TUM School of Computation, Information and Technology
Technische Universität München

Grounding Natural Language to 3D Scenes

Zhenyu Chen

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:
Prof. Dr. Rüdiger Westermann

Prüfer der Dissertation:

1. Prof. Dr. Matthias Nießner
2. Prof. Dr. Mohamed H. Elhoseiny

Die Dissertation wurde am 22.05.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 23.02.2024 angenommen.

Acknowledgement

Pursuing a doctoral degree is a long journey full of joy, bitterness, perseverance, and endeavor. Completing this dissertation concludes this wonderful journey, and marks the start of another great adventure beyond. I would not have come this far without all the continuous support and warmhearted encouragements from my supervisors, collaborators, colleagues, and my family.

First and foremost, I would like to thank my advisor Prof. Matthias Nießner, for offering me such a great chance at the one of the best labs in the world for cutting-edge 3D computer vision research, as well as for all the influential guidance for my career path. Furthermore, words cannot express my gratitude to my mentor Prof. Angel Chang, who taught me the most truthful meaning of perseverance, and offered me priceless help when I needed it the most. My sincere gratitude also extends to all my colaborators: Ali Gholami for the thoughtful discussions, and Qirui Wu for the continuous push until the last moment. I would also like to thank Prof. Rüdiger Westermann and Prof. Mohamed H. Elhoseiny for agreeing to serve as committee members of the dissertation defense.

Moreover, I am also grateful to all my colleagues, whose company makes this journey full of fun. A huge thank-you goes to Yawar Siddiqui for all the joyful conversations we shared, and being a friend; to Guy Gafni for the memes and jokes he made that got me out of the bitter failure of research; to Manuel Dahnert for offering many useful cultural insights; to Norman Müller for the funny chats that maintained my mentality; to Dejan Azinovic for offering help with writing unreservedly; to Artem Sevastopolsky for his photography tips; to Armen Avetisyan and Aljaz Bozic for the warmhearted support with WebGL; to Prof. Angela Dai, Dr. Yinyu Nie, Dr. Alexey Artemov, Dr. Georgi Georgiev, Ji Hou, Andreas Rössler, Andrei Burov, Barbara Rössle, Can Güeli, Jiapeng Tang, Lukas Höllein, Marc Benedi, Peter Kocsis, Shenhan Qian, Shivangi Aneja, Simon Giebenhain, Tobias Kirschstein, Yujin Chen, Junwen Huang, Yuchen Rao, Hao Yu, Ziya Erkoc, Christian Diller, Pablo Palafox, Alexey Bokhovkin, and David Rozenberszki, for their great company throughout the journey. Thanks to all the interns, Yang Li, Cheng Lin, Shijie Li, and Hassan Abu Alhaija from all over the world for making the summer full of fun and laughter. My gratitude also goes to Prof. Laura Leal-Taixe, Maxim Maximov, Mengyu Chu, You Xie, Junpeng Wang, and Qunjie Zhou, for the academic insights they shared, as well as Assia Franzmann and Susanne Weitz for the generous help with paperwork, and Christoph Weiler for the continuous technical support.

Saving the best for the last, I would like to thank my parents, Tao Chen and Yu Chen, for their unconditional support throughout my entire study. I would not have achieved this without your love. Finally, I would be remiss in not mentioning my partner Nan Hu, for her endless love, and for the unshakable belief in me, even in the most difficult times. Thank you for being a part of my life!

Abstract

Grounding natural language to 3D scenes is an essential research topic for many upcoming interactive robotic agents or AR/VR applications. In recent years, there has been tremendous breakthroughs in segmenting objects in images from language. However, these methods and datasets are restricted to 2D views, where the 3D extent of an object and its surrounding environment are incompletely modelled. This limitation hinders applications where it is critical to understand the complete 3D context and the physical size, e.g. interacting with objects in the indoor scenes. In this dissertation, we explore the possible deep-learning-based methods for text-driven scene understanding on RGB-D data.

First, we introduce the task of 3D object localization in RGB-D scans using natural language descriptions. As input, we assume a point cloud of a scanned 3D scene along with a free-form description of a specified target object. To address this task, we propose ScanRefer, learning a fused descriptor from 3D object proposals and encoded sentence embeddings. This fused descriptor correlates language expressions with geometric features, enabling regression of the 3D bounding box of a target object. We also introduce the ScanRefer dataset, containing 51, 583 descriptions of 11, 046 objects from 800 ScanNet scenes. ScanRefer is the first large-scale effort to perform object localization via natural language expression directly in 3D.

Then, we introduce the task of dense captioning in 3D scans from commodity RGB-D sensors. As input, we assume a point cloud of a 3D scene; the expected output is the bounding boxes along with the descriptions for the underlying objects. To address the 3D object detection and description problems, we propose Scan2Cap, an end-to-end trained method, to detect objects in the input scene and describe them in natural language. We use an attention mechanism that generates descriptive tokens while referring to the related components in the local context. Our method can effectively localize and describe 3D objects in scenes from the ScanRefer dataset, outperforming 2D baseline methods by a significant margin.

Recent work on dense captioning and visual grounding in 3D have achieved impressive results. Despite developments in both areas, the limited amount of available 3D vision-language data causes overfitting issues for 3D visual grounding and 3D dense captioning methods. Also, how to discriminatively describe objects in complex 3D environments is not fully studied yet. To address these challenges, we present D^{3}Net, an end-to-end neural speaker-listener architecture that can detect, describe and discriminate. Our method unifies dense captioning and visual grounding in 3D in a self-critical manner. Our method outperforms SOTA methods in both tasks on the ScanRefer dataset, surpassing the SOTA 3D dense captioning method by a significant margin.

Consequently, we discuss the limitations and potential future directions of our research.

Zusammenfassung

Das Verankern von natürlicher Sprache in 3D-Szenen ist ein wichtiges Forschungsthema für viele interaktive Roboteragenten oder AR/VR-Anwendungen. In den letzten Jahren gab es enorme Durchbrüche bei der Segmentierung von Objekten in Bildern aus der Sprache heraus. Diese Methoden und Datensätze beschränken sich jedoch auf 2D-Ansichten, in denen die 3D-Ausdehnung eines Objekts und seiner Umgebung unvollständig modelliert sind. Diese Begrenzung hindert Anwendungen, bei denen es entscheidend ist, den vollständigen 3D-Kontext und die physische Größe zu verstehen, z.B. beim Umgang mit Objekten in Innenräumen. In dieser Dissertation untersuchen wir mögliche Deep-Learning-basierte Methoden für textbasiertes Szenenverständnis auf RGB-D-Daten.

Zunächst führen wir die Aufgabe der 3D-Objektlokalisierung in RGB-D-Scans unter Verwendung von natürlicher Sprachbeschreibung ein. Als Eingabe nehmen wir eine Punktewolke einer gescannten 3D-Szene zusammen mit einer freien Beschreibung eines bestimmten Zielobjekts an. Um diese Aufgabe anzugehen, schlagen wir ScanRefer vor, das eine verschmolzene Beschreibung aus 3D-Objektvorschlägen und kodierten Satzembeddings lernt. Dieser verschmolzene Deskriptor korreliert Sprachausdrücke mit geometrischen Merkmalen und ermöglicht die Regression des 3D-Begrenzungsrahmens eines Zielobjekts. Wir stellen auch das ScanRefer-Datenset vor, das 51.583 Beschreibungen von 11.046 Objekten aus 800 ScanNet-Szenen enthält. ScanRefer ist der erste groß angelegte Versuch, eine Objektlokalisierung über natürliche Sprachausdrücke direkt in 3D durchzuführen.

Als nächstes stellen wir die Aufgabe des Dense Captioning in 3D-Scans von handelsüblichen RGB-D-Sensoren vor. Dabei nehmen wir als Eingabe einen Punktwolken-Datensatz einer 3D-Szene an und erwarten als Ausgabe die Begrenzungskästen (bounding boxes) zusammen mit Beschreibungen der darunterliegenden Objekte. Um die Probleme der 3D-Objekterkennung und -beschreibung anzugehen, schlagen wir Scan2Cap vor, eine end-to-end trainierte Methode, um Objekte in der Eingangsszene zu erkennen und in natürlicher Sprache zu beschreiben. Wir verwenden einen Aufmerksamkeitsmechanismus, der beschreibende Token generiert, während er auf die damit verbundenen Komponenten im lokalen Kontext Bezug nimmt. Unsere Methode kann 3D-Objekte in Szenen des ScanRefer-Datensatzes effektiv lokalisieren und beschreiben und übertrifft 2D-Baseline-Methoden deutlich.

Aktuelle Arbeiten zur dichten Bildbeschreibung und visuellen Verankerung in 3D haben beeindruckende Ergebnisse erzielt. Trotz Entwicklungen in beiden Bereichen führt die begrenzte Menge verfügbarer 3D-Vision-Sprachdaten zu Überanpassungsproblemen für 3D-Visuelle-Verankerungs- und 3D-Dichte-Bildbeschreibungsmethoden. Auch die Frage, wie Objekte in komplexen 3D-Umgebungen diskriminativ beschrieben werden können, ist noch nicht vollständig untersucht. Um diese Herausforderungen anzugehen, präsentieren wir D^{3}Net, eine end-to-end neuronale Sprecher-Hörer-Architektur, die erkennen, beschreiben und diskriminieren kann. Unser D3Net vereint die dichte Bildbeschreibung und visuelle Verankerung in 3D in einer selbstkritischen Weise. Unsere Methode übertrifft SOTA-Methoden in beiden Aufgaben auf dem ScanRefer-Datensatz und übertrifft die SOTA 3D-Dichte-Bildbeschreibungsmethode erheblich.

Infolgedessen diskutieren wir die Grenzen und möglichen zukünftigen Richtungen unserer Forschung.

Contents

Part I

Introduction

1 Introduction

Localizing objects in the physical world has been a very important research topic for computer vision since last century. The ultimate target is to develop an end-to-end system that can recognize physical entities, such as cats or pedestrians, from the given visual signals. Traditional object localization algorithms operate on image inputs to produce object bounding boxes, leveraging classic machine learning techniques such as Support Vector Machine (SVM) [1] or Scale-Invariant Feature Transform (SIFT) [2]. With the recent advent of the deep learning, the computer vision community has witnessed tremendous progress in localizing objects in images, empowered by large-scale image datasets such as ImageNet [3] and COCO [4]. Data-driven localization methods such as Faster R-CNN [5] and Mask R-CNN [6] have dominated the official benchmarks and incubated numerous downstream applications.

On the basis of the great success on localizing objects in images, language-guided object localization emerges. Such task is to localize a region described by a given referring expression in an image. The localization outputs are expected to be either a bounding box around the target object [7], or a segmentation mask over the target object [8], with the input description being short phrases [9], [10] or more complex descriptions [11]. This task is further extended to localize objects given a question as input [12] to encourage more interactivity between users and the intelligent system.

So far, recognizing entities upon language queries have achieved great success in image domain. However, localizing objects in images cannot provide the true physical extent of a real object, such as the size and the location in the environments. This shortage greatly hinders the development of the upcoming assistant robots or VR/AR applications, where knowing the 3D extent is critical for interacting with the objects. Thanks to the recent development of the commodity 3D sensors, large-scale RGB-D dataset, such as ScanNet [13] has been collected to enable the fundamental research for scene understanding in 3D. In this dissertation, we aim to extend modern deep learning techniques to enable object localization in 3D by learning the underlying spatial relationships in the 3D environment as well as in the language inputs.

We first show the possibility of using language guidance to localize objects in RGB-D scans with deep learning techniques. To this end, we introduce the task of 3D object localization in RGB-D scans using natural language descriptions, shortened as 3D Visual Grounding. As input, we assume a point cloud of a scanned 3D scene along with a free-form description of a specified target object. A bounding box for the desired object in such 3D scene is expected as output. To address this task, we propose the ScanRefer network, learning a fused descriptor from 3D object proposals and encoded sentence embeddings. This fused descriptor correlates language expressions with geometric features, enabling regression of the 3D bounding box of a target object. To train and benchmark our proposed ScanRefer network, we also introduce the ScanRefer dataset, containing $51,583$ descriptions of $11,046$ objects from 800 ScanNet [13] scenes. In this work, we demonstrate the first large-scale effort to perform object localization via natural language expression directly in 3D.

As understanding underlying 3D spatial relationships with the language guidance is critical to accurate object localizations in 3D scenes, explicitly addressing those relationships in free-form description inputs becomes the next challenge. Naturally, densely generating descriptions (which is also known as "Dense Captioning") for the objects and their 3D environments can directly model the spatial relationships in language. However, existing methods in images are limited to narrow viewpoints and fail to capture the complete 3D contexts. To tackle this problem, we introduce the task of dense captioning in 3D scans from commodity RGB-D sensors, shortened as 3D dense captioning. Specifically, as input, we assume a point cloud of a 3D scene; the expected output is the bounding boxes along with the descriptions for the underlying objects. To address the 3D object detection and description problems, we propose Scan2Cap, an end-to-end trained method, to detect objects in the input scene and describe them in natural language. We use an attention mechanism that generates descriptive tokens while referring to the related components in the local context. To reflect object relations (i.e. relative spatial relations) in the generated captions, we use a message passing graph module to facilitate learning object relation features. Our method can effectively localize and describe 3D objects in scenes from the ScanRefer dataset, outperforming the baseline methods by a significant margin.

Although previously introduced deep-learning-based methods have achieved impressive results, they are highly dependent on the data in terms of the richness of the semantic annotations of the RGB-D scans and the variety of the collected free-form descriptions. Such data-driven approaches are in general data-hungry. Therefore, training on limited amount of available 3D vision-language data can easily cause several overfitting issues for both aforementioned tasks in 3D. Besides the data shortage issue, although the previously proposed dense captioning method describes the spatial relationships, the generated descriptions often appear to be similar to each other. Those generated descriptions cannot be used by down-stream applications to uniquely identify objects in 3D environments. In the following, we focus on how to make the descriptions more discriminative in a data-efficient way. To address these challenges, we present D^{3}Net, an end-to-end neural architecture that can detect, describe and discriminate. Our D^{3}Net unifies dense captioning and visual grounding in 3D in a self-critical manner: a neural speaker module detects and describes target objects in a scene, and a neural listener module discriminates the candidate object proposals using the received description. Such self-critical property of D^{3}Net encourages generation of discriminative object captions and enables semi-supervised training on scan data with partially annotated descriptions. Our method outperforms SOTA methods for both tasks on the ScanRefer dataset, surpassing the SOTA 3D dense captioning method by a significant margin. As a conclusion, we show that the previously proposed visual grounding and dense captioning tasks in 3D are complementary to each other in nature. Our findings can encourage future opportunities in the language-drive 3D scene understanding and generic vision-language representation learning in 3D.

We start with introducing an essential language-driven 3D scene understanding task, namely 3D visual grounding. We propose our ScanRefer network for localizing 3D object with free-form description, and the first large-scale dataset for training and benchmark this task. Then, we introduce a revered task of 3D visual grounding, where we densely detect the object and generate their descriptions with respect to the 3D environments, referred to as 3D dense captioning. Finally, we focus on exploring the complementary nature of the proposed two tasks. As a summary, we make the following three contributions in the field of language-driven scene understanding:

- We introduced the new task 3D Visual Grounding for localizing objects in 3D environments with language-guided. We also proposed the ScanRefer network to learn a fused descriptor from 3D object proposals and encoded sentence embeddings. ScanRefer, a very first language-guided 3D object localization algorithm, which learns a fused descriptor from 3D object proposals and encoded sentence embeddings.

- We introduced the new task 3D Dense Captioning for densely detect objects and generate their descriptions in 3D environments. We also proposed Scan2Cap, the first 3D Dense Captioning algorithm, and show a significant improvement over the image-based methods.

- We explored the complementary property of 3D Visual Grounding and Dense Captioning, and proposed a novel speaker-listener architecture to generate discriminative descriptions and achieve semi-supervised training. We demonstrated that our method significantly improves the previous state-of-the-art results for both tasks.

1.1 Dissertation Overview

This thesis is structured in 7 chapters that are grouped into three parts as following:

- **Part I:** Introduction (Chapters 1–2)
  - Chapter 1 (Introduction) introduces the history and recent development of language-guided object localization and our contributions to the 3D community.
  - Chapter 2 (Theoretical Fundamentals) explains basic concepts on language-guided object localization to assist comprehending the thesis.
- **Part II:** Language-guided 3D Object Localization (Chapters 3–5)
  - Chapter 3 introduces our work ScanRefer on language-guided 3D object localization, which is a fundamental task towards understanding 3D environments with language guidance.

  - Chapter 4 introduces our work Scan2Cap on describing the objects in 3D scenes in natural language to explicitly model the underlying spatial relationships in 3D environments.
  - Chapter 5 introduces our work D^{3}Net on exploring the joint nature of the localizing and describing object in 3D environments with natural language.

- **Part III:** Conclusion & Outlook (Chapters 6–7)
  - Chapter 6 (Conclusion) summarizes our proposed methods and concludes our contributions.
  - Chapter 7 (Outlook) discusses the existing problems in our proposed methods and hints the potential direction.

1.2 Contributions

This thesis discusses the new challenges in localizing objects in 3D environments, including specific tasks such as 3D Visual Grounding and Dense Captioning, as well as the complementary nature of the aforementioned tasks. For 3D Visual Grounding, we propose ScanRefer [14], an end-to-end neural architecture to learn a fused descriptor from 3D object proposals and encoded sentence embeddings, showing significantly superior performance over the image-based baselines. ScanRefer also introduces the first large-scale vision-language dataset in 3D. As understanding the spatial relationships in the 3D environments is essential to accurate localizations, we propose Scan2Cap [15], an attention-based neural algorithm to densely generate descriptions for detected objects. Scan2Cap indicates the advantages of explicitly modelling the language expressions in 3D environments, demonstrating a significant improvement in comparison to image-based methods. Besides new challenges, data is a critical factor to strong neural networks. To tackle this problem, we propose D^{3}Net [16], a speaker-listener-based architecture that can detect, describe, and discriminate in a unified approach. In this work, the improvements for both tasks demonstrate the joint and complementary nature of those two 3D vision-language tasks. More specifically, this thesis is structured by publications and built by the following contributions:

- We introduce the new task 3D Visual Grounding to localize objects in 3D environments with language guidance. To tackle this task, we propose the ScanRefer network, the first end-to-end deep-learning-based approach for 3D Visual Grounding. The ScanRefer network is the first method that learns the joint descriptors between 3D object proposals and language queries. To train and benchmark the proposed method, we introduce the ScanRefer dataset, containing over $50,000$ descriptions for around $10,000$ objects in ScanNet scenes. The ScanRefer dataset is the first large-scale effort for the vision-language field in 3D. On the newly proposed ScanRefer benchmark, our method significantly outperforms all image-based methods. The implementation of the method and the data collection platform were done by the first author. Discussions with the co-authors led to the final paper [14].

- We introduce the new task 3D Dense Captioning to densely detect and describe objects in 3D environments using natural language, for directly modelling the spatial relationships in the language cues. To tackle this task, we propose Scan2Cap, the first neural approach to detect objects in the input scene and generate their natural language descriptions. Scan2Cap is the first work in 3D that generates object descriptions via a context-aware attention mechanism. To completely reflect the 3D spatial relationships, our method applies a message passing graph module to facilitate learning inter-object relational features. Scan2Cap is the first Dense Captioning method in 3D, and is shown to outperform image-based methods by a significant margin. The method development and implementation was done by the first author. Alternative baselines were provided by Ali Gholami. Discussions with the co-authors led to the final paper [15].

- We propose D^{3}Net, the first speaker-listener-based neural architecture in 3D that can detect, describe, and discriminate. In this work, our method unifies 3D Dense Captioning and Visual Grounding in a self-critical manner: a neural speaker module detects and describes target objects, and a neural listener module discriminates the candidate object proposals using the received descriptions. D^{3}Net is the first work in 3D vision-language that utilizes synthesized descriptions to achieve semi-supervised training. Our method outperforms SOTA methods for both 3D Visual Grounding and Dense Captioning by a significant margin. Our findings also reveal the complementary nature of both tasks, encouraging future opportunities for more generic vision-language representation learning in 3D. The method development and implementation was done by the first author. Discussions with the co-authors led to the final paper [16].

1.3 List of Publications

In this dissertation, the accepted version of the following IEEE publication is used:

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

The following publications are reproduced with permission from Springer Nature:

> Z. Chen, A. X. Chang, and M. Nießner, "Scanrefer: 3d object localization in rgb-d scans using natural language," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, Springer, 2020, pp. 202–221

> Z. Chen, Q. Wu, M. Nießner, and A. X. Chang, "D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, Springer, 2022, pp. 487–505

2 Fundamentals and Methods

3D Computer Vision is a popular and fast-evolving research area empowered by deep learning techniques. Connected with the language data, deep learning methods in 3D are enhanced with capability of jointly understanding the 3D vision signals and linguistic cues. In this dissertation, we discuss our proposed multimodal algorithms for language-guided 3D object localization. Specifically, we first introduce the different representations of 3D and language data that are utilized for training the neural networks in Section 2.1. Then, we introduce the relevant neural network architectures for processing the 3D and language data in Section 2.2, respectively. Finally, we introduce several 3D datasets (Section 2.3.1) as well as the popular tasks that are relevant to grounding natural language to 3D scenes (Section 2.3.2).

2.1 Representation

Representing the real-world data in the digital form is the foundation of modern deep-learning-based algorithms. In this dissertation, we mainly deal with signals from two sources: 3D physical world and word-based language sequences. The discrepancies between those two sources bring up many research questions, such as how to represent data from two significantly different domains as numerical signals, and how to discover the underlying shared information between the 3D and language inputs. In this section, we present a brief introduction to the common representations of 3D and language data.

2.1.1 3D Geometry Representation

2.1.1.1 Mesh

A mesh is a collection of vertices, edges, and faces that represents the 3D shape. Specifically, a mesh is usually stored as a list of vertices coordinates, and an index list of the faces corners. The faces are usually defined by triangles, quadrilaterals, or other convex polygons. In this dissertation, we primarily deal with triangle mesh data, where the faces are represented as triangles. Generally, triangle mesh is a very popular representation for 3D geometry in 3D computer vision and graphics. However, meshes also discretize the 3D geometry through triangulating the surface. As such, meshes with few faces often lack the details of the corresponding geometry. We visualize a triangle mesh in Figure 2.1.

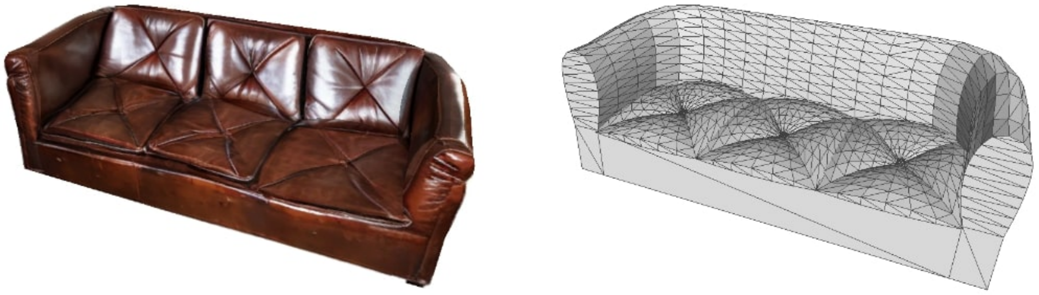


Figure 2.1: Triangle Mesh of a couch. We visualize the triangle connectivity on the right.

2.1.1.2 Point Cloud

A point cloud is a discrete set of points floating in the 3D space, where each point can be simply represented by the X, Y, and Z coordinates. In some cases, each point is often appended with additional features, such as the RGB values or the normal vector. In a nutshell, point cloud is the simplest and most memory-efficient representation for 3D entities in computer vision. However, point cloud is also an unstructured way for representing 3D shapes. For instance, even though two point clouds appear totally different from each other, they can still represent the same 3D shape. We visualize three different point clouds sampled from the mesh of the Stanford Bunny in Figure 2.2.

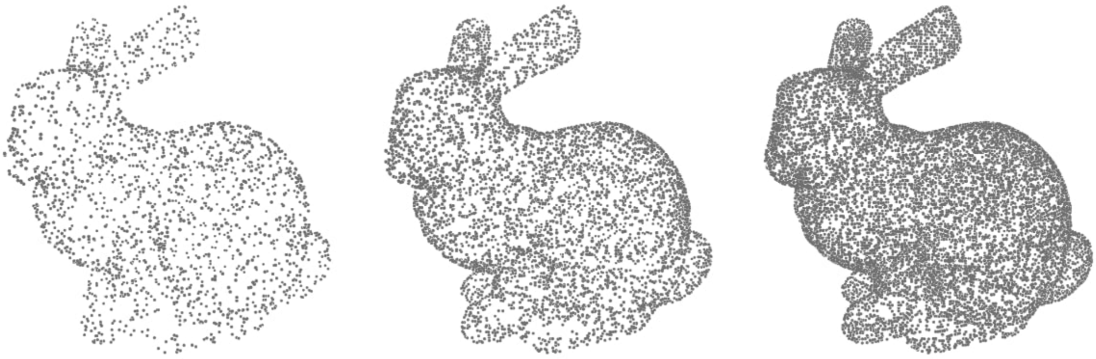


Figure 2.2: Point Cloud of the Stanford Bunny. A point cloud is a discrete set of points representing 3D geometry. Point clouds with different point coordinates and densities can still represent the same 3D shape.

There are different ways to acquire the point cloud data. For a given triangle mesh, point clouds can be generated by either taking the vertices of the mesh directly, or randomly choosing some mesh vertices as the output point cloud. Point clouds can also be produced by sampling the points on the mesh surface. In this dissertation, we operate on point clouds that are captured in real-world 3D indoor scenes via RGB-D cameras. On top of capturing color information as conventional cameras, the RGB-D cameras can also capture the depth values between the visible positions and the viewpoint in the current frustum. Utilizing the captured depth maps and the camera parameters, i.e. the intrinsic and extrinsic matrices, point clouds can be easily produced by back-projecting points from pixel space to 3D space. We visualize a point cloud generated by an RGB-D frame in Figure 2.3.

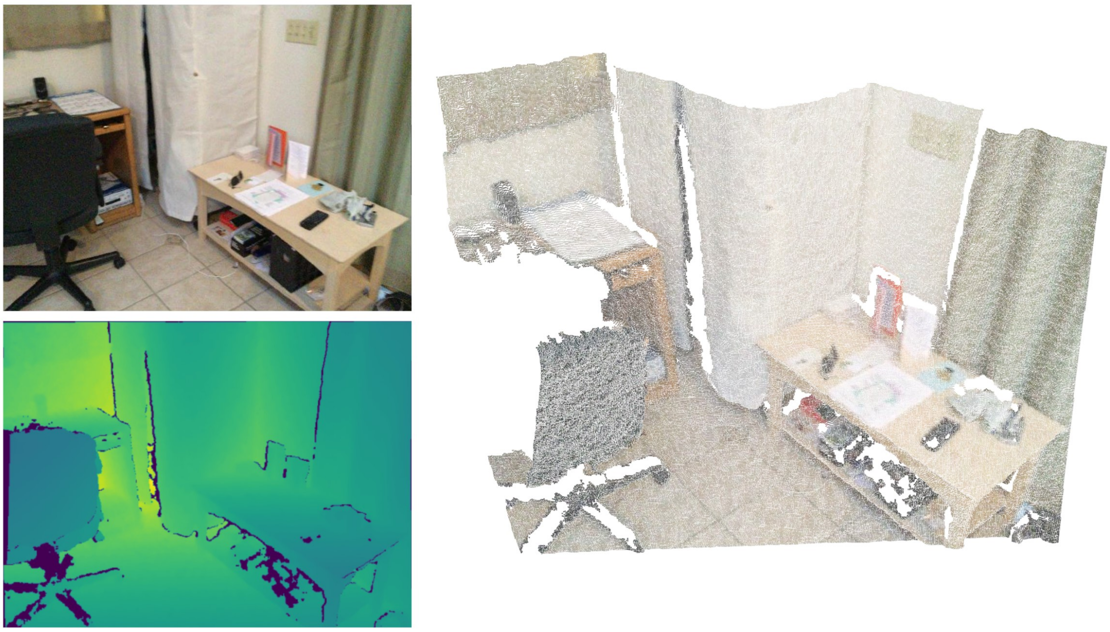


Figure 2.3: Point Cloud From RGB-D camera. We show an RGB frame (top left) and a depth map (bottom left) from the ScanNet [13]. A point cloud (right) can be generated by back-projecting the RGB values from pixel space to 3D space using the depth map.

