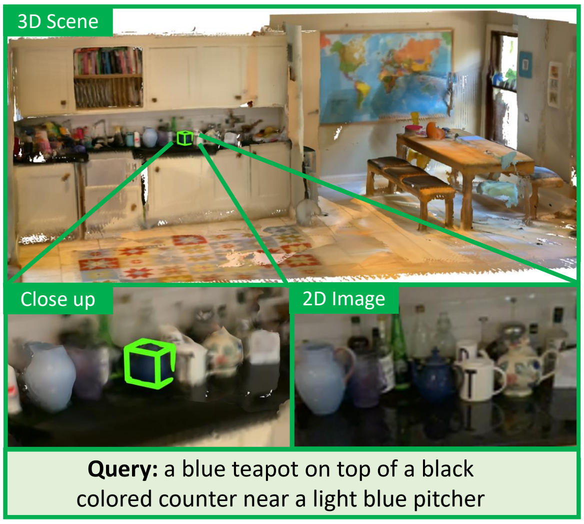
Abstract

3D referring expression comprehension is a task to ground text representations onto objects in 3D scenes. It is a crucial task for indoor household robots or augmented reality devices to localize objects referred to in user instructions. However, existing indoor 3D referring expression comprehension datasets typically cover larger object classes that are easy to localize, such as chairs, tables, or doors, and often overlook small objects, such as cooking tools or office supplies. Based on the recently proposed diverse and high-resolution 3D scene dataset of ARKitScenes, we construct the ARKitSceneRefer dataset focusing on small daily-use objects that frequently appear in real-world indoor scenes. ARKitSceneRefer contains $15k$ objects of $1,605$ indoor scenes, which are significantly larger than those of the existing 3D referring datasets, and covers diverse object classes of 583 from the LVIS dataset. In empirical experiments with both 2D and 3D state-of-the-art referring expression comprehension models, we observed the task difficulty of the localization in the diverse small object classes. ARKitSceneRefer dataset is available at: https://github.com/ku-nlp/ARKitSceneRefer

1 Introduction

3D referring expression comprehension (REC) is an essential task of understanding 3D scenes and localizing objects in scenes into easy-to-interpret text representations. It has numerous applications, such as robotics and augmented reality. Recently, sophisticated datasets have been proposed for this purpose (Chen et al., 2020; Wald et al., 2019; Qi et al., 2020). These datasets are based on object segmentations in 3D scenes and cover relatively large objects, such as furniture in indoor scenes.

However, when we develop robots that follow instructions and perform indoor household tasks, such robots are expected to find out and localize typically small objects that are required for household tasks. For example, for developing cooking robots, robots are expected to find cooking tools, foods, and ingredients. Similarly, to develop autonomous laundry robots, they need to find small objects such as clothes. We assume that finding these small objects in 3D scenes is meaningful for household robots, although these objects are difficult to be captured and often overlooked from the existing real-world scan-based 3D scene REC datasets.



Query: a blue teapot on top of a black colored counter near a light blue pitcher

Figure 1: An example of the ARKitSceneRefer dataset. We present the whole 3D scene, zoomed 3D scene, and 2D image zoomed for local structure.

In this study, we propose a new REC dataset named ARKitSceneRefer in 3D scenes in that we concentrate on the fineness and diversity of the objects referred to in instructions. ARKitSceneRefer is based on the recently proposed ARKitScenes dataset. ARKitScenes (Baruch et al., 2021) is a fine-grained photo-realistic 3D scan for diverse $1,661$ venues and $5,047$ scenes. Based on ARKitScenes, we extract small objects that are not covered in the previous 3D scene datasets well. We first apply the 2D object detector Detic (Zhou et al., 2022) with LVIS (Gupta et al., 2019) object classes for the video frames (i.e., 2D images) from which 3D

Findings of the Association for Computational Linguistics: EMNLP 2023, pages 784–799
December 6-10, 2023 ©2023 Association for Computational Linguistics

scenes in ARKitScenes are constructed. Next, we extract object labels and positions and map them to 3D scenes with ray-casting and clustering by DBSCAN (Ester et al., 1996). We confirm that most small objects are detected in 3D scenes with this approach. We then annotate referring expressions that people use to locate the objects via Amazon Mechanical Turk, while manually revising the incorrectly detected object labels. Figure 1 shows an example of our ARKitSceneRefer dataset. We finally collect $1,605$ scenes with 583 object classes and more than $15k$ objects and their corresponding referring expressions.

In addition, we conduct experiments with both 2D and 3D models to localize objects in ARKitSceneRefer. Our 2D models are based on the state-of-the-art 2D REC models MDETR (Kamath et al., 2021) and OFA (Wang et al., 2022). Our 3D models are based on an adaptation of the state-of-the-art 3D REC models ScanRefer (Chen et al., 2020) and 3DVG-Transformer (Zhao et al., 2021) on our dataset.

Our contributions are as follows: (i) creating the first object localization dataset concentrating on the small objects in daily indoor scenes upon the high-resolution 3D scene dataset of ARKitScenes, (ii) attaching more than $15k$ referring expressions with human annotations with a significantly large number of object classes, and (iii) comparisons with the state-of-the-art 2D and 3D REC models on ARKitSceneRefer.

