frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 04 January 2023
DOI 10.3389/frobt.2022.937772

OPEN ACCESS

EDITED BY
Kosmas Dimitropoulos,
Centre for Research and Technology Hellas (CERTH), Greece

REVIEWED BY
Nikos Grammalidis,
Centre for Research and Technology Hellas (CERTH), Greece
Sotiris Manitsaris,
Université de Sciences Lettres de Paris, France

*CORRESPONDENCE
Fethiye Irmak Doğan,
fidogan@kth.se

SPECIALTY SECTION
This article was submitted to Human-Robot Interaction, a section of the journal Frontiers in Robotics and AI

RECEIVED 06 May 2022
ACCEPTED 29 November 2022
PUBLISHED 04 January 2023

CITATION
Doğan FI, Melsión GI and Leite I (2023), Leveraging explainability for understanding object descriptions in ambiguous 3D environments.
Front. Robot. AI 9:937772.
doi: 10.3389/frobt.2022.937772

Leveraging explainability for understanding object descriptions in ambiguous 3D environments

Fethiye Irmak Doğan*, Gaspar I. Melsión and Iolanda Leite

Division of Robotics, Perception and Learning, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden

For effective human-robot collaboration, it is crucial for robots to understand requests from users perceiving the three-dimensional space and ask reasonable follow-up questions when there are ambiguities. While comprehending the users' object descriptions in the requests, existing studies have focused on this challenge for limited object categories that can be detected or localized with existing object detection and localization modules. Further, they have mostly focused on comprehending the object descriptions using flat RGB images without considering the depth dimension. On the other hand, in the wild, it is impossible to limit the object categories that can be encountered during the interaction, and 3-dimensional space perception that includes depth information is fundamental in successful task completion. To understand described objects and resolve ambiguities in the wild, for the first time, we suggest a method leveraging explainability. Our method focuses on the active areas of an RGB scene to find the described objects without putting the previous constraints on object categories and natural language instructions. We further improve our method to identify the described objects considering depth dimension. We evaluate our method in varied real-world images and observe that the regions suggested by our method can help resolve ambiguities. When we compare our method with a state-of-the-art baseline, we show that our method performs better in scenes with ambiguous objects which cannot be recognized by existing object detectors. We also show that using depth features significantly improves performance in scenes where depth data is critical to disambiguate the objects and across our evaluation dataset that contains objects that can be specified with and without the depth dimension.

KEYWORDS

explainability, resolving ambiguities, depth, referring expression comprehension (REC), real-world environments

1 Introduction

When humans and robots work on tasks as teammates, it is critical for robots to understand their human partners' natural language requests to successfully complete the task. During the task, the robot can encounter many challenges. For instance, when the robot is asked by its human partner to pick up an object, there can be misunderstandings caused by failures of speech recognition or the use of object descriptions that are unknown to the robot. Another challenge can be ambiguous requests (e.g., the human partner's object description might fit more than one object). In these cases, the robot should be able to make reasonable suggestions to its partner by using the familiar concepts in the request. For example, it should suggest the objects that fit the description instead of just saying it couldn't understand the request. While handling these challenges, the depth dimension also plays an important role for the robot. For instance, consider a robot located in the environment of **Figure 1**, helping a user pick up a described object. In this scenario, if a user asks the robot to pick up 'the mug next to the books,' it can aim to take the incorrect mug (i.e., the one in the blue bounding box) using the RGB scene because this mug is the closest to the books in 2D. Alternatively, if the robot can obtain the RGB-D scene and use the depth dimension to solve the problem, it can aim to take the correct mug (i.e., the one in the red bounding box), which is the closest to the books in 3D. Therefore, the depth dimension is critical in this scenario to understand the user's object descriptions.

People can identify objects with the help of referring expressions, which are phrases that describe the objects with their distinguishing features. In robotics, comprehending object descriptions has been studied extensively. Prior work has focused on situated dialogue systems (Kruijff et al., 2007; Zender et al., 2009), probabilistic graph models (Paul et al., 2016), and learning semantic maps (Kollar et al., 2013). Recent work on comprehending referring expressions has also employed models based on deep learning (Hatori et al., 2018; Shridhar and Hsu, 2018; Magassouba et al., 2019; Shridhar et al., 2020).

In this paper, we propose a method to comprehend users' expressions using deep neural networks' explainability in real-world, ambiguous environments. Although recent human-robot interaction (HRI) studies evaluate the importance of explainable AI for different tasks (Siau and Wang, 2018; Edmonds et al., 2019; Sridharan and Meadows, 2019; Tabrez and Hayes, 2019), to our knowledge, this is the first work using explainability to comprehend the user descriptions. Recent models on comprehension of user expressions demonstrate promising results, but they assume the target candidates in a scene are given (Magassouba et al., 2019), or these candidates can be obtained from the existing object detection (Hatori et al., 2018) or localization methods (Shridhar and Hsu, 2018; Shridhar et al., 2020). However, when robots are deployed in the real world, the encountered objects are not limited to the ones that can be detected by the state-of-art object detection or localization models, and it is not feasible to expand these models to localize every object category in a supervised fashion. Even when dealing with detectable object categories, due to environmental conditions such as poor illumination or cluttered scenes, these objects might not be possible to classify. In that case, when the described objects cannot be detected or localized, the existing solutions do not even consider these objects as target candidates. On the other hand, for a more general solution, our approach finds active areas of a scene using the explainability activations of an image captioning module, which is not trained on object-wise supervised fashion and learns a higher-level feature space. Therefore, our method does not require any detectable target candidates to suggest the described regions. This allows our system to handle various objects (including uncommon ones that may not be proposed by existing object detection or localization models) without putting any constraints on object categories or users' expressions.

In addition to focusing on limited object categories while comprehending referring expressions, most techniques in computer vision and robotics studies have relied on flat RGB images without using the depth dimension (Hatori et al., 2018; Yu et al., 2018; Magassouba et al., 2019). However, depth information plays a critical role in real-world environments, and it was recently shown that depth features could facilitate the comprehension of referring expressions (Mauceri et al., 2019). Consequently, there have been recent attempts to address this challenge using the three-dimensional feature space (i.e., 3D point clouds) (Achlioptas et al., 2020; Chen et al., 2020). Although these studies have shown promising results, in contrast to our system, they have still required candidate objects and selected the target object among the 3D object proposals. To our knowledge, our method is the first one to use explainability in RGB-D images to identify the described object regions in 3D environments.

In this work, we first find the active areas of an RGB scene using the explainability module (i.e., Grad-CAM (Selvaraju et al., 2017)), and then we use an unsupervised clustering technique (i.e., K-means) to find the active clusters. These active clusters are proposed as the regions that the robot needs to direct its attention to (**Grad-CAM RGB method** shown in **Figure 2**). Next, we extend this approach by providing the depth features in the input space and generating the RGB and depth activation heatmaps from Grad-CAM. Then, we obtain the combined activations showing the areas that are active in both of these heatmaps and cluster the combined activations (**Grad-CAM RGB-D method** shown in **Figure 5**). Our results show that the regions suggested by the Grad-CAM RGB method can be useful for resolving ambiguities. Moreover, compared to a state-of-the-art referring expression comprehension model

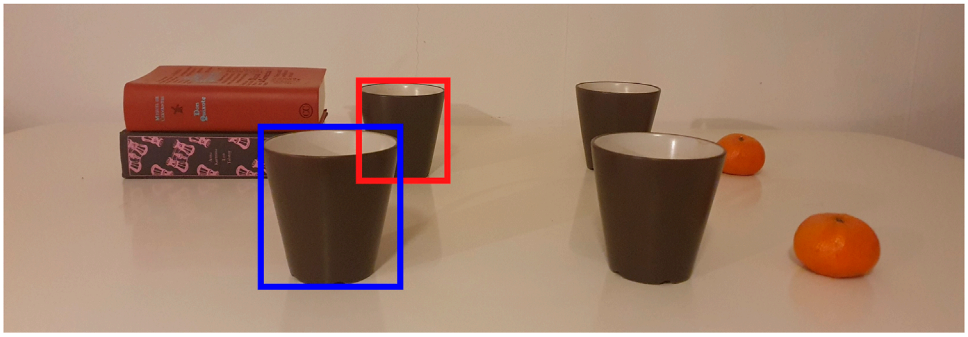

FIGURE 1
An example illustrating the motivation behind using depth to improve referring expression comprehension. In this example, when the user's object description is 'the mug next to the books', the robot can suggest the mug in the blue bounding box in RGB or the one in the red bounding box in RGB-D. Best viewed in color.

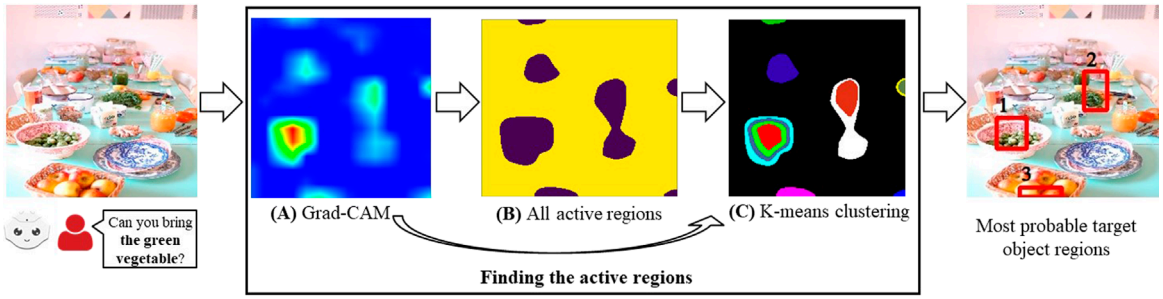

FIGURE 2
Overview of the Grad-CAM RGB method to find the described object regions for a given RGB scene and a referring expression (the bold part of the user's expression corresponds to the referring expression). The heatmap generated by Grad-CAM in **(A)**, all active regions in **(B)**, and the results from K-means clustering in **(C)**.

(i.e., MAttNet (Yu et al., 2018)), the Grad-CAM RGB method performs better in the scenes where several objects match the same description (e.g., multiple similar fishes) and where there are uncommon objects typically not recognized by off-the-shelf object detectors (e.g., an artichoke). Finally, we show that depth features employed by the Grad-CAM RGB-D method further enhance the performance in the scenes where the object descriptions are dependent on the depth dimension and in the evaluation dataset containing depth-dependent and independent features.

1.1 Contributions

Our contributions in this work can be summarized as follows:

- We propose using the explainability of image captioning to improve the effectiveness of referring expression comprehension (**Grad-CAM RGB Method**). To our knowledge, this is the first work employing explainability for comprehending user expressions to direct robots to described objects in the wild, without any restrictions such as detectable or localizable objects.
- We extend Grad-CAM RGB Method to take the depth dimension as an input, and we identify the target object regions from RGB-D images (**Grad-CAM RGB-D Method**). To our knowledge, our work is the first one comprehending referring expressions considering the depth of the objects using explainability.
- We examine the regions suggested by the Grad-CAM RGB method to determine whether these regions can be used for asking for clarification to resolve ambiguities.
- We compare the Grad-CAM RGB method with a state-of-the-art baseline in varied real-world images and show that our method performs better in challenging environments (i.e., scenes with uncommon and similar objects), which robots will more likely encounter in the real world.