2.1.1.3 Voxel Grid

A voxel grid is a 3D array representing volume elements on regular grids in 3D space. As with pixels in counterpart 2D images, the coordinates of voxels are not explicitly encoded within the stored values. To generate a voxel grid, the target 3D geometry is discretized with accordance to the desired resolution. In comparison to point clouds, voxel grids

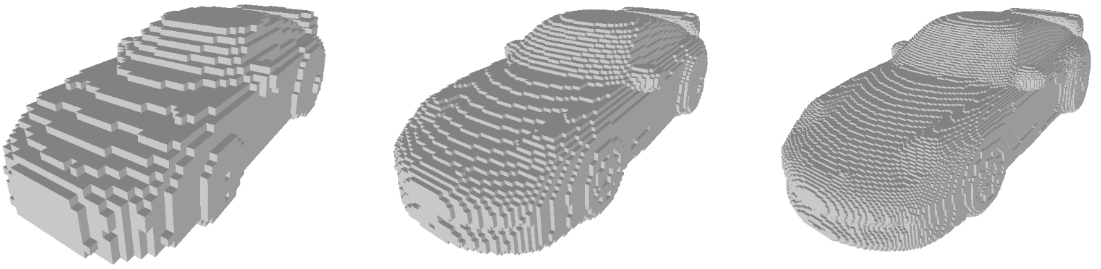


Figure 2.4: Voxel Grid Representation. We visualize voxel grids of a ShapeNet [17] car in resolution 64^3, 128^3, and 256^3. As the voxelization process discretizes the geometry, higher resolutions can reveal more details of the target 3D geometry. However, voxel grids with higher resolution also require significantly more memory for processing and storage, including numerous voxels representing free space.

are structured. That is, for the same 3D shape, the generated voxel grid is unique and identical at a specific resolution. Additionally, as voxel grids are the most similar representation to images (pixel grids), it is also easier to apply neural operands, such as convolutions, on voxels than on point clouds. However, the major downside of using voxel grids is the prohibitive memory requirement. With the increase of voxel resolutions, the storage size for voxel grids increases exponentially. Since empty space is the major components for most of the voxelized 3D geometries, alternative voxel representations such as sparse voxel grids are often adapted to reduce memory footprints.

There are many ways to generate voxel grids. One popular way to present voxel grids is to store the occupancy status of each volume in 3D space. This representation is call occupancy grid, where each voxel contains binary information whether the volume is occupied by the 3D shape. We visualize occupancy grids in different resolutions in Figure 2.4.

2.1.2 Language Representation

2.1.2.1 Bag-of-words Model

The bag-of-words model is a simple yet popular representation used in natural language processing and information retrieval. It represents a sentence or a document as a set of its words, where the frequencies of words are counted, but the grammar and the word order are discarded. In practice, the bag-of-words model is mainly used for generating features to characterize the language data. We show an example of representing a sentence with the bag-of-words model below.

```
sentence="John likes to watch movies. Mary likes movies too."
bag_of_words={"John":1, "likes":2, "to":1, "watch":1, "movies":2, "Mary":1, "too":1}
```

However, discarding the grammar and the word order eliminates the semantic meaning of the input expression. The following two sentences represent totally opposite semantic meanings while possessing exactly the same bad-of-words representations.

```
sentence_1="blue, not red"
bag_of_words_1={"blue":1, "red":1, "not":1}

sentence_2="red, not blue"
bag_of_words_2={"blue":1, "red":1, "not":1}
```

Generally, common words like “the” and “a” always appear with the highest frequencies but contribute little to representing the text inputs. One solution to this problem is to treat those words as “stop words” and exclude them throughout the encoding process. Another way is to properly normalize those frequencies with respect to the document.

2.1.2.2 N-gram Model

An n-gram is a consecutive sequence of n items in the input text. Those items can be letters, syllables, and words according to the application. Usually, an n-gram with 1 item is referred to as *unigram*; the one with 2 items is a *bigram*; the one with 3 items is called *trigram*. We show examples of n-grams extracted from a given sentence below.

```
sentence="John likes movies and music."

unigrams={"John", "likes","movies","music"}
bigrams={"John likes","likes movies","movies and","and music"}
trigrams={"John likes movies","likes movies and","movies and music"}
```

N-grams are widely used in natural language processing. They can also be used for modelling the sequence. More concretely, an n-gram model predicts an item x_i based on the other items $x_0, ..., x_{(i-1)}, x_{(i+1)}, ..., x_n$. This can be presented as $P(x_i|x_0, ..., x_{(i-1)}, x_{(i+1)}, ..., x_n)$ in probabilistic term. We discuss the application of n-grams in the following section.

2.1.2.3 Word Embeddings

In natural language processing, a word embedding is a vector representation of a word. More specifically, the word embedding vector encodes real values that represent the input word in a continuous vector space. In this vector space, embeddings of words with similar semantic meanings are expected to be closer to each other, while the dissimilar ones repel. We visualize some word embeddings in 2-dimensional space in Figure 2.5.

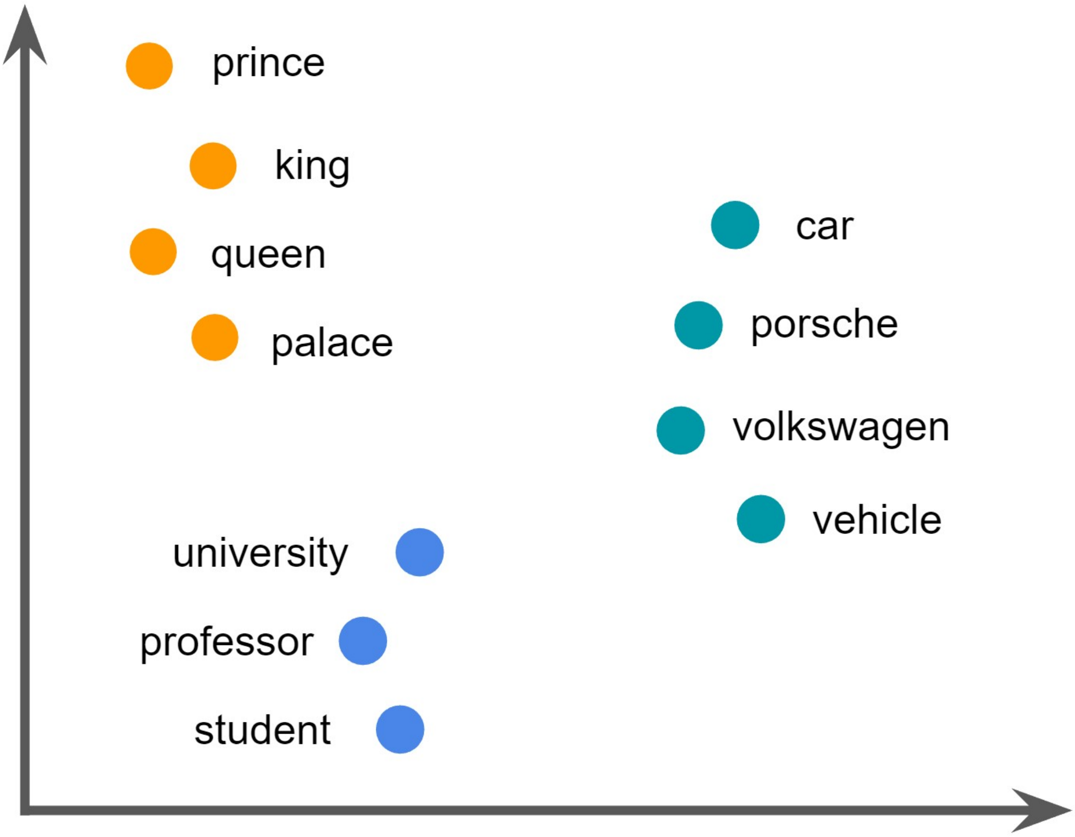


Figure 2.5: Word embeddings. We visualize the word embeddings in 2D space, where similar words are grouped together while the dissimilar ones repel.

To obtain the word embeddings with a given text corpus, one can represent each word as a continuous vector in accordance with the context in the n-gram. This process can be done in a self-supervised way, i.e. no labels for the text corpus is required. Another popular way to embed words is to learn the vector representations jointly with the downstream task. This is often an efficient approach if the word embeddings are only intended for the specified downstream task.

GloVe Embeddings GloVe ("Global Vectors for Word Representation") embeddings are a type of word embedding used in natural language processing [18]. Such embeddings encode the co-occurrence probability ratio between two words as vector differences. The intuition behind this approach is that words that appear in similar contexts are likely to have similar meanings. To create GloVe embeddings, the co-occurrence matrix is factorized using a matrix factorization algorithm, which produces word embeddings that capture the statistical relationships between words. Specifically, GloVe uses a weighted least squares objective J that minimizes the difference between the dot product of the vectors of two words and the logarithm of their number of co-occurrences:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w_j} + b_i + \tilde{b_j} - logX_{ij})^2 \tag{2.1}$$

where w_i and b_i are word vector and bias for word i, $\tilde{w_j}$ and $\tilde{b_j}$ are word vector and bias for context word j, X_{ij} is the number of times a word i occurs in the context of another word j, and f is a weighting function that assigns lower weights to rare and frequent co-occurrences.

2.2 Neural Feature Extractors

In the previous section, we introduce the 3D and language representations we use in this dissertation. Once all the representations are available, we apply different neural networks to extract high-level abstract features for the specific objectives. In this section, we introduce several neural networks that we use to process the 3D and language representations.

2.2.1 PointNet++

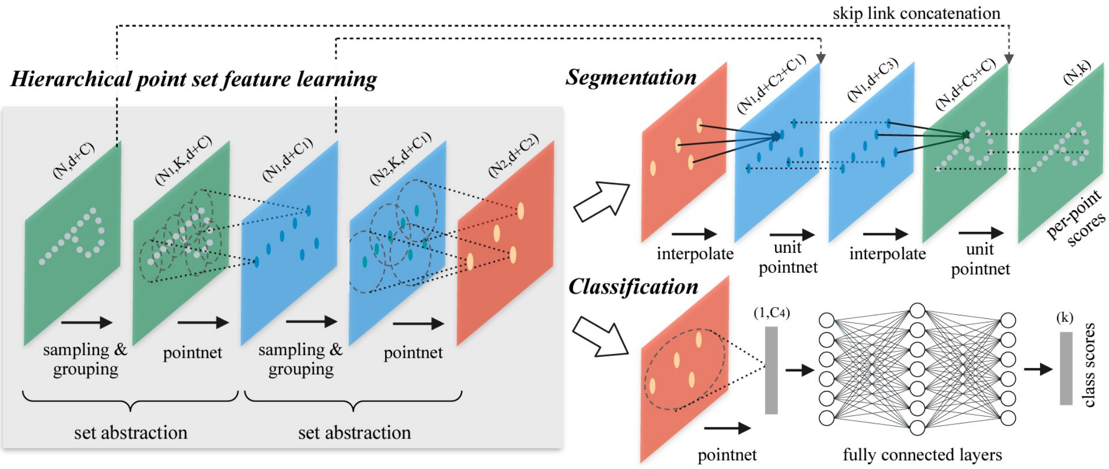


Figure 2.6: PointNet++ architecture. PointNet++ is a neural network architecture that operates directly on point clouds without any preprocessing. It is capable of handling point clouds of variable size and density. PointNet++ can be applied to many downstream tasks such as segmentation and classification. [19].

PointNet++ [19] is a neural network architecture designed for processing point clouds. It operates directly on PointNet++ without any preprocessing such as voxelization.

PointNet++ introduces a hierarchical neural network architecture that is able to capture features at different scales, allowing it to better handle complex and detailed point cloud data. The PointNet++ architecture consists of multiple modules, each of which processes a subset of the input point cloud. These modules are arranged in a hierarchical fashion, with higher-level modules processing features learned from lower-level modules. The output of each module is a set of learned features that are passed on to the next module. We demonstrate the PointNet++ architecture in Figure 2.6.

One of the key features of PointNet++ is its ability to handle point clouds of variable size and density. It accomplishes this through the use of a sampling and grouping technique, which selects a subset of points from the input cloud and groups them into small clusters. Each PointNet module operates on one of these clusters, allowing the network to process the entire point cloud efficiently.

In this dissertation, we apply PointNet++ as the feature extraction backbone for handling point cloud data. Its ability to handle variable-sized and density point clouds makes it a powerful tool for processing real-world 3D data, such as RGB-D scans from ScanNet [13].

2.2.2 SparseConv

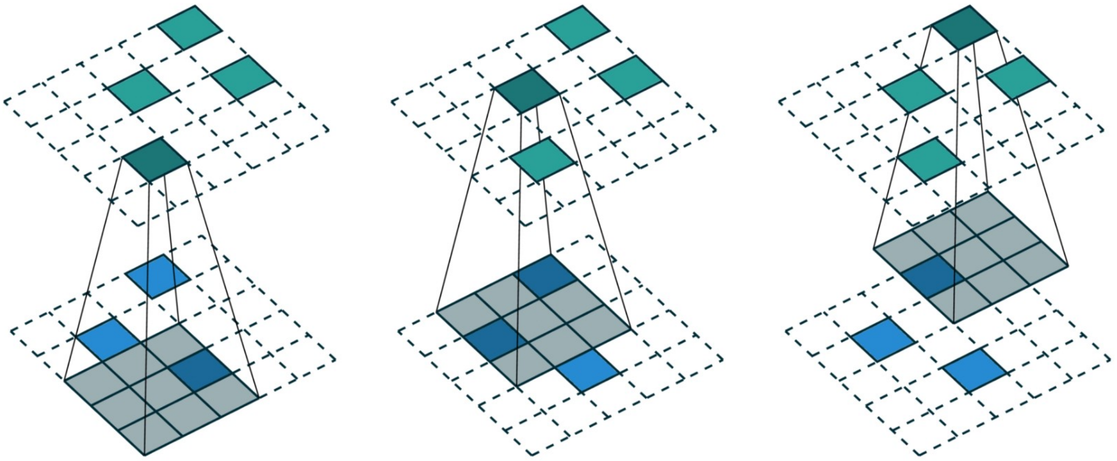


Figure 2.7: SparseConv operate on a sparse tensor. The convolution layer on a sparse tensor works similarly to that on a dense tensor. However, on a sparse tensor, convolution operates on a few specified grids according to the hash table. [20].

Although point clouds are memory-efficient for only storing points on structures, the sparse and unordered nature still lead to the performance bottleneck. In contrast, voxelizing 3D geometries as voxel grids can preserve more spatial details with regular shapes and uniform properties. As voxel grids are usually represented as 3D tensors in computer vision, one can easily apply 3D convolutional operands on them without any further processing. However, processing voxel grids with higher resolution requires significantly more storage and time. Additionally, processing voxel grids representing empty space is often inefficient in terms of the final objective, as a lot of computations are eventually wasted.

To tackle this problem, SparseConv ("Sparse Convolution") proposes to use a sparse tensor format for the input data, which only stores the non-zero values and their corresponding indices. Such sparse tensor is usually implemented via a hash table in practice. It then uses a specialized convolution operation that takes advantage of the sparsity of the input data to reduce the number of computations required. In Figure 2.7, we illustrate SparseConv operations on an input sparse tensor.

As sparse tensors can store voxel grids with high resolution efficiently, SparseConv usually produces much richer semantic features, resulting in better performance in downstream tasks such as 3D object classification and 3D semantic segmentation.

2.2.3 Transformers

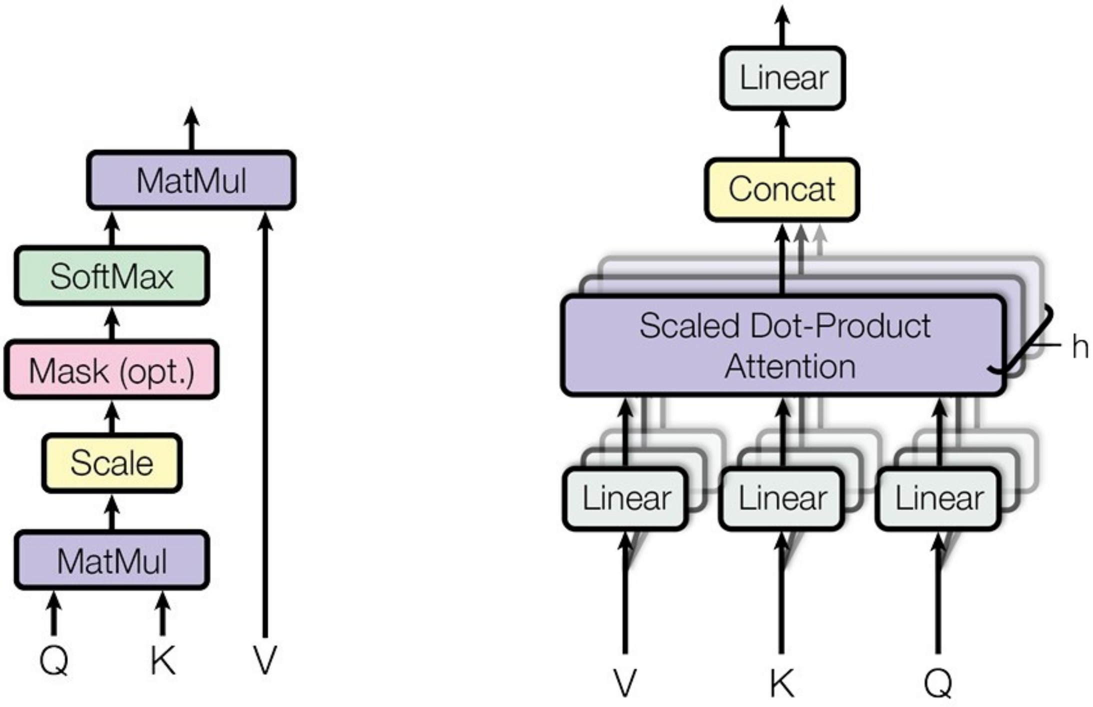
Recurrent neural networks usually suffer from long-term dependency issue, i.e. the networks often fail to remember important information hidden in the beginning of a long sequence. To tackle this issue, the Transformer architecture is proposed [21]. It is based on a self-attention mechanism, which allows the model to attend to different parts of the input sequence during the encoding and decoding phases.

As the foundation of Transformers, self-attention is a mechanism that allows the model to compute a weighted sum of the input elements based on their relevance to each other. This is done by computing attention weights for each element in the input sequence, based on its similarity to all other elements in the sequence. The resulting attention weights are used to compute a weighted sum of the input elements, which forms the output of the self-attention mechanism. Specifically, the input sequence is duplicated as three packed input tensors Q, K, and V. The output for a self-attention block is usually implemented as the following scaled dot product attention:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{Q^T K}{\sqrt{d}})V \tag{2.2}$$

where d is a hyperparameter scaling the dot product within the softmax function. We illustrate the self-attention mechanism in Figure 2.18a.

To enrich the information learned through self-attention mechanism, the Transformer uses multi-head attention, which computes multiple sets of attention weights in parallel. In practice, this is achieved by stacking multiple scaled dot-product attention modules, and map the concatenated attention features through a linear layer as the final output. We illustrate the multi-head attention block in Figure 2.18c.



(a) Scaled dot-product attention.

(b) Multi-head attention.

Figure 2.8: Basic modules of Transformers. Self-attention mechanism is the foundation of Transformers [21]. On the basis of self-attentions, multi-head attention modules improves the quality of the attention weights, which are used in Transformers to compute multiple sets of attention weights in parallel.

2.3 Related Datasets and Tasks

In this section, we introduce the relevant 3D and language datasets, as well as the adjunct tasks related to this dissertation.

2.3.1 Datasets

2.3.1.1 ScanNet

ScanNet [13] is a large-scale indoor scene dataset consists of RGB-D scans of various indoor environments, along with 3D annotations for a variety of semantic and geometric properties. The dataset includes over 1,500 scans of indoor environments, totaling over 2.5 million RGB-D frames. Each scan in the dataset is labeled with a variety of semantic categories, including walls, floors, ceilings, furniture, and objects. These labels were created using a combination of manual annotation and automatic segmentation algorithms. In addition to semantic labels, the dataset includes instance-level annotations for objects, such as chairs, tables, and beds. Besides segmentations, ScanNet also

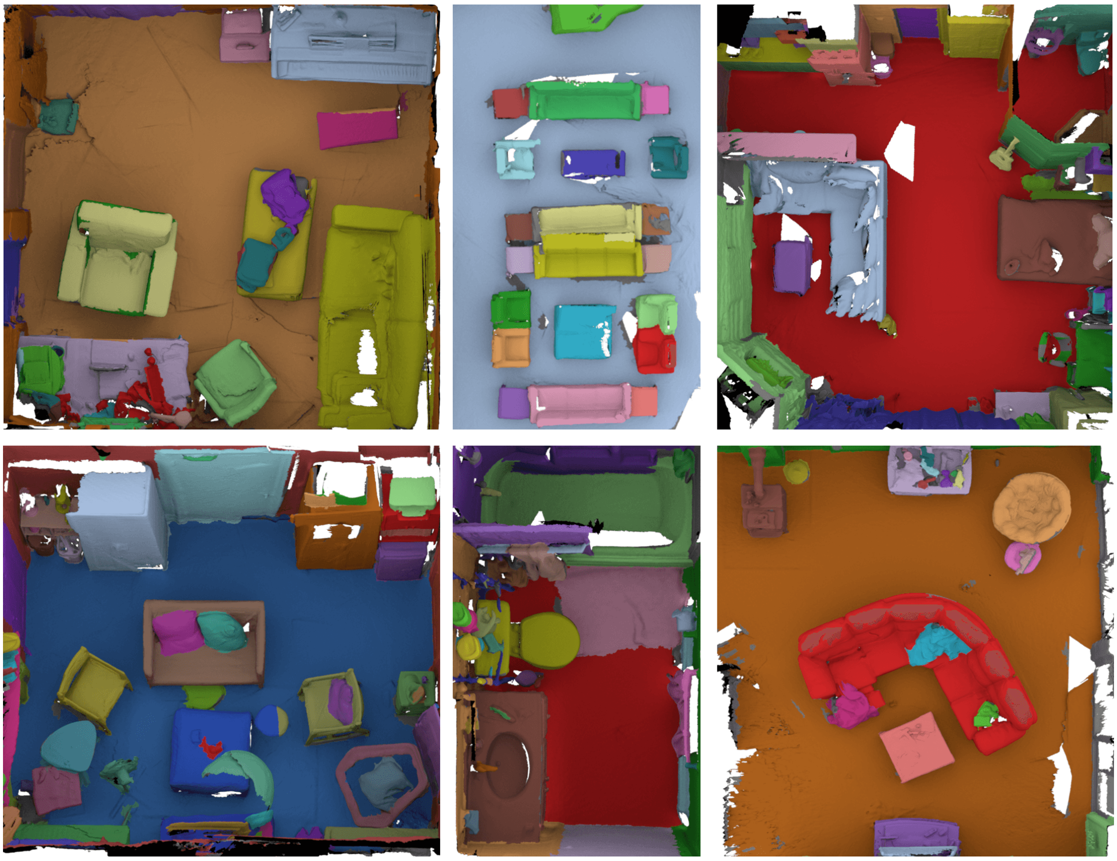


Figure 2.9: Rich annotations of the ScanNet scenes [13]. ScanNet provides instance-level semantic segmentations. Different colors represent different instances.

includes high-quality RGB-D scans captured using a variety of sensors. The scans have a resolution of 640x480 pixels and a depth resolution of 320x240 pixels. The ScanNet dataset has been used in a variety of computer vision and machine learning tasks, including 3D object detection, semantic segmentation, and scene reconstruction. It has also been used as a benchmark dataset for evaluating the performance of deep learning models on indoor scene understanding tasks. In this dissertation, we use ScanNet as the foundation for grounding language expression in 3D indoor environments. We demonstrate segmentation masks for ScanNet scenes in Figure 2.9.

2.3.1.2 RefCOCO

RefCOCO [9] (“ReferItGame Referring Expression Comprehension in Context”) is a dataset for the task of referring expression comprehension (which is also known as “visual grounding”), i.e. localizing an object in an input image with respect to an input text query. The RefCOCO dataset consists of images of everyday scenes, each annotated

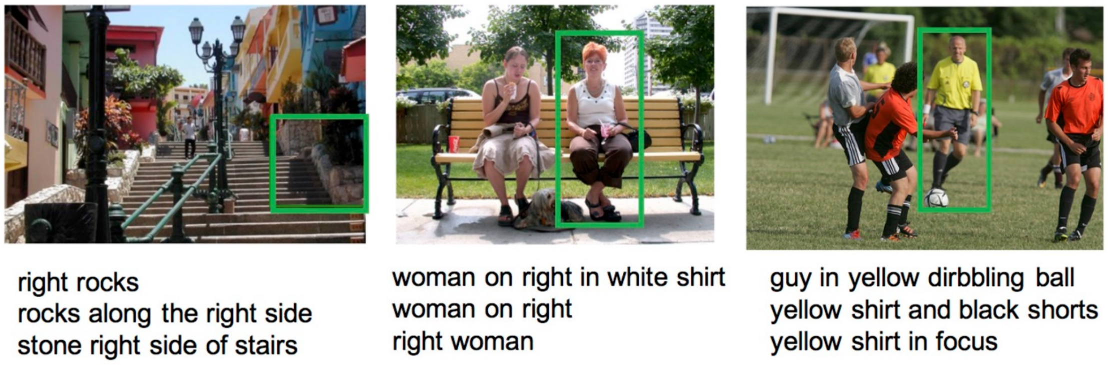

right rocks
rocks along the right side
stone right side of stairs
woman on right in white shirt
woman on right
right woman
guy in yellow dirbbling ball
yellow shirt and black shorts
yellow shirt in focus

Figure 2.10: Example of RefCOCO [9]. RefCOCO provides images and referring expressions for specific regions in paired images.

with one or more referring expressions. Referring expressions are short natural language phrases that refer to a specific object or region in the image, such as “right rocks” or “yellow shirt in focus”. RefCOCO provides various types of referring expressions, including spatial relations, color, size, and shape. It is widely used for training and evaluating methods for referring expression comprehension. In this dissertation, we explore referring expression comprehension task in the counterpart 3D environments. We show some examples of RefCOCO in Figure 2.10.

2.3.1.3 MSCOCO

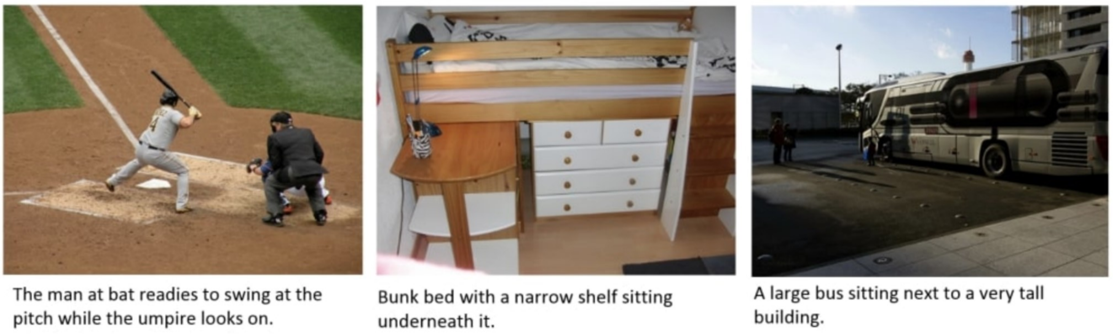

The man at bat readies to swing at the pitch while the umpire looks on.
Bunk bed with a narrow shelf sitting underneath it.
A large bus sitting next to a very tall building.

Figure 2.11: Example of MSCOCO [22]. RefCOCO provides images and associated captions.

MSCOCO [22] (“Microsoft Common Objects in Context”) dataset is a widely-used benchmark for image captioning, i.e. automatically generating descriptions for the input images. The MSCOCO dataset contains numerous images, each of which is annotated with five captions about different aspects of the given images. Those captions are annotated by human experts, ensuring the quality and diversity of the language data. The images in the dataset come from a wide range of sources and depict a diverse set of scenes and objects. The captions are also diverse, including a wide range of nouns, verbs, and adjectives. In this dissertation, we explore the possibility of generating captions for 3D environments. In contrast with image captioning on MSCOCO images, we specify the task objectives as the object in the 3D indoor scenes, which is different from describing the whole environment as in image captioning. We show some image-caption pairs of MSCOCO in Figure 2.11.

2.3.2 Tasks and Related Methods

2.3.2.1 3D Semantic Segmentation

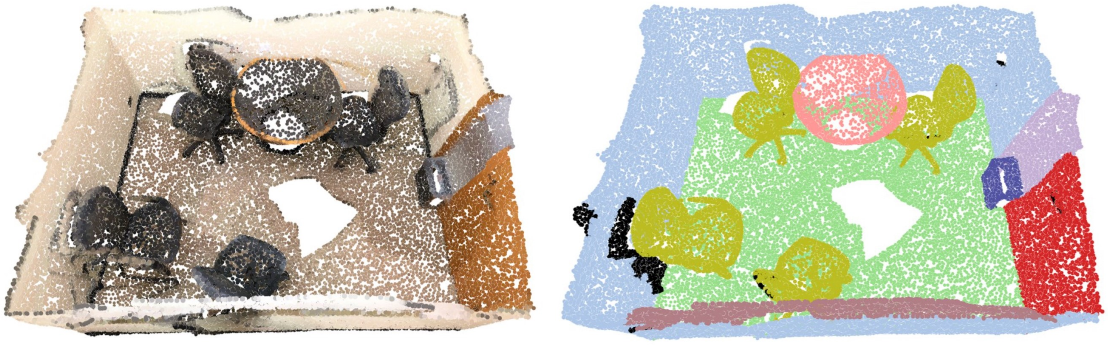


Figure 2.12: Semantic Segmentation in ScanNet [19]. We show the input point cloud (left) and the output per-point semantic labels (right).