2 Related Work

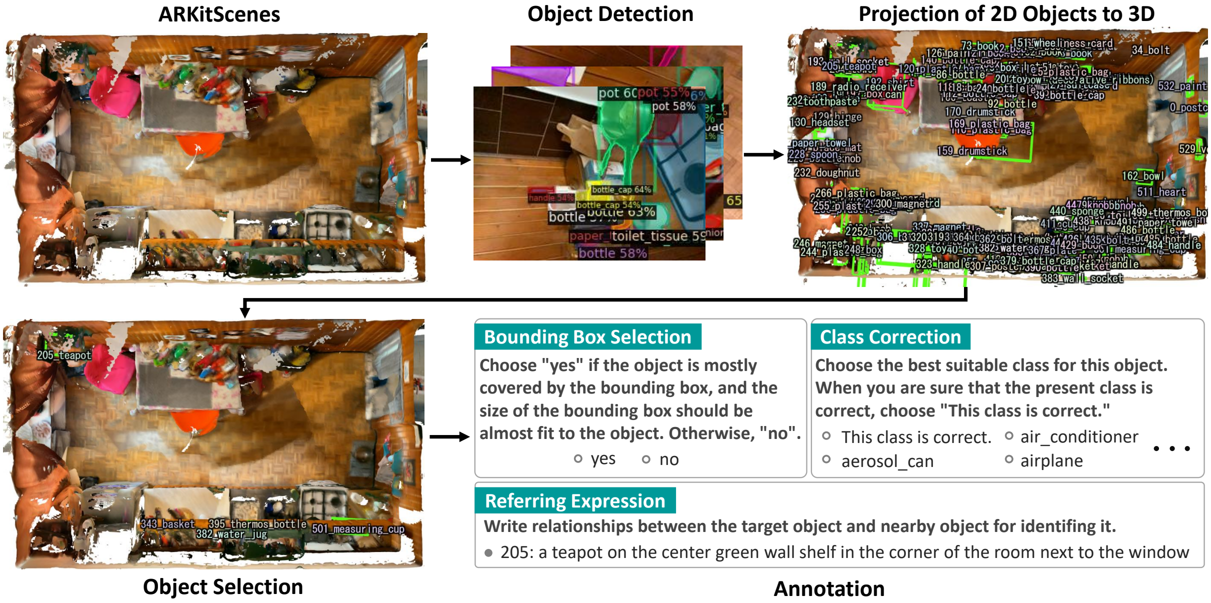
2.1 3D and Language

Recently, several photorealistic 3D indoor scene datasets (Nathan Silberman and Fergus, 2012; Song et al., 2017; Dai et al., 2017; Wald et al., 2019; Straub et al., 2019; Ramakrishnan et al., 2021; Rozenberszki et al., 2022) have been constructed. ScanNet (Dai et al., 2017) consists of $1,513$ RGB-D scans of 707 unique indoor environments with estimated camera parameters and semantic segmentation. 3RScan (Wald et al., 2019) consists of $1,482$ RGB-D scans of 478 environments across multiple time steps, including objects whose positions change over time and annotations of object instances and 6DoF mappings. ARKitScenes (Baruch et al., 2021) is the recently proposed high-resolution 3D scene dataset based on Apple's LiDER scanner. ARKitScenes consists of $5,047$ high-resolution RGB-D scans of $1,661$ unique indoor environments and provides high-quality depth maps and 3D-oriented bounding boxes.

Based on these 3D indoor-scene datasets, several language-related 3D scene understanding datasets have been proposed. For 3D visual grounding or *3D REC*, ScanRefer (Chen et al., 2020) and ReferIt3D (Achlioptas et al., 2020) have been proposed. These datasets are based on ScanNet and annotated with referring expressions for objects in 3D scenes. They are also used for the 3D dense captioning task. Similarly, the 3D question answering dataset ScanQA (Azuma et al., 2022) was proposed based on ScanNet. Yuan et al. (2022) extended 3D visual grounding to 3D phrase-aware grounding with phrase-level annotations from existing 3D visual grounding datasets (Chen et al., 2020; Achlioptas et al., 2020). Qi et al. (2020) annotated language instructions based on Matterport3D (Chang et al., 2017) and proposed remote embodied visual referring expression in real 3D indoor environments. Xu et al. (2022) proposed a large-scale 3D synthetic indoor dataset TO-Scene focusing on tabletop scenes. Unlike these datasets, our dataset focuses on a broader category of indoor small objects in real-world 3D scenes. Our dataset is more challenging because small objects are harder to recognize.

2.2 Referring Expression Comprehension

REC is the task of localizing a target object corresponding to a referring expression. In 2D REC (Kazemzadeh et al., 2014; Plummer et al., 2015; Yu et al., 2016; Mao et al., 2016), models find the target object region specified by textual referring expression in an image. Deng et al. (2021) use images with bounding boxes and queries for supervised REC. TransVG (Deng et al., 2021) is a transformer-based framework for 2D visual grounding, outperforming existing one-stage and two-stage methods. These fully supervised REC, However, depends on large annotated datasets. Weakly supervised methods (Liu et al., 2019; Sun et al., 2021) don't require manually annotated bounding boxes and unsupervised methods (Jiang et al., 2022) that require neither manually annotated bounding boxes nor queries have also been studied. Pseudo-Q (Jiang et al., 2022) proposed a method for generating pseudo queries with objects, attributes, and spatial relationships as key components, outperforming the weakly supervised



Choose "yes" if the object is mostly covered by the bounding box, and the size of the bounding box should be almost fit to the object. Otherwise, "no".
○ yes ○ no

Choose the best suitable class for this object. When you are sure that the present class is correct, choose "This class is correct."
○ This class is correct. ○ air_conditioner
○ aerosol_can ○ airplane . . .

Write relationships between the target object and nearby object for identifing it.
- 205: a teapot on the center green wall shelf in the corner of the room next to the window

Figure 2: Overview of our dataset construction pipeline. Green boxes in the 3D scene represent 3D bounding boxes.

methods.

Recently, pre-training on large vision-and-language datasets become popular in image-understanding tasks. Many existing 2D REC methods (Li* et al., 2022; Yang et al., 2022; Subramanian et al., 2022; Kamath et al., 2021; Wang et al., 2022) relied on some pre-trained models. MDETR (Kamath et al., 2021) is an end-to-end text-modulated detector derived from DETR (Carion et al., 2020) and achieved good performances on scene understanding tasks. OFA (Wang et al., 2022) is a unified sequence-to-sequence pre-trained model that unifies multiple multi-modal tasks such as image captioning, VQA, visual grounding, and text-to-image generation. OFA achieved state-of-the-art performances on several vision-and-language tasks, including the REC. In addition to 2D REC, Video-REC (Li et al., 2017; Chen et al., 2019) become a major task. First-person vision REC of RefEgo (Kurita et al., 2023) shares the similar difficulties with 3D REC problems. Both OFA and MDETR are utilized in Kurita et al. (2023).

Compared to 2D and video REC, 3D REC is an emerging task. Two-stage (Chen et al., 2020; Zhao et al., 2021; Yuan et al., 2021) and single-stage (Luo et al., 2022) methods have been proposed for 3D REC. Two-stage methods generate object proposals and then match them with the query. These methods have the disadvantage that they don't take the query into account when generating object proposals. To address this disadvantage, single-stage methods conduct language-aware key point selection. Other approaches (Chen et al., 2022; Jain et al., 2022; Wu et al., 2022) have been proposed for further improvements of the matching. D^{3}Net (Chen et al., 2022) unified dense captioning and REC in a self-critical manner. In this study, we adapt two-stage models for the proposed task.

3 Dataset

We describe the methods to construct the ARKitSceneRefer dataset in this section.

3.1 Data Collection

We construct the ARKitSceneRefer dataset based on ARKitScenes (Baruch et al., 2021), which is a large-scale 3D indoor scene dataset. ARKitScenes has comparably higher resolutions in 3D scenes, and thus it is suitable for our task that targets small object localization. Note that, in 3D scenes, some small objects often become unclear and difficult to recognize. However, most of them can be clearly detected and classified in corresponding 2D images. The performance of object detection in 2D images has been improved significantly, making it possible to find small objects in 2D images. Therefore, in this study, we detect target objects in the video frames (i.e., 2D images) where the 3D scene is constructed and then localize them in 3D scenes. Figure 2 shows our dataset construction pipeline. In the following subsections, we describe each step in detail.