- We show that using the depth dimension in the Grad-CAM RGB-D method improves the performance in scenes where the target objects are described with the spatial relations dependent on the depth features and in the whole evaluation dataset, which contains object features both dependent and independent of the depth dimension.

2 Background

2.1 Referring expression comprehension

In human-robot collaborative settings, referring expression comprehension (REC) is crucial, where the physical nature of the interaction allows humans to expand their natural language expressions with visual cues. Recent advances have taken advantage of this characteristic to improve the comprehension of referring expressions from humans (Mao et al., 2016; Yu et al., 2018; Magassouba et al., 2019; Kamath et al., 2021). In addition, it is common that the expressions given by humans are ambiguous and difficult to interpret by the robot, where interactive methods for the robot to clarify them have been shown to improve the success rate of comprehension (Hatori et al., 2018; Shridhar and Hsu, 2018). However, these methods are usually limited to the pre-learned object categories of the vision algorithm. The work from Shridhar et al. (2020) avoided the use of predefined object categories, but it was still restricted to the target candidates obtained from the DenseCap object localization module (Johnson et al., 2016). Hence, we present a novel approach employing explainability to solve the task of comprehending referring expressions that removes the dependency on using an object detection module that limits the results to the learned object categories.

2.1.1 Spatial referring expressions

It is common that referring expressions contain relational concepts between multiple entities in the scene, and its exploitation has been shown to improve the capability of the models to comprehend those expressions (Zender et al., 2009; Nagaraja et al., 2016; Hu et al., 2017; Shridhar et al., 2020). In particular, these relationships tend to be spatial relations from the point of reference of the user and the robot must be able to cope with this kind of descriptions in order to resolve any ambiguities there might be to eventually identify the right entity in the scene (Ding et al., 2021; Venkatesh et al., 2021; Roh et al., 2022). Ding et al. (2021) present a transformer-based architecture combining the language features with a vision-guided attention framework to model the global context in a multi-modal fashion. Nagaraja et al. (2016) provided CNN features to LSTMs to model spatial relationships between a region and its context regions. Shridhar et al. (2020) proposes a two-stage approach, first generating descriptions of the candidate objects and then finding the best match with the object in the expression.

2.1.2 Using depth for REC

While identifying the spatial relationships among objects, depth information has been shown to improve the task performance (Birmingham et al., 2018). Consequently, studies on referring expression comprehension have also focused on resolving this problem in three-dimensional feature space (Yuan et al., 2021; Thomason et al., 2022). For instance, 3D Point Clouds were used as an input to select the target objects among the detected object candidates (Chen et al., 2020) or segmented 3D instances (Achlioptas et al., 2020). Further, Mauceri et al. (2019) proposed an RGB-D dataset with referring expressions and evaluated this dataset with proof-of-concept experiments. In their experiments, they modified the referring expression generation model of Mao et al. (2016) to take the depth dimension as an input in addition to RGB features. They also used this generation method for comprehension by maximizing the probability of generating the input expression for candidate bounding boxes. Their findings showed pioneering results for our work: additional depth features enhanced the model's performance. However, their method assumed that the candidate bounding boxes were given or could be obtained by object box proposal systems, but our method does not require any candidate proposals thanks to leveraging explainability of image captioning activations.

2.2 Explainability

Explainability has been claimed to offer a viable solution to make intelligent systems more fair and accountable (Barredo Arrieta et al., 2020). There have been several techniques presented in the academia to make machine learning models to be more interpretable (Guidotti et al., 2018), and they have been curated to the variety of existing models, e.g., classifiers (Ribeiro et al., 2016), image captioning (Selvaraju et al., 2017), natural language processing (Alonso et al., 2021), and reinforcement learning (Madumal et al., 2020). There are multiple uses that explainable systems may have, depending on the step of the development and deployment cycle to which the explainability is being leveraged. For instance, explanations may be useful for developers in order to debug their models and be able to understand better their functioning to correct them in the best way possible (Kulesza et al., 2015), for field experts using AI-based systems e.g. to aid in medical diagnosis (Watson et al., 2019), or as integrated part to improve a system's performance (Hendricks et al., 2018), but also it is crucial for consumers of the technology to understand how the systems work e.g. in bank loan applications and the 'right to explanation' from the latest data privacy standards (Adadi and Berrada, 2018).

The broad audience of the field has caught the attention of researchers from a variety of areas that raised concern about the viability of the current explainability solutions to be usable for the general public (Abdul et al., 2018; Wang et al., 2019), stating a clear mismatch between the technical advances and the appropriate practices in the way explanations are presented to the users (Miller et al., 2017). Miller (2019) established specific lines of investigation based on research from the social sciences on explanation that could help make Explainable AI (XAI) systems to be more human-centered, and several works have used it as foundation to study explainability in applications closer to the end-user with an effort to understand their preferences (Ehsan and Riedl, 2020; Liao et al., 2020). In this work, we present a novel use of explainability that enables the robot to act in a more human-centered way when recognizing users' expressions of its surroundings.

2.2.1 Explainability in human-robot interaction

The embodiment and social factors of HRI add a new dimension to the importance of designing and using explainability with a human-centered approach for robotic applications (Han et al., 2021). The physically embodied nature of robots give them the capacity to expand the interaction to different levels, by using social cues and multiple modalities to convey their explanations. Although there have been advances in explainability techniques that make use of multimodal explanations combining visual approaches with text (Park et al., 2018), currently the majority of explainable embodied agents do not take advantage of it, and most researchers opt for using only lexical utterances for the robot to deliver its explanations (Wallkötter et al., 2021).

Explainability has been shown to be an important tool to use in human-robot collaboration settings. For instance, during an interactive robot learning scenario, explainability may help the human teacher to make better decisions based on the robot's explanations (Chao et al., 2010; Edmonds et al., 2019), and increase the predictability of the robot's actions to facilitate collaboration between humans and robots on a shared task (Tabrez and Hayes, 2019). Other examples in the HRI community use explainability for non-experts to understand the causes of unexpected failures in robotic systems (Das et al., 2021). We want to contribute to this body of work by leveraging explainability in specific robotic applications.

2.2.2 Using explainability for advancing the System's functioning

Explainability has also been used for advancing the systems' functioning (Ross et al., 2017; Hendricks et al., 2018; Li et al., 2018; Selvaraju et al., 2019). Recent computer vision studies have demonstrated the potential of interpretability to expand the use of explainability beyond the original concept of transparency by using explanations to improve models' intrinsic functioning. For instance, Selvaraju et al. (2019) aligned the visual explanations obtained from Grad-CAM (Selvaraju et al., 2017) with the human attention heatmaps to improve task accuracy in image captioning and visual question answering tasks. Hendricks et al. (2018) used a similar approach to force a captioning model to generate gender-specific words based on the person region of the image instead of the biased reasons given by gender-stereotyped datasets. Similarly, Ross et al. (2017) improved model generalization by constraining explanations with input gradient penalties.

In other domains, human attention maps have been aligned with the explanations provided by Grad-CAM to improve visual grounding in vision and language tasks (Selvaraju et al., 2019). Further, Li et al. (2018) presented a method to generate more accurate explanations (i.e., attention maps) through supervision in an end-to-end fashion while training the network. In line with enhancing the intrinsic functioning, our work leverages explainability to improve human-robot collaboration, using Grad-CAM (Selvaraju et al., 2017) saliency maps to direct the robot's attention to the appropriate regions described by the user.

3 Proposed method

In this section, we present our method finding the described regions from RGB scenes (Grad-CAM RGB method), and also how we extend this approach to also consider the depth features (Grad-CAM RGB-D method).

3.1 Grad-CAM RGB method

For a given RGB scene and a referring expression provided by a human using natural language, we aim to find the bounding boxes that show the described objects. To achieve this, we first use Grad-CAM (Selvaraju et al., 2017) to find the active areas in the scene, and then we use unsupervised clustering to find different clusters in these active areas. From these active clusters, we generate the bounding boxes most likely to belong to the target object regions (**Figure 2**).

3.1.1 Obtaining heatmap activations

We use the image captioning module of Grad-CAM (Selvaraju et al., 2017) to find active areas of a scene. The module takes a scene and an expression as an input, and it generates a heatmap $\mathcal{H}$ as an output. This heatmap shows the relevant regions in the scene. In order to obtain the heatmap, the module uses the pre-trained NeuralTalk2 image captioning model (Karpathy and Fei-Fei, 2015) and finds the gradient of the

caption's log probability with respect to the final convolutional layer. Then, the module uses these gradients to provide visual explanations.

When different captions are provided for the same image, different areas become active depending on the items in the captions (e.g., different objects). In our work, these captions correspond to referring expressions, and we find the active areas specified by the referring expression (**Figure 2A**).

Using the NeuralTalk2 image captioning model with Grad-CAM has the advantage of not being restricted to specific object categories. We achieve this because the NeuralTalk2 method was trained on a dataset (i.e., MSCOCO (Lin et al., 2014) with five captions per image collected from crowd workers) that describes scenes with many different features, not restricted to object categories. Thanks to varied scene descriptions encountered during the training of NeuralTalk2, when an object category is unknown (i.e., not in MSCOCO object categories), the higher-level feature space learned by NeuralTalk2 and visualized by Grad-CAM can be used to show the active areas that fit the given description. For instance, in **Figure 3**, when the expression is 'the blue sky', the highlighted region of Grad-CAM shows the sky, although the sky is not in the object categories of the MSCOCO. In that case, the color information is helpful for NeuralTalk2 to determine what to search for in the image. In this example, the existing works that first detect the candidate objects and select the target object among these candidates fail if they do not detect the sky, which is typically not recognized by off-the-shelf object detectors. On the other hand, by using the Grad-CAM activations of the NeuralTalk2 captioning method, we can consider the sky as a candidate region using the additional features given in the object description.

3.1.2 Clustering heatmap

After finding the active areas in a scene, we aim to cluster them. These clusters can be interpreted as different regions belonging to candidate objects so the robot can direct its attention to the right part of the scene. To achieve this, we first find the total number of active areas in the heatmap and use this value to determine the number of the resulting active clusters. Consequently, we use K-means clustering to identify those clusters.

3.1.2.1 Finding the number of clusters

In order to determine the number of clusters in K-means clustering, we find the number of unconnected areas that are active in the heatmap $\mathcal{H}$. We first define a set $\mathcal{U}$ where its values are 1 for active pixels and 0 otherwise:

$$\mathcal{U} = \left\{ |p_r > T_h \text{ or } p_g > T_h|, \quad \forall p \in \mathcal{H} \right\}, \tag{1}$$

where $|.|$ sets the value as 1 when the condition is correct and 0 otherwise. Additionally, p_r and p_g show the normalized intensity values of each pixel p for the red and green channels. We set the threshold T_h as 0.9 to only consider the pixels with high activation. A smaller value of this threshold can drastically increase the number of clusters by considering low-activation areas. With our formulation, $\mathcal{U}$ corresponds to all active areas in the heatmap. The visualization of $\mathcal{U}$ can be seen in **Figure 2B**.