3D Semantic Segmentation is a fundamental problem in 3D scene understanding, with applications in robotics, autonomous driving, and AR/VR. It is the task of assigning a label to each point in a 3D point cloud, where the label indicates the semantic category of the object or surface represented by that point. People usually use IoU ("Intersection over Union") to evaluate the performance of the semantic segmentation method. The IoU score is defined as:

$$\text{IoU} = \frac{B \cap \hat{B}}{B \cup \hat{B}} \tag{2.3}$$

where B is the set of points with ground truth labels and $\hat{B}$ is the predicted ones. We visualize an input point cloud and the point cloud with predicted semantic labels in Figure 2.12.

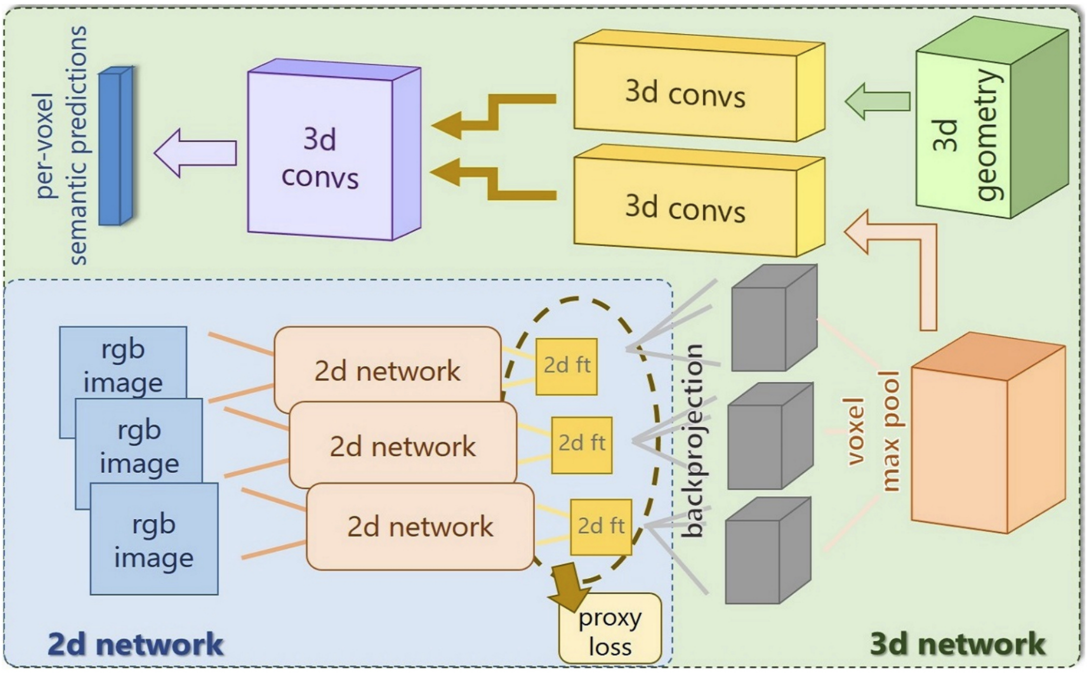


Figure 2.13: 3DMV [23] architecture. To compensate the poorly reconstructed 3D parts, RGB features are back-projected to the corresponding 3D locations.

3DMV Pure geometric inputs for 3D semantic segmentation usually suffer from poor 3D reconstruction quality. To tackle this issue, 3DMV [23] introduces a cross-modality learning method (3DMV) for this task. The key idea of 3DMV is to use encoded RGB frames as auxiliary input to compensate the poorly reconstructed 3D parts, which are represented as voxel grids in practice. Specifically, 3D convolutional neural modules are used to extract geometric features, while a 2D network encodes RGB features. The RGB features are back-projected to the corresponding 3D locations via camera intrinsic and extrinsic matrices. The whole architecture is visualized in Figure 2.13.

2.3.2.2 3D Object Detection

3D Object Detection is an important task in computer vision that involves detecting and localizing objects in 3D space, which is usually from point clouds. This task presents several unique challenges compared to object detection in counterpart 2D images. For example, 3D data is often sparse and irregular, requiring specialized techniques for processing and analysis. 3D Object Detection has many upcoming applications, including autonomous driving, assistant robots, and AR/VR. In this dissertation, we treat 3D Object Detection as the fundamental technique for localizing objects in 3D scenes. We visualize an input point cloud and the detected object bounding boxes in Figure 2.14.

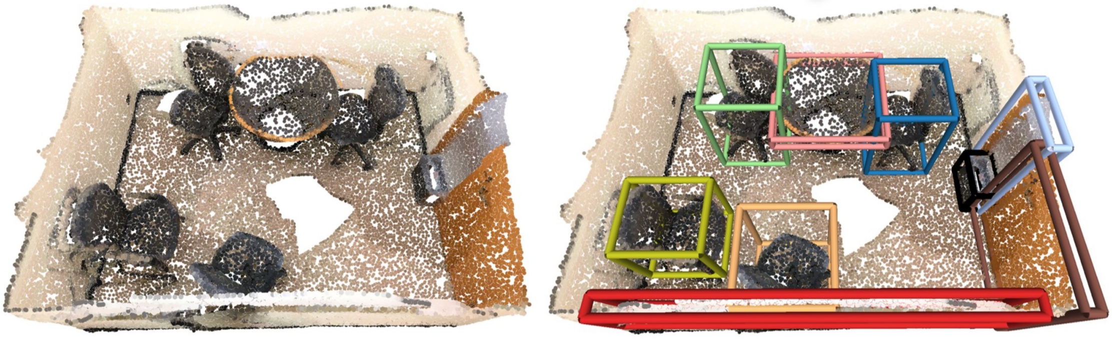


Figure 2.14: Object Detection in ScanNet [24]**.** We show the input point cloud (left) and the output bounding boxes (right).

One of the most popular metrics for object detection is mAP@k. Here, k is the previously introduced IoU score. Then, mAP@k is the mean of the AP scores for all object categories in the dataset, thresholded by given IoU score k. AP ("Average Precision") is a single-number metric that summarizes the precision-recall curve. Specifically, AP is calculated as the area under the precision-recall curve. In this dissertation, we also adapt this widely used metric to evaluate the performance of our localization network.

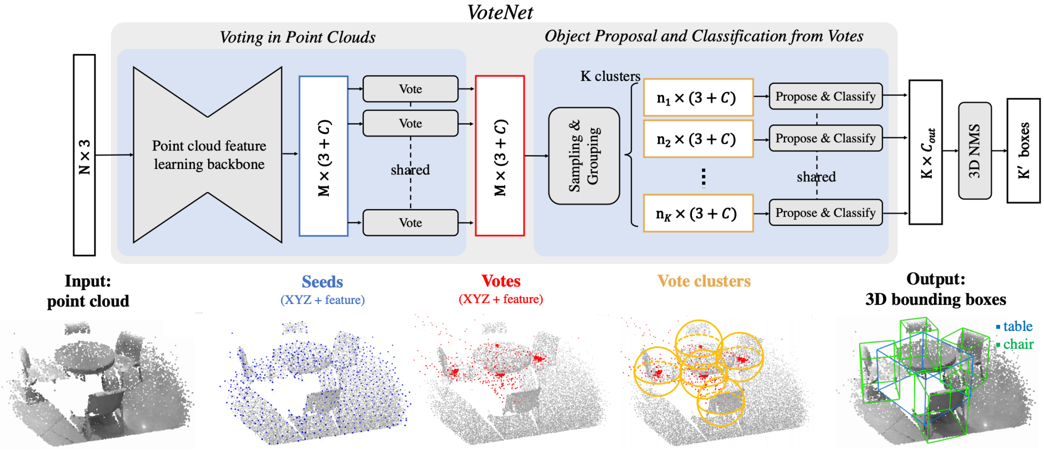


Figure 2.15: VoteNet [24] **Architecture**. VoteNet takes point clouds as input and predict object bounding boxes from a set of point clusters as output.

VoteNet VoteNet [24] is a deep-learning-based 3D object detection architecture that operates directly on point clouds. Leveraging a PointNet++ backbone, VoteNet extracts rich low-level features directly from the input point cloud data. The key idea of VoteNet is to use Hough Voting algorithm that involves voting in 3D space to identify clusters of points that are likely to belong to the same object. VoteNet generates a set of object proposals, which are candidate regions in the scene that may contain objects of interest. Then, for each object proposal, VoteNet performs object classification as well as bounding box regression to determine the type and size of the candidate object. The training of VoteNet is done in an end-to-end manner. In this dissertation, we take VoteNet as the basic backbone for predicting bounding boxes of 3D objects in RGB-D scans. We visualize the VoteNet architecture in Figure 2.15.

2.3.2.3 3D Instance Segmentation

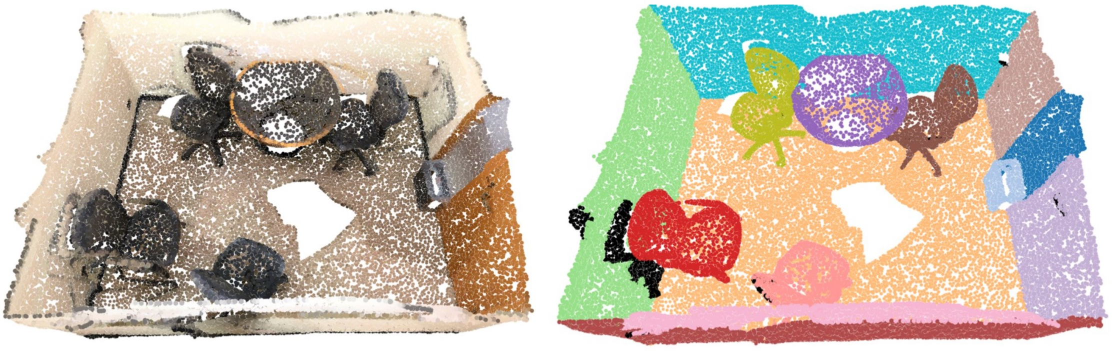


Figure 2.16: Instance Segmentation in ScanNet [25]. We show the input point cloud (left) and the segmented instance labels (right), where each color corresponds to an individual object.

3D Instance Segmentation is the task of not only assigning semantic labels to each point in a 3D point cloud, but also grouping points together into instances of individual objects. The output of a 3D instance segmentation algorithm is a set of segmented objects, where each object is assigned a unique instance ID. 3D instance segmentation is a challenging problem due to the complexity and variability of 3D scenes, as well as the need to distinguish between different instances of the same semantic category. There are mainly two types of instance segmentation algorithms: 1) top-down, which first detects object bounding boxes then segment the 3D shapes, as in Hou et al. [26]; 2) bottom-up, which first performs semantic segmentation then cluster points into instances, as in PointGroup [25]. This task is evaluated similarly to 3D Object Detection. The only difference is that the evaluation of 3D Instance Segmentation is based on points rather than bounding boxes. In this dissertation, we also adapt 3D Instance Segmentation into the object localization pipeline, as it expects much denser and detailed per-point localizations in comparison to bounding box predictions. We visualize an input point cloud and the predicted instance labels in Figure 2.16.

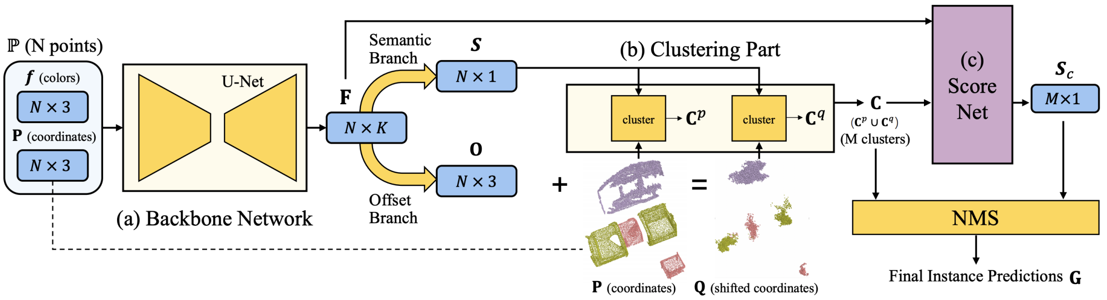


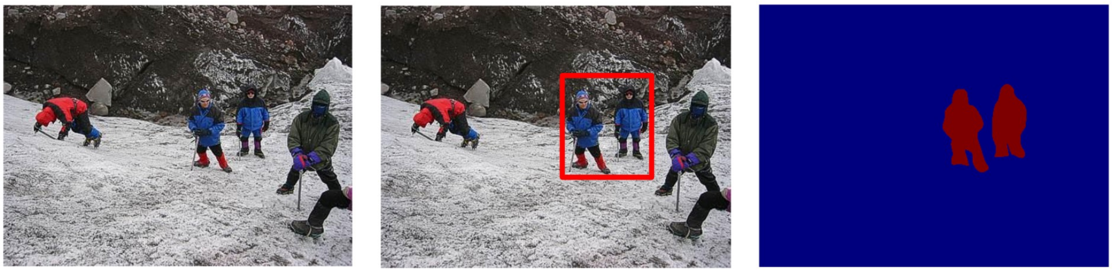
Figure 2.17: PointGroup architecture [25]. PointGroup takes a point cloud as input, then generates instances by a clustering module, a ScoreNet module, and a NMS ("Non-Maximum Suppression") module.

PointGroup PointGroup [25] is a SparseConv-based end-to-end approach that generates accurate and efficient instance segmentation results on large-scale 3D point clouds. It takes the point cloud as input and converts it into a sparse voxel grid. Then, a SparseConv [20] backbone extract per-voxel features, which are mapped back to the original point cloud as per-point features. Afterward, a grouping module uses a clustering algorithm similar to VoteNet [24] to group points within each object proposal into instances. PointGroup applies a small network called ScoreNet to predict the confidence for whether the object proposals are valid instances. A NMS ("Non-Maximum Suppression") module is applied during inference to filter out redundant instance predictions. In this dissertation, we utilize a localization backbone analogous to PointGroup. We show the PointGroup architecture in Figure [25].

2.3.3 Visual Grounding

Visual Grounding refers to the process of linking language or textual expression to specific regions in an image. It involves identifying and localizing objects, regions, or events referred to in the language, and establishing a correspondence between these elements and their visual representations. Visual Grounding plays a crucial role in the joint field of natural language processing and computer vision.

There are various approaches to achieve Visual Grounding, including the ones based on object detection methods and instance segmentation networks, as shown in Figure 2.18. To enhance the multimodal features learned from images and texts, recent approaches often apply attention mechanisms [21]. These methods enable the model to identify and locate visual features in the input, and to establish correspondences between these



(a) Input image. (b) Grounded image region. (c) Grounded image masks.

Figure 2.18: Visual Grounding. For a given image and a query "people in blue coat", the visual grounding system usually predicts the bounding box for the desired region, or mark the region via a segmentation mask.

features and the textual information in the output. In a nutshell, the accuracy and effectiveness of visual grounding methods depend on the quality and richness of the visual features and the textual input, as well as the high-level multimodal features extracted from both modalities.

To evaluate the performance of Visual Grounding methods, people usually use a thresholded accuracy, where the positive predictions have higher intersection over union (IoU) with the ground truths than the thresholds. The Visual Grounding accuracy is denoted as Acc@kIoU, where k is the threshold for the IoU score. In practice, k is usually set to 0.25 and 0.5 for most experiments. We also adapt such thresholded accuracy in this dissertation.

SCRC SCRC ("Spatial Context Recurrent ConvNet") is an end-to-end approach for visual grounding in images. Specifically, it follows a two-stage approach: it first detects a set of image regions as the candidates, then it uses a recurrent neural network to score the candidate regions according to the input query. This approach can effectively identify desired image regions using the language cues. In this dissertation, we discuss its feasibility by lifting SCRC to much more challenging 3D scenarios. We illustrate the SCRC pipeline in Figure 2.19.

One-Stage Visual Grounding Previous visual grounding methods usually take a two-stage approach, i.e. producing the bounding box proposals first, then picking the most likely box in accordance with the input text query. Such approach is often bottle-necked by the inference speed. Additionally, the grounding performance is also capped by the quality of the region proposals. For example, if the target region is never detected during the first stage, it is impossible to match the text query with the correct image region.

To tackle this issue, Yang et al. [27] propose a one-stage visual grounding approach. The key idea is to directly predict the grounded bounding box given the input image and query. Specifically, the one-stage network extracts the image features as dense feature maps, which are then fused together with encoded text features. Such fused features are processed through a grounding module to predict the bounding box size

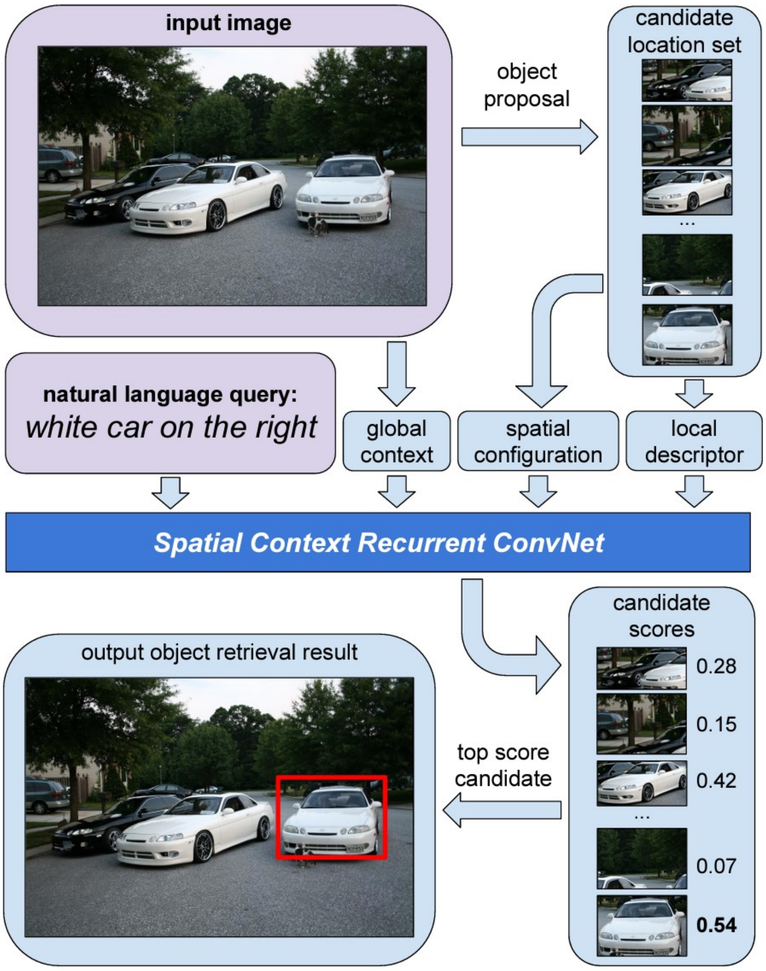


Figure 2.19: SCRC pipeline [7]. SCRC retrieves candidate image regions by scoring each region according to the text query.

and confidence for matching with the input query. This way, one-stage visual grounding network achieves fast inference while showing the state-of-the-art performance at the time of publication. We visualize this architecture in Figure 2.20.

2.3.4 Dense Captioning

The goal of image captioning is to teach an intelligent system to understand the content of an image and express that understanding in natural language. In contrast, dense captioning is a multimodal task that involves generating multiple captions for different regions or objects in an image. Unlike image captioning, dense captioning aims to provide a more detailed description of the contents of an image by describing the different regions

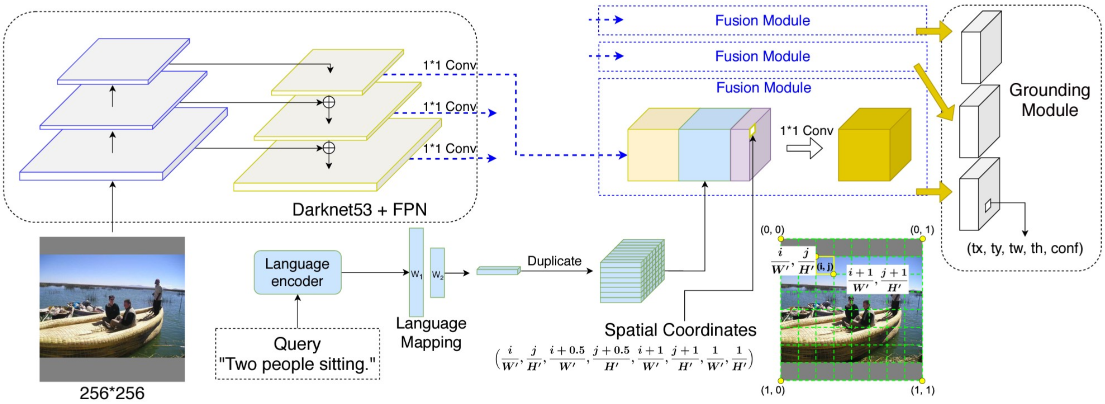


Figure 2.20: One-stage visual grounding pipeline [27]. Taking an image and a text query as input, the one-stage approach detects and identifies the desired image region in a one-stage manner.

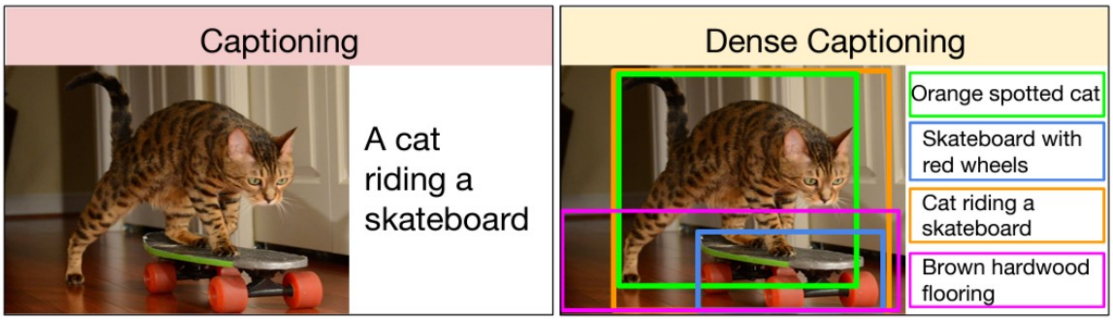


Figure 2.21: A comparison between image captioning and dense captioning [28]. The key difference between image captioning and dense captioning is that the former usually expects a single description for the whole image, while the latter expects dense pairs of image regions and descriptions as output.

or objects within it. To showcase the difference between image captioning and dense captioning, we show a comparison in Figure 2.21. Dense captioning can be seen as a combination of object detection and image captioning. One of the challenges of dense captioning is how to effectively combine object detection and captioning algorithms to generate accurate and diverse descriptions.

Dense Captioning methods are evaluated with a joint metric of object detection and image captioning. More concretely, an mAP thresholded by both IoU score and a specific image captioning metric is applied. Such mAP is measured through computing all paired threshold values.

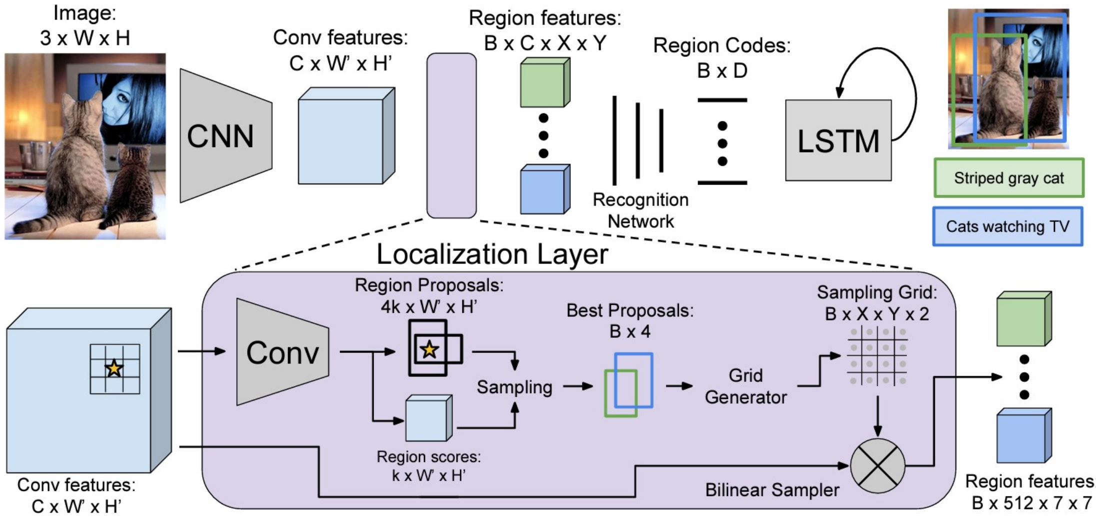


Figure 2.22: DenseCap pipeline [28]. DenseCap detects object regions and iteratively generate natural language descriptions for all objects.

DenseCap DenseCap [28] is the first work that propose a dense captioning algorithm that combines object detection and image captioning to generate natural language descriptions of the objects and regions in an image. The algorithm consists of two main components: a fully convolutional localization network (FCLN) for object detection and a recurrent neural network (RNN) for caption generation. Specifically, the FCLN module directly predicts bounding boxes and object scores for each location in a feature map. Then, the RNN is trained to generate natural language descriptions taking the object region features as input, which is analogous to conventional image captioning methods. In this dissertation, we discuss in depth the feasibility of performing dense captioning in 3D scenes, while highlighting the key challenging of doing so in 3D environments. The DenseCap pipeline is visualized in Figure 2.22.

Part II

Grounding Natural Language to 3D Scenes

3 3D Object Localization in RGB-D Scans using Natural Language

This chapter introduces the following paper:

Z. Chen, A. X. Chang, and M. Nießner, "Scanrefer: 3d object localization in rgb-d scans using natural language," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, Springer, 2020, pp. 202–221

Abstract of the paper We introduce the task of 3D object localization in RGB-D scans using natural language descriptions. As input, we assume a point cloud of a scanned 3D scene along with a free-form description of a specified target object. To address this task, we propose **ScanRefer**, learning a fused descriptor from 3D object proposals and encoded sentence embeddings. This fused descriptor correlates language expressions with geometric features, enabling regression of the 3D bounding box of a target object. We also introduce the ScanRefer dataset, containing $51,583$ descriptions of $11,046$ objects from 800 ScanNet [13] scenes. ScanRefer is the first large-scale effort to perform object localization via natural language expression directly in 3D.

Contribution The method development and implementation was done by the first author. Discussions with the co-authors led to the final paper.

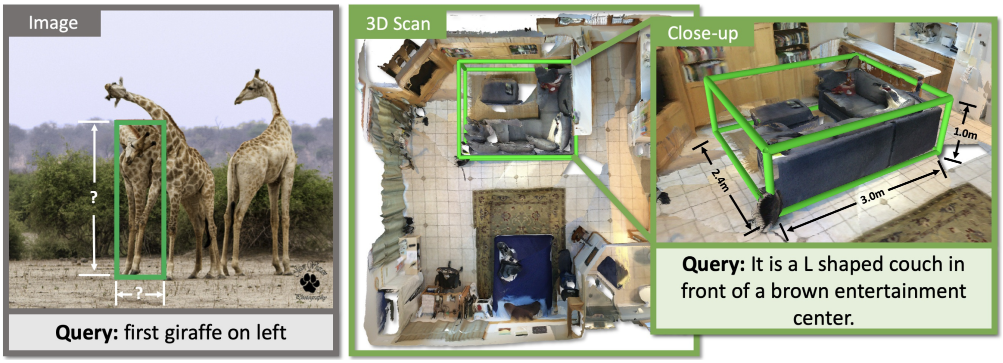


Image
?
?
Query: first giraffe on left
3D Scan
Close-up
1.0m
2.4m
3.0m
Query: It is a L shaped couch in front of a brown entertainment center.

Figure 3.1: We introduce the task of object localization in 3D scenes using natural language. Given as input a 3D scene and a natural language expression, we predict the bounding box for the target 3D object (right). The counterpart 2D task (left) does not capture the physical extent of the 3D objects.

3.1 Introduction

In recent years, there has been tremendous progress in both semantic understanding and localization of objects in 2D images from natural language (also known as visual grounding). Datasets such as ReferIt [9], RefCOCO [29], and Flickr30K Entities [10] have enabled the development of various methods for visual grounding in 2D [7], [8], [11]. However, these methods and datasets are restricted to 2D images, where object localization fails to capture the true 3D extent of an object (see Fig. 4.1, left). This is a limitation for applications ranging from assistive robots to AR/VR agents where understanding the global 3D context and the physical size is important, e.g., finding objects in large spaces, interacting with them, and understanding their spatial relationships. Early work by Kong et al. [30] looked at coreference in 3D, but was limited to single-view RGB-D images.