3.1.1 Object Detection

In this step, we detect objects in video frames from which the 3D scene is constructed. The results of object detection are used to select target objects in the dataset. Detic (Zhou et al., 2022) is used as the object detection model. Detic is trained on both object detection and image classification datasets. With that, Detic expands the vocabulary of object detection, reduces the performance gap between rare classes and all classes in standard LVIS (Gupta et al., 2019) benchmarks, and achieves state-of-the-art performance. Detic provides more detailed class information than conventional models trained on MSCOCO (Chen et al., 2015), which is helpful for the next class selection step, and the instance segmentation corresponding to each 2D bounding box, which is helpful for obtaining 3D bounding boxes. We used pre-trained Detic in LVIS, MSCOCO, and ImageNet-21K.[1] Because the same object appears in chronologically close video frames, it is unnecessary to perform object detection on all frames. Therefore, we conduct object detection only for $1/10$ frames uniformly sampled from all frames.

3.1.2 Projection of 2D Objects to 3D

After we obtain the bounding box of objects detected by Detic, we project the position of the object point in video frames into the world coordinate of the 3D space using the provided intrinsic and extrinsic parameters of ARKitScenes. We first project the camera position for each video frame into the world coordinate with the extrinsic parameters. The position of the detected objects is then projected by the intrinsic parameters of the camera. Here, two problems exist: the distance between the camera and the object remains unknown, and the projections of the same object from multiple video frames don't always converge on a single point because of the noise of the projection parameters. It is also important to isolate bounding boxes for different objects because the Detic results often contain multiple same-class-label objects in a video frame.

Therefore, we apply the ray-casting and simple clustering-based approach for summarizing multiple projections for a single object in the 3D scene. We use the ray-casting from the camera point to the ARKitScenes mesh to obtain the first intersection of the mesh and the ray to the target object. Here we use the line from the camera point to the center of the bounding boxes as the "ray." By doing so, we obtain multiple ray-mesh intersections for each scene. We then make clusters by DBSCAN (Ester et al., 1996) to summarize the intersections and create object points in 3D space. For clustering, we make clusters for the same class-label objects. We also impose a threshold of 0.05m of distance for making a single cluster to DBSCAN to keep the resolution of small objects in scenes. As a result of clustering, we obtain 68.25 clusters in a single scene on average on the validation set. Note that we didn't use the clusters that consist of fewer numbers of intersections for further analyses and annotations.

Finally, we use 2D instance segmentations in the images corresponding to each 3D object point to assign 3D bounding boxes. Similar to obtaining 3D object points, we use ray-casting for each pixel of the instance segmentation to project it onto the 3D scene. To reduce computational complexity, ray casting is only conducted for pixels whose positions are divisible by 5. To eliminate noise, the top 5% and bottom 5% of the projected instance segmentation coordinates are removed, and the minimum and maximum coordinates within that range are considered as the 3D bounding box. Note that the 3D bounding boxes are assumed to be parallel to each axis because an object rotation is not taken into account in ARKitSceneRefer.

3.1.3 Object Selection

The object selection step consists of three sub-steps: class selection, scene selection, and target object selection. Firstly, as large objects and small objects are mixed in the results of object detection, we conduct class selection to select small objects. We choose the class of small objects based on the criteria that they can be grasped with both hands (e.g., coffee maker, laptop computer, and microwave oven). As a result of object selection, the number of object classes decreases from the Detic object detection results of $1,151$ to 794. Next, we conduct scene selection to guarantee the diversity of 3D scenes. We select one scene per room. In the same room, the scene with the largest number of objects and more than 20 objects is selected. As a result, the number of scenes decreases from $5,047$ to $1,615$. Finally, we conduct target object selection to decide on the objects to be annotated. The number of target objects is set to 20 for each 3D scene. In order to increase the number of objects in rarely detected classes, instead of choosing tar-

[1] https://dl.fbaipublicfiles.com/detic/Detic_LCOCOI21k_CLIP_SwinB_896b32_4x_ft4x_max-size.pth

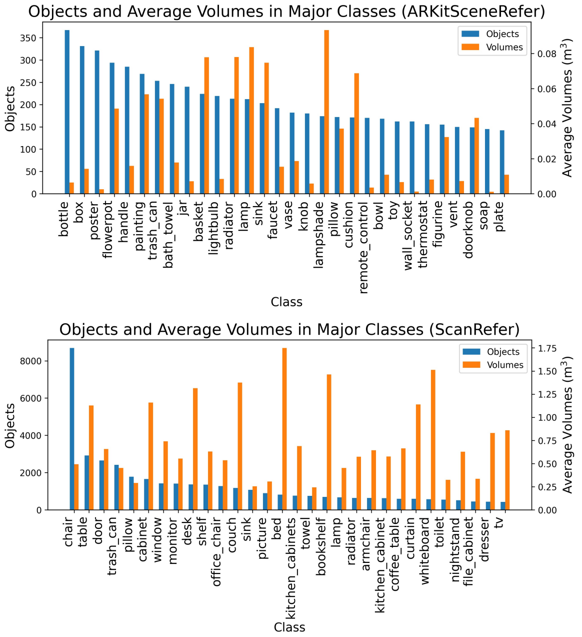


Figure 3: The distributions of the number of objects and average volumes in major 30 object classes for ARKitSceneRefer (upper) and ScanRefer (bottom).

get objects randomly, we select objects from low-frequent classes to high-frequent classes according to the detected class frequency. As a result, the number of the entire object class of ARKitSceneRefer becomes 612.

3.1.4 Annotation

With the above steps, we obtain the position information of small objects from 3D scenes without any human annotations. Then we annotate referring expressions on Amazon Mechanical Turk as the last step for constructing our dataset. Before the annotation, we conduct a qualification test and create a worker pool to ensure the quality of our dataset. In the annotation website, We present workers a 3D scene viewer with 3D bounding boxes made in Sec. 3.1.2, three images containing the target object with 2D bounding boxes detected by Detic, and the class of the target object. The images are randomly selected from video frames containing large detected bounding boxes from target objects. A 2D bounding box for the target object is shown in each image, aiming to make it easier for workers to recognize the location of the target object. The 3D bounding boxes of all objects and all target objects of each task are shown in the 3D scene so that workers can clearly recognize the target object and other objects. While referring to the 3D scene and 2D images with target objects, workers are asked to annotate referring expressions for the target objects; the referring expression for a target object must be clear enough to distinguish the target object from other objects. One task contains five target objects. Workers are restricted to countries of USA, Canada, UK, and Australia. Workers are also asked to choose 3D bounding boxes in terms of whether their position is almost correct as 3D bounding boxes sometimes become noisy. As a result, workers answered 48.15% of the 3D bounding boxes as correct. The wrong bounding boxes are excluded from the final dataset. Furthermore, as the class obtained from Detic is sometimes wrong, the workers are asked to choose the correct class in the case that the class originally shown in the interface was wrong. The class vocabulary used for class correction is also LVIS, which is the same as the one used in Detic. As a result, workers modified 40.72% classes throughout the annotation tasks, and the number of classes in our dataset became 583. Note that 4.93% small objects wrongly detected by Detic are modified by the worker to larger object classes, and thus not all annotated target objects belong to small object classes.[2]