After finding all active areas $\mathcal{U}$, we compute the number of unconnected ones to determine the number of clusters. To this end, we consider the 2D connectivity of pixels. Concretely, two pixels are considered neighbors if they have horizontal, vertical, or diagonal connectivity, and their activations are the same (i.e., either 0 or 1). While computing the number of unconnected

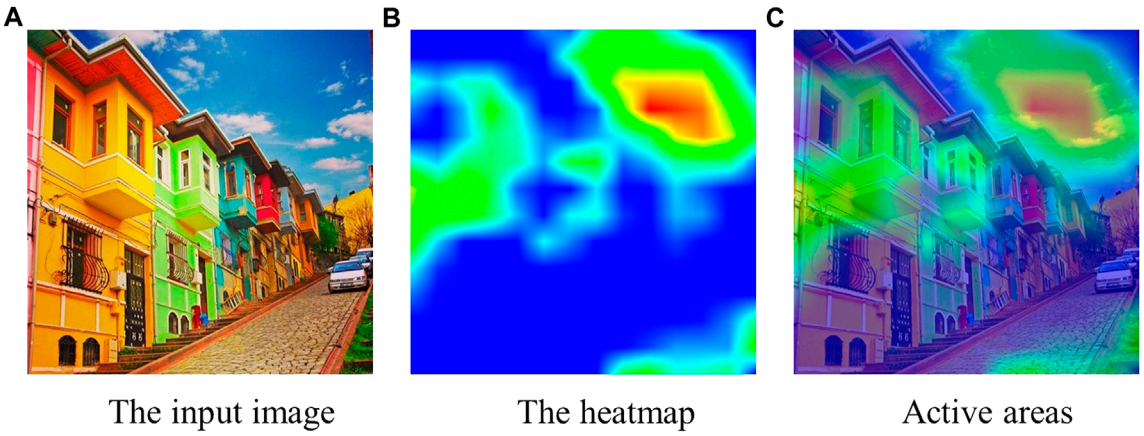

FIGURE 3
The input image (in **(A)**), the heatmap from Grad-CAM (in **(B)**), and the activations aligned with the original image (in **(C)**) when the expression is "the blue sky." The Grad-CAM heatmap highlights the sky using the color features, although the sky is not in the object categories of the MSCOCO dataset.

areas, we discard an area if it is very small (experimentally set as less than 150 pixels), and we consider the background to be another region. The calculated number of unconnected regions, n, is provided as the number of clusters for the K-means clustering algorithm.

3.1.2.2 Using K-means clustering

For some activations in heatmaps, it can be challenging to determine whether close active areas belong to the same cluster. In these cases, the neighboring method explained in Section 3.1.2.1 can not be directly applied to separate the active regions into different clusters. For instance, in Figure 4, it is not possible to determine which active area belongs to which cluster by only checking their connectivity, given the activations of different regions overlap. To address this problem, we use K-means clustering.

In order to cluster each pixel p, we consider the following features: $f(p) = \{p_x, p_y, p_r, p_g, p_b\}$. In our formulation, p_x and p_y are the normalized horizontal and vertical coordinates of pixel p. p_r, p_g and p_b represent the normalized intensity values of the red, green and blue channels.

First, we apply a Gaussian filter to the heatmap $\mathcal{H}$ to smooth the image. The Gaussian kernel's width and height are set as 11, and the smoothed image is represented as $\mathcal{H}_g$.

We define another set $\mathcal{W}$ such that every element in $\mathcal{W}$ corresponds to a pixel p and contains $f(p)$ if p is active or zeros if p is inactive:

$$\mathcal{W} = \left\{ \| p_r > T_m \text{ or } p_g > T_m \|, \quad \forall p \in \mathcal{H}_g \right\}, \tag{2}$$

where $\|.\|$ sets the value as $f(p)$ when the condition is correct, and 0s otherwise. We set threshold T_m as 0.5 because we do not need to consider regions with low activation.

After finding the number of clusters, n, and features for each pixel in $\mathcal{W}$, we cluster $\mathcal{W}$ using the K-means algorithm. The centroids of the clusters are initialized randomly, and they are updated by minimizing the within-cluster sum-of-squares. The maximum number of iterations for the algorithm is set to 300.

After obtaining the clusters from the K-means algorithm, we check whether there are unconnected regions within the same cluster. If a cluster has unconnected regions, we separate these regions into different clusters using 2D neighboring connectivity, as described in Section 3.1.2.1. Also, we discard a cluster if it is too small (< 150 pixels). Therefore, the total number of clusters can be different than the n value.

We represent all of the obtained clusters as C and each cluster in C as c_i–see Figure 2C for visualization of C. We calculate the activation of each cluster $c_i \in C$ using the channel intensities in $\mathcal{H}$:

$$a_{c_i} \leftarrow \frac{1}{n_{c_i}} \sum_{\forall p \in c_i} \left(w_r \times p_r + w_g \times p_g \right), \quad \text{for } c_i \in C, \tag{3}$$

where p_r and p_g are the normalized red and green channel intensities in $\mathcal{H}$, and n_{c_i} represents the number of pixels in region c_i. Further, w_r and w_g are the activation weights for the red and green channels. We experimentally set w_r as 0.7 and w_g as 0.3. w_r has a higher weight than w_g because red channels reflect more about the activation in our heatmap.

After finding activation value a_{c_i} for each c_i, we sort the clusters in descending order of their activation levels. We represent these sorted clusters as C_{sorted}. For each $c_i \in C_{sorted}$, we obtain the smallest bounding boxes covering c_i. The obtained bounding boxes are represented as B_{sorted}, and we consider B_{sorted} as the candidate bounding boxes most likely to belong to the described object.

The overall procedure of the Grad-CAM RGB method is summarized in Algorithm 1 (see Supplementary Material for an alternative solution).

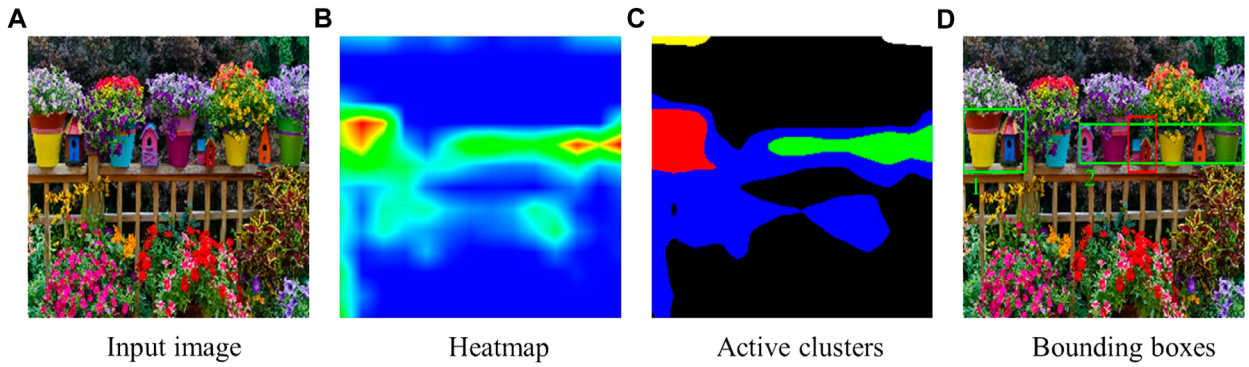

FIGURE 4
When the expression is "the red house between the pink and yellow flower pots," **(A)** the input image, **(B)** the heatmap from Grad-CAM, **(C)** the active clusters from K-means, and **(D)** the suggested candidate bounding boxes (the green ones are the candidates and the red one is the target object). The activations of different regions overlap, so the heatmap is not straightforward to cluster without using K-means clustering.

Input: an RGB scene and a referring expression.
Output: B_{sorted}, the candidate bounding boxes belonging to the described object.
Generate the heatmap $\mathcal{H}$ using Grad-CAM for the given scene and the referring expression
Set $\mathcal{U}$ to sbe the all active areas in $\mathcal{H}$ (Eq. 1)
Let n to be the number of disconnected areas in $\mathcal{U}$
Obtain $\mathcal{H}_g$ by applying a Gaussian filter to $\mathcal{H}$
Let $\mathcal{W}$ to contain the feature vectors of pixels in $\mathcal{H}_g$ (Eq. 2)
Cluster $\mathcal{W}$ using K-means clustering with n number of clusters
Set C to be the clusters obtained from K-means clustering
Calculate the activation a_{c_i} for each cluster $c_i \in C$ (Eq. 3)
Obtain C_{sorted} by sorting C in terms of the cluster activations
Set B_{sorted} to be the smallest bounding boxes covering each cluster in C_{sorted}
Provide B_{sorted} as the candidate bounding boxes belonging to the described object

Algorithm 1. Grad-CAM RGB method.

3.2 Grad-CAM RGB-D method

To obtain the described RGB-D scene regions for a given expression, we propose to extend the Grad-CAM RGB method presented in Section 3.1 with depth features. To achieve this, we first generate the activation heatmap of RGB and depth channels using Grad-CAM. Then, we find the combined activations showing the common active areas in these channels. Finally, we apply K-means clustering to the combined activations and suggest the bounding boxes covering the clusters with the highest activations as the regions belonging to the described objects. (See Figure 5 for an overview.)

3.2.1 Obtaining heatmap activations

To obtain the active parts of RGB-D scenes, we use the NeuralTalk2 image captioning module of Grad-CAM as in Section 3.1.1. The NeuralTalk2 image captioning model was trained on RGB images, but thanks to its rich feature space, the Grad-CAM activations of the captioning model can also generate useful activations for the depth dimension of the scenes. For instance, in Figure 6, heatmap activations of NeuralTalk2 in RGB image are not accurate enough to identify 'the microwave closer to the table'. On the other hand, the heatmap of the depth image forces these activations towards the described area. Therefore, in this case, using the depth heatmap together with the RGB one can help to highlight the correct areas.

After observing the depth heatmap can help to identify the areas described by a user, as in Figure 6, we provide an RGB-D scene to Grad-CAM through its RGB channels and depth dimension. Therefore, we obtain two different heatmaps, one from RGB denoted as $\mathcal{H}^{RGB}$ and another from the depth denoted as $\mathcal{H}^{depth}$. For instance, in Figure 5A, the image in the back in the first row shows $\mathcal{H}^{RGB}$ and the image in the back in the second row visualizes the $\mathcal{H}^{depth}$.

In the heatmap representation, higher intensities in the red channel show higher activations, and higher values in the blue channels denote lower heatmap activations as before. We represent each pixel's normalized RGB channel intensities as $\{p_r^{RGB}, p_g^{RGB}, p_b^{RGB}\}$ and $\{p_r^{depth}, p_g^{depth}, p_b^{depth}\}$ for $\mathcal{H}^{RGB}$ and $\mathcal{H}^{depth}$ respectively.