In this work, we address these shortcomings by proposing the task of object localization using natural language directly in 3D space. Specifically, we develop a neural network architecture that localizes objects in 3D point clouds given natural language descriptions referring to the underlying objects; i.e., for a given text description in a 3D scene, we predict a corresponding 3D bounding box matching the best-described object. To facilitate the task, we collect the ScanRefer dataset, which provides natural language descriptions for RGB-D scans in ScanNet [13]. In total, we acquire $51,583$ descriptions of $11,046$ objects. To the best of our knowledge, our ScanRefer dataset is the first large-scale effort that combines 3D scene semantics and free-form descriptions. In summary, our contributions are as follows:

- We introduce the task of localizing objects in 3D environments using natural language descriptions.
- We provide the ScanRefer dataset containing $51,583$ human-written free-form descriptions of $11,046$ objects in 3D scans.

- We propose a neural network architecture for localization based on language descriptions that directly fuses features from 2D images and language expressions with 3D point cloud features.
- We show that our end-to-end method outperforms the best 2D visual grounding method that simply backprojects its 2D predictions to 3D by a significant margin (9.04 Acc@0.5IoU vs. 22.39 Acc@0.5IoU).

3.2 Related Work

Grounding Referring Expressions in Images. There has been much work connecting images to natural language descriptions across tasks such as image captioning [31]–[34], text-to-image retrieval [35], [36], and visual grounding [7], [11], [37]. The task of visual grounding (with variants also known as referring expression comprehension or phrase localization) is to localize a region described by a given referring expression, the query. Localization can be specified by a 2D bounding box [9]–[11] or a segmentation mask [8], with the input description being short phrases [9], [10] or more complex descriptions [11]. Recently, Acharya et al [12] proposed visual query detection where the input is a question. The focus of our work is to lift this task to 3D, focusing on complex descriptions that can localize an unique object in a scene.

Existing methods focus on predicting 2D bounding boxes [7], [29], [35], [37]–[42] and some predict segmentation masks [8], [43]–[47]. A two-stage pipeline is common, where first an object detector, either unsupervised [48] or pretrained [5], is used to propose regions of interest, and then the regions are ranked by similarity to the query, with the highest scoring region provided as the final output. Other methods address the referring expression task with a single stage end-to-end network [8], [27], [49]. There are also approaches that incorporate syntax [50], [51], use graph attention networks [52]–[54], speaker-listener models [11], [55], weakly supervised methods [56]–[58] or tackle zero-shot settings for unseen nouns [59].

However, all these methods operate on 2D image datasets [9], [10], [29]. A recent dataset [60] integrates RGB-D images but lacks the complete 3D context beyond a single image. Qi et al. [61] study referring expressions in an embodied setting, where semantic annotations are projected from 3D to 2D bounding boxes on images observed by an agent. Our contribution is to lift NLP tasks to 3D by introducing the first large-scale effort that couples free-form descriptions to objects in 3D scans. Tab. 3.1 summarizes the difference between our ScanRefer dataset and existing 2D datasets.

Object Detection in 3D. Recent work on 3D object detection on volumetric grids [26], [62]–[65] has been applied to several 3D RGB-D datasets [13], [66], [67]. As an alternative to regular grids, point-based methods, such as PointNet [68] or PointNet++ [19], have been used as backbones for 3D detection and/or object instance segmentation [69], [70]. Recently, Qi et al. [24] introduced VoteNet, a 3D object detection method for point clouds based on Hough Voting [71]. Our approach extracts geometric features in a similar fashion, but backprojects 2D feature information since the color signal is useful for describing 3D objects with natural language.

dataset	#objects	#expressions	AvgLeng	data format	3D context
ReferIt [9]	96,654	130,364	3.51	image	-
RefCOCO [29]	50,000	142,209	3.50	image	-
Google RefExp [11]	49,820	95,010	8.40	image	-
SUN-Spot [60]	3,245	7,990	14.04	image	depth
REVERIE [61]	4,140	21,702	18.00	image	panoramic image
ScanRefer (ours)	**11,046**	**51,583**	**20.27**	**3D scan**	**depth, size, location, etc.**

Table 3.1: Comparison of referring expression datasets in terms of the number of objects (#objects), number of expressions (#expressions), average lengths of the expressions, data format and the 3D context.

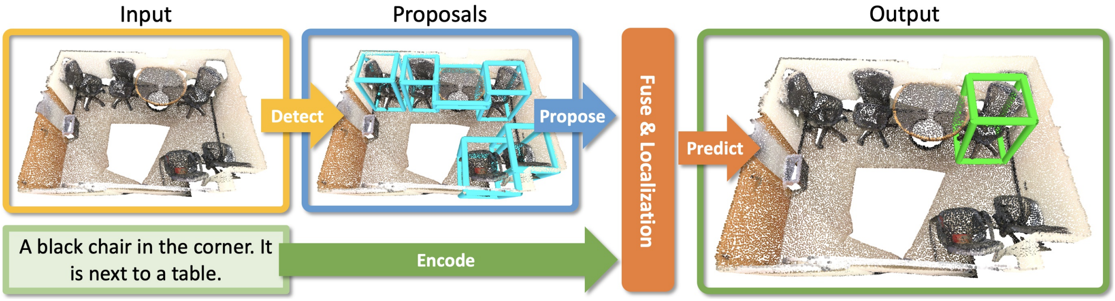


Figure 3.2: Our task: ScanRefer takes as input a 3D scene point cloud and a description of an object in the scene, and predicts the object bounding box.

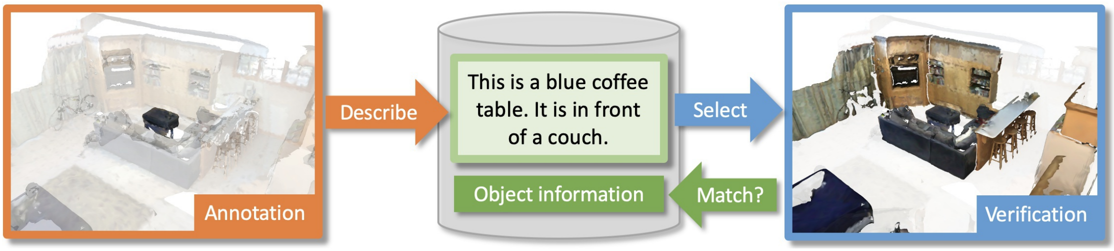


Figure 3.3: Our data collection pipeline. The annotator writes a description for the focused object in the scene. Then, a verifier selects the objects that match the description. The selected object is compared with the target object to check that it can be uniquely identified by the description.

3D Vision and Language. Vision and language research is gaining popularity in image domains (e.g., image captioning [32]–[34], [72], image-text matching [73]–[77], and text-to-image generation [77]–[79]), but there is little work on vision and language in 3D. Chen et al. [80] learn a joint embedding of 3D shapes from ShapeNet [17] and corresponding natural language descriptions. Achlioptas et al. [81] disambiguate between different objects using language. Recent work has started to investigate grounding of language to 3D by identifying 3D bounding boxes of target objects for simple arrangements of primitive shapes of different colors [82]. Instead of focusing on isolated objects, we consider large 3D RGB-D reconstructions that are typical in semantic 3D scene understanding.

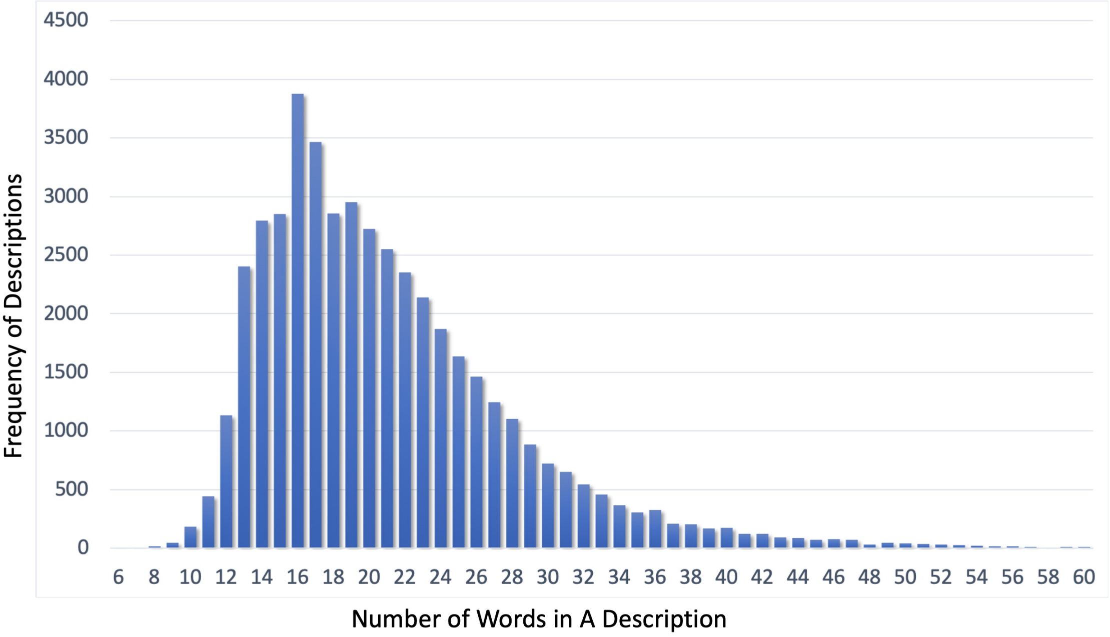


Figure 3.4: Description lengths.

A closely related work by Kong et al. [30] studied the problem of coreference in text description of single-view RGB-D images of scenes, where they aimed to connect noun phrases in a scene description to 3D bounding boxes of objects.

3.3 Task

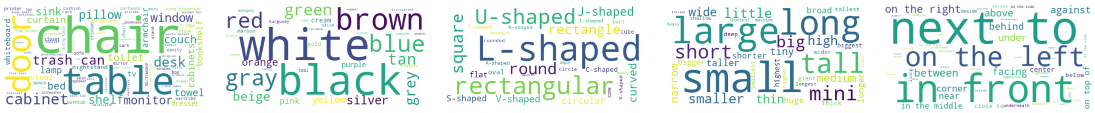
We introduce the task of object localization in 3D scenes using natural language (Fig. 3.2). The input is a 3D scene and free-form text describing an object in the scene. The scene is represented as a point cloud with additional features such as colors and normals for each point. The goal is to predict the 3D bounding box of the object that matches the input description.

3.4 Dataset

The ScanRefer dataset is based on ScanNet [13] which is composed of 1,613 RGB-D scans taken in 806 unique indoor environments. We provide 5 descriptions for each object in each scene, focusing on complete coverage of all objects that are present in the reconstruction. Here, we summarize the annotation process and statistics of our dataset (see supplement for more details).

Number of descriptions	51,583
Number of scenes	800
Number of objects	11,046
Number of objects per scene	13.81
Number of descriptions per scene	64.48
Number of descriptions per object	4.67
Size of vocabulary	4,197
Average length of descriptions	20.27

Table 3.2: ScanRefer dataset statistics.



(a) (b) (c) (d) (e)

Figure 3.5: Word clouds of terms for (a) object names (b) colors (c) shapes (d) sizes, and (e) spatial relations for the ScanRefer dataset. Bigger fonts indicate more frequent terms in the descriptions.

3.4.1 Data Collection

We deploy a web-based annotation interface on Amazon Mechanical Turk (AMT) to collect object descriptions in the ScanNet scenes. The annotation pipeline consists of two stages: i) description collection, and ii) verification (Fig. 3.3). From each scene, we select objects to annotate by restricting to indoor furniture categories and excluding structural objects such as "Floor" and "Wall". We manually check the selected objects are recognizable and filter out objects with reconstructions that are too incomplete or hard to identify.

Annotation The 3D web-based UI shows each object in context. The workers see all objects other than the target object faded out and a set of captured image frames to compensate for incomplete details in the reconstructions. The initial viewpoint is random but includes the target object. Camera controls allow for adjusting the camera view to better examine the target object. We ask the annotator to describe the appearance of the target and its spatial location relative to other objects. To ensure the descriptions are informative, we require the annotator to provide at least two full sentences. We batch and randomize the tasks so that each object is described by five different workers.

Verification We recruit trained workers (students) to verify that the descriptions are discriminative and correct. Verifiers are shown the 3D scene and a description, and are asked to select the objects (potentially multiple) in the scene that match the description. Descriptions that result in the wrong object or multiple objects are filtered out. Verifiers also correct spelling and wording issues in the description when necessary. We filter out 2,823 invalid descriptions that do not match the target objects and fix writing issues for 2,129 descriptions.

1. There is a brown wooden chair placed right against the wall.
2. This is a triangular shape table. The table is near the armchair.
3. The little nightstand. The nightstand is on the right of the bed.
4. This is a short trash can. It is in front of a taller trash can.
5. The couch is the biggest one below the picture. The couch has three seats and is brown.
6. This is a gray desk chair. This chair is the last one on the side closest to the open door.
7. The kitchen counter is covering the lower cabinets. The kitchen counter is under the upper cabinets that are mounted above.
8. This is a round bar stool. It is third from the wall.

Table 3.3: Examples from our dataset illustrating different types of phrases such as attributes (1-8) and parts (5), comparatives (4), superlatives (5), intra-class spatial relations (6), inter-class spatial relations (7) and ordinal numbers (8).

3.4.2 Dataset Statistics

We collected 51,583 descriptions for 800 ScanNet scenes[1]. On average, there are 13.81 objects, 64.48 descriptions per scene, and 4.67 descriptions per object after filtering (see Tab. 3.2 for basic statistics, Tab. 3.3 for sample descriptions, and Fig. 3.4 for the distribution of the description lengths). The descriptions are complex and diverse, covering over 250 types of common indoor objects, and exhibiting interesting linguistic phenomena. Due to the complexity of the descriptions, one of the key challenges of our task is to determine what parts of the description describe the target object, and what parts describe neighboring objects. Among those descriptions, 41,034 mention object attributes such as color, shape, size, etc. We find that many people use spatial language (98.7%), color (74.7%), and shape terms (64.9%). In contrast, only 14.2% of the descriptions convey size information. Fig 3.5 shows commonly used object names and attributes. Tab. 3.3 shows interesting expressions, including comparatives ("taller") and superlatives ("the biggest one"), as well as phrases involving ordinals such as "third from the wall". Overall, there are 672 and 2,734 descriptions with comparative and superlative phrases. We provide more detailed statistics in the supplement.

3.5 Method

Our architecture consists of two main modules: 1) detection & encoding; 2) fusion & localization (Fig. 5.3). The detection & encoding module encodes the input point cloud and description, and outputs the object proposals and the language embedding, which are fed into the fusion module to mask out invalid object proposals and produce the fused features. Finally, the object proposal with the highest confidence predicted by the localization module is chosen as the final output.

[1] 6 scenes are excluded since they do not contain any objects to describe

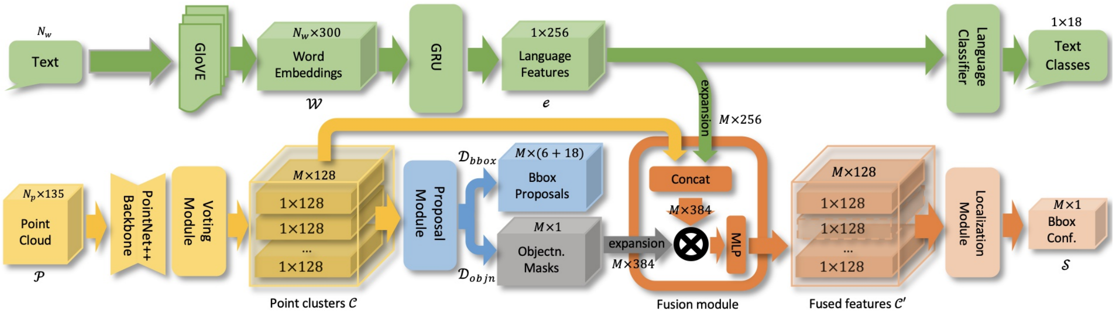


Figure 3.6: ScanRefer architecture: The PointNet++ [19] backbone takes as input a point cloud and aggregates it to high-level point feature maps, which are then clustered and fused as object proposals by a voting module similar to Qi et al. [24]. Object proposals are masked by the objectness predictions, and then fused with the sentence embedding of the input descriptions, which is obtained by a GloVE [18] + GRU [85] embedding. In addition, an extra language-to-object classifier serves as a proxy loss. We apply a softmax function in the localization module to output the confidence scores for the object proposals.

3.5.1 Data Representations

Point clouds We randomly sample N_P vertices of one scan from ScanNet as the input point cloud $\mathcal{P} = \{(p_i, f_i)\}$, where $p_i \in \mathcal{R}^3$ represents the point coordinates in 3D space and f_i stands for additional point features such as colors and normals. Note that the point coordinates p_i provides only geometrical information and does not contain other visual information such as color and texture. Since descriptions of objects do refer to attributes such as color and texture, we incorporate visual appearance by adapting the feature projection scheme in Dai et al. [23] to project multi-view image features $v_i \in \mathcal{R}^{128}$ to the point cloud. The image features are extracted using a pre-trained ENet [83]. Following Qi et al. [24], we also append the height of the point from the ground and normals to the new point features $f_i' \in \mathcal{R}^{135}$. The final point cloud data is prepared offline as $\mathcal{P}' = \{(p_i, f_i')\} \in \mathcal{R}^{N_P \times 135}$. We set N_P to 40,000 in our experiments.
Descriptions We tokenize the input description with SpaCy [84] and the N_W tokens to 300-dimensional word embedding vectors $\mathcal{W} = \{w_j\} \in \mathcal{R}^{N_W \times 300}$ using pretrained GloVE word embeddings [18].

3.5.2 Network Architecture

Our method takes as input the preprocessed point cloud $\mathcal{P}'$ and the word embedding sequence $\mathcal{W}$ representing the input description and outputs the 3D bounding box for the proposal which is most likely referred to by the input description. Conceptually, our localization pipeline consists of the following four stages: detection, encoding, fusion and localization.
Detection As the first step in our network, we detect all probable objects in the given point cloud. To construct our detection module, we adapt the PointNet++ [19] backbone

and the voting module in Qi et al. [24] to process the point cloud input and aggregate all object candidates to individual clusters. The output from the voting module is a set of point clusters $\mathcal{C} \in \mathcal{R}^{M \times 128}$ representing all object proposals with enriched point features, where M is the upper bound of the number of proposals. Next, the proposal module takes in the point clusters and processes those clusters to predict the objectness mask $\mathcal{D}_{\text{objn}} \in \mathcal{R}^{M \times 1}$ and the axis-aligned bounding boxes $\mathcal{D}_{\text{bbox}} \in \mathcal{R}^{M \times (6+18)}$ for all M proposals, where each $\mathcal{D}^i_{\text{bbox}} = (c_x, c_y, c_z, r_x, r_y, r_z, l)$ consists of the box center c, the box lengths r and a vector $l \in \mathcal{R}^{18}$ representing the semantic predictions.

Encoding The sequences of word embedding vectors of the input description are fed into a GRU cell [85] to aggregate the textual information. We take the final hidden state $e \in \mathcal{R}^{256}$ of the GRU cell as the final language embedding.

Fusion The outputs from the previous detection and encoding modules are fed into the fusion module (orange block in Fig. 5.3, see supplemental for details) to integrate the point features together with the language embeddings. Specifically, each feature vector $c_i \in \mathcal{R}^{128}$ in the point cluster $\mathcal{C}$ is concatenated with the language embedding $e \in \mathcal{R}^{256}$ as the extended feature vector, which is then masked by the predicted objectness mask $\mathcal{D}^i_{\text{objn}} \in \{0, 1\}$ and fused by a multi-layer perceptron as the the final fused cluster features $C' = \{c'_i\} \in \mathcal{R}^{M \times 128}$.

Localization The localization module aims to predict which of the proposed bounding boxes corresponds to the description. Point clusters with fused cluster features $\mathcal{C}' = \{c'_i\}$ are processed by a single layer perceptron to produce the raw scores of how likely each box is the target box. We use a softmax function to squash all the raw scores into the interval of $[0, 1]$ as the localization confidences $S = \{s_i\} \in \mathcal{R}^{M \times 1}$ for the proposed M bounding boxes.

3.5.3 Loss Function

Localization loss For the predicted localization confidence $s_i \in [0, 1]$ for object proposal $\mathcal{D}^i_{\text{bbox}}$, the target label is represented as $t_i \in \{0, 1\}$. Following the strategy of Yang et al. [27], we set the label t_j for the j^{th} box that has the highest IoU score with the ground truth box as 1 and others as 0. We then use a cross-entropy loss as the localization loss $\mathcal{L}_{\text{loc}} = -\sum_{i=1}^{M} t_i \log(s_i)$.

Object detection loss We use the same detection loss $\mathcal{L}_{det}$ as introduced in Qi et al. [24] for object proposals $\mathcal{D}^i_{\text{bbox}}$ and $\mathcal{D}^i_{\text{objn}}$: $\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{vote-reg}} + 0.5\mathcal{L}_{\text{objn-cls}} + \mathcal{L}_{\text{box}} + 0.1\mathcal{L}_{\text{sem-cls}}$, where $\mathcal{L}_{\text{vote-reg}}$, $\mathcal{L}_{\text{objn-cls}}$, $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{sem-cls}}$ represent the vote regression loss (defined in Qi et al. [24]), the objectness binary classification loss, box regression loss and the semantic classification loss for the 18 ScanNet benchmark classes, respectively. We ignore the bounding box orientations in our task and simplify $\mathcal{L}_{\text{box}}$ as $\mathcal{L}_{\text{box}} = \mathcal{L}_{\text{center-reg}} + 0.1\mathcal{L}_{\text{size-cls}} + \mathcal{L}_{\text{size-reg}}$, where $\mathcal{L}_{\text{center-reg}}$, $\mathcal{L}_{\text{size-cls}}$ and $\mathcal{L}_{\text{size-reg}}$ are used for regressing the box center, classifying the box size and regressing the box size, respectively. We refer readers to Qi et al. [24] for more details.

Language to object classification loss To further supervise the training, we include an object classification loss based on the input description. We consider the 18 ScanNet benchmark classes (excluding the label “Floor” and “Wall”). The language to object classification loss $\mathcal{L}_{\text{cls}}$ is a multi-class cross-entropy loss.

Final loss The final loss is a linear combination of the localization loss, object detection loss and the language to object classification loss: $\mathcal{L} = \alpha\mathcal{L}_{\text{loc}} + \beta\mathcal{L}_{\text{det}} + \gamma\mathcal{L}_{\text{cls}}$, where α, β and γ are the weights for the individual loss terms. After fine-tuning on the validation split, we set those weights to 1, 10, and 10 in our experiments to ensure the loss terms are roughly of the same magnitude.

3.5.4 Training and Inference

Training During training, the detection and encoding modules propose object candidates as point clusters, which are then fed into the fusion and localization modules to fuse the features from the previous module and predict the final bounding boxes. We train the detection backbone end-to-end with the detection loss. In the localization module, we use a softmax function to compress the raw scores to $[0, 1]$. The higher the predicted confidence is, the more likely the proposal will be chosen as output. To filter out invalid object proposals, we use the predicted objectness mask to ensure that only positive proposals are taken into account. We set the maximum number of proposals M to 256 in practice.

Inference Since there can be overlapping detections, we apply a non-maximum suppression module to suppress those overlapping proposals in the inference step. The remaining object proposals are fed into the localization module to predict the final score for each proposal. The number of object proposals is less than the upper bound M in the training step.

Implementation Details We implement our architecture using PyTorch and train the model end-to-end using ADAM [86] with a learning rate of 1e$-$3. We train the model for roughly $130,000$ iterations until convergence. To avoid overfitting, we set the weight decay factor to 1e$-$5 and apply data augmentations to our training data. For point clouds, we apply rotation about all three axes by a random angle in $[-5°, 5°]$ and randomly translate the point cloud within 0.5 meters in all directions. We rotate around all axes (not just up), since the ground alignment in ScanNet is imperfect.

3.6 Experiments

Train/Val/Test Split. Following the official ScanNet [13] split, we split our data into train/val/test sets with 36,665, 9,508 and 5,410 samples respectively, ensuring disjoint scenes for each split. Results and analysis are conducted on the val split (except for results in Tab. 4.1 bottom). The test set is hidden and will be reserved for the ScanRefer benchmark.

Metric. To evaluate the performance of our method, we measure the thresholded accuracy where the positive predictions have higher intersection over union (IoU) with the ground truths than the thresholds. Similar to work with 2D images, we use Acc@kIoU as our metric, where the threshold value k for IoU is set to 0.25 and 0.5 in our experiments.

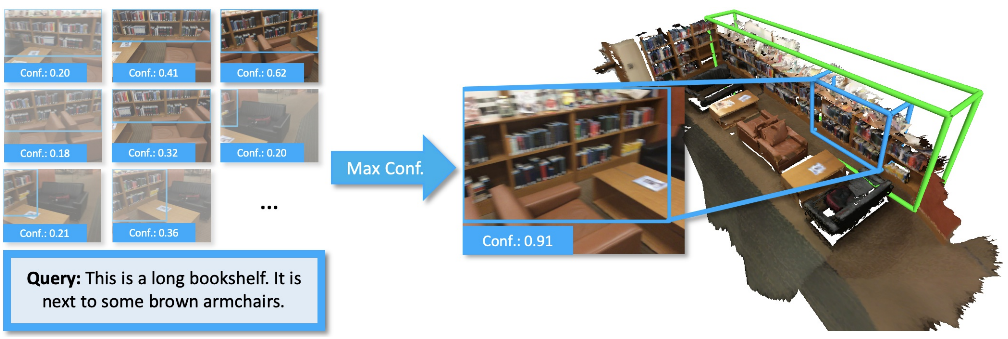


Figure 3.7: Object localization in an image using a 2D grounding method and back-projecting the result to the 3D scene (blue box) vs. directly localizing in the 3D scene (green box). Grounding in 2D images suffers from the limited view of a single frame, which results in inaccurate 3D bounding boxes.

Baselines. We design several baselines by 1) evaluating our language localization module on ground truth bounding boxes, 2) adapting 3D object detectors, and 3) adapting 2D referring methods to 3D using back-projection.

OracleCatRand & OracleRefer: To examine the difficulty of our task, we use an oracle with ground truth bounding boxes of objects, and predict the box by simply selecting a random box that matches the object category (OracleCatRand) or our trained fusion and localization modules (OracleRefer).

VoteNetRand & VoteNetBest: From the predicted object proposals of the VoteNet backbone [24], we select one of the bounding box proposals, either by selecting a box randomly with the correct semantic class label (VoteNetRand) or the best matching box given the ground truth (VoteNetBest). VoteNetBest provides an upper bound on how well the object detection component works for our task, while VoteNetRand provides a measure of whether additional information beyond the semantic label is required.

SCRC & One-stage: 2D image baselines for referring expression comprehension by extending SCRC [7] and One-stage [27] to 3D using back-projection. Since 2D referring expression methods operate on a single image frame, we construct a 2D training set by using the recorded camera pose associated with each annotation to retrieve the frame from the scan video with the closest camera pose. At inference time, we sample frames from the scans (using every 20th frame) and predict the target 2D bounding boxes in each frame. We then select the 2D bounding box with the highest confidence score from the bounding box candidates and project it to 3D using the depth map for that frame (see Fig. 3.7).

Ours: We compare our full end-to-end model against using a pretrained VoteNet backbone with a trained GRU [85] for selecting a matching bounding box.

3.6.1 Task Difficulty

To understand how informative the input description is beyond capturing the object category, we analyze the performance of the methods on "unique" and "multiple" subsets

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
OracleCatRand (GT boxes + RandCat)	100.00	100.00	18.09	17.84	29.99	29.76
OracleRefer (GT boxes + GRU)	74.09	73.55	32.57	32.00	40.63	40.06
VoteNetRand (VoteNet[24] + RandCat)	34.34	19.35	5.73	2.81	10.00	5.28
VoteNetBest (VoteNet[24] + Best)	88.85	85.50	46.63	46.42	55.10	54.33
SCRC [7] + backproj	24.03	9.22	17.77	5.97	18.70	6.45
One-stage [27] + backproj	29.32	22.82	18.72	6.49	20.38	9.04
Ours (VoteNet[48] + GRU)	55.09	37.66	26.37	16.03	32.49	20.53
Ours (end-to-end)	**63.04**	**39.95**	**28.91**	**18.17**	**35.53**	**22.39**
Test results (ScanRefer benchmark)						
OracleRefer (GT boxes + GRU)	72.37	71.84	31.81	31.26	39.69	39.13
VoteNetBest (VoteNet[24] + Best)	86.78	83.85	45.54	45.33	53.82	53.07
Ours (VoteNet[48] + GRU)	57.67	36.96	28.31	15.16	34.90	20.05
Ours (end-to-end)	**62.90**	**40.31**	**30.88**	**16.54**	**38.06**	**21.87**

Table 3.4: Comparison of localization results obtained by our ScanRefer and baseline models. We measure percentage of predictions whose IoU with the ground truth boxes are greater than 0.25 and 0.5. We also report scores on "unique" and "multiple" subsets; unique means that there is only a single object of its class in the scene. We outperform all baselines by a significant margin.

with 1,875 and 7,663 samples from val split, respectively. The "unique" subset contains samples where only one unique object from a certain category matches the description, while the "multiple" subset contains ambiguous cases where there are multiple objects of the same category. For instance, if there is only one refrigerator in a scene, it is sufficient to identify that the sentence refers to a refrigerator. In contrast, if there are multiple objects of the same category in a scene (e.g., chair), the full description must be taken into account. From the OracleCatRand baseline, we see that information from the description, other than the object category, is necessary to disambiguate between multiple objects (see Tab. 4.1 Acc@0.5IoU multiple). From the OracleRefer baseline, we see that using our fused language module, we are able to improve beyond over selecting a random object of the same category (multiple Acc@0.5IoU increases from 17.84% to 32.00%), but we often fail to identify the correct object category (unique Acc@0.5IoU drops from 100.0% to 73.55%).