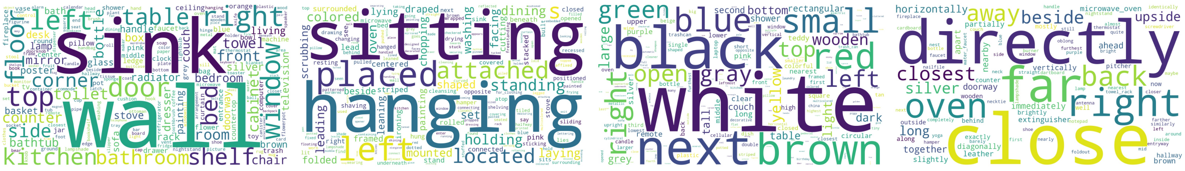
3.2 Dataset Statistics

Finally, we collected 9.69 objects on average for each scene in $1,605$ 3D scenes in ARKitScenes and $15,553$ referring expressions for these objects. Each object has a 14.43 average length of the referring expression. The referring expression covers 538 classes of indoor objects. Table 1 shows a comparison of our data with existing 3D referring expression datasets. Our dataset is significantly larger than existing datasets in two aspects, 3D scenes, and object classes. Figure 3 shows distributions of the number of objects and average volumes of each class in major 30 classes comparing our dataset with ScanRefer. We can see that ScanRefer includes many object classes for relatively large objects, such as "table" and "door." Compared to ScanRefer, ARKitSceneRefer includes many object classes for small objects, such as "bottle" and "box." Moreover, the volumes in ARKitSceneRefer are not more than $0.10\mathrm{m}^3$, while the volumes in ScanRefer are significantly greater than $0.10\mathrm{m}^3$. The distribution indicates that our dataset is successfully focused on small objects. Figure 4 shows the most commonly used words for nouns, verbs, adjectives, and adverbs classified by NLTK (Loper

[2] See Appendix A for our annotation interface.

Dataset	Environment	Objects	Expressions	Average Length	Scenes	Venues	Object Class
Sr3D (Achlioptas et al., 2020)	ScanNet	8,863	**83,572**	9.68	1,273	613	76
Nr3D (Achlioptas et al., 2020)	ScanNet	5,878	41,503	11.32	641	641	76
ScanRefer (Chen et al., 2020)	ScanNet	11,046	51,583	**20.27**	800	800	279
ARKitSceneRefer	**ARKitScenes**	**15,553**	15,553	14.43	**1,605**	**1,605**	**583**

Table 1: Overview of 3D referring expression comprehension datasets.



(a) Noun (b) Verb (c) Adjective (d) Adverb

Figure 4: Word clouds of (a) noun (b) verb (c) adjective (d) adverb for the ARKitSceneRefer. Bigger fonts mean high frequencies in the referring expressions.

Split	Expressions	Scenes	Object Class
Train	11,197	1,144	534
Val	2,732	285	363
Test	1,624	176	310

Table 2: ARKitSceneRefer dataset statistics.

and Bird, 2002). In our dataset, “wall” and “sink” are commonly used as nouns, “hanging” and “sitting” as verbs, “white” and “black” as adjectives, and “close” and “directly” as adverbs. Note that NLTK rarely fails to classify the part of speech, such as “oven” classified into adverbs. We further split our dataset into training, validation, and test sets. Table 2 shows the statistics of our dataset after the split.[3]

4 Model

Following the previous 3D referring expression studies (Chen et al., 2020; Zhao et al., 2021; Luo et al., 2022), we compare 2D to 3D REC models.[4]

4.1 2D Models

Our 2D models are based on MDETR (Kamath et al., 2021) and OFA (Wang et al., 2022), which are state-of-the-art 2D REC models. We first apply 2D REC models, which take a video frame and a referring expression as input and predict the bounding box corresponding to the referring expression in the video frame. Then the centers of the predicted bounding boxes in video frames are projected onto the 3D scene and clustered using the same method presented in Sec. 3.1.2. Finally, the center of the cluster with the most points is regarded as the center of the predicted target object on the 3D scene. Note that 2D models can’t predict 3D bounding boxes because these models don’t generate 2D instance segmentation maps.

4.2 3D Models

Our 3D models are based on ScanRefer (Chen et al., 2020) and 3DVG-Transformer (Zhao et al., 2021), which are state-of-the-art 3D REC models. We customize both ScanRefer and 3DVG-Transformer to fit to our task. Specifically, we don’t adopt the vote regression loss introduced in ScanRefer because there are no fine-grained instance segmentation labels in ARKitScenes, which means we define the object detection loss $\mathcal{L}_{\text{det}}$ as $\mathcal{L}_{\text{det}} = 0.5\mathcal{L}_{\text{objn-cls}} + \mathcal{L}_{\text{box}} + 0.1\mathcal{L}_{\text{sem-cls}}$, where $\mathcal{L}_{\text{objn-cls}}$, $\mathcal{L}_{\text{box}}$, and $\mathcal{L}_{\text{sem-cls}}$ respectively represent the objectness binary classification loss, box regession loss, and semantic classification loss, all of which are introduced in ScanRefer. Our loss function is defined as followings:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{loc}} + \beta\mathcal{L}_{\text{det}} + \gamma\mathcal{L}_{\text{cls}} \quad (1)$$

where $\mathcal{L}_{\text{loc}}$ and $\mathcal{L}_{\text{cls}}$ respectively represent the localization loss and the language-to-object classification loss, all of which are introduced in ScanRefer, and α, β, and γ represent the weights for each loss. Note that the loss function of 3DVG-Transformer are based on ScanRefer, but the weights are customized. We use the same weights introduced in ScanRefer and 3DVG-Transformer.

[3] More details are provided in Appendix B and C.

[4] See Appendix D for a formulation of the task.

5 Experiments

5.1 Evaluation Metrics

Following ScanRefer and Refer360° (Cirik et al., 2020), we employ two evaluation metrics. The first metric is IoU@k, where the predicted 3D bounding box is considered correct if its Intersection over Union (IoU) with the ground truth 3D bounding box is equal to or greater than the threshold value k. This metric has been widely adopted in existing studies on REC. We set the threshold values k to 0.05, 0.15, 0.25, and 0.5. The second metric is Dist@l, which considers the predicted 3D bounding box as correct if the distance between its center and the center of the ground truth 3D bounding box is equal to or less than the threshold value l. We use threshold values l of 0.1, 0.3, and 0.5. Note that units of IoU@k and Dist@l are percentiles.