3.2.2 Combining RBG and depth activations

After obtaining the activation heatmaps $\mathcal{H}^{RGB}$ and $\mathcal{H}^{depth}$, we find the intersecting area of the active parts in the heatmaps. First, we check the channel intensities of each pixel for both $\mathcal{H}^{RGB}$ and $\mathcal{H}^{depth}$. When red or green channel intensities are higher than a threshold T_{rgb} (experimentally set as 0.39) for both of the pixels in $\mathcal{H}^{RGB}$ and $\mathcal{H}^{depth}$, we assume that the corresponding pixel in their intersection heatmap $\mathcal{H}^{int}$ is also active. In that case, we take the mean of each channel in $\mathcal{H}^{RGB}$ and $\mathcal{H}^{depth}$ to set the corresponding pixel intensities $\{p_r^{int}, p_g^{int}, p_b^{int}\}$ in $\mathcal{H}^{int}$:

$$p_r^{int} \leftarrow \frac{1}{2}\left(p_r^{RGB} + p_r^{depth}\right), \tag{4}$$

$$p_g^{int} \leftarrow \frac{1}{2}\left(p_g^{RGB} + p_g^{depth}\right), \tag{5}$$

$$p_b^{int} \leftarrow \frac{1}{2}\left(p_b^{RGB} + p_b^{depth}\right). \tag{6}$$

If the red and green channels of a pixel in $\mathcal{H}^{RGB}$ or $\mathcal{H}^{depth}$ are lower than T_{rgb}, we set the corresponding pixel in $\mathcal{H}^{int}$ as inactive, i.e., we set $\{p_r^{int}, p_g^{int}, p_b^{int}\}$ as $\{0, 0, 1\}$ since the highest intensity in blue channel shows an inactive pixel. The second row of Figure 5B shows an example visualization of $\mathcal{H}^{int}$.

3.2.3 Clustering heatmap

After obtaining $\mathcal{H}^{int}$ showing the activation intersection of $\mathcal{H}^{RGB}$ and $\mathcal{H}^{depth}$, we cluster $\mathcal{H}^{int}$ to find the active regions in the RGB-D scene. To achieve this, we first obtain the number of clusters and then use this number for K-means clustering to identify the active clusters.

To obtain the number of clusters n from $\mathcal{H}^{int}$, we calculated the number of unconnected areas in $\mathcal{H}^{int}$ following the procedure

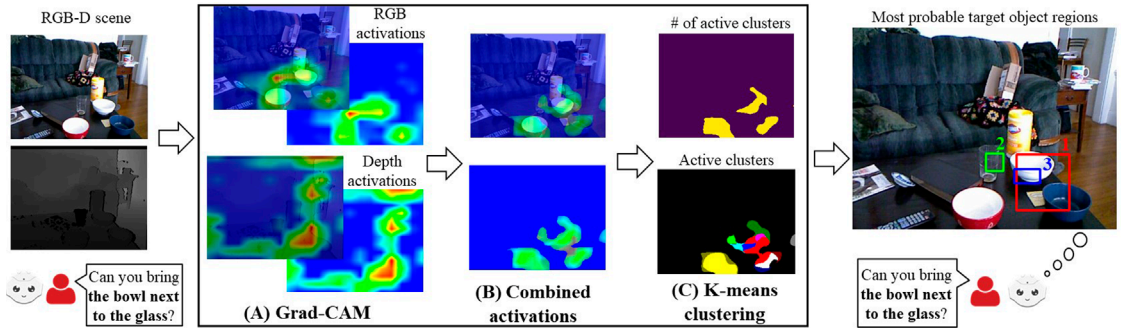

FIGURE 5
For a given RGB-D scene and a referring expression (i.e., the bold part of the user expression), the overview of the Grad-CAM RGB-D method to obtain the bounding boxes containing the target object regions. The RGB and depth heatmaps generated by Grad-CAM in **(A)**, combined activations in **(B)**, and the results from K-means clustering in **(C)**.

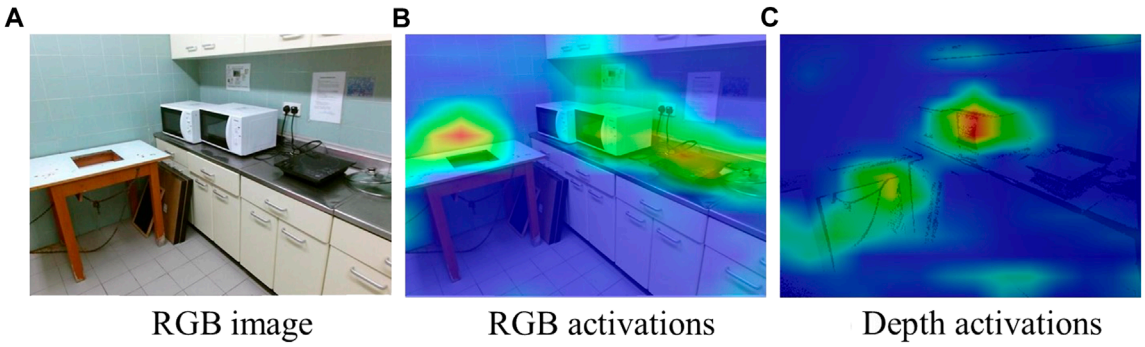

FIGURE 6
The RGB image (in **(A)**), the heatmap activations of RGB image (in **(B)**), and depth activations (in **(C)**) when the expression is "the microwave closer to the table."

explained in Section 3.1.2.1 – see the first row of Figure 5C for the visualization of number of active clusters. The computed number n is provided as the number of clusters to the K-means clustering.

After finding the cluster count n, we apply K-Means clustering to determine the active clusters. We first apply a Gaussian filter to $\mathcal{H}^{int}$ as in Section 3.1.2.2 and obtained $\mathcal{H}^{int}_g$. Then, we define a feature vector for each pixel in $\mathcal{H}^{int}_g$. After the Gaussian smoothing, if a pixel is active (i.e., the red or blue channel has a value higher than 0.5, as before), the feature vector of the pixel contains the five features descried in Section3.1.2.2 and also the depth feature:

$$f(p^{int}) = \{p^{int}_x, p^{int}_y, p^{int}_z, p^{int}_r, p^{int}_g, p^{int}_b\}, \quad (7)$$

where these features correspond to the pixel's coordinates in the x and y-axes, its corresponding depth value obtained from the input RGB-D scene, and its pixel intensities in red, blue, and green channels, respectively. All of these feature values are normalized in the zero to one range. Alternatively, if a pixel is not active after smoothing, the feature vector is set as $\{0,0,0,0,0,0\}$, as before.

Using the pixels' features and the calculated number of clusters n, we cluster the pixels of $\mathcal{H}^{int}_g$ with K-means clustering following the procedure explained in Section 3.1.2.2—see the second row of Figure 5C for the visualization of example clusters. After the K-means clustering, we calculate the activation a_{c_i} of each cluster c_i using the Eq. 3, and sort the clusters from the highest activation to the lowest. Finally, we suggest the bounding boxes B_{sorted} covering the sorted clusters as the candidate bounding boxes containing the target object.

Algorithm 2 summarizes the overall procedure of the Grad-CAM RGB-D method.

4 Experiments and results

To evaluate the Grad-CAM RGB and RGB-D methods, we conducted two sets of experiments. First, to assess the Grad-CAM RGB method efficacy, we selected a state-of-the-art referring expression comprehension method as a baseline (i.e., MAttNet (Yu et al., 2018)), gathered varied real-world images, and compared the results of both methods on these images. Then,

Input: an RGB-D scene and a referring expression.
Output: B_{sorted}, the candidate bounding boxes belonging to the described object.
1 Generate the heatmap activations $\mathcal{H}^{RGB}$ and $\mathcal{H}^{depth}$ using Grad-CAM
2 Find the heatmap $\mathcal{H}^{int}$ showing the common active areas of $\mathcal{H}^{RGB}$ and $\mathcal{H}^{depth}$ (Eqs **4**, **5**, **6**)
3 Count the number of unconnected areas (n) of active pixels in $\mathcal{H}^{int}$
4 Obtain $\mathcal{H}_g^{int}$ by applying a Gaussian filter to $\mathcal{H}^{int}$
5 Collect the feature vector of each pixel in $\mathcal{H}_g^{int}$ (Eq. **7**)
6 Find the clusters by employing K-means clustering to the feature vector with n number of clusters
7 Follow the steps between the lines 7–10 in **Algorithm 1**, and obtain B_{sorted}
8 Suggest B_{sorted} as candidate bounding boxes showing the target object regions

Algorithm 2. Grad-CAM RGB-D Method.

to analyze the impacts of depth features, we compared the Grad-CAM RGB-D method with the Grad-CAM RGB method on another dataset containing depth dependent and independent features.

4.1 MattNet baseline

For a given RGB scene and referring expression, MAttNet first obtains the candidate objects using an object detection module. Then, the method checks how well the expression fits each of the candidate objects. Finally, the candidate object that best fits the expression is considered the target object. To compare the Grad-CAM RGB method with MAttNet, we sort the candidate bounding boxes by how well they fit the expression. Similar to Grad-CAM RGB output, the bounding boxes ordered from the most likely to the least likely are considered MAttNet's candidate bounding boxes belonging to the described object.

4.2 Data collection

4.2.1 MTurk dataset

To compare the Grad-CAM RGB method with MAttNet, we gathered a dataset of 25 images containing indoor and outdoor scenes (12 images from SUN (Xiao et al., 2010), 8 images from Google Images, 4 images from Doğan et al. (2019), and 1 image from SUN RGB-D (Song et al., 2015)). These images are classified as easy (7 images), medium (8 images), and hard (10 images) difficulty levels. An image is labeled as easy if there are only a few objects in total, they are commonly known objects (e.g., bottle, book, mouse, *etc.*), and the number of same-type objects is 2 (i.e., only one distractor per object). If the objects are common, but the number of distractors is at least three per object, the image is classified in the medium category. The images in the hard group contain many objects with distractors and some objects that are not so common (e.g., radish, papaya, and artichoke). Since MAttNet uses Mask R-CNN (He et al., 2017) for extracting objects, we determine an object as common if it is part of the list of instance categories of Mask R-CNN (i.e., 90 types of objects), so a fair comparison is ensured. Next, one target object per image is annotated by a person blind to our research questions (female, 29 years old). She was instructed to draw a bounding box around an object she would consider difficult to describe.

Thereafter, we used Amazon Mechanical Turk (AMT) to collect written expressions describing the target objects in the images. We asked AMT workers to provide an unambiguous description of the target object such that it could be differentiated from other similar objects in the image and gave them some examples. We asked them to describe the objects to a robot in order to collect descriptions that simulate interactions between a user and a robot (e.g., a user requests an object from a robot). For each interaction, each user could describe an object using its various features or refer to an object in relation to other objects. For example, different AMT workers described the object in **Figure 8D** as 'the brown vegetable on the top right', 'the purple vegetable right next to the mushrooms', and 'the turnip to the right of the eggplant'. To account for this variability, we gathered 10 expressions describing the same target object in the same image. In total, we obtained 250 expressions–see **Figure 8** for some examples.