3.6.2 Quantitative Analysis

We evaluate the performance of our model against baselines on the val and the hidden test split of ScanRefer which serves as the ScanRefer benchmark (see Tab. 4.1). Note that for all results using Ours and VoteNet for object proposal, we take the average of 5 differently seeded subsamplings (of seed points and vote points) during inference (see supplemental for more details on experimental variance). Training the detection backbone jointly with the localization module (end-to-end) leads to a better performance when compared to the model trained separately (VoteNet[24] + GRU). However, as the

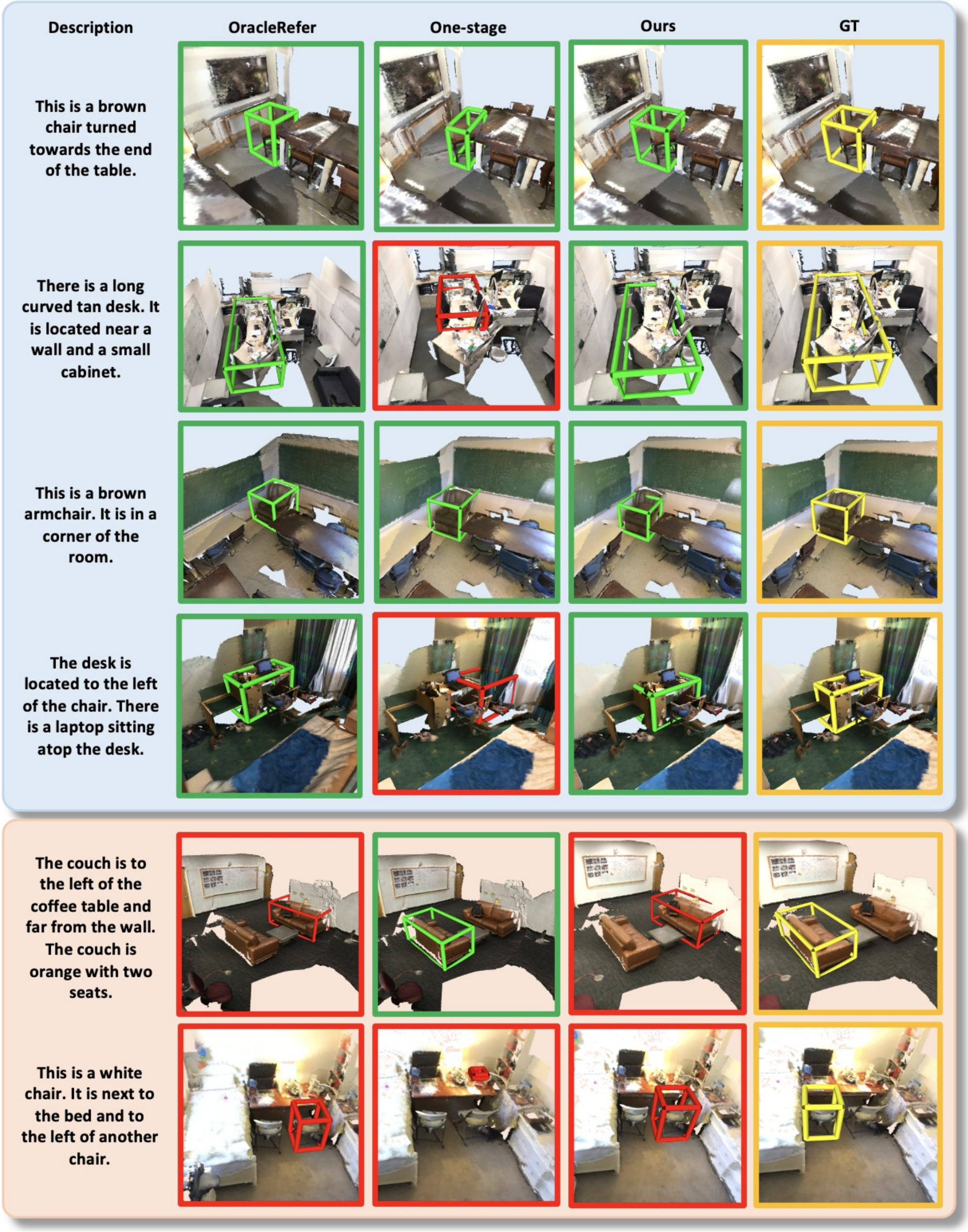

Description
OracleRefer
One-stage
Ours
GT
This is a brown chair turned towards the end of the table.
There is a long curved tan desk. It is located near a wall and a small cabinet.
This is a brown armchair. It is in a corner of the room.
The desk is located to the left of the chair. There is a laptop sitting atop the desk.
The couch is to the left of the coffee table and far from the wall. The couch is orange with two seats.
This is a white chair. It is next to the bed and to the left of another chair.

Figure 3.8: Qualitative results from baseline methods and ScanRefer. Predicted boxes are marked green if they have an IoU score higher than 0.5, otherwise they are marked red. We show examples where our method produced good predictions (blue block) as well as failure cases (orange block). Image best viewed in color.

accuracy gap between VoteNetBest and ours (end-to-end) indicates, there is still room for improving the match between language inputs and the visual signals. For the val

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Ours (xyz)	50.83	31.81	24.38	13.98	29.51	17.43
Ours (xyz+rgb)	51.22	32.09	24.50	14.51	29.68	17.92
Ours (xyz+rgb+normals)	54.24	33.71	25.44	15.53	31.05	19.05
Ours (xyz+multiview)	56.69	35.32	25.83	14.26	31.63	19.75
Ours (xyz+multiview+normals)	55.27	35.51	25.95	16.29	31.64	20.02
Ours (xyz+lobjcls)	58.92	35.01	28.27	16.99	34.21	20.49
Ours (xyz+rgb+lobjcls)	60.11	37.89	27.21	16.49	33.59	20.65
Ours (xyz+rgb+normals+lobjcls)	60.54	39.19	26.95	16.69	33.47	21.06
Ours (xyz+multiview+lobjcls)	61.16	39.02	26.49	16.69	34.71	21.87
Ours (xyz+multiview+normals+lobjcls)	**63.04**	**39.95**	**28.91**	**18.17**	**35.53**	**22.39**

Table 3.5: Ablation study with different features. We measure the percentages of predictions whose IoU with the ground truth boxes are greater than 0.25 and 0.5. Unique means that there is only a single object of its class in the scene.

split, we also include additional experiments on the 2D baselines and a comparison with VoteNetRand. With just category information, VoteNetRand is able to perform relatively well on the "unique" subset, but has trouble identifying the correct object in the "multiple" case. However, the gap between the VoteNetRand and OracleCatRand for the "unique" case shows that 3D object detection still need to be improved. Our method is able to improve over the bounding box predictions from VoteNetRand, and leverages additional information in the description to differentiate between ambiguous objects. It adapts better to the 3D context compared to the 2D methods (SCRC and One-stage) which is limited by the view of a single frame (see Fig. 3.7 and Fig. 4.6).

3.6.3 Qualitative Analysis

Fig. 4.6 shows results produced by OracleRefer, One-stage, and our method. The successful localization cases in the green boxes show our architecture can handle the semantic correlation between the scene contexts and the textual descriptions. In contrast, even provided with a pool of ground truth proposals, OracleRefer sometimes still fails to predict correct bounding boxes, while One-stage is limited by the single view and hence cannot produce accurate bounding boxes in 3D space. The failure case of OracleRefer suggests that our fusion & localization module can still be improved. Some failure cases of our method are displayed in the orange block in Fig. 4.6, indicating that our architecture cannot handle all spatial relations to distinguish between ambiguous objects.

3.6.4 Ablation Studies

We conduct an ablation study on our model to examine what components and point cloud features contribute to the performance (see Tab. 3.5).

Does a language-based object classifier help? To show the effectiveness of the extra supervision on input descriptions, we conduct an experiment with the language to object classifier (+lobjcls) and without. Architectures with a language to object classifier outperform ones without it. This indicates that it is helpful to predict the category of the target object based on the input description.

Do colors help? We compare our method trained with the geometry and multi-view image features (xyz+multiview+lobjcls) with a model trained with only geometry (xyz+lobjcls) and one trained with RGB values from the reconstructed meshes (xyz+rgb+lobjcls). ScanRefer trained with geometry and pre-processed multi-view image features outperforms the other two models. The performance of models with color information are higher than those that use only geometry.

Do other features help? We include normals from the ScanNet meshes to the input point cloud features and compare performance against networks trained without them. The additional 3D information improves performance. Our architecture trained with geometry, multi-view features, and normals (xyz+multiview+ normals+lobjcls) achieves the best performance among all ablations.

3.7 Conclusion

In this work, we introduce the task of localizing a target object in a 3D point cloud using natural language descriptions. We collect the ScanReferdataset which contains 51,583 unique descriptions for 11,046 objects from 800 ScanNet [13] scenes. We propose an end-to-end method for localizing an object with a free-formed description as reference, which first proposes point clusters of interest and then matches them to the embeddings of the input sentence. Our architecture is capable of learning the semantic similarities of the given contexts and regressing the bounding boxes for the target objects. Overall, we hope that our new dataset and method will enable future research in the 3D visual language field.

4 Generating Descriptions for 3D Objects in RGB-D Scans

This chapter introduces the following paper:

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

Abstract of the paper We introduce the task of dense captioning in 3D scans from commodity RGB-D sensors. As input, we assume a point cloud of a 3D scene; the expected output is the bounding boxes along with the descriptions for the underlying objects. To address the 3D object detection and description problems, we propose Scan2Cap, an end-to-end trained method, to detect objects in the input scene and describe them in natural language. We use an attention mechanism that generates descriptive tokens while referring to the related components in the local context. To reflect object relations (i.e. relative spatial relations) in the generated captions, we use a message passing graph module to facilitate learning object relation features. Our method can effectively localize and describe 3D objects in scenes from the ScanRefer dataset, outperforming 2D baseline methods by a significant margin (**27.61% CiDEr@0.5IoU** improvement).

Contribution The method development and implementation was done by the first author. Discussions with the co-authors led to the final paper.

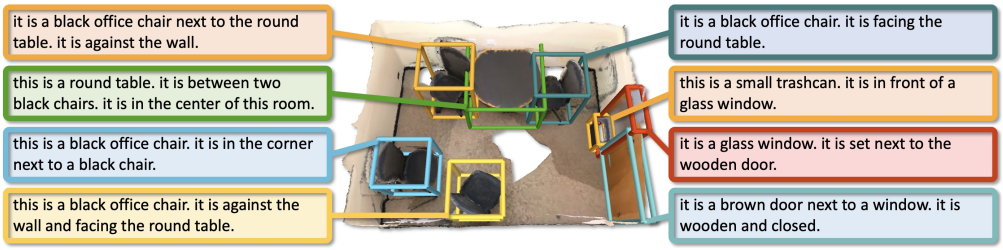

it is a black office chair next to the round table. it is against the wall.
this is a round table. it is between two black chairs. it is in the center of this room.
this is a black office chair. it is in the corner next to a black chair.
this is a black office chair. it is against the wall and facing the round table.
it is a black office chair. it is facing the round table.
this is a small trashcan. it is in front of a glass window.
it is a glass window. it is set next to the wooden door.
it is a brown door next to a window. it is wooden and closed.

Figure 4.1: We introduce the task of dense captioning in RGB-D scans with a model that can densely localize objects in a 3D scene and describe them using natural language in a single forward pass.

4.1 Introduction

The intersection of visual scene understanding [5], [6] and natural language processing [21], [87] is a rich and active area of research. Specifically, there has been a lot of work on image captioning [32]–[34], [72], [88] and the related task of dense captioning [28], [32], [89]–[92]. In dense captioning, individual objects are localized in an image and each object is described using natural language. So far, dense captioning work has operated purely on 2D visual data, most commonly single-view images that are limited by the field of view. Images are inherently viewpoint specific and scale agnostic, and fail to capture the physical extent of 3D objects (i.e. the actual size of the objects) and their locations in the environment.

In this work, we introduce the new task of dense captioning in 3D scenes. We aim to jointly localize and describe each object in a 3D scene. We show that leveraging the 3D information of an object such as actual object size or object location results in more accurate descriptions.

Apart from the 2D constraints in images, even seminal work on dense captioning suffers from *aperture* issues [90]. Object relations are often neglected while describing scene objects, which makes the task more challenging. We address this problem with a graph-based attentive captioning architecture that jointly learns object features and object relation features on the instance level, and generates descriptive tokens. Specifically, our proposed method (referred to as Scan2Cap) consists of two critical components: 1) *Relational Graph* facilitates learning the object features and object relation features using a message passing neural network; 2) *Context-aware Attention Captioning* generates the descriptive tokens while attending to the object and object relation features. In summary, our contribution is fourfold:

- We introduce the 3D dense captioning task to densely detect and describe 3D objects in RGB-D scans.
- We propose a novel message passing graph module that facilitates learning of the 3D object features and 3D object relation features.

- We propose an end-to-end trained method that can take 3D object features and 3D object relation features into account when describing the 3D object in a single forward pass.
- We show that our method outperforms 2D-3D back-projected results of 2D captioning baselines by a significant margin (**27.61%**).

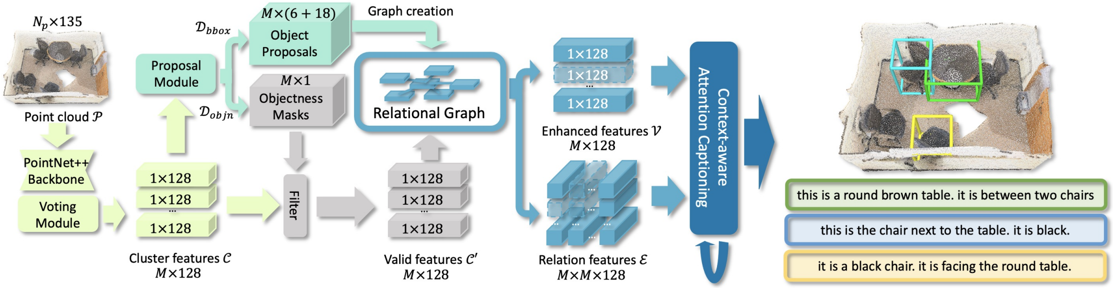

N_p×135
Point cloud $\mathcal{P}$
PointNet++ Backbone
Voting Module
Proposal Module
$\mathcal{D}_{bbox}$
$\mathcal{D}_{objn}$
M×(6 + 18)
Object Proposals
M×1
Objectness Masks
Graph creation
Relational Graph
1×128
1×128
...
1×128
Cluster features $\mathcal{C}$
M×128
Filter
1×128
1×128
...
1×128
Valid features $\mathcal{C}'$
M×128
1×128
1×128
...
1×128
Enhanced features $\mathcal{V}$
M×128
Relation features $\mathcal{E}$
M×M×128
Context-aware Attention Captioning
this is a round brown table. it is between two chairs
this is the chair next to the table. it is black.
it is a black chair. it is facing the round table.

Figure 4.2: Scan2Cap takes as input a point cloud to generate the cluster features $\mathcal{C}$ for the proposal module, using a backbone following PointNet++ [19] and a voting module similar to [24]. The proposal module predicts the object proposals $\mathcal{D}_{\text{bbox}}$ as well as the objectness masks $\mathcal{D}_{\text{objn}}$, which are later used for filtering the cluster features as the valid features $\mathcal{C}^{'}$. A graph is then constructed using the object proposals and the valid cluster features. The relational graph module takes in the graph and outputs the enhanced object features $\mathcal{V}$ and the relation features $\mathcal{C}^{'}$. As the last step, the context-aware attention captioning module, inspired by [88], generates descriptive tokens for each object proposal using the enhanced features and the relation features.

4.2 Related work

4.2.1 3D Object Detection

There are many methods for 3D object detection on 3D RGB-D datasets [13], [66], [67], [93]. Methods utilizing 3D volumetric grids have achieved impressive performance [63]–[65], [94], [95]. At the same time, methods operating on point clouds serve as an alternative and also achieve impressive results. For instance, [24] use a Hough voting scheme to aggregate points and generate object proposals while using a PointNet++ [19] backbone. Following this work, [96] recently proposed a pipeline to jointly perform voting in both point clouds and associated images. Our method builds on these works as we utilize the same backbone for processing the input geometry; however, we back-project multi-view image features to point clouds to leverage the original RGB input, since appearance is critical for accurately describing the target objects in the scene.

4.2.2 Image Captioning

Image captioning has attracted a great deal of interest [32]–[34], [72], [88], [97]–[99]. Attention based captioning over grid regions [34], [72] and over detected objects [88], [100] allows focusing on specific image regions while captioning. One recent trend is the attempt to capture relationships between objects using attention and graph neural networks [101]–[103] or transformers [104]. We build on these ideas to propose a 3D captioning network with graphs that capture object relations in 3D.

The dense captioning task introduced by [28] is closely related to our task. This task is a variant of image captioning where captions are generated for all detected objects. While achieving impressive results, this method does not consider the context outside of the salient image regions. To tackle this issue, [89] include the global image feature as context to the captioning input. [91] explicitly model the relations between detected regions in the image. Due to the limited view of a single image, prior work on 2D images could not capture the large context available in 3D environments. In contrast, we focus on decomposing the input 3D scene and capturing the appearance and spatial information of the objects in the 3D environment.

4.2.3 3D Vision and Language

While the joint field of vision and language has received much attention in the image domain, in tasks such as image captioning [32]–[34], [72], [88], [97]–[99], dense captioning [28], [89], [91], text-to-image generation [77]–[79], visual grounding [7], [11], [37], vision and language in 3D is still not well-explored. [80] introduces a dataset which consists of descriptions for ShapeNet [17] objects, enabling text-to-shape generation and shape captioning. On the scene level, [14] propose a dataset for localizing object in ScanNet [13] scenes using natural language expressions. Concurrently, [105] propose another dataset for distinguishing fine-grained objects in ScanNet scenes using natural language queries. This work enables research on connecting natural language to 3D environments, and inspires our work to densely localize and describe 3D objects with respect to the scene context.

4.3 Task

We introduce the task of dense captioning in 3D scenes. The input for this task is a point cloud of a scene, consisting of the object geometries as well as several additional point features such as RGB values and normal vectors. The expected output is the object bounding boxes for the underlying instances in the scene and their corresponding natural language descriptions.

4.4 Method

We propose an end-to-end architecture on the input point clouds to address the 3D dense description generation task. Our architecture consists of the following main components:

1) detection backbone; 2) relational graph; 3) context-aware attention captioning. As Fig. 5.3 shows, our network takes a point cloud as input, and generates a set of 3D object proposals using the detection module. A relational graph module then enhances object features using contextual cues and provides object relation features. Finally, a context-aware attention module generates descriptions from the enhanced object and relation features.

4.4.1 Data Representation

As input to the detection module, we assume a point cloud $\mathcal{P}$ of a scan from ScanNet consisting of the geometry coordinates and additional point features capturing the visual appearance and the height from the ground. To obtain the extended visual point features, we follow [14] and adapt the feature projection scheme of [23] to back-project multi-view image features to the point cloud as additional features. The image features are extracted using a pre-trained ENet [83]. Following [24], we also append the height of the point from the ground to the new point features. As a result, we represent the final point cloud data as $\mathcal{P} = \{(p_i, f_i)\} \in \mathcal{R}^{N_P \times 135}$, where $p_i \in \mathcal{R}^3, i = 1, ..., N_P$ are the coordinates and $f_i \in \mathcal{R}^{132}$ are the additional features.

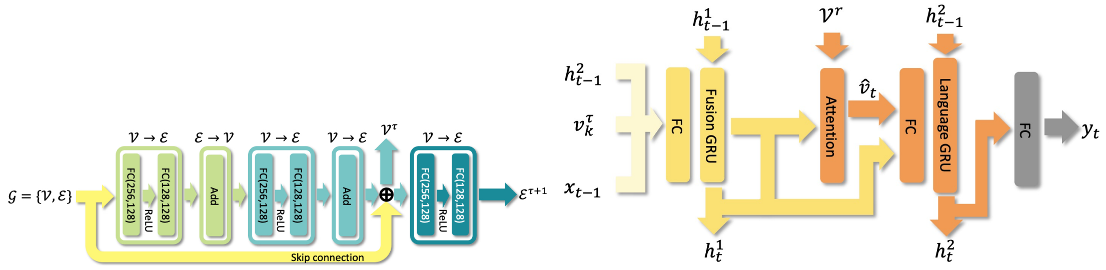
4.4.2 Detection Backbone

As the first step in our network, we detect all probable objects in the given point cloud with the back-projected multi-view image features discussed in 4.4.1. To construct our detection module, we adapt the PointNet++ [19] backbone and the voting module in VoteNet [24] to aggregate all object candidates to individual clusters. The output from the voting module is a set of point clusters $\mathcal{C} \in \mathcal{R}^{M \times 128}$ representing all object proposals with enriched point features, where M is the upper bound of the number of proposals. Next, the proposal module takes in the point clusters to predict the objectness mask $\mathcal{D}_{\text{objn}} \in \mathcal{R}^{M \times 1}$ and the axis-aligned bounding boxes $\mathcal{D}_{\text{bbox}} \in \mathcal{R}^{M \times (6+18)}$ for all M proposals, where each $\mathcal{D}^i_{\text{bbox}} = (c_x, c_y, c_z, r_x, r_y, r_z, l)$ consists of the box center c, the box lengths r and a vector $l \in \mathcal{R}^{18}$ representing the semantic predictions.

4.4.3 Relational Graph

Describing the object in the scene often involves its appearance and spatial location with respect to nearby objects. Therefore, we propose a relational graph module equipped with a message passing network to enhance the object features and extract the object relation features. We create a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where we treat the object proposals as nodes in the graph and relationship between objects as edges. For the edges, we consider only the nearest K objects surrounding each object. We use standard neural message passing [106] where the message passing at graph step τ is defined as follows:

$$\mathcal{V} \rightarrow \mathcal{E} : g^{\tau+1}_{i,j} = f^{\tau}([g^{\tau}_i, g^{\tau}_j - g^{\tau}_i]) \tag{4.1}$$



(a) Relational graph module.

(b) Context-aware attention captioning module.

Figure 4.3: (a) Context enhancement module takes in the scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and produces the enhanced object features $\mathcal{V}^\tau$ and object relation features $\mathcal{E}^{\tau+1}$; (b) At time step t, the context-aware captioning module takes in the enhanced features v^τ_k of the target object and generates the next token y_t with the help of attention mechanism on the attention context features $\mathcal{V}^r$.

where $g^\tau_i \in \mathcal{R}^{128}$ and $g^\tau_j \in \mathcal{R}^{128}$ are the features of nodes i and j at graph step τ. $g^{\tau+1}_{i,j} \in \mathcal{R}^{128}$ denotes the message between nodes i and j at the next graph step $\tau + 1$. $[\cdot, \cdot]$ concatenates two vectors. $f^\tau(\cdot)$ is a learnable non-linear function, which is in practice set as an MLP. The aggregated node features from messages after every message passing step is defined as $\mathcal{E} \rightarrow \mathcal{V} : g^{\tau+1}_i = \sum^K_{k=1} g^\tau_{i,k}$. We take the node features $\mathcal{V}^\tau$ in the last graph step τ as the output enhanced object features. We append an additional message passing layer after the last graph step and use the learned message $\mathcal{E}^{\tau+1}$ as the output object relation features. An MLP is attached to the output message passing layer to predict the angular deviations between two objects. We illustrate the relational graph module in Fig. 4.3a.

4.4.4 Context-aware Attention Captioning

Inspired by [88], we design a context aware attention captioning module which takes both the enhanced object features and object relation features and generates the caption one token at a time, as shown in Fig. 4.3b.

Fusion GRU. At time-step t of caption generation, we first concatenate three vectors as the fused input feature u^1_{t-1}: GRU hidden state from time-step $t - 1$ denoted as $h^2_{t-1} \in R^{512}$, enhanced object feature $v^\tau_k \in R^{128}$ of the k^{th} object and GloVE [18] embedding of the token generated at $t - 1$ denoted as $x_t = W_e y_{t-1} \in R^{300}$. The Fusion GRU handles the fused input feature u^1_{t-1} and delivers the hidden state h^1_t to the attention module.

Attention module. Unlike the attention module in [88] which only considers object features, we include both the enhanced object features $\mathcal{V}^\tau = \{v^\tau_i\} \in \mathcal{R}^{M \times 128}$ as well as the object relation features $e_{k,j} \in \mathcal{R}^{128}$. We add each object relation feature $e_{k,j}$ between the object k and its neighbor j to the corresponding enhanced object feature v_j of the j^{th} object as the final attention context feature set $\mathcal{V}^r = \{v^r_1, ..., v^r_k, ..., v^r_M\}$. Intuitively, the attention module will attend to the neighbor objects and their associated relations with the current object. We define the intermediate attention distribution $\alpha_t \in \mathcal{R}^{M\times 128}$ over the context features as:

$$\alpha_t = \text{softmax}((\mathcal{V}^r W_v + 1_h h_{t-1}^{1T} W_h) W_a) 1_a \tag{4.2}$$

where $W_a \in \mathcal{R}^{128\times 1}$, $W_v \in \mathcal{R}^{128\times 128}$, $W_h \in \mathcal{R}^{512\times 128}$ are learnable parameters. $1_h \in \mathcal{R}^{M\times 1}$ and $1_a \in \mathcal{R}^{1\times 128}$ are identity matrices. Finally, the attention module outputs the aggregated context vector $\hat{v}_t = \sum_{i=1}^{M} \mathcal{V}_i^r \odot \alpha_{ti}$ to represent the attended object and inter-object relation.

Language GRU. We then concatenate the hidden state h_{t-1}^1 of the Fusion GRU in last time step and the aggregated context vector $\hat{v}_t$, and process them with a MLP as the fused feature u_t^2. The language GRU takes in the fused input u_t^2 and delivers the hidden state h_t^2 to the output MLP to predict token y_t at the current time step t.

4.4.5 Training Objective

Object detection loss. We use the same detection loss $\mathcal{L}_{det}$ as introduced in [24] for object proposals $\mathcal{D}_{\text{bbox}}$ and $\mathcal{D}_{\text{objn}}$: $\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{vote-reg}} + 0.5\mathcal{L}_{\text{objn-cls}} + \mathcal{L}_{\text{box}} + 0.1\mathcal{L}_{\text{sem-cls}}$, where $\mathcal{L}_{\text{vote-reg}}$, $\mathcal{L}_{\text{objn-cls}}$, $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{sem-cls}}$ represent the vote regression loss (defined in [24]), the objectness binary classification loss, box regression loss and the semantic classification loss for the 18 ScanNet benchmark classes, respectively. We ignore the bounding box orientations in our task and simplify $\mathcal{L}_{\text{box}}$ as $\mathcal{L}_{\text{box}} = \mathcal{L}_{\text{center-reg}} + 0.1\mathcal{L}_{\text{size-cls}} + \mathcal{L}_{\text{size-reg}}$, where $\mathcal{L}_{\text{center-reg}}$, $\mathcal{L}_{\text{size-cls}}$ and $\mathcal{L}_{\text{size-reg}}$ are used for regressing the box center, classifying the box size and regressing the box size, respectively. We refer readers to [24] for more details.

Relative orientation loss. To stabilize the learning process of the relational graph module, we apply a relative orientation loss $\mathcal{L}_{\text{ad}}$ on the message passing network as a proxy loss. We discretize the output angular deviations ranges from $0°$ to $180°$ into 6 classes, and use a cross entropy loss as our classification loss. We construct the ground truth labels using the transformation matrices of the aligned CAD models in Scan2CAD [107], and mask out objects not provided in Scan2CAD in the loss function.

Description loss. The main objective loss constrains the description generation. We apply a conventional cross entropy loss function $\mathcal{L}_{\text{des}}$ on the generated token probabilities, as in previous work [32]–[34].

Final loss. We combine all three loss terms in a linear manner as our final loss function:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{det}} + \beta\mathcal{L}_{\text{ad}} + \gamma\mathcal{L}_{\text{des}} \tag{4.3}$$

where α, β and γ are the weights for the individual loss terms. After fine-tuning on the validation split, we set those weights to $\alpha = 10$, $\beta = 1$, and $\gamma = 0.1$ in our experiments to ensure the loss terms are roughly of the same magnitude.

4.4.6 Training and Inference

In our experiments, we randomly select 40,000 points from ScanNet mesh vertices. During training, we set the upper bound of the number of object proposals as $M = 256$. We only use the unmasked predictions corresponding to the provided objects in Scan2CAD for minimizing the relative orientation loss, as stated in 4.4.5. To optimize the description loss, we select the generated description of the object proposal with the largest IoU with the ground truth bounding box. During inference, we apply a non-maximum suppression module to suppress overlapping proposals.