5.2 Settings

2D Models We used MDETR and OFA$_{\text{large}}$ fine-tuned on RefCOCOg (Mao et al., 2016). We compared the following methods:

- MDETR-random and OFA-random: We randomly sampled input 1/10 video frames used to construct the 3D scene of ARKitScenes. Note that the target object may not appear in the randomly sampled video frames. If the target object is not contained in a video frame, the 2D REC models may localize irrelevant regions, leading to noises.

- OFA-Detic: This is a heuristic-based method. OFA-Detic conducted object detection on video frames by Detic, and then used only video frames that contained the detected class appearing in the referring expression. If no class is included in the referring expression, we used the same randomly sampled video frames as OFA-random. Note that as we also used Detic for dataset construction, this method is biased. We leave the comparison of using other object detectors for video frame selection as future work.

We used DBSCAN (Ester et al., 1996) algorithm for clustering. We set the maximum distance between points in a cluster to 0.02m, and the minimum number of points that make up a cluster to 1.

3D Models We used NLTK (Loper and Bird, 2002) for tokenization. We used GloVe (Pennington et al., 2014) to convert tokens in a referring expression to word embeddings. Then all word embeddings in the referring expression are concatenated and input to the 3D models. ScanRefer was trained for 200 epochs on the batch size of 32, and 3DVG-Transformer was trained for 500 epochs on the batch size of 8. The initial learning rate was $1e{-}3$, and AdamW (Loshchilov and Hutter, 2019) was used for optimization. Following ScanRefer, we applied random flipping, random rotation in the range of [-5°,5°], random scaling in the range of [$e^{-0.1}$, $e^{0.1}$], and random translation in the range of [-0.5m, 0.5m] to point clouds for data augmentation. The input features for the 3D models were xyz coordinates, colors, and normals, where the number of vertices in the point cloud was $200,000$. All experiments were conducted on 1 NVIDIA A100 GPU.

Method	Split	Dist@0.1	Dist@0.3	Dist@0.5
MDETR-random	val	7.43	13.68	17.16
OFA-random	val	7.97	14.34	17.60
OFA-Detic	val	**13.39**	**25.51**	**31.62**
MDETR-random	test	7.63	14.28	17.73
OFA-random	test	7.82	15.57	18.78
OFA-Detic	test	**13.48**	**27.95**	**35.46**

Table 3: Localization results by 2D models of MDETR and OFA.

5.3 Quantitative Analysis

Tables 3 and 4 present the evaluation results of 2D and 3D models on ARKitSceneRefer, respectively.[5]

IoU For 3D models, 3DVG-Transformer outperformed ScanRefer by a large margin. However, both of these models achieved lower performance than that on previous datasets. For example, in terms of IoU@0.25, 3DVG-Transformer achieved 45.90% on the ScanRefer validation set while only 2.21% on our validation set, which suggested that our dataset is insanely difficult compared to the existing datasets.

Dist Comparing 2D models of MDETR-random and OFA-random to 3D models for Dist@0.1 and Dist@0.3, MDETR-random and OFA-random outperformed 3D models. However, 3D models were comparable to 2D models for Dist@0.5. This is because the 3D models can make predictions based on the entire 3D scene. Even if the target object is not recognizable, the approximate position can be guessed. OFA-Detic significantly outperformed all

[5] More discussions can be found in Appendix E.

Method	Split	IoU@0.05	IoU@0.15	IoU@0.25	IoU@0.5	Dist@0.1	Dist@0.3	Dist@0.5
ScanRefer	val	2.97	1.17	0.54	0.02	1.09	9.84	18.55
3DVG-Transformer	val	**5.69**	**3.65**	**2.21**	**0.30**	**2.81**	**11.70**	**17.93**
ScanRefer	test	3.00	1.13	0.46	0.03	1.02	9.26	17.82
3DVG-Transformer	test	**6.41**	**3.54**	**2.20**	**0.41**	**2.90**	**12.82**	**19.18**

Table 4: Localization results by 3D models of ScanRefer and 3DVG-Transformer.

other methods, indicating the importance of selecting video frames that contain the target object for the 2D model.

5.4 Qualitative Analysis

Figure 5 shows the comparison of localization results of 3DVG-Transformer and OFA-Detic. In the leftmost example, both 3DVG-Transformer and OFA-Detic successfully localized the referred object. Relatively large objects that are unique in the scene were easy to localize accurately. However, in the second example from the left, only 3DVG-Transformer successfully localized the referred object. This suggests that 2D models, which rely on local information from video frames, struggled to consider the entire 3D scene simultaneously, resulting in overlooking relatively small objects. In the third example from the left, only OFA-Detic successfully localized the referred object. This suggests that 3D localizing models faces difficulties in accurately localizing quite small objects such as bottles. In the rightmost example, both 3DVG-Transformer and OFA-Detic failed to localize the referred object. This suggests that objects in complicated scenes are still difficult to localize even with current best models.

6 Conclusion

In this paper, we introduced a new 3D REC dataset, ARKitSceneRefer, for small objects. ARKitSceneRefer consists of $15,553$ referring expressions for $1,605$ scenes in ARKitScenes. We found that conventional 3D models cannot get high accuracy on our dataset. We also confirmed that the performance of the 2D models varied significantly depending on the input video frames. In the future, we plan to use the confidence scores of 2D models for image selection. We hope that our dataset will be useful in the 3D REC community.

7 Limitations

Dataset ARKitSceneRefer only provides one referring expression per object, which is less than in previous works. Additionally, some objects in the 3D scenes of ARKitScenes fail to reconstruct accurately, which is common to ScanNet, resulting in missing parts or low resolution.