We gathered such a dataset to evaluate the Grad-CAM RGB method's performance in different conditions. The easy and medium difficulty images represent the typical computer vision datasets for referring expression comprehension (e.g., RefCOCO dataset (Yu et al., 2016) which contains MSCOCO (Lin et al., 2014) images where MAttNet and NeuralTalk2 were trained). In these scenes, the total object categories are limited (91 novel object categories for COCO images) and detectable by existing object detectors. On the other hand, in the hard category dataset, the object categories go beyond the existing datasets, and this dataset represents the scenes that can be encountered in the wild. Therefore, this three-level difficulty dataset enables us to observe the behavior of the methods in many interactions at different difficulty levels. Further, neither NeuralTalk2 nor MAttNet were trained on our collected scenes and expressions,

which helps us to better evaluate the methods' generalization capacities.

4.2.2 SUN RGB-D dataset

To compare the Grad-CAM RGB and RGB-D methods, we gathered another dataset with 70 scenes from SUN RGB-D (Song et al., 2015). This dataset contains various real-world scenes collected from different spatial contexts (e.g., living room, bedroom, bathroom, office, *etc.*). Moreover, for each scene, we selected a target object with at least one distractor (i.e., the objects that are in the same object category as the target object). Further, for each target object, we collected an expression describing the target object in a natural and unambiguous manner. In the end, we obtained a dataset with 70 images and 70 expressions referring to the target objects.

Half of our dataset (35 images) was considered to be depth independent, and the remaining half was labeled as depth dependent. In depth independent category, the target objects were described with features that were not tied to depth dimensions (e.g., the spatial relations such as 'to the left', 'to the right' or other object features such as the color or object type. In contrast, the depth dependent category images needed the depth dimension to disambiguate the target objects. Therefore, the expressions used to describe the target objects were dependent on their three-dimensional distances (e.g., the expressions contained depth-dependent spatial relations such as 'close by', 'next to', 'in front of', *etc.*) – see **Figure 10** for some example images and expressions).

We collected such a dataset because we aim to assess the impacts of using depth features for dept dependent and independent environments. The instances that we collected for this purpose enable us to manipulate the environment's depth dependence for a detailed comparison of the Grad-CAM RGB and RGB-D methods. Moreover, the equal proportion of instances for each category ensures the fair evaluation of the methods' overall performance.

4.3 Evaluation procedure

The candidate bounding boxes are obtained from the MTurk dataset using Grad-CAM RGB and MattNet, and from the SUN RGB-D dataset using the Grad-CAM RGB and RGB-D methods. The first three candidates from each method are considered for computing a matching score with the target object bounding box. To calculate the matching score, S_i, we use $1 - L_{DIoU}$, where L_{DIoU} (defined by Zheng et al. (2020)) represents the matching loss function between two bounding boxes. Therefore, S_i is:

$$S_i \leftarrow \frac{area\left(b_i \cap b_{target}\right)}{area\left(b_i \cup b_{target}\right)} - \frac{d^2}{c^2}, \tag{8}$$

where b_i is the candidate bounding box and b_{target} is the box of the target object. d represents the normalized distance between the centers of b_i and b_{target}, and c is the normalized diagonal length of the smallest box covering b_i and b_{target}.

In Eq. **8**, the first term gives a higher score for a higher intersection of the boxes, and the second term penalizes the distance between their center of masses. The matching score S_i can vary in $[-1,1]$ interval. The first of the three candidates that results in $S_i > 0$ is accepted as the candidate box showing the region belonging to the target object. In the case of all three candidates having a score lower than zero, we report it as none of the candidate boxes belonging to the target object.

In the first evaluation with the MTurk dataset, the same steps were applied to the Grad-CAM RGB method and MatNett for the 250 expressions. Both methods could find at least three candidate boxes in all cases, except for MAttNet in one instance. That image belongs to the easy category, and it was able to find the target object for the first two candidates without affecting the reported results.

In the second evaluation with the SUN RGB-D dataset, the same steps were applied to the Grad-CAM RGB and RGB-D methods for 70 expressions, and both of the methods could suggest at least three candidate boxes.

4.4 Results

4.4.1 Grad-CAM RGB vs. MattNet baseline

In this section, we present our results comparing the Grad-CAM RGB method with the MattNet baseline in the MTurk dataset for 250 expressions. **Figure 7** presents how often the target object matched with the first three candidates for all images at each level of difficulty. In **Figure 8**, we show the first candidates suggested by the two methods for the same images and target objects.

4.4.1.1 All MTurk dataset

We first compared the Grad-CAM RGB method with MAttNet for how many times the target object from the 250 user expressions matched the first, second, or third candidate bounding boxes according to the S_i score from Eq **8**. A Chi-Square test did not find any significant differences between the methods, $\chi^2\ (3, N = 500) = 4.34, p = .23$. Most often, the target object was not matched with any of the first three candidate bounding boxes proposed by the two models (i.e., the mode was "none" of the candidates for both methods). In **Figure 7A**, we can see that both methods showed similar trends for different candidates, and the number of times that the methods generated candidate bounding boxes that matched the target object were similar.

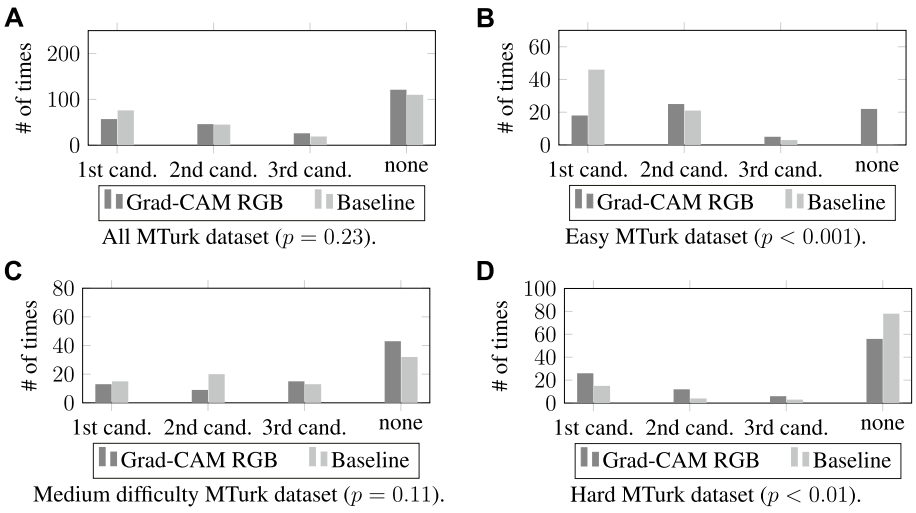

FIGURE 7
The number of times the Grad-CAM RGB method and the Mattnet baseline generated candidate bounding boxes that matched the target object by difficulty level. All MTurk Dataset in **(A)**, easy images in **(B)**, medium difficulty images in **(C)**, and hard images in **(D)**.

4.4.1.2 Easy MTurk dataset

We examined the results for the easy images with 70 expressions (Figure 7B). We conducted a two-sided Fisher's exact test (the minimum expected value was less than 5 for some cells, so the Chi-Square test couldn't be applied). The results showed significant differences (Fisher's exact test value: 40.29, $N = 140$, $p < 0.001$. Most often, the target object was matched with the first candidate bounding box for the MAttNet baseline and second candidate for the Grad-CAM RGB method–see Figure 7B). Examining the first candidate, the baseline found the target objects more often than the Grad-CAM RGB method did. Moreover, there were no cases where none of the baseline's first three candidates was correct, while the Grad-CAM RGB method had 22 cases.

4.4.1.3 Medium difficulty MTurk dataset

For the medium difficulty images, we evaluated the results for 80 expressions. A Chi-Square test did not identify a significant difference between the methods (χ^2 $(3, N = 160) = 6.07, p = .11$, the mode was "none" of the candidates for both methods). Figure 7C shows that the number of times finding the target boxes was similar for the first and third candidates for both methods. The results from both methods were slightly different for the second and the last items, but these differences were not significant.

4.4.1.4 Hard MTurk dataset

We compared the Grad-CAM RGB method with the baseline for the hard category scenes for 100 expressions (Figure 7D). We again conducted a two-sided Fisher's exact test, that showed significant differences (Fisher's exact test value: 11.44, $N = 200$, $p = .009$, the mode was "none" of the candidates for both methods). The results indicate that the Grad-CAM RGB method found the target object in its first, second, and third candidates more often than MAttNet. Also, the baseline had a higher number of cases for which no candidate was correct.

4.4.2 Grad-CAM RGB-D vs. Grad-CAM RGB

We compared the Grad-CAM RGB-D method with the Grad-CAM RGB method considering the number of times the target object matched with the candidate bounding boxes in the SUN RGB-D dataset for different depth dependencies–see Figure 9. Further, we provided some qualitative examples showing the first candidate bounding boxes suggested by both methods for the depth independent and dependent categories (Figure 10).

4.4.2.1 All SUN RGB-D dataset

We first evaluated our results by considering the whole SUN RGB-D dataset (70 images). Figure 9A shows that the Grad-CAM RGB-D method found the target object more often in its first and second candidates compared to the Grad-CAM RGB method. Moreover, the cases where none of the first three candidates matched with the target object were rarer in the Grad-CAM RGB-D method. Further analysis of these results with a Chi-Squared test showed that these differences were significant (χ^2 $(3, N = 140) = 16.06, p = .001$; the mode is the first candidate for both methods, i.e., the candidate most often matched with the target object was the first candidate).