4.4.7 Implementation Details

We implement our architecture using PyTorch [108] and train end-to-end using ADAM [86] with a learning rate of 1e−3. We train the model for $90,000$ iterations until convergence. To avoid overfitting, we set the weight decay factor to 1e−5 and apply data augmentation to our training data. Following ScanRefer [14], the point cloud is rotated by a random angle in $[-5°, 5°]$ about all three axes and randomly translated within 0.5 meters in all directions. Since the ground alignment in ScanNet is imperfect, the rotation is around all axes (not just up). We truncate descriptions longer than 30 tokens and add SOS and EOS tokens to indicate the start and end of the description.

	Captioning	Detection	C@0.25IoU	B-4@0.25IoU	M@0.25IoU	R@0.25IoU	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	mAP@0.5IoU
2D-3D Proj.	2D	Mask R-CNN	18.29	10.27	16.67	33.63	8.31	2.31	12.54	25.93	10.50
3D-2D Proj.	2D	VoteNet	19.73	17.86	19.83	40.68	11.47	8.56	15.73	31.65	31.83
VoteNetRetr [24]	3D	VoteNet	15.12	18.09	19.93	38.99	10.18	13.38	17.14	33.22	31.83
Ours	3D	VoteNet	**56.82**	**34.18**	**26.29**	**55.27**	**39.08**	**23.32**	**21.97**	**44.78**	**32.21**

Table 4.1: Comparison of 3D dense captioning results obtained by Scan2Cap and other baseline methods. We average the scores of the conventional captioning metrics, e.g. CiDEr [109], with the percentage of the predicted bounding boxes whose IoU with the ground truth are greater than 0.25 and 0.5. Our method outperforms all baselines with a remarkable margin.

	Cap	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU
OracleRetr2D	2D	20.51	20.17	23.76	50.98
Oracle2Cap2D	2D	58.44	37.05	28.59	61.35
OracleRetr3D	3D	33.03	23.36	25.80	52.99
Oracle2Cap3D	3D	**67.95**	**41.49**	**29.23**	**63.66**

Table 4.2: Comparison of 3D dense captioning results obtained by our method and other baseline methods with GT detections. We average the scores of the conventional captioning metrics with the percentage of the predicted bounding boxes whose IoU with the ground truth are greater than 0.5. Our method with GT bounding boxes outperforms all variants with a remarkable margin.

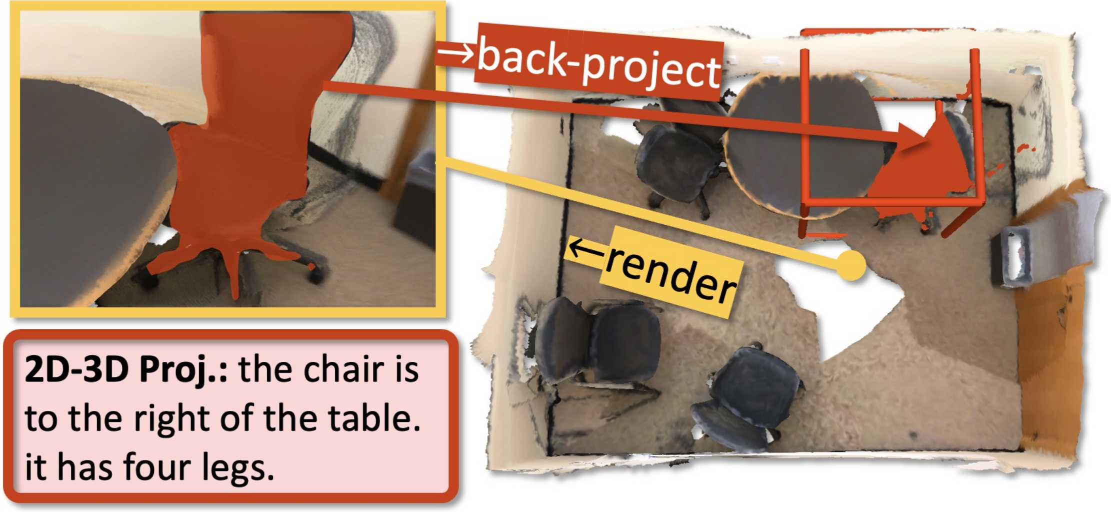

→back-project
←render
2D-3D Proj.: the chair is to the right of the table. it has four legs.

Figure 4.4: In 2D-3D Proj, we first generate a description for each detected object in a rendered viewpoint. Then we back-project the object mask to the 3D space to evaluate the caption with our proposed caption evaluation metric.

4.5 Experiments

Dataset. We use the ScanRefer [14] dataset which consists of 51,583 descriptions for 11,046 objects in 800 ScanNet [13] scenes. The descriptions contain information about the appearance of the objects (e.g. "this is a black wooden chair"), and the spatial relations between the annotated object and nearby objects (e.g. "the chair is placed at the end of the long dining table right before the TV on the wall").

Train&val splits. Following the official ScanRefer [14] benchmark split, we divide our data into train/val sets with 36,665 and 9,508 samples respectively, ensuring disjoint scenes for each split. Results and analysis are conducted on the val split, as the hidden test set is not officially available.

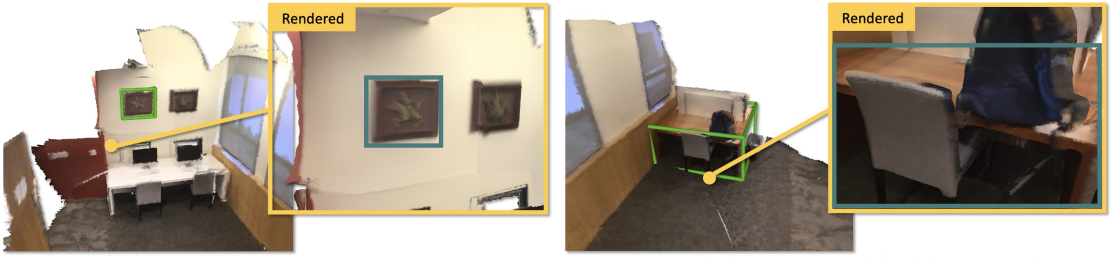
Metrics. To jointly measure the quality of the generated description and the detected bounding boxes, we evaluate the descriptions by combining standard image captioning metrics such as CiDEr [109] and BLEU [110], with Intersection-over-Union (IoU) scores between predicted bounding boxes and the target bounding boxes. We define our combined metrics as $m@k\text{IoU} = \frac{1}{N}\sum_{i=0}^{N} m_i u_i$, where $u_i \in \{0, 1\}$ is set to 1 if the IoU score for the i^{th} box is greater than k, otherwise 0. We use m to represent the captioning metrics CiDEr [109], BLEU-4 [110], METEOR [111] and ROUGE [112], abbreviated as C, B-4, M, R, respectively. N is the number of ground truth or detected object bounding boxes. We use mean average precision (mAP) thresholded by IoU as the object detection metric.

Skylines with ground truth input. To examine the upper limit of our proposed 3D dense captioning task, we use the ground truth (GT) object bounding boxes for generating object descriptions using our method and retrieval based approaches. We compare the performance of captioning in 3D with existing 2D-based captioning methods. For our 2D-based baselines, we generate descriptions for the 2D renders of the reconstructed ScanNet [13] scenes using the recorded viewpoints in ScanRefer [14].
Oracle2Cap3D We use ground truth 3D object bounding box features instead of detection backbone predictions to generate object descriptions. The relational graph and context-aware attention captioning module learn and generate corresponding captioning for each object. We use the same hyper-parameters with the Scan2Cap experiment.
OracleRetr3D We use the ground truth 3D object bounding box features in the val split to obtain the description for the most similar object features in the train split.
Oracle2Cap2D We first concatenate the global image and target object features and feed it to a caption generation method similar to [33]. In addition to [33], we try a memory augmented meshed transformer [104]. Surprisingly, the former performs better (see supplementary for details). We suspect that this performance gap is due to noisy 2D input and the size of our dataset, which does not allow for training complex methods (e.g. transformers) to their maximum potential. The target object bounding boxes are extracted using rendered ground truth instance masks and their features are extracted using a pre-trained ResNet-101 [113].
OracleRetr2D Similar to *OracleRetr3D*, use ground truth 2D object bounding box features in the val split to retrieve the description from the most similar train split object.

Baselines. We design experiments that leverage the detected object information in the input for description generation. Additionally, we show how existing 2D-based captioning methods perform in our newly proposed task.
***VoteNetRetr* [24]** Similar to *OracleRetr3D*, but we use the features of the 3D bounding boxes detected using a pre-trained VoteNet [24].
2D-3D Proj We first detect the object bounding boxes in rendered images using a pre-trained Mask R-CNN [6] with a ResNet-101 [113] backbone, then feed the 2D object bounding box features to our description generation module similar to [33]. We evaluate the generated captions in 3D by back-projecting the 2D masks to 3D using inverse camera extrinsics (see Fig. 4.4).
3D-2D Proj We first detect the object bounding boxes in scans using a pre-trained VoteNet [24], then project the bounding boxes to the rendered images. The 2D bounding box features are fed to our captioning module which uses the same decoding scheme as in [33].

4.5.1 Quantitative Analysis

We compare our method with the baseline methods on the official val split of ScanRefer [14]. As there is no direct prior work on this newly proposed task, we divide



OracleRetr2D: this is a brown picture. it is above a cabinet.

OracleRetr3D: this is a framed print hanging on the wall in the kitchen. it is above the stove.

Oracle2Cap2D: the picture is above the toilet. it is square.

Oracle2Cap3D: the picture is above the white desk. it is a dark and framed.

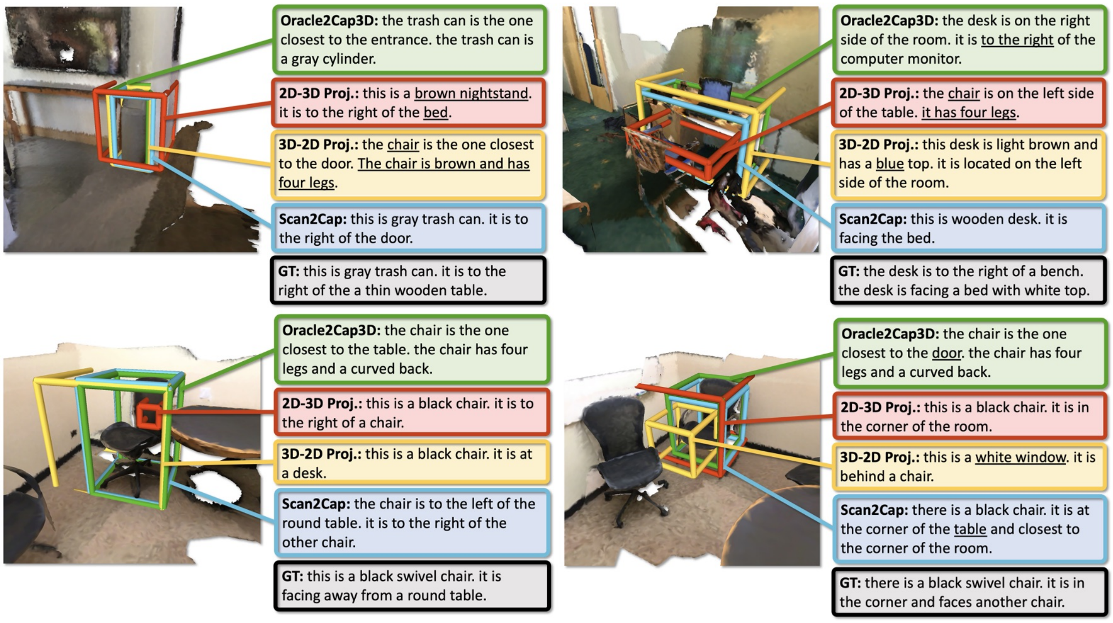
OracleRetr2D: the bed is in the middle of the room. it has a brown headboard.

OracleRetr3D: this is a metal desk. it can be seen from the door.

Oracle2Cap2D: this is a brown desk. it is to the left of a table.

Oracle2Cap3D: the brown desk is next to the window. it is a brown rectangle.

Figure 4.5: Qualitative results from skylines with GT input with inaccurate parts of the generated caption underscored. Captioning in 3D benefits from the richness of 3D context, while captioning with 2D information fails to capture the details of the local physical environment. Best viewed in color.



Oracle2Cap3D: the trash can is the one closest to the entrance. the trash can is a gray cylinder.

2D-3D Proj.: this is a brown nightstand. it is to the right of the bed.

3D-2D Proj.: the chair is the one closest to the door. The chair is brown and has four legs.

Scan2Cap: this is gray trash can. it is to the right of the door.

GT: this is gray trash can. it is to the right of the a thin wooden table.

Oracle2Cap3D: the desk is on the right side of the room. it is to the right of the computer monitor.

2D-3D Proj.: the chair is on the left side of the table. it has four legs.

3D-2D Proj.: this desk is light brown and has a blue top. it is located on the left side of the room.

Scan2Cap: this is wooden desk. it is facing the bed.

GT: the desk is to the right of a bench. the desk is facing a bed with white top.

Oracle2Cap3D: the chair is the one closest to the table. the chair has four legs and a curved back.

2D-3D Proj.: this is a black chair. it is to the right of a chair.

3D-2D Proj.: this is a black chair. it is at a desk.

Scan2Cap: the chair is to the left of the round table. it is to the right of the other chair.

GT: this is a black swivel chair. it is facing away from a round table.

Oracle2Cap3D: the chair is the one closest to the door. the chair has four legs and a curved back.

2D-3D Proj.: this is a black chair. it is in the corner of the room.

3D-2D Proj.: this is a white window. it is behind a chair.

Scan2Cap: there is a black chair. it is at the corner of the table and closest to the corner of the room.

GT: there is a black swivel chair. it is in the corner and faces another chair.

Figure 4.6: Qualitative results from baseline methods and Scan2Cap with inaccurate parts of the generated caption underscored. Scan2Cap produces good bounding boxes with descriptions for the target appearance and their relational interactions with objects nearby. In contrast, the baselines suffers from poor bounding box predictions or limited view and produces less informative captions. Best viewed in color.

description generation into: 1) generating the object bounding boxes and descriptions in 2D input, and back-projecting the bounding boxes to 3D using camera parameters; 2) directly generating object bounding boxes with descriptions in 3D space. As shown in

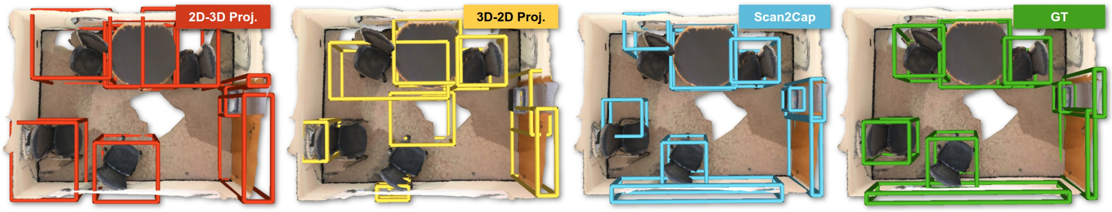


Figure 4.7: Comparison of object detections of baseline methods and Scan2Cap. 2D-3D Proj. suffers from the detection performance gap between image and 3D space. Scan2Cap produces better bounding boxes compared to 3D-2D Proj. due to the end-to-end fine-tuning.

	Cap	Acc (Category)	Acc (Attribute)	Acc (Relation)
Oracle2Cap2D	2D	69.00	67.42	37.00
Oracle2Cap3D	3D	85.15 (+**16.15**)	72.22 (+**4.80**)	76.24 (+**39.24**)
Ours	3D	84.16 (+**15.16**)	64.21 (-3.21)	69.00 (+**32.00**)

Table 4.3: Manual analysis of the generated captions obtained by skyline methods with GT input and ours. We measure the accuracy of three different aspects (object categories, appearance attributes and spatial relations) in the generated captions. Compared to captioning in 2D, captioning directly in 3D better capture these aspects in descriptions, especially for describing spatial relations in the local environment.

Tab. 4.1, describing the detected objects in 3D results in a big performance boost compared to the back-projected 2D approach (39.08% compared to 11.47% on C@0.5IoU). When using ground truth, descriptions generated with 3D object bounding boxes (*Oracle2Cap3D*) effectively outperform their counterparts that use 2D object bounding box information (*Oracle2Cap2D*), as shown in Tab. 4.2. The performance gap between our method and *Oracle2Cap3D* indicates that the detection backbone can be further improved as a potential future work.

	C@0.25IoU	B-4@0.25IoU	M@0.25IoU	R@0.25IoU	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	mAP@0.5IoU
Ours (fixed VoteNet)	56.20	**35.14**	26.14	**55.71**	33.87	20.11	20.48	42.33	31.83
Ours (end-to-end)	**56.82**	34.18	**26.29**	55.27	**39.08**	**23.32**	**21.97**	**44.78**	**32.21**

Table 4.4: Ablation study with a fixed pre-trained VoteNet [24] and an end-to-end fine-tuned VoteNet. We compute standard captioning metrics with respect to the percentage of the predicted bounding box whose IoU with the ground truth are greater than 0.25 and 0.5. Higher values are better.

4.5.2 Qualitative Analysis

We see from Fig. 4.5 that the captions retrieved by OracleRetr2D hallucinate objects that are not there, while Oracle2Cap2D provides inaccurate captions that fail to capture correct local context. In contrast, the captions from Oracle2Cap3D are longer and

	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU
VoteNet [24]+GRU [85]	34.31	21.42	20.13	41.33
VoteNet [24]+CAC	36.15	21.58	20.65	41.78
VoteNet [24]+RG+CAC	**39.08**	**23.32**	**21.97**	**44.78**

Table 4.5: Ablation study with different components in our method: VoteNet [24] + GRU [85], which is similar to "show and tell" [33]; VoteNet + Context-aware Attention Captioning (CAC); VoteNet + Relational Graph (RG) + Context-aware Attention Captioning (CAC), namely Scan2Cap. We compute standard captioning metrics with respect to the percentage of the predicted bounding boxes whose IoU with the ground truth are greater than 0.5. The higher the better. Clearly, our method with attention mechanism and graph module is shown to be effective.

capture relationships with the surrounding objects, such as "above the white desk" and "next to the window". Fig. 4.6 show the qualitative results of Oracle2Cap3D, 2D-3D Proj, 3D-2D Proj and our method (Scan2Cap). Leveraging the end-to-end training, Scan2Cap is able to predict better object bounding boxes compared to the baseline methods (see Fig. 4.6 top row). Aside from the improved quality of object bounding boxes, descriptions generated by our method are richer when describing the relations between objects (see second row of Fig. 4.6).

Provided with the ground truth object information, Oracle2Cap3D can include even more details in the descriptions. However, there are mistakes with the local surroundings (see the sample in the right column in Fig. 4.6), indicating there is still room for improvement. In contrast, image-based 2D-3D Proj. suffers from limitations of the 2D input and fails to produce good bounding boxes with detailed descriptions. Compared to our method, 3D-2D Proj. fails to predict good bounding boxes because of the lack of a fine-tuned detection backbone, as shown in Fig. 4.7.

4.5.3 Analysis and Ablations

Is it better to caption in 3D or 2D? One question we want to study is whether it is better to caption in 3D or 2D. Therefore, we conduct a manual analysis on 100 randomly selected descriptions generated by Oracle2Cap2D, Oracle2Cap3D and our method. In this analysis, we manually check if those descriptions correctly capture three important aspects for indoor objects: object categories, appearance attributes and spatial relations. As demonstrated in Tab. 5.5, directly captioning objects in 3D captures those aspects more accurately when comparing Oracle2Cap3D with Oracle2Cap2D, especially for describing the spatial relations. However, the accuracy drop on object attributes from Oracle2Cap2D to our method (-3.21%) shows the detection backbone can still be improved.

Does context-aware attention captioning help? We compare our model with the basic description generation component (GRU) introduced in [33] and our model with the context-aware attention captioning (CAC) as discussed in Sec. 4.4.4. The model equipped with the context-aware captioning module outperforms its counterpart without attention mechanism on all metrics (see the first row vs. the second row in Tab. 4.5).

Does the relational graph help? We evaluate the performance of our method against our model without the proposed relational graph (RG) and/or the context-aware attention captioning (CAC). As shown in Tab. 4.5, our model equipped with the context enhancement module (third row) outperforms all other ablations.

Does end-to-end training help? We show in Tab. 4.4 the effectiveness of fine-tuning the pretrained VoteNet end-to-end with the description generation objective. We observe that end-to-end training of the network allows for gradient updates from our relative orientation loss and description generation loss that compensate for detection errors. While the fine-tuned VoteNet detection backbone delivers similar detection results, its performance on describing objects outperforms its fixed ablation by a big margin on all more demanding metrics (see columns for metrics m@0.5IoU in Tab. 4.4).

4.6 Conclusion

In this work, we introduce the task of dense description generation in RGB-D scans. We propose an end-to-end trained architecture to localize the 3D objects in the input point cloud and generate descriptions for them in natural language. Thus, we address the 3D localization and description generation problems at the same time. We apply an attention-based captioning pipeline equipped with a message passing network to generate descriptive tokens while referring to related components in the local context. Our architecture effectively localizes and describes 3D objects, outperforming 2D-based dense captioning methods on the 3D dense description generation task by a large margin. Nevertheless, our method struggles to capture complex relations like ordinal counting. For instance, our method only predicts “the round chair next to another wooden chair”, while the ground truth “the *third* round chair from the wall” reveals more fine-grained spatial relations, indicating possibilities for improvement. Overall, we hope that our work will enable future research in 3D vision and language.

5 Unifying 3D Object Localization and Describing in RGB-D Scans

This chapter introduces the following paper:

Z. Chen, Q. Wu, M. Nießner, and A. X. Chang, "D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, Springer, 2022, pp. 487–505

Abstract of the paper Recent work on dense captioning and visual grounding in 3D have achieved impressive results. Despite developments in both areas, the limited amount of available 3D vision-language data causes overfitting issues for 3D visual grounding and 3D dense captioning methods. Also, how to discriminatively describe objects in complex 3D environments is not fully studied yet. To address these challenges, we present D^{3}Net, an end-to-end neural speaker-listener architecture that can **d**etect, **d**escribe and **d**iscriminate. Our D^{3}Net unifies dense captioning and visual grounding in 3D in a self-critical manner. This self-critical property of D^{3}Net encourages generation of discriminative object captions and enables semi-supervised training on scan data with partially annotated descriptions. Our method outperforms SOTA methods in both tasks on the ScanRefer dataset, surpassing the SOTA 3D dense captioning method by a significant margin.

Contribution The method development and implementation was done by the first author. Discussions with the co-authors led to the final paper.

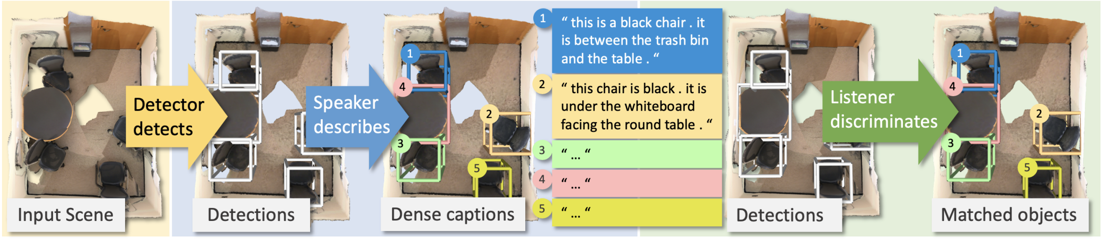

Input Scene
Detector detects
Detections
Speaker describes
1
2
3
4
5
Dense captions
1 " this is a black chair . it is between the trash bin and the table . "
2 " this chair is black . it is under the whiteboard facing the round table . "
3 " ... "
4 " ... "
5 " ... "
Detections
Listener discriminates
Matched objects

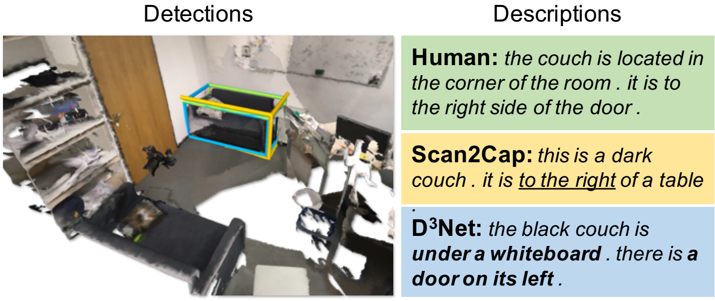
Figure 5.1: We introduce D^{3}Net, an end-to-end neural speaker-listener architecture that can **d**etect, **d**escribe and **d**iscriminate. D^{3}Net also enables semi-supervised training on ScanNet data with partially annotated descriptions.

5.1 Introduction

Recently, there has been increasing interest in bridging 3D visual scene understanding [13], [24], [66], [67], [93]–[95] and natural language processing [21], [87], [114]–[116]. The task of 3D visual grounding [14], [117], [118] localizes 3D objects described by natural language queries. 3D dense captioning proposed by [15] is the reverse task where we generate descriptions for 3D objects in RGB-D scans. Both tasks enable applications such as assistive robots and natural language control in AR/VR systems.

However, existing work on 3D visual grounding [14], [117]–[120] and dense captioning [15], [121] treats the two problems as separate, with *detect-then-dis-criminate* or *detect-then-describe* being the common strategies for tackling the two tasks. Separating the two complementary tasks hinders holistic 3D scene understanding where the ultimate goal is to create models that can infer: 1) what are the objects; 2) how to describe each object; 3) what object is being referred to through natural language. The disadvantages of having separated strategies are twofold. First, the detect-then-describe strategy often struggles to describe target objects in a discriminative way. In Fig. 5.2, the generated descriptions from Scan2Cap [15] fail to uniquely describe the target objects, especially in scenes with several similar objects. Second, existing 3D visual grounding methods [14], [118] in the detect-then-discriminate strategy suffer from severe overfitting issue, partly due to the small amount of 3D vision-language data [14], [119] which is limited compared to counterpart 2D datasets such as MSCOCO [4].

To address these issues, we propose an end-to-end self-critical solution, D^{3}Net, to enable discriminability in dense caption generation and utilize the generated captions improve localization. Relevant work in image captioning [122], [123] tackles similar issues where the generated captions are indiscriminative and repetitive by explicitly reinforcing discriminative caption generation with an image retrieval loss. Inspired by this scheme, we introduce a speaker-listener strategy, where the captioning module "speaks" about the 3D objects, while the localization module "listens" and finds the targets. Our proposed speaker-listener architecture can **d**etect, **d**escribe and **d**iscriminate, as illustrated in Fig. 5.1. The key idea is to reinforce the speaker to generate discriminative descriptions so that the listener can better localize the described targets given those descriptions.



Human: *the couch is located in the corner of the room . it is to the right side of the door .*

Scan2Cap: *this is a dark couch . it is to the right of a table*

D^3Net: *the black couch is* ***under a whiteboard*** *. there is* ***a door on its left*** *.*

Figure 5.2: Prior work [15] struggle to produce discriminative object captions. Also, captions often appear to be template-based. In contrast, our D^3Net generates discriminative object captions.

This approach brings another benefit. Since the speaker-listener architecture self-critically generates and discriminates descriptions, we can train on scenes without any object descriptions. We see further improvements in 3D dense captioning and 3D visual grounding performance when using this additional data alongside annotated scenes. This can allow for semi-supervised training on RGB-D scans beyond the ScanNet dataset. To summarize, our contributions are:

- We introduce a unified speaker-listener architecture to generate discriminative object descriptions in RGB-D scans. Our architecture allows for a semi-supervised training scheme that can alleviate data shortage in the 3D vision-language field.
- We study how the different components impact performance and find that having a strong detector is essential, and that by jointly optimizing the detector, speaker, and listener we can improve detection as well as 3D dense captioning and visual grounding.
- We show that our method outperforms the state-of-the-art for both 3D dense captioning and 3D visual grounding method by a significant margin.

5.2 Related Work

Vision and language in 3D. Recently, there has been growing interest in grounding language to 3D data [14], [80], [81], [119], [124]–[126]. [14] and [119] introduce two complementary datasets consisting of descriptions of real-world 3D objects from ScanNet [13] reconstructions, named ScanRefer and ReferIt3D, respectively. ScanRefer proposes the joint task of detecting and localizing objects in a 3D scan based on a textual description, while ReferIt3D is focused on distinguishing 3D objects from the same semantic class given ground-truth bounding boxes. [117] localize objects by decomposing input queries into fine-grained aspects, and use PointGroup [25] as their visual backbone. However, the frozen detection backbone is not fine-tuned together with the localization module. [118] propose a transformer-based architecture with a VoteNet [24] backbone to handle multimodal contexts during localization. Despite the improved matching module, their work still suffers from poor quality detections due to the weak 3D detector. We show that fine-tuning an improved 3D detector is essential to getting good predictions and good localization performance. [15] introduce the task of densely detecting and captioning objects in RGB-D scans. Recently, [121] aggregate the 2D features to point cloud to generate faithful object descriptions. Although their methods can effectively detect objects and generate captions w.r.t. their attributes, the quality of the bounding boxes and the discriminability of the captions are inadequate. Our method explicitly handles the discriminability of the generated captions through a self-critical speaker-listener architecture, resulting in the state-of-the-art performance in both 3D dense captioning and 3D visual grounding tasks.