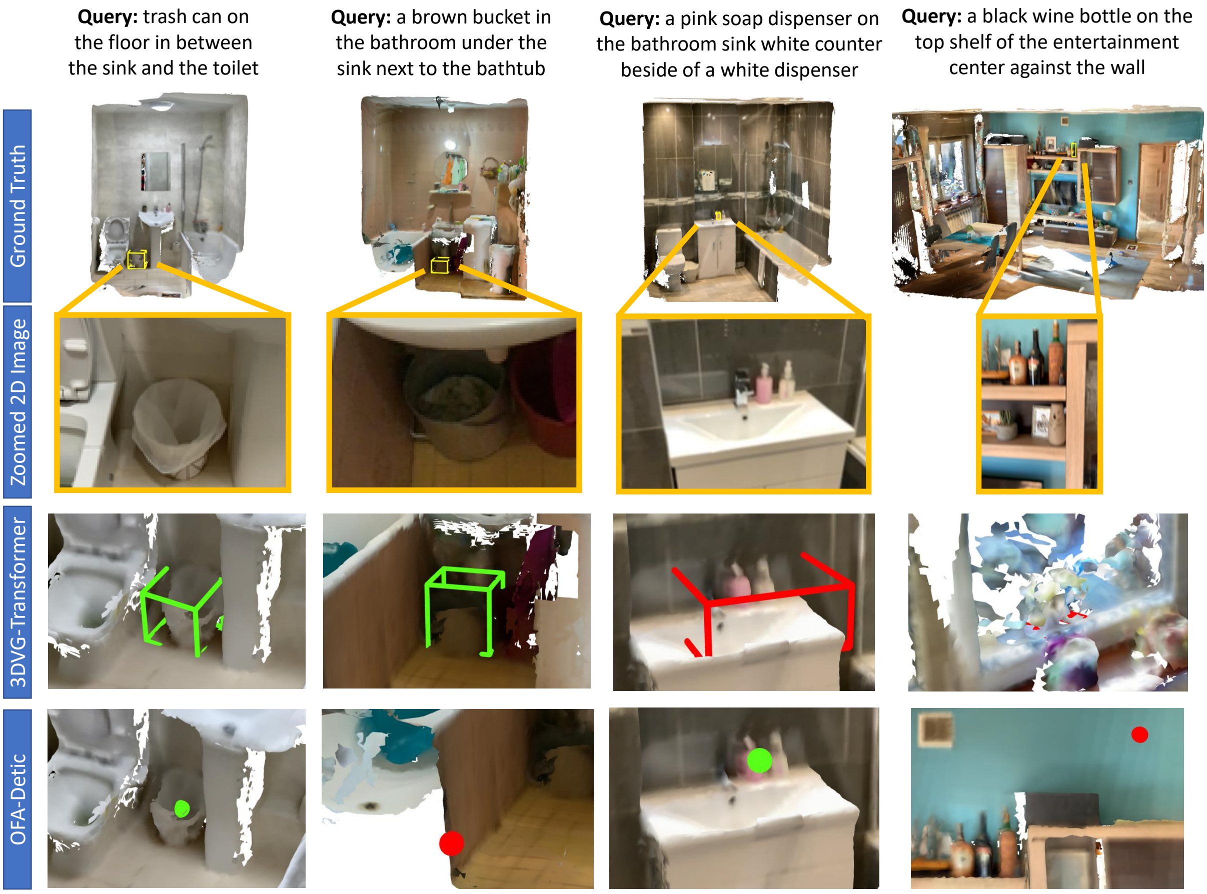
Human Annotation To ensure the quality of the dataset, we conducted a qualification test to gather highly skilled workers. However, still, subjectivity rarely leads to occasional errors in human annotations. Particularly in this paper, selecting accurate 3D bounding boxes is susceptible to such influences.

2D Models In this paper, we utilized off-the-shelf 2D models that were fine-tuned on RefCOCOg. These models already exhibit impressive performance, but we can expect further improvement on our task by fine-tuning them on our dataset. In our experiments, we employed simple heuristic video frame selection methods. It can potentially enhance accuracy to implement more optimized video frame selection methods tailored to our task.

3D Models ARKitScenes lacks semantic segmentation maps, which leads to the omission of the vote regression loss employed by ScanRefer and 3DVG-Transformer. Consequently, in our experiments, there is a possibility that the object detector is not fully optimized. However, there has been significant progress in recent research on 3D scene understanding (Peng et al., 2023; Yang et al., 2023). Leveraging these advancements to generate high-quality pseudo-labels could improve 3D model performance.

8 Ethical Statements

The annotations were conducted on Amazon Mechanical Turk, and we ensured that the workers received fair compensation, taking into account market rates. As we utilized existing data and didn't collect any newly scanned data, the workers' privacy can be well protected. The annotation process was conducted in compliance with the procedures established by our institution.



Query: trash can on the floor in between the sink and the toilet
Query: a brown bucket in the bathroom under the sink next to the bathtub
Query: a pink soap dispenser on the bathroom sink white counter beside of a white dispenser
Query: a black wine bottle on the top shelf of the entertainment center against the wall
Ground Truth
Zoomed 2D Image
3DVG-Transformer
OFA-Detic

Figure 5: Qualitative analysis of the localization results of the 3DVG-Transformer and OFA-Detic models. Yellow, green, and red represent ground truth, correct predictions, and incorrect predictions, respectively.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP23H03454 and JP22K17983, and by JST PRESTO Grant Numjber JPMJPR20C2.

References

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. 2020. ReferIt3D: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Proceedings of the 16th European Conference on Computer Vision (ECCV)*.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoki Kawanabe. 2022. Scanqa: 3d question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. 2021. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 213–229, Berlin, Heidelberg. Springer-Verlag.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. 2020. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer.

Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X. Chang. 2022. D3net: A speaker-listener architecture for semi-supervised dense captioning and visual grounding in rgb-d scans. In *ECCV*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Zhenfang Chen, Lin Ma, Wenhan Luo, and Kwan-Yee Kenneth Wong. 2019. Weakly-Supervised Spatio-Temporally Grounding Natural Sentence in Video. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. 2020. Refer360°: A referring expression recognition dataset in 360° images. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7189–7202, Online. Association for Computational Linguistics.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. 2021. Transvg: End-to-end visual grounding with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1769–1779.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Agrim Gupta, Piotr Dollar, and Ross Girshick. 2019. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. 2022. Bottom up top down detection transformers for language grounding in images and point clouds. In *ECCV*.

Haojun Jiang, Yuanze Lin, Dongchen Han, Shiji Song, and Gao Huang. 2022. Pseudo-q: Generating pseudo language queries for visual grounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. Mdetr - modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1780–1790.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar. Association for Computational Linguistics.

Shuhei Kurita, Naoki Katsura, and Eri Onami. 2023. Refego: Referring expression comprehension dataset from first-person perception of ego4d. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15214–15224.

Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. 2022. Grounded language-image pre-training. In *CVPR*.

Zhenyang Li, Ran Tao, Efstratios Gavves, Cees G. M. Snoek, and Arnold W. M. Smeulders. 2017. Tracking by Natural Language Specification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Li Su, and Qingming Huang. 2019. Knowledge-guided pairwise reconstruction network for weakly supervised referring expression grounding. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 539–547, New York, NY, USA. Association for Computing Machinery.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 2022. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16454–16463.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *CVPR*.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. 2023. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2641–2649.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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. 2021. 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*.

David Rozenberszki, Or Litany, and Angela Dai. 2022. Language-grounded indoor 3d semantic segmentation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*.

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. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*.

Sanjay Subramanian, William Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. 2022. ReCLIP: A strong zero-shot baseline for referring expression comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5198–5215, Dublin, Ireland. Association for Computational Linguistics.

Mingjie Sun, Jimin Xiao, Eng Gee Lim, Si Liu, and John Y. Goulermas. 2021. Discriminative triad matching and reconstruction for weakly referring expression grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4189–4195.

Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. 2019. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7658–7667.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23318–23340. PMLR.

Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. 2022. Eda: Explicit text-decoupling and dense alignment for 3d visual and language learning.

Mutian Xu, Pei Chen, Haolin Liu, and Xiaoguang Han. 2022. To-scene: A large-scale dataset for understanding 3d tabletop scenes. In *ECCV*.

Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. 2023. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *ECCV*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 69–85.

Zhihao Yuan, Xu Yan, Zhuo Li, Xuhao Li, Yao Guo, Shuguang Cui, and Zhen Li. 2022. Toward explainable and fine-grained 3d grounding through referring textual phrases. *arXiv preprint arXiv:2207.01821*.

Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. 2021. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1791–1800.

Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 2021. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2928–2937.

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. 2022. Detecting twenty-thousand classes using image-level supervision. In *ECCV*.

A Mturk Annotation Interface

Figure 6 present an example of our annotation interface. Workers can see 3D scenes, object classes, and 3D bounding boxes. They can also rotate and zoom 3D scenes interactively.

B Unique/Multiple Objects

One of the major challenges for 3D REC is multiple objects with the same class as the target object can appear in the same scene. Basically, it's hard to determine unique/multiple objects precisely because we sampled target objects before revising classes from Detic. Instead, we can determine unique/multiple objects on the dataset before the target object selection, which means the analysis is a little noisy. We confirmed that class-unique and class-multiple objects in the whole dataset are 61.5% and 38.5%, respectively. As a result, the model should focus on not only object words but also other descriptions (e.g., other objects and relationships) in order to achieve higher scores. Furthermore, the performances of unique and multiple objects on the test set by 3DVG-Transformer were 1.75% and 2.16% on IoU@0.25 and 12.69% and 13.00% on Dist@0.3, respectively. It might be because of the difference in the amount of the data. Specifically, we selected low-frequency objects so that our dataset could cover extensive object classes, leading to the fact that the average number of objects in the whole dataset was 94.11 for class-unique and 199.45 for class-multiple, respectively.

C Human Score on the Small Dataset

To verify the dataset quality, we randomly tested 50 samples from val and test datasets. We carefully checked whether the objects were detectable and classified them into five categories: (i) we can localize objects in 3D scenes without video frames (ii) we can localize objects in 3D scenes referring to video frames. (iii) we can't localize objects because of the ambiguity of referring expressions. (iv) we can't localize objects because of the incorrectness of bounding boxes. (v) It's hard to localize objects. We confirmed that 29 of 50 objects can be localized by referring expressions and 3D scenes. Furthermore, an additional five objects can be localized by using video frames. This result indicated that the performances of 3D REC models were much lower than those of humans. We also confirmed that 42 of 50 objects ((i)+(ii)+(iii)) are detectable, which certified the quality of small object representations. Although ARKitScenes provides high-quality 3D scenes, we didn't use them in this paper because of the lack of computational resources. Indeed, we confirmed that we would not be able to see high-quality 3D scenes on the web browser. Moreover, not only our paper but also other papers (e.g., ScanRefer and 3DVG-Transformer) adopted downsampling of 3D scenes before feeding them into models to reduce the GPU memory. We believe that future improvements in computational resources would make it possible to handle high-quality 3D scenes while we can use the same annotations of ARKitSceneRefer.

D Task

We introduce a text-based small object localization task in 3D scenes. Our task is completely different from existing 3D referring expression tasks in terms of the object size, which means existing tasks mostly focus on large objects. In our task, the input of the 3D REC model is a fine-grained 3D indoor scene and a referring expression that clearly describes the target object. 3D scenes are represented by xyz coordinates, colors, and normals. The model predicts the 3D bounding box of the target object as:

$$\mathrm{3DVG}_{small}(\mathrm{Scene}, \mathrm{Query}) = \mathrm{Box} \qquad (2)$$

where $\mathrm{Box} \in \mathbb{R}^6$ represents the xyz coordinates, height, width, and depth of the 3D bounding box.

E Discussion

Number of Points on 3D Scenes When the 3D models localize small objects, the number of vertices in the point cloud should be a very important parameter compared to localizing large objects. As shown in Table 5, we investigated how the number of vertices in the point cloud affects the performance. We used 3DVG-Transformer because its performance was better than ScanRefer. However, the performance was comparable as we reduced the number of vertices. This is because the object detectors employed by 3D models were not optimized because of the lack of semantic segmentation maps.

3D Features Chen et al. (2020) found that using color and normal information improves performance in 3D models. Therefore, we conducted experiments to verify this claim in our task. We

Points	Split	IoU@0.05	IoU@0.15	IoU@0.25	IoU@0.5	Dist@0.1	Dist@0.3	Dist@0.5
50,000	test	6.82	4.31	2.57	0.23	2.72	**12.95**	**20.35**
100,000	test	**7.27**	**4.72**	**3.04**	**0.49**	**3.26**	12.82	18.90
200,000	test	6.41	3.54	2.20	0.41	2.90	12.82	19.18

Table 5: Comparison of different numbers of vertices in the point cloud for the 3DVG-Transformer model.

trained the model with four features: (i) coordinates (xyz), (ii) coordinates and colors (xyz+rgb), (iii) coordinated and normals (xyz+normal), (iv) coordinates, colors, and normals (xyz+rgb+normal), for the 3DVG-Transformer model. As shown in Table 6, rgb features were not effective in our task. This is because our dataset focuses on small objects and handles a wide range of object classes, making it more difficult to associate rgb features with objects. By contrast, normal features were slightly effective in our task.

Additionally, we compared 3DVG-Transformer trained with coordinates, colors, and normals (xyz+color+normal) with a model trained with coordinates, colors, normals, and multiview features (xyz+color+normal+multivew). For the reduction of computational costs, we changed the number of points, epochs, and object proposals from $200,000$ to $25,000$, from 500 to 200, and from $1,024$ to 256, respectively. As shown in Table 7, 3DVG-Transformer (xyz+color+normal) was comparable to 3DVG-Transformer (xyz+color+normal+multiview). This showed that the importance of 3D scenes might be comparable with video frames.

Upper Bound of 2D Models We focused on that MDETR and OFA achieve high accuracy in 2D REC tasks, but the performance is significantly lower in our task. Therefore, we investigated the upper bound of the OFA-based 2D model, which outperforms the MDETR-based model, as shown in Table 3. We further conducted experiments on the following two settings:

- OFA-oracle: We conducted object detection on video frames by Detic, and then used only video frames with the detected class corresponding with the one annotated in the ARKitSceneRefer.

- OFA-upper: We used only the video frames used for annotating referring expressions by crowdsourcing workers.

As shown in Table 8, OFA-oracle and OFA-upper were superior to OFA-random significantly. OFA-oracle was slightly superior to OFA-Detic because many referring expressions include the object class in themselves. The results showed that if we use video frames with target objects, the model can be further improved in our task.