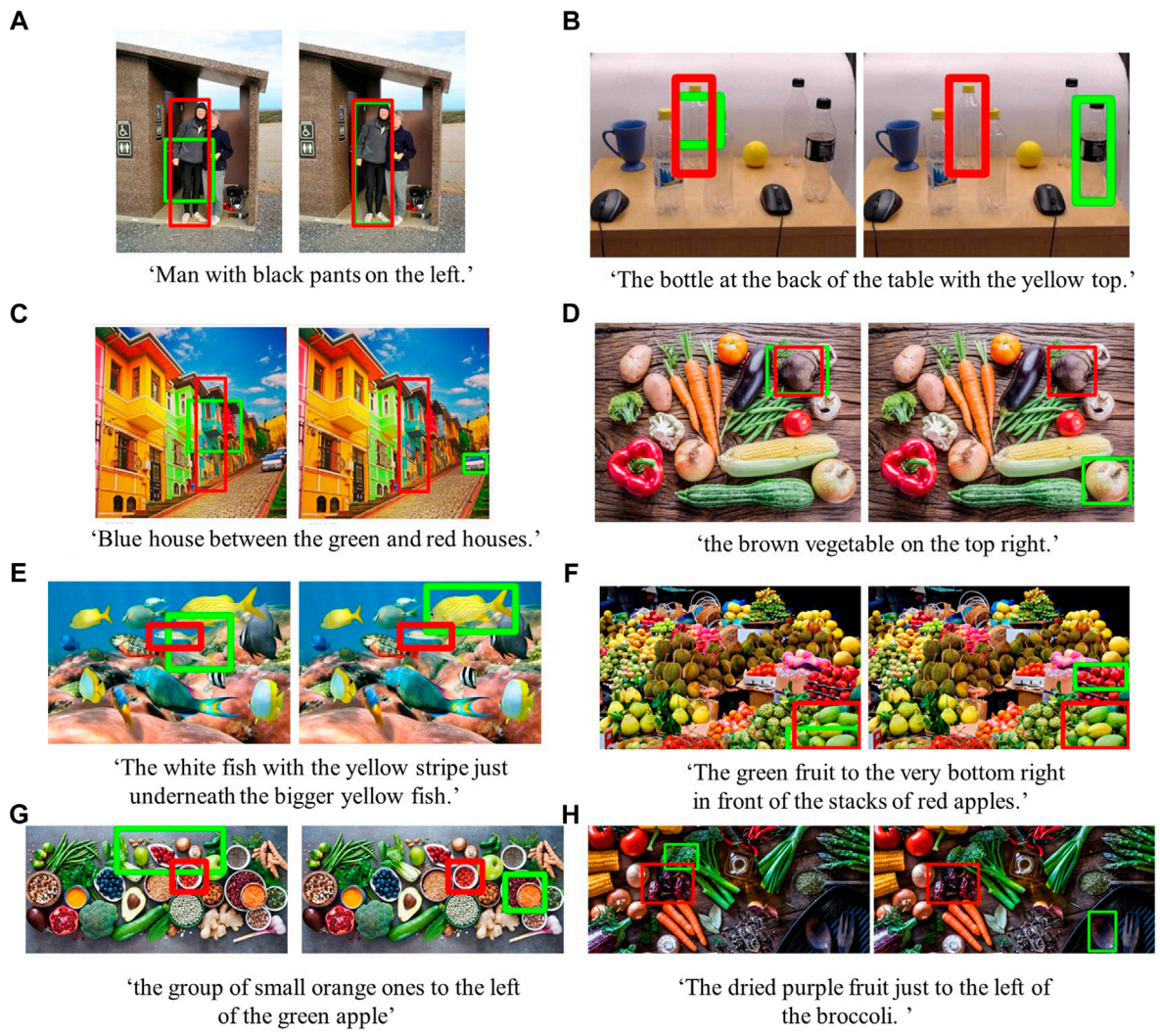

FIGURE 8
Examples of easy **(A)**, medium **(B)**, and hard **(C,D,E,F,G, H)** MTurk dataset images with original expressions collected from AMT workers describing the target objects in red boxes. The green boxes indicate the first proposed candidate object from the Grad-CAM RGB method (on the left) and the MAttNet baseline (on the right). Best viewed in color.

4.4.2.2 Depth independent SUN RGB-D dataset

To assess the impacts of depth features, we also examined the results in the depth independent category (35 images), where the target object descriptions did not depend on depth. Figure 9B shows that the Grad-CAM RGB-D method's first and second candidates matched with the target object more often, and the Grad-CAM RGB-D method failed less while suggesting the regions belonging to the target object. However, when we examined the results with Fisher's exact test (a Chi-Squared test could not be applied because some cells had a minimum expected value of fewer than five), we did not observe any significant differences between methods (Fisher's exact test value: 5.59, $N = 70$, $p = 0.12$, the mode is the first candidate for both methods).

4.4.2.3 Depth dependent SUN RGB-D dataset

Finally, we evaluated the impacts of using the depth dimension for the depth dependent category (35 images), where the descriptions of the target objects' were tied to their depth features. The results shown in Figure 9C demonstrated that the regions identified by the Grad-CAM RGB-D method in its first, second, or third candidates matched with the target object more often compared to the RGB method. Further, the

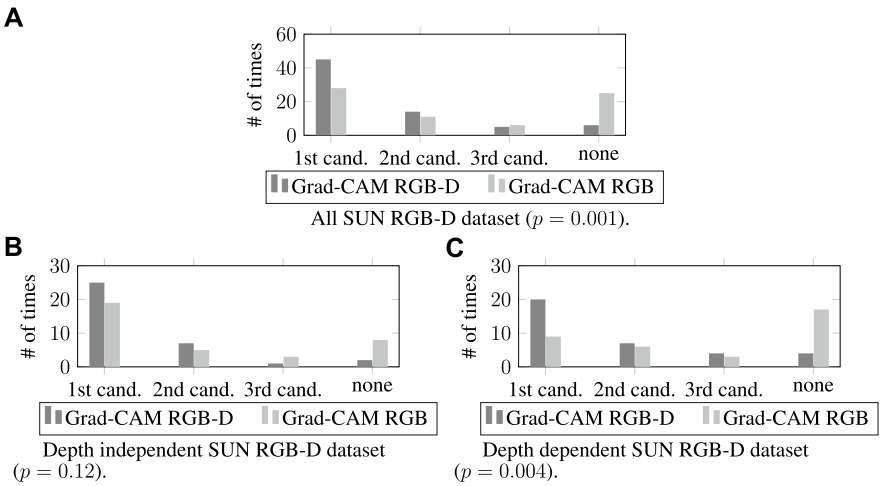

FIGURE 9
The number of times that the generated candidate bounding boxes matched with the target objects for the all SUN RGB-D dataset (in **(A)**), depth independent (in **(B)**), and dependent (in **(C)**) categories.

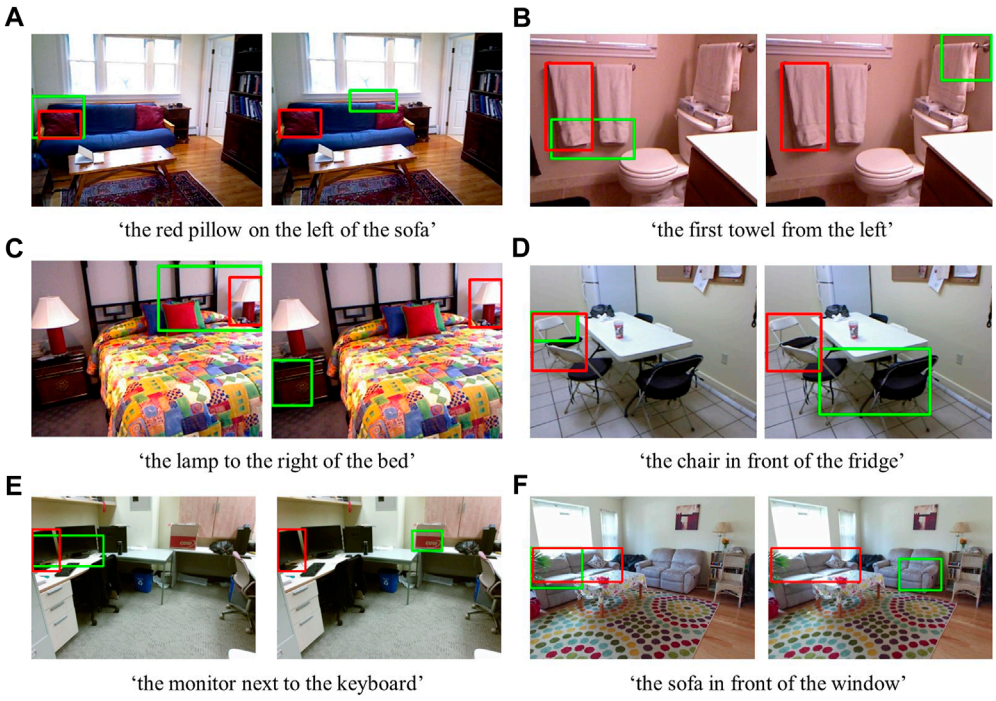

FIGURE 10
Examples from the depth independent **(A,B, C)** and depth dependent **(D,E, F)** SUN RGB-D dataset. The red bounding boxes show the target objects (ground truth), and the green boxes show the first candidates from the Grad-CAM RGB-D method (on the left) and the Grad-CAM RGB method (on the right) suggested for the given expressions. Best viewed in color.

Grad-CAM RGB-D method had fewer cases where none of its first three candidates matched the target object. To assess these results' significance, we ran another Fisher's exact test. The result of this analysis showed that the differences were significant (Fisher's exact test value: 12.67, $N = 70$, $p = 0.004$; the mode is the first candidate for the Grad-CAM RGB-D method and none of the first three candidates for the Grad-CAM RGB method).

5 Discussion

In this section, we discuss our results where we compared the Grad-CAM RGB method with the MattNet baseline on the MTurk dataset, and also the analysis obtained from the evaluation of the Grad-CAM RGB and RGB-D methods on the SUN-RGB-D dataset.

First of all, MAttNet performs significantly better than the Grad-CAM RGB method for easy MTurk images. This was expected because there are few objects in the images, the number of distractors per object is only one, and the objects are commonly known. Therefore, the chance level for MAttNet to predict the target is very high (i.e., $1/n$ where n is the total number of detected objects). The chance level is lower for the Grad-CAM RGB method because it focuses on the activation of each pixel, not the detected object boxes.

The results for hard MTurk images show that the Grad-CAM RGB method performs significantly better than the MAttNet baseline at suggesting regions belonging to the target object. This shows that the Grad-CAM RGB method can be employed when MattNet fails to identify target objects in challenging environments where there are many objects with distractors and also uncommon objects. In these environments, the users mostly referred to the uncommon objects using features such as color, shape, general category (e.g., vegetable instead of radish), and their spatial relationships with known objects nearby. On the other hand, in the easy and medium difficulty MTurk images, the users described the objects primarily using the objects' exact names because they are familiar. Therefore, the results indicate that the Grad-CAM RGB method performs better than MAttNet when the descriptions are based on an object's features instead of its name.

We did not expect to observe significant differences for the all MTurk dataset and medium difficulty MTurk images because our goal with the Grad-CAM RGB method is not an overall performance improvement, given that it does not simplify the problem to select the target object among the suggested candidates. Instead, we aim to suggest a method that can work better *in the wild* (e.g., with uncommon objects and ambiguities). Therefore, the hard MTurk dataset is crucial for the evaluation of such a system. Results on this dataset are critical for human-robot collaboration because it is impossible to assume that the robot is familiar with all of the different ways that users will use when referring to objects in the real world. In these cases, the Grad-CAM RGB method successfully suggests regions by using known concepts. For instance, in **Figure 8D**, if the robot doesn't know the concept of a vegetable, it can still predict a region by looking for something brown and on the top right. In other words, the Grad-CAM RGB method can handle the unknown objects in the expressions by employing explainability of image captioning and looking for which input features (i.e., which pixels of the image in our case) contribute more to the output. However, handling unknown objects is more difficult for the MAttNet baseline because there should be a detected bounding box to consider an object as a candidate.

From the qualitative results of the MTurk dataset, we observe that the Grad-CAM RGB method focuses on the regions which are important for the given expression. For instance, in **Figure 8A** from the easy MTurk images, the Grad-CAM RGB method finds a bounding box focused on the pants of the man because the expression includes this information. From the same example, we also notice that the bounding box suggested by the Grad-CAM RGB method does not entirely cover the man, but MAttNet provides more precise bounding boxes in such cases (commonly known objects with fewer ambiguities) by being based on an object detector. On the other hand, when there are uncommon objects (e.g., papayas in **Figure 8F**), relying on important regions of the scene, not only specified by object categories but also object features, enables the Grad-CAM RGB method to find regions that better fit expressions than MAttNet. Even in the failure cases shown in **Figures 8G,H** (reported as none in **Figure 7**), the suggested regions are still sensible. For instance, in **Figure 8H**, the suggested bounding box focuses on the broccoli because the expression includes this information. This is crucial because our goal with the Grad-CAM RGB method is to endow robots with the ability to direct their attention to the right part of the scene in the wild and determine the regions to ask for an efficient follow-up clarification instead of asking the user to repeat the whole request again.