Generating captions in images. Image captioning has attracted a great deal of interest [32]–[34], [72], [88], [97]–[99], [127]. Recent work [122], [123] suggest that traditional encoder-decoder-based image captioning methods suffer from the discriminability issues. [122] propose an additional image retrieval branch to reinforce discriminative caption generation. [123] propose a reinforcement learning method to train not only on annotated web images, but also images without any paired captions. In contrast to generating captions for the entire image, in the dense captioning task we densely generate captions for each detected object in the input image [28], [89], [91]. Although such methods are effective for generating captions in 2D images, directly applying such training techniques on 3D dense captioning can lead to unsatisfactory results, since the captions involve 3D geometric relationships. In contrast, we work directly on 3D scene input dealing with object attributes as well as 3D spatial relationships.

Grounding referential expressions in images. There has been tremendous progress in the task of grounding referential expressions in images, also known as visual grounding [7]–[11], [37]. Given an image and a natural language text query as input, the target object is either localized by a bounding box [7], [37], or a segmentation mask [8]. These methods have achieved great success in the image domain. However, they are not designed to deal with 3D geometry inputs and handle complex 3D spatial relationships. Our proposed method directly decomposes the 3D input data with a sparse convolutional detection backbone, which produces accurate object proposals as well as semantically rich features.

Speaker-listener models for grounding. The speaker-listener model is a popular architecture for pragmatic language understanding, where a line of research explores how the context and communicative goals affect the linguistics [128], [129]. Recent work use neural speaker-listener architectures to tackle referring expression generation [11], [55], [130], vision-language navigation [131], and shape differentiation [81]. [11] construct a CNN-LSTM architecture optimized by a softmax loss to directly discriminate the generated referential expressions. There is no separate neural listener module compared with our method. [130] and [55] introduce a LSTM-based neural listener in the speaker-listener pipeline, but generating the referential expression is not directly supervised via the listener model, but rather trained via a proxy objective. In contrast, our method directly optimizes the Transformer-based neural listener for the visual grounding task by

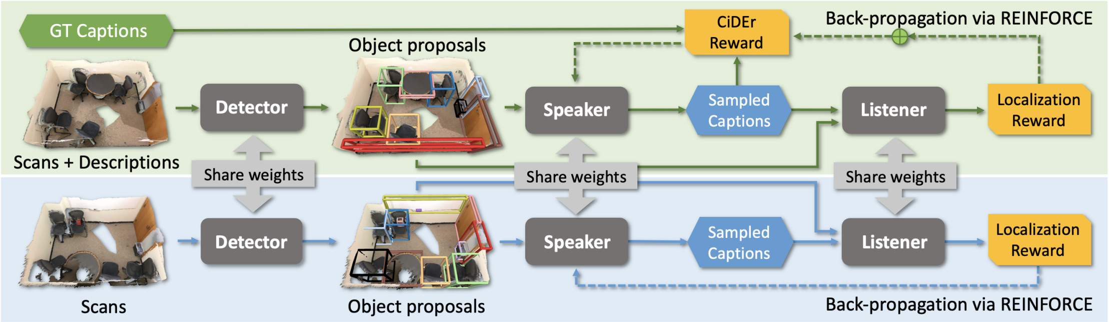


Figure 5.3: D^{3}Net architecture. We input point clouds into the *detector* to predict object proposals. Then, those proposals are fed into the speaker to generate captions that *describes* each object. To *discriminate* the object described by each caption, the listener matches the generated captions with object proposals. The captioning and localization results are back-propagated via REINFORCE [132] as rewards through the dashed lines. D^{3}Net also enables end-to-end training on point clouds with no GT object descriptions (bottom blue block).

discriminating the generated object captions without any proxy training objective. Similarly, [81] includes a pretrained and frozen listener in the training objective, while ours enables joint end-to-end optimization for both the speaker and listener via policy gradient algorithm. We experimentally show our method to be effective for semi-supervised learning in the two 3D vision-language tasks.

5.3 Method

D^{3}Net has three components: a 3D object detector, the speaker (captioning) module, and the listener (localization) module. Fig. 5.3 shows the overall architecture and training flow. The point clouds are fed into the detector to predict object proposals. The speaker takes object proposals as input to produce captions. To increase caption discriminability, we match these captions with object proposals via the listener. Caption quality is measured by the CIDEr **vedantam2015CIDEr** scores and the listener loss, which are back-propagated via REINFORCE [132] as rewards to the speaker. Our architecture can handle scenes without ground-truth (GT) object descriptions by reinforcing the speaker with the listener loss only.

5.3.1 Modules

Detector. We use PointGroup [25] as our detector module. PointGroup is a relatively simple model for 3D instance segmentation that achieves competitive performance on the ScanNet benchmark. We use ENet to augment the point clouds with multi-view features, following [23]. PointGroup uses a U-Net architecture with a SparseConvNet backbone to encode point features, cluster the points, and uses ScoreNet, another U-Net structure, to score each cluster. We take the cluster features after ScoreNet as the encoded object features. We refer readers to the original paper [25] for more details. The object bounding boxes are determined by taking the minimum and maximum points in the point clusters, and are produced as final outputs of our detector module.

Speaker. We base our speaker on the dense captioning method introduced by [15]. Our speaker module has two submodules: 1) a relational graph module, which is responsible for learning object-to-object spatial location relationships; 2) a context-aware attention captioning module, which attentively generates descriptive tokens with respect to the object attributes as well as the object-to-object spatial relationships.

Listener. For the listener, we follow the architecture introduced by [14] but replace the multi-modal fusion module with the transformer-based multi-modal fusion module of [118]. Our listener module has two submodules: 1) a language encoding module with a GRU cell; 2) a transformer-based multi-modal fusion module similar to [118], which attends to elements in the input query descriptions and the detected object proposals. As in [14], we also incorporate a language object classifier to discriminate the semantics of the target objects in the input query descriptions.

5.3.2 Training Objective

The three modules are designed to be trained in an end-to-end fashion (see 5.3). In this section, we describe the loss for each module, and how they are combined for the overall loss.

Detection loss. We use the instance segmentation loss introduced in PointGroup [25] to train the 3D backbone. The detection loss is composed of four parts: $L_{\text{det}} = L_{\text{sem}} + L_{\text{o\_reg}} + L_{\text{o\_dir}} + L_{\text{c\_score}}$. L_{sem} is a cross-entropy loss supervising semantic label prediction for each point. $L_{\text{o\_reg}}$ is a L_1 regression loss constraining the learned point offsets belonging to the same cluster. $L_{\text{o\_dir}}$ constrains the direction of predicted offset vectors, defined as the means of minus cosine similarities. It helps regress precise offsets, particularly for boundary points of large-size objects, since these points are relatively far from the instance centroids. $L_{\text{c\_score}}$ is another binary cross-entropy loss supervising the predicted objectness scores.

Listener loss. The listener loss is composed of a localization loss L_{loc} and a language-based object classification loss L_{lobjcls}. To obtain the localization loss L_{loc}, we first require a target bounding box. We use the detected bounding box with the highest IoU with the GT bounding box as the target bounding box. Then, a cross-entropy loss L_{loc} is applied to supervise the matching score prediction. In the end-to-end training scenario, the detected bounding boxes associated with the generated descriptions from the speaker are treated as the target bounding boxes. The language object classification loss is a cross-entropy loss L_{lobjcls} to supervise the classification based on the input description. The target classes are consistent with the ScanNet 18 classes, excluding structural objects such as “floor” and “wall”.

Speaker loss using MLE training objective. The speaker loss is a standard captioning loss from maximum likelihood estimation (MLE). During training, provided with a pair of GT bounding box and the associated GT description, we optimize the description associated with the predicted bounding box which has the highest IoU score with the current GT bounding box. We first treat the description generation task as a sequence prediction task, factorized as: $L_{\text{spk-XE}}(\theta) = -\sum_{t=1}^{T} \text{log}p(\hat{c_t}|\hat{c_1}, ..., \hat{c_{t-1}}; I, \theta)$, where $\hat{c_t}$ denotes the generated token at step t; I and θ represent the visual signal and model parameter, respectively. The token $\hat{c_t}$ is sampled from the probability distribution over the pre-defined vocabulary. The generation process is performed by greedy decoding or beam search in an autoregressive manner, and we use the argmax function to sample each token.

Joint loss using REINFORCE training objective. We use REINFORCE to train the detector-speaker-listener jointly. We first describe the enhanced speaker-loss, $L_{\text{spk-R}}$ that is trained using reinforcement learning to produce discriminative captions. We then describe the overall loss used in end-to-end training. Following prior work [55], [99], [122], [123], [133], [134], generating descriptions is treated as a reinforcement learning task. In the setting of reinforcement learning, the speaker module is treated as the "agent", while the previously generated words and the input visual signal I are the "environment". At step t, generating word $\hat{c_t}$ by the speaker module is deemed as the "action" taken with the policy p_θ, which is defined by the speaker module parameters θ. Specifically, with the generated description $\hat{C} = \{c_1, ..., c_T\}$, the objective is to maximize the reward function $R(\hat{C}, I)$. We apply the "REINFORCE with baseline" algorithm following [99] to reduce the variance of this loss function, where a baseline reward $R(C^*, I)$ of the description C^* independent of $\hat{C}$ is introduced. We apply beam search to sample descriptions and choose the greedily decoded descriptions as the baseline. The simplified policy gradient is:

$$L_{\text{spk-R}}(\theta) \approx -(R(\hat{C}, I) - R(C^*, I)) \sum_{t=1}^{T} \text{log}p(\hat{c_t}|I, \theta) \tag{5.1}$$

Rewards. As the word-level sampling through the argmax function is non-differentiable, the subsequent listener loss cannot be directly back-propagated through the speaker module. A workaround is to use the gumbel softmax re-parametrization trick [135]. Following the training scheme of [123] and [122], the listener loss can be inserted into the REINFORCE reward function to increase the discriminability of generated referential descriptions. Specifically, given the localization loss L_{loc} and the language object classification loss L_{lobjcls}, the reward function $R(\hat{C})$ is the weighted sum of the CIDEr score of the sampled description and the listener-related losses:

$$R(\hat{C}, I) = R^{\text{CIDEr}}(\hat{C}, I) - \alpha[L_{\text{loc}}(\hat{C}) + \beta L_{\text{lobjcls}}(\hat{C})] \tag{5.2}$$

where α and β are the weights balancing the CIDEr reward and the listener rewards. We empirically set them to 0.1 and 1 in our experiments, respectively. To stabilize the training, the reward related to the baseline description $R(C^*)$ should be formulated analogously. Note that there should be no gradient calculation and back-propagation for the baseline C^*. For scenes with no GT descriptions provided, the CIDEr reward is cancelled in the reward function, which in this case becomes $R(\hat{C}, I) = -\alpha[L_{\text{loc}}(\hat{C}) + \beta L_{\text{lobjcls}}(\hat{C})]$.

Relative orientation loss. Following [15], we adopt the relative orientation loss on the message passing module as a proxy loss. The object-to-object relative orientations ranging from 0° to 180° are discretized into 6 classes. We apply a simple cross-entropy loss L_{ori} to supervise the relative orientation predictions.

Overall loss. We combine loss terms in our end-to-end joint training objective as: $L = L_{\text{det}} + L_{\text{spk-R}} + 0.3L_{\text{ori}}$.

5.3.3 Training

We use a stage-wise training strategy for stable training. We first pretrain the detector backbone on all training scans in ScanNet via the detector loss L_{det}. We then train the dense captioning pipeline with the pretrained detector and a newly initialized speaker end-to-end via the detector loss and the speaker MLE loss $L_{\text{spk-XE}}$. After the speaker MLE loss converges, we train the visual grounding pipeline with the fine-tuned frozen detector and the listener via the listener loss L_{loc}. Finally, we fine-tune the entire speaker-listener architecture with the overall loss L.

5.3.4 Inference

During inference, we use the detector and the speaker to do 3D dense captioning and the listener to do visual grounding. The detector first produces object proposals, and the speaker generates a description for each object proposal. We take the minimum and maximum coordinates in the predicted object instance masks to construct the bounding boxes. For the object proposals that are assigned to the same ground truth, we keep only the one with the highest IoU with the GT bounding box. When evaluating the detector itself, the non-maximum suppression is applied.

5.4 Experiments

5.4.1 Dataset

We use the ScanRefer [14] dataset consisting of around 51k descriptions for over 11k objects in 800 ScanNet [13] scans. The descriptions include information about the appearance of the objects, as well as the object-to-object spatial relationships. We follow the official split from the ScanRefer benchmark for training and validation. We report our visual grounding results on the validation split and benchmark results on the hidden test set[1]. Our dense captioning results are on the validation split due to the lack of the test grounding truth. We also conduct experiments on the ReferIt3D dataset [119] (please see the supplemental).

[1] `http://kaldir.vc.in.tum.de/scanrefer_benchmark`

5.4.2 Semi-supervised Training with Extra Data

As the scans in ScanRefer dataset are only a subset of scans in ScanNet, we extend the training set by including all re-scans of the same scenes for semi-supervised training. Unlike the scans in ScanRefer, these re-scans do not have per object descriptions. We can control how much extra data to use by randomly sampling (with replacement) from the set of re-scans. We experiment with augmenting our data with 0.1 to 1 times the amount of annotated data as extra data. During training, we randomly select detected objects in the sampled extra scans for subsequent dense captioning and visual grounding. For the complete 'extra' scenario, we use a comparable amount (1x) of extra data as the annotated data in ScanRefer.

5.4.3 Implementation Details

We implement the PointGroup backbone using the Minkowski Engine [20] (see supplement). For the backbone, we train using Adam [86] with a learning rate of 2e-3, on the ScanNet train split with batch size 4 for 140k iterations, until convergence. For data augmentation, we follow [25], randomly applying jitter, mirroring about the YZ-plane, and rotation about the Z axis (up-axis) to each point cloud scene. We then use the Adam optimizer with learning rate 1e-3 to train the detector and the listener on the ScanRefer dataset with batch size 4 for 60k iterations, until convergence. Each scan is paired with 8 descriptions (i.e. 4 scans and 32 descriptions per batch iteration). Then, we combine the trained detector with the newly initialized speaker on the ScanRefer dataset for the 3D dense captioning task, where the weights of the detector are frozen. We again use Adam with learning rate 1e-3, with the training process converging within 14k iterations. All our experiments are conducted on a RTX 3090, and all neural modules are implemented using PyTorch [136].

5.4.4 Quantitative Results

3D dense captioning and detection. Tab. 5.1 compares our 3D dense captioning and object detection results against the baseline methods Scan2Cap [15] and X-Trans2Cap [121]. Leveraging the improved PointGroup based detector, our speaker model trained with the conventional MLE objective (Ours (MLE)) outperforms Scan2Cap and X-Trans2Cap by a large margin in all metrics. As expected, training with the CIDEr reward (Ours (CIDEr)) significantly improves the CIDEr score. We note that other captioning metrics are also improved, but the detection mAP@0.5 remains similar. Training with object localization reward (Ours (CIDEr+loc.)) improves both captioning and detection further due to the improved discriminability during description generation. Note that if we use a frozen pretrained listener (Ours (CIDEr+fixed loc.)), the improvement is not as significant as when we allow the listener weights to be fine-tuned (Ours (CIDEr+loc.)). Our full model with the full listener reward incorporates an additional language object classification loss (Ours (CIDEr+loc.+lobjcls.)) and further improves the performance for both tasks.

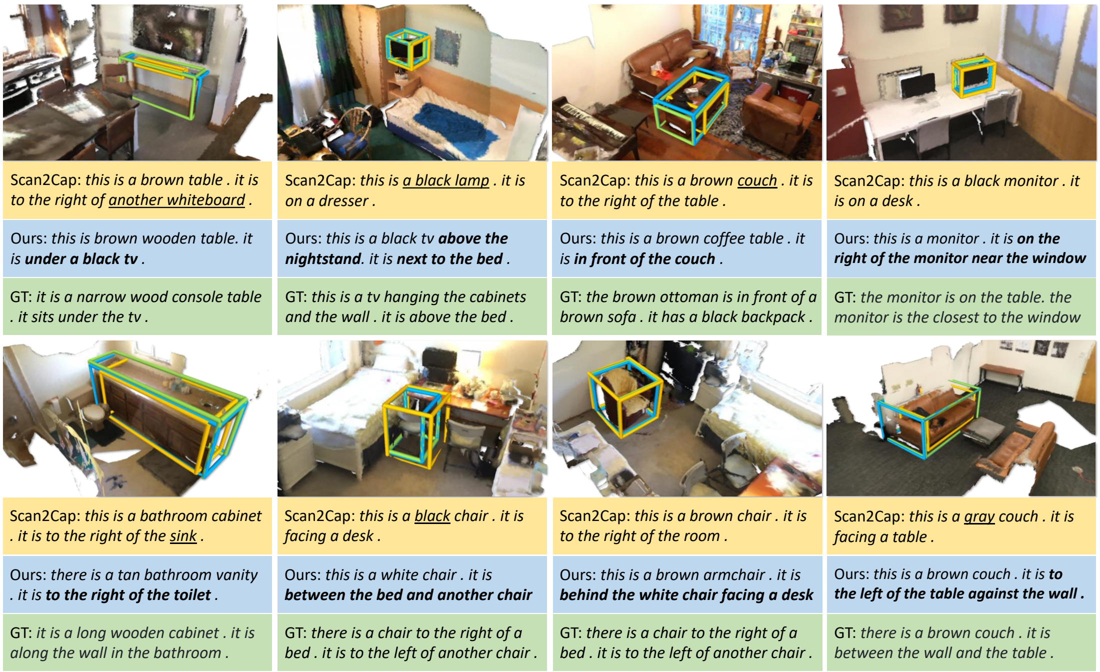
	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	mAP@0.5
Scan2Cap [15]	39.08	23.32	21.97	44.78	32.21
X-Trans2Cap [121]	43.87	25.05	22.46	44.97	35.31
Ours (MLE)	46.07	30.29	24.35	51.67	50.93
Ours (CIDEr)	57.88	32.64	24.86	52.26	51.01
Ours (CIDEr+fixed loc.)	58.93	33.36	25.12	52.62	51.04
Ours (CIDEr+loc.)	61.30	34.50	25.25	52.80	52.07
Ours (CIDEr+loc.+lobjcls.)	61.50	35.05	25.48	53.31	52.58
Ours (w/ 0.1x extra data)	61.91	35.03	25.38	53.25	52.64
Ours (w/ 0.5x extra data)	62.36	35.54	25.43	53.67	53.17
Ours (w/ 1x extra data)	**62.64**	**35.68**	**25.72**	**53.90**	**53.95**

Table 5.1: Quantitative results on 3D dense captioning and object detection. As in [15], we average the conventional captioning evaluation metrics with the percentage of the predicted bounding boxes whose IoU with the GTs are higher than 0.5. Our speaker model outperforms the baseline Scan2Cap without training via REINFORCE, while training with CIDEr reward further boosts the dense captioning performance. We also showcase the effectiveness of training with additional scans with no description annotations. Our speaker-listener architecture trained with 1x extra data achieves the best performance.

	Val Acc@0.5IoU			Test Acc@0.5IoU		
	Unique	Multiple	Overall	Unique	Multiple	Overall
ScanRefer [14]	53.51	21.11	27.40	43.53	20.97	26.03
TGNN [120]	56.80	23.18	29.70	58.90	25.30	32.80
InstanceRefer [117]	66.83	24.77	32.93	66.69	26.88	35.80
3DVG-Trans [118]	60.64	28.42	34.67	55.15	29.33	35.12
3DVG-Trans+ [118]	-	-	-	57.87	**31.02**	37.04
Ours (w/o fine-tuning)	70.35	27.11	35.58	65.79	27.26	35.90
Ours	**72.04**	**30.05**	**37.87**	**68.43**	30.74	**39.19**

Table 5.2: Quantitative results on 3D visual grounding. We adapt the evaluation setting as in [14]. "Unique" means there is only one object belongs to a specific class in the scene, while "multiple" represents the cases where more than one object from a specific class can be found in the scene. Clearly, our base visual grounding network outperforms all baselines even before being put into the speaker-listener architecture. After the speaker-listener fine-tuning, our method achieves the state-of-the-art performance on the ScanRefer validation set and the public benchmark. Note that 3DVG-Trans+ is an unpublished extension of 3DVG-Trans [118] which appears only on the public benchmark.

Does additional data help? As our method allow for training the listener with scans without language data, we investigate the effectiveness of training with additional Scan-



Scan2Cap: *this is a brown table . it is to the right of another whiteboard .*

Ours: *this is brown wooden table. it is* ***under a black tv*** *.*

GT: *it is a narrow wood console table . it sits under the tv .*

Scan2Cap: *this is a black lamp . it is on a dresser .*

Ours: *this is a black tv* ***above the nightstand****. it is* ***next to the bed*** *.*

GT: *this is a tv hanging the cabinets and the wall . it is above the bed .*

Scan2Cap: *this is a brown couch . it is to the right of the table .*

Ours: *this is a brown coffee table . it is* ***in front of the couch*** *.*

GT: *the brown ottoman is in front of a brown sofa . it has a black backpack .*

Scan2Cap: *this is a black monitor . it is on a desk .*

Ours: *this is a monitor . it is* ***on the right of the monitor near the window***

GT: *the monitor is on the table. the monitor is the closest to the window*

Scan2Cap: *this is a bathroom cabinet . it is to the right of the sink .*

Ours: *there is a tan bathroom vanity . it is* ***to the right of the toilet*** *.*

GT: *it is a long wooden cabinet . it is along the wall in the bathroom .*

Scan2Cap: *this is a black chair . it is facing a desk .*

Ours: *this is a white chair . it is* ***between the bed and another chair***

GT: *there is a chair to the right of a bed . it is to the left of another chair .*

Scan2Cap: *this is a brown chair . it is to the right of the room .*

Ours: *this is a brown armchair . it is* ***behind the white chair facing a desk***

GT: *there is a chair to the right of a bed . it is to the left of another chair .*

Scan2Cap: *this is a gray couch . it is facing a table .*

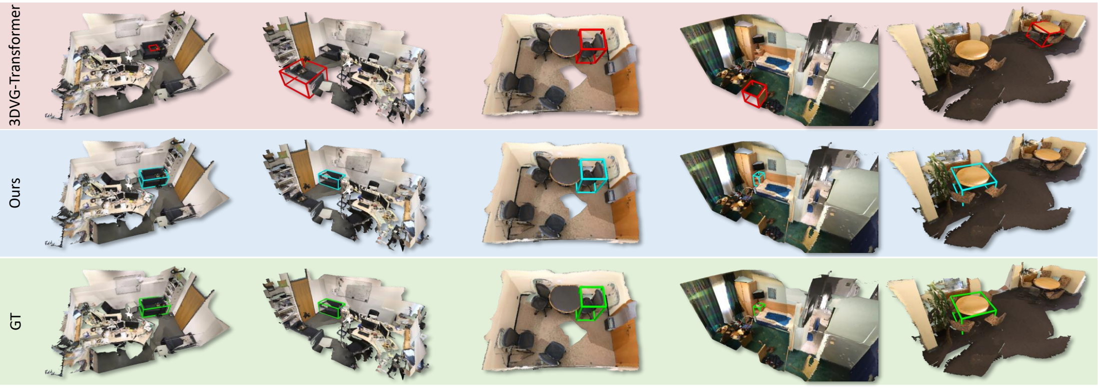
Ours: *this is a brown couch . it is* ***to the left of the table against the wall .***

GT: *there is a brown couch . it is between the wall and the table .*

Figure 5.4: Qualitative results in 3D dense captioning task from Scan2Cap [15] and our method. We underline the inaccurate words and mark the spatially discriminative phrases in bold. Our method qualitatively outperforms Scan2Cap in producing better object bounding boxes and more discriminative descriptions.

Net data that have not been annotated with descriptions. We vary the amount of extra scan data (without descriptions) from 0.1x to 1x of fully annotated data and train our full model with CIDEr and full listener reward (loc.+lobjcls.). Our results (last three rows of Tab. 5.1), show that our semi-supervised training strategy can leverage the extra data to improve both dense captioning and object detection.

3D visual grounding. Tab. 5.2 compares our results against prior 3D visual grounding methods ScanRefer [14], TGNN [120], InstanceRefer [117] and 3DVG-Transformer [118], and 3DVG-Trans+, an unpublished extension. Our method trained only with the detection loss and the listener loss ("Ours w/o fine-tuning"), i.e. without the speaker-listener setting, outperforms all the previous methods in the "Unique" and "Overall" scenarios. We find the improved fusion module together with the improved detector is sufficient to outperform 3DVG-Trans. Due to the improved detector, our method can distinguish objects in the "Unique" case, where the semantic labels play an important role. Meanwhile, 3DVG-Trans [118] still outperforms our base listener when discriminating objects from the same class ("Multiple" case). Our end-to-end speaker-listener (last row) outperforms all previous method including 3DVG-Trans.



Query: This is a black couch. It is located next to a tall shelf and there is a fan in front of it.

Query: A black couch in the corner of the room. There is an information board above it.

Query: This is a black chair. It is between the trash bin and the table.

Query: The nightstand is brown and is in the bedroom. It's at the end of the bed below the TV.

Query: It is a light brown table surrounded by four chairs. It is to the left in the room by the plant.

Figure 5.5: 3D visual grounding results using 3DVG-Transformer [118] and our method. 3DVG-Transformer fails to accurately predict object bounding boxes, while our method produces accurate bounding boxes and correctly distinguishes target objects from distractors.

5.4.5 Qualitative Analysis

3D dense captioning. Fig. 5.4 compares our results with object captions from Scan2Cap [15]. Descriptions generated by Scan2Cap cannot uniquely identify the target object in the input scenes (see the yellow block on the bottom right). Also, Scan2Cap produces inaccurate object bounding boxes, which affects the quality of object captions (see the yellow block on the top left). Compared to captions from Scan2Cap, our method produces more discriminative object captions that specifies more spatial relations (see bolded phrases in the blue blocks).

3D visual grounding. Fig. 5.5 compares our results with 3DVG-Transformer [118]. Though 3DVG-Transformer is able to pick the correct object, it suffers from poor object detections and is constrained by the performance of the VoteNet-based detection backbone (see the first column). Our method is capable of selecting the queried objects while also predicting more accurate object bounding boxes.

5.4.6 Analysis and Ablation Studies

Does better detection backbone help? From Tab. 5.1, we see that using a better detector can significant improve performance. We further examine the effect of using different detection backbones (VoteNet and PointGroup) compared to GT bounding boxes in Tab. 5.3. For each detection backbone, we use four variants of our method: the models trained without the joint speaker-listener architecture, and the speaker-listener architecture trained with CIDEr reward, listener reward and extra ScanNet data. The results with GT boxes show the effectiveness of our speaker-listener architecture, when detections are perfect. The large improvement from VoteNet [24] to PointGroup [25] show

Method	Detection	mAP@0.5	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	Unique Acc@0.5IoU	Multiple Acc@0.5IoU	Overall Acc@0.5IoU
Ours (MLE)	GT	100.00	71.41	42.95	29.67	64.93	88.45	36.46	46.03
Ours (CIDEr)	GT	100.00	94.80	47.92	30.80	**66.34**	-	-	-
Ours (CIDEr+lis.)	GT	100.00	95.62	47.65	**30.93**	66.31	89.76	36.85	47.14
Ours (CIDEr+lis.+extra)	GT	100.00	**96.31**	**48.20**	30.80	66.10	**89.86**	**40.66**	**48.17**
Ours (MLE)	VoteNet	32.21	39.08	23.32	21.97	**44.78**	56.41	21.11	27.95
Ours (CIDEr)	VoteNet	37.66	46.88	25.96	22.10	44.69	-	-	-
Ours (CIDEr+lis.)	VoteNet	38.03	47.32	24.76	21.66	43.62	57.90	20.73	28.03
Ours (CIDEr+lis.+extra)	VoteNet	**38.82**	**48.38**	**26.09**	**22.15**	44.74	**58.40**	**21.66**	**29.25**
Ours (MLE)	PointGroup	47.19	46.07	30.29	24.35	51.67	70.35	27.11	35.58
Ours (CIDEr)	PointGroup	52.44	57.88	32.64	24.86	52.26	-	-	-
Ours (CIDEr+lis.)	PointGroup	52.58	61.50	35.05	25.48	53.31	71.04	27.40	35.62
Ours (CIDEr+lis.+extra)	PointGroup	**53.95**	**62.64**	**35.68**	**25.72**	**53.90**	**72.04**	**30.05**	**37.87**

Table 5.3: Quantitative results on object detection, dense captioning and visual grounding in RGB-D scans. We train our method using different detection backbones as well as the ground truth bounding boxes. Our method trained with CIDEr and listener reward as well as the additional data outperforms the pretrained speaker and listener models.

	detection	Unique Acc@0.5IoU	Multiple Acc@0.5IoU	Overall Acc@0.5IoU
Scan2Cap [15]	VN [24]	80.52	29.95	39.08
Ours (w/ CIDEr & lis.)	PG [25]	81.16	30.22	41.62
Ours (w/ CIDEr & lis. & extra)	PG [25]	**81.27**	**30.33**	**41.73**
Ours (w/ CIDEr & lis.)	GT	89.76	38.53	48.07
Ours (w/ CIDEr & lis. & extra)	GT	**90.29**	**40.66**	**49.71**

Table 5.4: We automatically evaluate the discriminability of the generated object descriptions. A pretrained neural listener similar to [118] is fed with the GT object features and the descriptions generated by Scan2Cap [15] as well as our method. Higher grounding accuracy indicates better discriminability, especially in the "multiple" case. To alleviate noisy detections, the evaluation results on the descriptions generated from the GT object features are also presented. Our method generates more discriminative descriptions compared to Scan2Cap.

the importance of a better detection backbone. The gap between GT and VoteNet/-PointGroup shows there is room for further improvement.