Input	Split	IoU@0.05	IoU@0.15	IoU@0.25	IoU@0.5	Dist@0.1	Dist@0.3	Dist@0.5
xyz	test	6.60	4.18	2.36	0.16	2.57	12.46	18.30
xyz+rgb	test	6.19	3.90	2.51	0.34	2.62	11.09	17.78
xyz+normal	test	**7.04**	**4.33**	**2.58**	0.40	**3.43**	**14.33**	**22.24**
xyz+rgb+normal	test	6.41	3.54	2.20	**0.41**	2.90	12.82	19.18

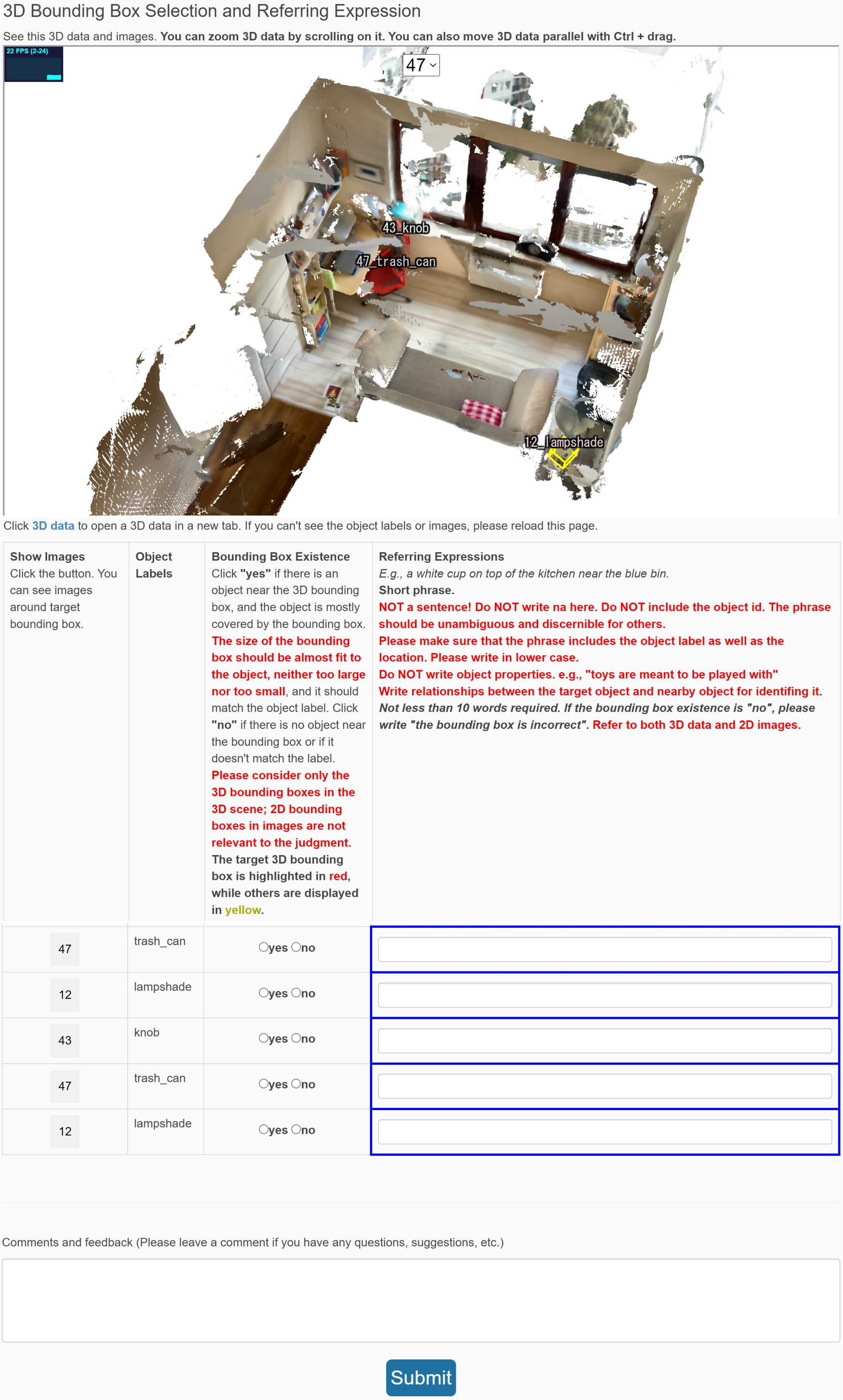
Table 6: Comparison of color and normal features for the 3DVG-Transformer model.

Input	Split	IoU@0.05	IoU@0.15	IoU@0.25	IoU@0.5	Dist@0.1	Dist@0.3	Dist@0.5
xyz+rgb+normal	test	**3.26**	**2.08**	**1.25**	**0.16**	**1.15**	**7.21**	**12.14**
xyz+rgb+normal+multiview	test	2.62	1.65	0.92	0.08	1.01	5.74	10.38

Table 7: Comparison of multiview features for the 3DVG-Transformer model. Please note that the experimental settings are different from others.

Method	Split	Dist@0.1	Dist@0.3	Dist@0.5
OFA-random	test	7.82	15.57	18.78
OFA-Detic	test	13.48	27.95	35.46
OFA-oracle	test	16.25	34.29	42.30
OFA-upper	test	**32.94**	**63.30**	**76.10**

Table 8: Oracle and upper bound results for the OFA-based 2D model.



3D Bounding Box Selection and Referring Expression

See this 3D data and images. **You can zoom 3D data by scrolling on it. You can also move 3D data parallel with Ctrl + drag.**

Click 3D data to open a 3D data in a new tab. If you can't see the object labels or images, please reload this page.

Show Images Click the button. You can see images around target bounding box.	**Object Labels**	**Bounding Box Existence** Click **"yes"** if there is an object near the 3D bounding box, and the object is mostly covered by the bounding box. **The size of the bounding box should be almost fit to the object, neither too large nor too small**, and it should match the object label. Click **"no"** if there is no object near the bounding box or if it doesn't match the label. **Please consider only the 3D bounding boxes in the 3D scene; 2D bounding boxes in images are not relevant to the judgment. The target 3D bounding box is highlighted in red, while others are displayed in yellow.**	**Referring Expressions** *E.g., a white cup on top of the kitchen near the blue bin.* **Short phrase.** **NOT a sentence! Do NOT write na here. Do NOT include the object id. The phrase should be unambiguous and discernible for others.** **Please make sure that the phrase includes the object label as well as the location. Please write in lower case.** **Do NOT write object properties. e.g., "toys are meant to be played with"** **Write relationships between the target object and nearby object for identifing it.** ***Not less than 10 words required. If the bounding box existence is "no", please write "the bounding box is incorrect". Refer to both 3D data and 2D images.***
47	trash_can	○**yes** ○**no**	
12	lampshade	○**yes** ○**no**	
43	knob	○**yes** ○**no**	
47	trash_can	○**yes** ○**no**	
12	lampshade	○**yes** ○**no**	

Comments and feedback (Please leave a comment if you have any questions, suggestions, etc.)

Submit

Figure 6: Example of the annotation interface. We provided an opportunity to indicate if they are unable to locate the object at all.