In line with our goal, our qualitative results from the MTurk dataset support that if there are ambiguities in the environment, the Grad-CAM RGB method can be used to ask for further clarifications by only focusing on the active clusters instead of the whole image. For example, when we asked AMT workers to describe objects as if describing them to a robot (i.e., to obtain object descriptions simulating natural language user requests), there were ambiguities in their descriptions. For instance, in **Figure 8E**, the worker's description fits both of the small white fishes, and the bounding box obtained from the Grad-CAM RGB method contains the parts of both fishes. In another example, when the description is the green vegetables in **Figure 2**, the Grad-CAM RGB method finds the active clusters on the green vegetables for the first two candidates. Also, in **Figure 4**, when the red birdhouse is described, the Grad-CAM RGB method finds the most active regions on the birdhouses. Therefore, these examples demonstrate that the robot can ask the user to clarify the request by only considering these active regions instead of taking into account the whole images (e.g. in **Figure 8E**, the robot can ask 'do you mean the fish on the left or on the right?'). In brief, focusing on active clusters can improve the efficiency of human-robot collaboration.

When we compared the Grad-CAM RGB and RGB-D methods to see whether using depth features improves the system

performance, the quantitative evaluation for depth independent SUN RGB-D dataset demonstrated that using the depth of the objects did not result in significant differences. In this category, similar performances from the Grad-CAM RGB and RGB-D methods were expected because the target object descriptions are not dependent on the depth dimension. However, the system performance was significantly improved for the whole SUN RGB-D dataset and the depth dependent category. Further, the improvement was even more distinct for the depth dependent instances. The performance advancements in this category, which was collected to simulate depth-dependent environments, show that considering depth is critical in real-world applications of referring expression comprehension. In these applications, the objects are located in three-dimensional feature space, and finding the described object can be impossible without their depth features. In such cases, when the robot is comprehending the user's expressions, the Grad-CAM RGB-D method can be used for successful human-robot collaboration.

Our quantitative results from the all SUN RGB-D dataset and depth dependent category also demonstrated that the Grad-CAM RGB-D method could identify the target objects in its first candidate more often than the Grad-CAM RGB method could. Furthermore, the number of failures (i.e., none of the first three candidates matched with the target object) was significantly fewer for the Grad-CAM RGB-D method in these cases. These findings imply that, in a real-world environment, the robot would find the described objects more often in its first selection without opting for its latter candidates, and it would make fewer mistakes if the depth dimension were provided in its input space. This suggests that using depth while comprehending users' expressions improves the task accuracy and efficiency of human-robot collaboration.

In our qualitative results from the depth independent SUN RGB-D category, we show the first candidate bounding boxes suggested by the Grad-CAM RGB-D and RGB methods in **Figures 10A,B,C**. Even though we did not observe significant differences in our quantitative results for this category, the qualitative results show some of the examples in which the RGB-D method (on the left) suggested the regions matching the described objects better than the RGB method (on the right). Although some bounding boxes from the Grad-CAM RGB-D method do not exactly cover the target objects, the suggested regions are still sensible. For instance, the region suggested in **Figure 10C** partially contains the lamp and the bed when the expression is 'the lamp to the right of the bed'. However, the region suggested by the Grad-CAM RGB method is towards the incorrect lamp. Therefore, significant differences between methods for this category might be obtained with further analysis of the suggested regions by using different matching scores or asking users to evaluate these proposed regions.

In our qualitative results for the depth dependent SUN RGB-D category (**Figures 10D,E,F**), we show the first candidate bounding boxes obtained from the Grad-CAM RGB-D (on the left) and RGB methods (on the right). We observe that the regions suggested by the Grad-CAM RGB-D method fit better to the target object. In these examples, the lack of depth features misleads the Grad-CAM RGB method to select the distractor objects. For example, in **Figure 10D**, when the expression is 'the chair in front of the fridge,' the Grad-CAM RGB method highlighted the incorrect chairs, which can be considered in front of the fridge in 2D. However, the Grad-CAM RGB-D method can handle these situations using the additional features obtained from the depth dimension. These examples demonstrate the significance of the depth features for accurate comprehension of referring expressions in real-world environments.

6 Conclusion and future work

We propose the Grad-CAM RGB method to point the robot's attention in the regions of a scene described by a user to improve human-robot collaboration in the wild and also suggest extending this method to Grad-CAM RGB-D considering the depth features. Our methods find the regions belonging to the described objects using explainability. In the Grad-CAM RGB method, the region activations of an RGB scene are found using Grad-CAM, and then we use K-means clustering to obtain the active clusters. On the other hand, the Grad-CAM RGB-D method uses Grad-CAM to generate the activation heatmaps of RGB channels and the depth dimension, and then the combined activations, obtained from the common active parts of the heatmaps, are clustered to find the active clusters showing the target object. Our qualitative results from the Grad-CAM RGB method demonstrate that the regions suggested by this method can be used to resolve ambiguities. Moreover, through our evaluation, we show that the Grad-CAM RGB method works better than a state-of-art baseline for scenes with uncommon objects and multiple distractors. Finally, we demonstrate that using the depth dimension in the Grad-CAM RGB-D method significantly improves the performance in depth dependent and the whole evaluation dataset, which includes all of the depth dependent and independent category instances.

There could be several extensions of our work. We have already deployed the Grad-CAM RGB method in a robot to evaluate the efficiency of the interaction while resolving the ambiguities by asking follow-up clarifications (Doğan et al., 2022). This interaction can be further examined with the perspective of explainable robotics (Setchi et al., 2020) considering how users' perception of the robot is affected by the given visual explanations of the system predictions. Additionally, although we use the NeuralTalk2 image captaining model to obtain the activation heatmaps, our approach is applicable to other CNN-based image captioning models, such as Huang et al. (2019) and Jiang et al. (2018). Therefore,

future research can make use of our method and utilize other state-of-art captioning techniques to possibly improve the presented accuracies. Further, our system can be expanded by taking into account the aspects of visual attention studies (e.g., the importance of surrounding context (Itti and Koch, 2001) or correlation between the visual attention and gaze (Borji and Itti, 2013; Zaraki et al., 2014)). Moreover, the Grad-CAM module can be used to take the three dimensions (i.e., an RGB-D scene) as an input instead of obtaining RGB and depth activations separately. In this case, the challenge can be training an image captioning network that performs well in 3D scenes to visualize the RGB-D gradient activations. Although there are recent attempts to address the image captioning task in 3D (e.g., Chen et al. (2021)), these studies focus on relatively small datasets compared to MSCOCO, and the varied scene descriptions of MSCOCO enable our approach to work for uncommon object categories. If RGB-D gradient activations can be obtained from such a rich dataset, our method can be applied to them to obtain the described object regions without putting any restrictions on object categories. Finally, 3D point clouds can be provided in the input space instead of RGB-D images, and the performance of the robot can be evaluated further with and without depth features. This interaction can also be examined for the user's trust and reliance on the system predictions, which are critical measures for explainable robotics.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Ethics statement

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required for this study in accordance with the national legislation and the institutional requirements.

Author contributions

FD worked on the development and implementation of the proposed methods, data collection, data analyses, and writing the first draft of the manuscript. GM contributed to MTurk Study for data collection, execution of the online study, and took part in the writing process. IL supervised the research, took part in the development of algorithms, helped with the study design, and contributed to the paper writing. All authors contributed to the manuscript revision; they read and approved the final version.

Funding

This work was partially funded by grants from the Swedish Research Council (2017-05189), the Swedish Foundation for Strategic Research (SSF FFL18-0199), the S-FACTOR project from NordForsk, the Digital Futures Research Center, the Vinnova Competence Center for Trustworthy Edge Computing Systems and Applications at KTH, and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Acknowledgments

We are grateful to Grace Hung for her voluntary contributions to the data collection, and Liz Carter for her valuable comments.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2022.937772/full#supplementary-material

References

Abdul, A., Vermeulen, J., Wang, D., Lim, B. Y., and Kankanhalli, M. (2018). "Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda," in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, April 2018 (New York, NY: Association for Computing Machinery), 1–18. CHI '18. doi:10.1145/3173574.3174156

Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., and Guibas, L. (2020). "Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes," in *16th European conference on computer vision (ECCV)* (Springer International Publishing), 422–440.

Adadi, A., and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6, 52138–52160. doi:10.1109/ACCESS.2018.2870052

Alonso, J. M., Barro, S., Bugarín, A., van Deemter, K., Gardent, C., Gatt, A., et al. (2021). "Interactive natural language technology for explainable artificial intelligence," in *Trustworthy AI - integrating learning, optimization and reasoning. Lecture notes in computer science*. Editors F. Heintz, M. Milano, and B. O'Sullivan (Cham: Springer International Publishing), 63–70. doi:10.1007/978-3-030-73959-1_5

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* 58, 82–115. doi:10.1016/j.inffus.2019.12.012

Birmingham, B., Muscat, A., and Belz, A. (2018). "Adding the third dimension to spatial relation detection in 2d images," in Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 2018 (New York, NY: Association for Computational Linguistics), 146–151.

Borji, A., and Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 185–207. doi:10.1109/TPAMI.2012.89

Chao, C., Cakmak, M., and Thomaz, A. L. (2010). "Transparent active learning for robots," in Proceedings of the 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Osaka, Japan, March 2010 (IEEE), 317–324. doi:10.1109/HRI.2010.5453178

Chen, D. Z., Chang, A. X., and Nießner, M. (2020). "Scanrefer: 3d object localization in rgb-d scans using natural language," in Proceedings of the 16th European Conference on Computer Vision (ECCV), August 2020 (Springer, Cham), 202–221.

Chen, Z., Gholami, A., Niessner, M., and Chang, A. X. (2021). "Scan2cap: Context-aware dense captioning in rgb-d scans," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE), 3193–3203.

Das, D., Banerjee, S., and Chernova, S. (2021). "Explainable AI for robot failures: Generating explanations that improve user assistance in fault recovery," in Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction, New York, NY, USA, March 2021 (New York, NY: Association for Computing Machinery), 351–360. HRI '21. doi:10.1145/3434073.3444657

Ding, H., Liu, C., Wang, S., and Jiang, X. (2021). "Vision-Language transformer and query generation for referring segmentation," in Proceedings of the IEEE/CVF International Conference on Computer Vision (IEEE), 16321–16330.

Doğan, F. I., Kalkan, S., and Leite, I. (2019). "Learning to generate unambiguous spatial referring expressions for real-world environments," in Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, November 2019 (IEEE), 4992–4999. doi:10.1109/IROS40897.2019.8968510

Doğan, F. I., Torre, I., and Leite, I. (2022). "Asking follow-up clarifications to resolve ambiguities in human-robot conversation," in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, Sapporo, Japan, March 2022 (IEEE), 461–469.

Edmonds, M., Gao, F., Liu, H., Xie, X., Qi, S., Rothrock, B., et al. (2019). A tale of two explanations: Enhancing human trust by explaining robot behavior. *Sci. Robot.* 4, eaay4663. doi:10.1126/scirobotics.aay4663

Ehsan, U., and Riedl, M. O. (2020). "Human-centered explainable AI: Towards a reflective sociotechnical approach," in *HCI international 2020 - late breaking papers: Multimodality and intelligence. Lecture notes in computer science*. Editors C. Stephanidis, M. Kurosu, H. Degen, and L. Reinerman-Jones (Cham: Springer International Publishing), 449–466. doi:10.1007/978-3-030-60117-1_33

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Comput. Surv.* 51, 1–42. doi:10.1145/3236009

Han, Z., Phillips, E., and Yanco, H. A. (2021). The need for verbal robot explanations and how people would like a robot to explain itself. *ACM Trans. Hum. Robot. Interact.* 10, 1–42. doi:10.1145/3469652

Hatori, J., Kikuchi, Y., Kobayashi, S., Takahashi, K., Tsuboi, Y., Unno, Y., et al. (2018). "Interactively picking real-world objects with unconstrained spoken language instructions," in Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, May 2018 (IEEE), 3774–3781. doi:10.1109/ICRA.2018.8460699

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, Venice, Italy, October 2017 (IEEE), 2961–2969.

Hendricks, L. A., Burns, K., Saenko, K., Darrell, T., and Rohrbach, A. (2018). "Women also snowboard: Overcoming bias in captioning models," in *Computer vision – eccv 2018. Lecture notes in computer science*. Editors V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (Cham: Springer International Publishing), 793–811. doi:10.1007/978-3-030-01219-9_47

Hu, R., Rohrbach, M., Andreas, J., Darrell, T., and Saenko, K. (2017). "Modeling relationships in referential expressions with compositional modular networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE), 1115–1124.

Huang, L., Wang, W., Chen, J., and Wei, X.-Y. (2019). "Attention on attention for image captioning," in Proceedings of the IEEE/CVF international conference on computer vision (IEEE), 4634–4643.

Itti, L., and Koch, C. (2001). Computational modelling of visual attention. *Nat. Rev. Neurosci.* 2, 194–203. doi:10.1038/35058500

Jiang, W., Ma, L., Jiang, Y.-G., Liu, W., and Zhang, T. (2018). "Recurrent fusion network for image captioning," in Proceedings of the European conference on computer vision (ECCV), October 2018 (IEEE), 499–515.

Johnson, J., Karpathy, A., and Fei-Fei, L. (2016). "Densecap: Fully convolutional localization networks for dense captioning," in Proceedings of the IEEE conference on computer vision and pattern recognition (IEEE), 4565–4574.

Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., and Carion, N. (2021). "Mdetr - modulated detection for end-to-end multi-modal understanding," in Proceedings of the IEEE/CVF International Conference on Computer Vision, October 2021 (IEEE), 1780–1790.

Karpathy, A., and Fei-Fei, L. (2015). "Deep visual-semantic alignments for generating image descriptions," in Proceedings of the IEEE conference on computer vision and pattern recognition, April 2015 (IEEE), 3128–3137.

Kollar, T., Perera, V., Nardi, D., and Veloso, M. (2013). "Learning environmental knowledge from task-based human-robot dialog," in Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 2013 (IEEE), 4304–4309.

Kruijff, G.-J. M., Lison, P., Benjamin, T., Jacobsson, H., and Hawes, N. (2007). "Incremental, multi-level processing for comprehending situated dialogue in human-robot interaction," in *Symposium on language and robots*.

Kulesza, T., Burnett, M., Wong, W.-K., and Stumpf, S. (2015). "Principles of explanatory debugging to personalize interactive machine learning," in Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI '15, New York, NY, USA, January 2015 (New York, NY: Association for Computing Machinery), 126–137. doi:10.1145/2678025.2701399

Li, K., Wu, Z., Peng, K. C., Ernst, J., and Fu, Y. (2018). "Tell me where to look: Guided attention inference network," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Feb 2018 (IEEE), 9215–9223. doi:10.1109/CVPR.2018.00960

Liao, Q. V., Gruen, D., and Miller, S. (2020). "Questioning the AI: Informing design practices for explainable AI user experiences," in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, April 2020 (New York, NY, USA: Association for Computing Machinery), 1–15. CHI '20. doi:10.1145/3313831.3376590

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft coco: Common objects in context," in *European conference on computer vision* (Springer), 740–755.

Madumal, P., Miller, T., Sonenberg, L., and Vetere, F. (2020). Explainable reinforcement learning through a causal lens. *Proc. AAAI Conf. Artif. Intell.* 34, 2493–2500. doi:10.1609/aaai.v34i03.5631

Magassouba, A., Sugiura, K., Quoc, A. T., and Kawai, H. (2019). Understanding natural language instructions for fetching daily objects using gan-based multimodal target–source classification. *IEEE Robot. Autom. Lett.* 4, 3884–3891. doi:10.1109/LRA.2019.2926223

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. (2016). "Generation and comprehension of unambiguous object descriptions," in Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, June 2016 (IEEE), 11–20.

Mauceri, C., Palmer, M., and Heckman, C. (2019). "Sun-spot: An rgb-d dataset with spatial referring expressions," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Seoul, Korea (South), October 2019 (IEEE).

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* 267, 1–38. doi:10.1016/j.artint.2018.07.007

Miller, T., Howe, P., and Sonenberg, L. (2017). *Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences. arXiv:1712.00547 [cs].*

Nagaraja, V. K., Morariu, V. I., and Davis, L. S. (2016). "Modeling context between objects for referring expression understanding," in *Computer vision – eccv 2016*. Editors B. Leibe, J. Matas, N. Sebe, and M. Welling (Cham: Springer International Publishing), 792–807.

Park, D. H., Hendricks, L. A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., et al. (2018). "Multimodal explanations: Justifying decisions and pointing to the evidence," in Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 2018 (IEEE), 8779–8788. doi:10.1109/CVPR.2018.00915

Paul, R., Arkin, J., Roy, N., and M Howard, T. (2016). "Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators," in *Rss*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, New York, USA, August 2016 (New York, NY: Association for Computing Machinery), 1135–1144. doi:10.1145/2939672.2939778

Roh, J., Desingh, K., Farhadi, A., and Fox, D. (2022). "LanguageRefer: Spatial-Language model for 3D visual grounding," in Proceedings of the 5th Conference on Robot Learning, June 2022 Editor F. Aleksandra, H. David, and N. Gerhard (PMLR), 1046–1056.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. (2017). "Right for the right reasons: Training differentiable models by constraining their explanations," in *International joint conference on artificial intelligence* (California: International Joint Conferences on Artificial Intelligence Organization), 2662–2670. doi:10.24963/ijcai.2017/371

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). "Grad-cam: Visual explanations from deep networks via gradient-based localization," in Proceedings of the IEEE international conference on computer vision, 618–626.

Selvaraju, R. R., Lee, S., Shen, Y., Jin, H., Ghosh, S., Heck, L., et al. (2019). "Taking a HINT: Leveraging explanations to make vision and language models more grounded," in Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), October 2019 (IEEE), 2591–2600. doi:10.1109/ICCV.2019.00268

Setchi, R., Dehkordi, M. B., and Khan, J. S. (2020). Explainable robotics in human-robot interactions. *Procedia Comput. Sci.* 176, 3057–3066. doi:10.1016/j.procs.2020.09.198

Shridhar, M., and Hsu, D. (2018). "Interactive visual grounding of referring expressions for human-robot interaction," in *Proceedings of robotics: Science and systems* (Pittsburgh, Pennsylvania. doi:10.15607/RSS.2018.XIV.028

Shridhar, M., Mittal, D., and Hsu, D. (2020). Ingress: Interactive visual grounding of referring expressions. *Int. J. Robotics Res.* 39, 217–232. doi:10.1177/0278364919897133

Siau, K., and Wang, W. (2018). Building trust in artificial intelligence, machine learning, and robotics. *Cut. Bus. Technol. J.* 31 (2), 47–53.

Song, S., Lichtenberg, S. P., and Xiao, J. (2015). "Sun rgb-d: A rgb-d scene understanding benchmark suite," in Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, June 2015 (IEEE).

Sridharan, M., and Meadows, B. (2019). Towards a theory of explanations for human–robot collaboration. *Kunstl. Intell.* 33, 331–342. doi:10.1007/s13218-019-00616-y

Tabrez, A., and Hayes, B. (2019). "Improving human-robot interaction through explainable reinforcement learning," in Proceedings of the 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Daegu, Korea (South), March 2019 (IEEE), 751–753.

Thomason, J., Shridhar, M., Bisk, Y., Paxton, C., and Zettlemoyer, L. (2022). "Language grounding with 3D objects," in Proceedings of the 5th Conference on Robot Learning (PMLR), June 2022 (IEEE), 1691–1701.

Venkatesh, S. G., Biswas, A., Upadrashta, R., Srinivasan, V., Talukdar, P., and Amrutur, B. (2021). "Spatial reasoning from natural language instructions for robot manipulation," in Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, June 2021 (IEEE), 11196–11202. doi:10.1109/ICRA48506.2021.9560895

Wallkötter, S., Tulli, S., Castellano, G., Paiva, A., and Chetouani, M. (2021). Explainable embodied agents through social cues: A review. *ACM Trans. Hum. Robot. Interact.* 10, 1–24. doi:10.1145/3457188

Wang, D., Yang, Q., Abdul, A., and Lim, B. Y. (2019). "Designing theory-driven user-centric explainable AI," in Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, May 2019 (New York, NY: Association for Computing Machinery), 1–15. CHI '19. doi:10.1145/3290605.3300831

Watson, D. S., Krutzinna, J., Bruce, I. N., Griffiths, C. E., McInnes, I. B., Barnes, M. R., et al. (2019). Clinical applications of machine learning algorithms: Beyond the black box. *BMJ* 364, l886. doi:10.1136/bmj.l886

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). "Sun database: Large-scale scene recognition from abbey to zoo," in Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, June 2010 (IEEE).

Yu, L., Poirson, P., Yang, S., Berg, A. C., and Berg, T. L. (2016). "Modeling context in referring expressions," in *European conference on computer vision* (Springer), 69–85.

Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., et al. (2018). "Mattnet: Modular attention network for referring expression comprehension," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2018 (IEEE), 1307–1315.

Yuan, Z., Yan, X., Liao, Y., Zhang, R., Wang, S., Li, Z., et al. (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, March 2021 (IEEE), 1791–1800.

Zaraki, A., Mazzei, D., Giuliani, M., and De Rossi, D. (2014). Designing and evaluating a social gaze-control system for a humanoid robot. *IEEE Trans. Hum. Mach. Syst.* 44, 157–168. doi:10.1109/THMS.2014.2303083

Zender, H., Kruijff, G.-J. M., and Kruijff-Korbayová, I. (2009). "Situated resolution and generation of spatial referring expressions for robotic assistants," in Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 2009 (IEEE).

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. (2020). Distance-iou loss: Faster and better learning for bounding box regression. *Proc. AAAI Conf. Artif. Intell.* 34, 12993–13000. doi:10.1609/aaai.v34i07.6999