Are the generated descriptions more discriminative? To check whether the speaker-listener architecture generates more discriminative descriptions, we conduct an automatic evaluation via a reverse task. In this task, we feed the generated descriptions and GT bounding boxes into a pretrained neural listener model similar to [118]. The predicted visual grounding results are evaluated in the same way as in our 3D visual grounding experiments. Higher grounding accuracy indicates better discrimination, especially in the "Multiple" case. Results (Tab. 5.4) show that our speaker-listener architecture generates more discriminative descriptions compared to Scan2Cap [15]. The discrimination is further improved when training with extra ScanNet data. To disentan-

	Acc (Category)	Acc (Attribute)	Acc (Relation)
Scan2Cap [15]	84.10	64.21	69.00
Ours (MLE)	88.00 (+3.84)	74.73 (+10.53)	69.00 (+0.00)
Ours (CIDEr)	88.89 (+4.73)	75.00 (+10.79)	68.00 (-1.00)
Ours (CIDEr+lis.)	90.91 (+6.75)	77.38 (+13.17)	75.00 (+6.00)
Ours (CIDEr+lis.+extra)	92.93 **(+8.77)**	80.95 **(+16.74)**	78.57 **(+9.57)**

Table 5.5: Manual analysis of captions generated by Scan2Cap [15] and variants of our method. We measure accuracy in three different aspects: object categories, appearance attributes and spatial relations. Our method generates more accurate descriptions in all aspects, especially for describing spatial relations.

gle the affect of imperfectly predicted bounding boxes, we also train and evaluate our method with GT boxes (see last two rows in Tab. 5.4). We see that our semi-supervised speaker-listener architecture generates more discriminative descriptions.

Does the listener help with captioning? The third to the sixth rows in Tab. 5.1 measure the benefit of training the speaker together with the listener (Ours (CIDEr+loc.) and Ours (CIDEr+loc.+lobjcls.)) rather than training the speaker alone (Ours (CIDEr)). Training with the listener improves all captioning metrics. Also, training jointly with an unfrozen listener (Ours (CIDEr+loc.) leads to a better performance when compared with the variant with a pretrained and frozen listener (Ours (CIDEr+fixed loc.), which is similar to [81]. Additionally, as the detector is not only fine-tuned with the speaker but also with the listener, the additional supervision from the listener helps with the detection performance as well.

To analyze the quality of the generated object captions, we asked 5 students to perform a fine-grained manual analysis of the captions. Each student was presented with a batch of 100 randomly selected object captions with associated objects highlighted in the 3D scene. The student are then asked to indicate if the respective aspects were included and correctly described. The manual analysis results in Tab. 5.5 shows that our method generates more accurate descriptions compared to Scan2Cap. In particular, training with the listener and extra ScanNet data produces more accurate spatial relations in the descriptions. The results of fine-grained manual analysis complements the automatic captioning evaluation metric. While metrics such as CIDEr captures the overall similarity of the generated sentences against the references, the accuracies in Tab. 5.5 measures the correctness of the decomposed visual attributes.

Does the speaker help with grounding? Tab. 5.2 compares grounding results between a pretrained listener (Ours w/o fine-tuning) and a fine-tuned speaker-listener model (Ours). Although the grounding performance drops in the “Unique” subset, the improvements in “Multiple” suggests better discriminability in tougher and ambiguous scenarios.

5.5 Conclusion

We present D^{3}Net, an end-to-end speaker-listener architecture that can **d**etect, **d**escribe and **d**iscriminate. Specifically, the speaker iteratively generates descriptive tokens given the object proposals detected by the detector, while the listener discriminates the object proposals in the scene with the generated captions. The self-discriminative property of D^{3}Net also enables semi-supervised training on ScanNet data without the annotated descriptions. Our method outperforms the previous SOTA methods in both tasks on ScanRefer, surpassing the previous SOTA 3D dense captioning method by a significant margin. Our architecture can serve as an initial step towards leveraging unannotated 3D data for language and 3D vision. Overall, we hope that our work will encourage more future research in 3D vision and language.

Part III

Conclusion & Outlook

6 Conclusion

This dissertation investigates a very important research topic: Grounding Natural Language to 3D Scenes. We mainly focus on three problems: Localizing 3D Objects in RGB-D Scans using Natural Language, Generating Descriptions for 3D Objects and Unifying 3D Object Localization and Describing in RGB-D Scans. Each of these problems were introduced in Part II, and we present concluding remarks in the following.

ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language In Chapter 3, we introduce the task of localizing a target object in a 3D point cloud using natural language descriptions. We collect the ScanReferdataset, which contains 51,583 unique descriptions for 11,046 objects from 800 ScanNet [13] scenes. We propose an end-to-end method for localizing an object with a free-formed description as reference, which first proposes point clusters of interest and then matches them to the embeddings of the input sentence. Our architecture is capable of learning the semantic similarities of the given contexts and regressing the bounding boxes for the target objects. Overall, we hope that our new dataset and method will enable future research in the 3D visual language field.

Scan2Cap: Context-aware Dense Captioning in RGB-D Scans In Chapter 4, we introduce the task of dense description generation in RGB-D scans. We propose an end-to-end trained architecture to localize the 3D objects in the input point cloud and generate descriptions for them in natural language. Thus, we address the 3D localization and description generation problems at the same time. We apply an attention-based captioning pipeline equipped with a message passing network to generate descriptive tokens while referring to related components in the local context. Our architecture effectively localizes and describes 3D objects, outperforming 2D-based dense captioning methods on the 3D dense description generation task by a large margin. Overall, we hope that our work will enable future research in 3D vision and language.

D^{3}Net: A Unified Speaker-Listener Architecture for 3D Dense Captioning and Visual Grounding In Chapter 5, we present D^{3}Net, an end-to-end speaker-listener architecture that can **d**etect, **d**escribe and **d**iscriminate. Specifically, the speaker iteratively generates descriptive tokens given the object proposals detected by the detector, while the listener discriminates the object proposals in the scene with the generated captions. The self-discriminative property of D^{3}Net also enables semi-supervised training on ScanNet data without the annotated descriptions. Our method outperforms the previous SOTA methods in both tasks on ScanRefer, surpassing the previous SOTA 3D dense captioning method by a significant margin. Our architecture can serve as an initial step towards leveraging unannotated 3D data for language and 3D vision. Overall, we hope that our work will encourage more future research in 3D vision and language.

7 Limitations and Future Work

ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language In this work, we introduce the task of localizing a target object in a 3D point cloud using natural language descriptions. We propose the ScanRefer network for localizing an object with a free-formed description as reference, which first proposes point clusters of interest and then matches them to the embeddings of the input sentence. Since the last hidden state of a recurrent neural network is used as the final language feature, the underlying importance within words is not handled in this work. One potential improvement is to apply self-attention mechanism over the text inputs to enhance the language feature learning, which we would like to leave for future research.

Scan2Cap: Context-aware Dense Captioning in RGB-D Scans In Scan2Cap, we introduce the task of dense description generation in RGB-D scans. We propose an end-to-end trained architecture to localize the 3D objects in the input point cloud and generate descriptions for them in natural language. We apply an attention-based captioning pipeline equipped with a message passing network to generate descriptive tokens while referring to related components in the local context. Nevertheless, our method struggles to capture complex relations like ordinal counting. For instance, our method only predicts "the round chair next to another wooden chair", while the ground truth "the third round chair from the wall" reveals more fine-grained spatial relations, indicating possibilities for improvement. We would like to leave it for future research.

D^{3}Net: A Unified Speaker-Listener Architecture for 3D Dense Captioning and Visual Grounding In D^{3}Net, we propose an end-to-end neural architecture that can detect, describe, and discriminate. Our architecture facilitate self-critical learning. This is done by using a speaker module to iteratively generates descriptive tokens given the object proposals detected by the detector, and using a listener module to discriminate the object proposals in the scene with the generated captions. However, our method applies a reinforcement learning algorithm to approximate sample the gradients for a non-differentiable objective function. Such approach is hard to optimize in practice due to a huge variance in the sampled gradients. To enhance the training stability and facilitate the joint learning, one possible direction is to utilize multimodal transformer architecture for self-discriminative training. We would like to leave this thought to future work.

Bibliography

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[4] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[7] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell, "Natural language object retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4555–4564.

[8] R. Hu, M. Rohrbach, and T. Darrell, "Segmentation from natural language expressions," in *European Conference on Computer Vision*, Springer, 2016, pp. 108–124.

[9] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "ReferItGame: Referring to objects in photographs of natural scenes," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 787–798.

[10] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2641–2649.

[11] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 11–20.

[12] M. Acharya, K. Jariwala, and C. Kanan, "VQD: Visual query detection in natural scenes," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[13] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] Z. Chen, A. X. Chang, and M. Nießner, "Scanrefer: 3d object localization in rgb-d scans using natural language," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, Springer, 2020, pp. 202–221.

[15] Z. Chen, A. Gholami, M. Nießner, and A. X. Chang, "Scan2cap: Context-aware dense captioning in rgb-d scans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3193–3203.

[16] Z. Chen, Q. Wu, M. Nießner, and A. X. Chang, "D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, Springer, 2022, pp. 487–505.

[17] A. X. Chang, T. Funkhouser, L. Guibas, *et al.*, "ShapeNet: An information-rich 3D model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[18] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[20] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal ConvNets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[21] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[22] X. Chen, H. Fang, T.-Y. Lin, *et al.*, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[23] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 452–468.

[24] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[25] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "PointGroup: Dual-set point grouping for 3D instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4867–4876.

[26] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4421–4430.

[27] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, and J. Luo, "A fast and accurate one-stage approach to visual grounding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4683–4693.

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

[29] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," in *European Conference on Computer Vision*, Springer, 2016, pp. 69–85.

[30] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler, "What are you talking about? text-to-image coreference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3558–3565.

[31] A. Karpathy, A. Joulin, and L. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," in *Advances in neural information processing systems*, 2014, pp. 1889–1897.

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

[33] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[34] K. Xu, J. Ba, R. Kiros, *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[35] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5005–5013.

[36] Y. Huang, Q. Wu, C. Song, and L. Wang, "Learning semantic concepts and order for image and sentence matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6163–6171.

[37] L. Yu, Z. Lin, X. Shen, *et al.*, "MAttNet: Modular attention network for referring expression comprehension," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1307–1315.

[38] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in *European Conference on Computer Vision*, Springer, 2016, pp. 817–834.

[39] L. Wang, Y. Li, J. Huang, and S. Lazebnik, "Learning two-branch neural networks for image-text matching tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 394–407, 2018.

[40] B. A. Plummer, P. Kordas, M Hadi Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik, "Conditional image-text embedding networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 249–264.

[41] P. Dogan, L. Sigal, and M. Gross, "Neural sequential phrase grounding (SeqGROUND)," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4175–4184.

[42] X. Liu, Z. Wang, J. Shao, X. Wang, and H. Li, "Improving referring expression grounding with cross-modal attention-guided erasing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1950–1959.

[43] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, and A. Yuille, "Recurrent multimodal interaction for referring image segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1271–1280.

[44] R. Li, K. Li, Y.-C. Kuo, *et al.*, "Referring image segmentation via recurrent refinement networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5745–5753.

[45] E. Margffoy-Tuay, J. C. Pérez, E. Botero, and P. Arbeláez, "Dynamic multimodal instance segmentation guided by natural language queries," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 630–645.

[46] L. Ye, M. Rochan, Z. Liu, and Y. Wang, "Cross-modal self-attention network for referring image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 502–10 511.

[47] D.-J. Chen, S. Jia, Y.-C. Lo, H.-T. Chen, and T.-L. Liu, "See-through-text grouping for referring image segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7454–7463.

[48] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*, Springer, 2014, pp. 391–405.

[49] A. Nguyen, T.-T. Do, I. Reid, D. G. Caldwell, and N. G. Tsagarakis, "Object captioning and retrieval with natural language," *arXiv preprint arXiv:1803.06152*, 2018.

[50] D. Liu, H. Zhang, F. Wu, and Z.-J. Zha, "Learning to assemble neural module tree networks for visual grounding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4673–4682.

[51] R. Hong, D. Liu, X. Mo, X. He, and H. Zhang, "Learning to compose and reason with language tree structures for visual grounding," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[52] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel, "Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1960–1968.

[53] S. Yang, G. Li, and Y. Yu, "Cross-modal relationship inference for grounding referring expressions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4145–4154.

[54] S. Yang, G. Li, and Y. Yu, "Dynamic graph attention for referring expression comprehension," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4644–4653.

[55] L. Yu, H. Tan, M. Bansal, and T. L. Berg, "A joint speaker-listener-reinforcer model for referring expressions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7282–7290.

[56] F. Xiao, L. Sigal, and Y. Jae Lee, "Weakly-supervised visual grounding of phrases with linguistic structures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5945–5954.

[57] F. Zhao, J. Li, J. Zhao, and J. Feng, "Weakly supervised phrase localization with multi-scale anchored transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5696–5705.

[58] S. Datta, K. Sikka, A. Roy, K. Ahuja, D. Parikh, and A. Divakaran, "Align2Ground: Weakly supervised phrase grounding guided by image-caption alignment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[59] A. Sadhu, K. Chen, and R. Nevatia, "Zero-shot grounding of objects from natural language queries," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4694–4703.

[60] C. Mauceri, M. Palmer, and C. Heckman, "SUN-Spot: An RGB-D dataset with spatial referring expressions," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[61] Y. Qi, Q. Wu, P. Anderson, M. Liu, C. Shen, and A. v. d. Hengel, "REVERIE: Remote embodied visual referring expression in real indoor environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[62] J. Hou, A. Dai, and M. Nießner, "3D-SIC: 3D semantic instance completion for RGB-D scans," *arXiv preprint arXiv:1904.12012*, 2019.

[63] J. Lahoud, B. Ghanem, M. Pollefeys, and M. R. Oswald, "3D instance segmentation via multi-task metric learning," *arXiv preprint arXiv:1906.08650*, 2019.

[64] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "PanopticFusion: Online volumetric semantic mapping at the level of stuff and things," *arXiv preprint arXiv:1903.01177*, 2019.

[65] C. Elich, F. Engelmann, J. Schult, T. Kontogianni, and B. Leibe, "3D-BEVIS: Birds-eye-view instance segmentation," *arXiv preprint arXiv:1904.02199*, 2019.

[66] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.

[67] A. Chang, A. Dai, T. Funkhouser, *et al.*, "Matterport3D: Learning from RGB-D data in indoor environments," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2017.

[68] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[69] B. Yang, J. Wang, R. Clark, *et al.*, "Learning object bounding boxes for 3D instance segmentation on point clouds," *arXiv preprint arXiv:1906.01140*, 2019.

[70] F. Engelmann, T. Kontogianni, and B. Leibe, "Dilated point convolutions: On the receptive field of point convolutions," *arXiv preprint arXiv:1907.12046*, 2019.

[71] P. V. Hough, "Machine analysis of bubble chamber pictures," in *Conf. Proc.*, vol. 590914, 1959, pp. 554–558.

[72] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 375–383.

[73] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, 2014, pp. 7–16.

[74] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

[75] S. Li, T. Xiao, H. Li, W. Yang, and X. Wang, "Identity-aware textual-visual matching with latent co-attention," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1890–1899.

[76] Y. Huang, W. Wang, and L. Wang, "Instance-aware image and sentence matching with selective multimodal LSTM," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2310–2318.

[77] J. Gu, J. Cai, S. R. Joty, L. Niu, and G. Wang, "Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7181–7189.

[78] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[79] S. Sharma, D. Suhubdy, V. Michalski, S. E. Kahou, and Y. Bengio, "ChatPainter: Improving text to image generation using dialogue," *arXiv preprint arXiv:1802.08216*, 2018.

[80] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese, "Text2Shape: Generating shapes from natural language by learning joint embeddings," in *Proc. Asian Conference on Computer Vision (ACCV)*, 2018.

[81] P. Achlioptas, J. Fan, R. X. Hawkins, N. D. Goodman, and L. J. Guibas, "ShapeGlot: Learning language for shape differentiation," in *Proc. International Conference on Computer Vision (ICCV)*, 2019.

[82] M. Prabhudesai, H.-Y. F. Tung, S. A. Javed, M. Sieb, A. W. Harley, and K. Fragkiadaki, "Embodied language grounding with implicit 3D visual feature representations," *arXiv preprint arXiv:1910.01210*, 2019.

[83] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[84] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," To appear, 2017.

[85] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[86] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[87] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[88] P. Anderson, X. He, C. Buehler, *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[89] L. Yang, K. Tang, J. Yang, and L.-J. Li, "Dense captioning with joint inference and visual context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2193–2202.

[90] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, and J. Shao, "Context and attribute grounded dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6241–6250.

[91] D.-J. Kim, J. Choi, T.-H. Oh, and I. S. Kweon, “Dense relational captioning: Triple-stream networks for relationship-based captioning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6271–6280.

[92] X. Li, S. Jiang, and J. Han, “Learning object context for dense captioning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8650–8657.

[93] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, “SceneNN: A scene meshes dataset with annotations,” in *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, 2016, pp. 92–101.

[94] J. Hou, A. Dai, and M. Nießner, “3D-SIS: 3D semantic instance segmentation of RGB-D scans,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4421–4430.

[95] J. Hou, A. Dai, and M. Nießner, “RevealNet: Seeing behind objects in RGB-D scans,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2098–2107.

[96] C. R. Qi, X. Chen, O. Litany, and L. J. Guibas, “ImVoteNet: Boosting 3D object detection in point clouds with image votes,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2020.

[97] J. Donahue, L. Anne Hendricks, S. Guadarrama, *et al.*, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[98] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu, and T. Zhang, “Recurrent fusion network for image captioning,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 499–515.

[99] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical sequence training for image captioning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7008–7024.

[100] J. Lu, J. Yang, D. Batra, and D. Parikh, “Neural baby talk,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7219–7228.

[101] L. Gao, B. Wang, and W. Wang, “Image captioning with scene-graph based semantic concepts,” in *Proceedings of the International Conference on Machine Learning and Computing*, 2018, pp. 225–229.

[102] T. Yao, Y. Pan, Y. Li, and T. Mei, “Exploring visual relationship for image captioning,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 684–699.

[103] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 685–10 694.

[104] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 578–10 587.

[105] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. Guibas, "ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[106] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the International Conference on Machine Learning*, 2017.

[107] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "Scan2CAD: Learning CAD model alignment in RGB-D scans," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019.

[108] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[109] R. Vedantam, C Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[110] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[111] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[112] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[113] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[114] T. B. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[115] Y. Liu, M. Ott, N. Goyal, *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[116] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[117] Z. Yuan, X. Yan, Y. Liao, *et al.*, "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*, 2021, pp. 1791–1800.

[118] L. Zhao, D. Cai, L. Sheng, and D. Xu, "3DVG-Transformer: Relation modeling for visual grounding on point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2928–2937.

[119] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. Guibas, "ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes," in *European Conference on Computer Vision*, Springer, 2020, pp. 422–440.

[120] P.-H. Huang, H.-H. Lee, H.-T. Chen, and T.-L. Liu, "Text-guided graph neural networks for referring 3D instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[121] Z. Yuan, X. Yan, Y. Liao, *et al.*, *X-Trans2Cap: Cross-modal knowledge transfer using transformer for 3D dense captioning*, arXiv:2203.00843, 2022. arXiv: `2203.00843 [cs.CV]`.

[122] R. Luo, B. Price, S. Cohen, and G. Shakhnarovich, "Discriminability objective for training descriptive captions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6964–6974.

[123] X. Liu, H. Li, J. Shao, D. Chen, and X. Wang, "Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 338–354.

[124] X. Wu, H. Averbuch-Elor, J. Sun, and N. Snavely, "Towers of babel: Combining images, language, and 3D geometry for learning multimodal vision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[125] J. Roh, K. Desingh, A. Farhadi, and D. Fox, "LanguageRefer: Spatial-language model for 3D visual grounding," in *Proceedings of the Conference on Robot Learning*, 2021.

[126] J. Thomason, M. Shridhar, Y. Bisk, C. Paxton, and L. Zettlemoyer, "Language grounding with 3D objects," in *Proceedings of the Conference on Robot Learning*, 2021.

[127] O. Sidorov, R. Hu, M. Rohrbach, and A. Singh, "Textcaps: A dataset for image captioning with reading comprehension," in *European Conference on Computer Vision*, Springer, 2020, pp. 742–758.

[128] P. Cole and J. L. Morgan, "Syntax and semantics. volume 3: Speech acts," *Tijdschrift Voor Filosofie*, vol. 39, no. 3, 1977.

[129] D. Golland, P. Liang, and D. Klein, "A game-theoretic approach to generating spatial descriptions," in *Proceedings of the 2010 conference on empirical methods in natural language processing*, 2010, pp. 410–419.

[130] R. Luo and G. Shakhnarovich, “Comprehension-guided referring expressions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7102–7111.

[131] D. Fried, R. Hu, V. Cirik, *et al.*, “Speaker-follower models for vision-and-language navigation,” in *Advances in Neural Information Processing Systems*, 2018.

[132] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[133] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” *arXiv preprint arXiv:1511.06732*, 2015.

[134] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, “Generating visual explanations,” in *European conference on computer vision*, Springer, 2016, pp. 3–19.

[135] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” *arXiv preprint arXiv:1611.01144*, 2016.

[136] A. Paszke, S. Gross, F. Massa, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035.

Appendix

A Open-source Code & Videos

A.1 ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language

- Project: `https://daveredrum.github.io/ScanRefer/`
- Source Code: `https://github.com/daveredrum/ScanRefer`
- Video: `https://www.youtube.com/watch?v=T9J5t-UEcNA`

A.2 Scan2Cap: Context-aware Dense Captioning in RGB-D Scans

- Project: `https://daveredrum.github.io/Scan2Cap/`
- Source Code: `https://github.com/daveredrum/Scan2Cap`
- Video: `https://www.youtube.com/watch?v=AgmIpDbwTCY`

A.3 D3Net: A Unified Speaker-Listener Architecture for 3D Dense Captioning and Visual Grounding

- Project: `https://daveredrum.github.io/D3Net/`
- Source Code: `https://github.com/daveredrum/D3Net`
- Video: `https://www.youtube.com/watch?v=mIPNzoVOGN4`

B Authored and Co-authored Publications

1. Z. Chen, A. X. Chang, and M. Nießner, "Scanrefer: 3d object localization in rgb-d scans using natural language," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, Springer, 2020, pp. 202–221

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

3. Z. Chen, Q. Wu, M. Nießner, and A. X. Chang, "D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, Springer, 2022, pp. 487–505

C Original Publications

ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language

Dave Zhenyu Chen[1]   Angel X. Chang[2]   Matthias Nießner[1]

[1]Technical University of Munich   [2]Simon Fraser University

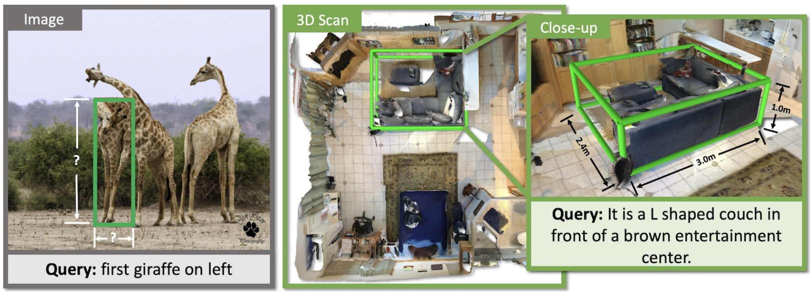


Image
?
?
Query: first giraffe on left
3D Scan
Close-up
1.0m
2.4m
3.0m
Query: It is a L shaped couch in front of a brown entertainment center.

Fig. 1: We introduce the task of object localization in 3D scenes using natural language. Given as input a 3D scene and a natural language expression, we predict the bounding box for the target 3D object (right). The counterpart 2D task (left) does not capture the physical extent of the 3D objects.

Abstract. We introduce the task of 3D object localization in RGB-D scans using natural language descriptions. As input, we assume a point cloud of a scanned 3D scene along with a free-form description of a specified target object. To address this task, we propose **ScanRefer**, learning a fused descriptor from 3D object proposals and encoded sentence embeddings. This fused descriptor correlates language expressions with geometric features, enabling regression of the 3D bounding box of a target object. We also introduce the ScanRefer dataset, containing $51,583$ descriptions of $11,046$ objects from 800 ScanNet [9] scenes. ScanRefer is the first large-scale effort to perform object localization via natural language expression directly in 3D [1].

1 Introduction

In recent years, there has been tremendous progress in both semantic understanding and localization of objects in 2D images from natural language (also known as visual grounding). Datasets such as ReferIt [28], RefCOCO [71], and

[1] Project page: `https://daveredrum.github.io/ScanRefer/`

Flickr30K Entities [47] have enabled the development of various methods for visual grounding in 2D [23, 22, 39]. However, these methods and datasets are restricted to 2D images, where object localization fails to capture the true 3D extent of an object (see Fig. 1, left). This is a limitation for applications ranging from assistive robots to AR/VR agents where understanding the global 3D context and the physical size is important, e.g., finding objects in large spaces, interacting with them, and understanding their spatial relationships. Early work by Kong et al. [31] looked at coreference in 3D, but was limited to single-view RGB-D images.

In this work, we address these shortcomings by proposing the task of object localization using natural language directly in 3D space. Specifically, we develop a neural network architecture that localizes objects in 3D point clouds given natural language descriptions referring to the underlying objects; i.e., for a given text description in a 3D scene, we predict a corresponding 3D bounding box matching the best-described object. To facilitate the task, we collect the ScanRefer dataset, which provides natural language descriptions for RGB-D scans in ScanNet [9]. In total, we acquire $51,583$ descriptions of $11,046$ objects. To the best of our knowledge, our ScanRefer dataset is the first large-scale effort that combines 3D scene semantics and free-form descriptions. In summary, our contributions are as follows:

- We introduce the task of localizing objects in 3D environments using natural language descriptions.
- We provide the ScanRefer dataset containing $51,583$ human-written free-form descriptions of $11,046$ objects in 3D scans.
- We propose a neural network architecture for localization based on language descriptions that directly fuses features from 2D images and language expressions with 3D point cloud features.
- We show that our end-to-end method outperforms the best 2D visual grounding method that simply backprojects its 2D predictions to 3D by a significant margin (9.04 Acc@0.5IoU vs. 22.39 Acc@0.5IoU).

2 Related Work

Grounding Referring Expressions in Images. There has been much work connecting images to natural language descriptions across tasks such as image captioning [27, 26, 59, 64], text-to-image retrieval [61, 25], and visual grounding [23, 39, 70]. The task of visual grounding (with variants also known as referring expression comprehension or phrase localization) is to localize a region described by a given referring expression, the query. Localization can be specified by a 2D bounding box [28, 47, 39] or a segmentation mask [22], with the input description being short phrases [28, 47] or more complex descriptions [39]. Recently, Acharya et al [1] proposed visual query detection where the input is a question. The focus of our work is to lift this task to 3D, focusing on complex descriptions that can localize an unique object in a scene.

dataset	#objects	#expressions	AvgLeng	data format	3D context
ReferIt [28]	96,654	130,364	3.51	image	-
RefCOCO [71]	50,000	142,209	3.50	image	-
Google RefExp [39]	49,820	95,010	8.40	image	-
SUN-Spot [41]	3,245	7,990	14.04	image	depth
REVERIE [52]	4,140	21,702	18.00	image	panoramic image
ScanRefer (ours)	**11,046**	**51,583**	**20.27**	**3D scan**	**depth, size, location, etc.**

Table 1: Comparison of referring expression datasets in terms of the number of objects (#objects), number of expressions (#expressions), average lengths of the expressions, data format and the 3D context.

Existing methods focus on predicting 2D bounding boxes [23, 55, 61, 60, 46, 71, 70, 12, 37] and some predict segmentation masks [22, 35, 33, 40, 69, 6]. A two-stage pipeline is common, where first an object detector, either unsupervised [74] or pretrained [54], is used to propose regions of interest, and then the regions are ranked by similarity to the query, with the highest scoring region provided as the final output. Other methods address the referring expression task with a single stage end-to-end network [22, 43, 68]. There are also approaches that incorporate syntax [36, 17], use graph attention networks [62, 66, 67], speaker-listener models [39, 72], weakly supervised methods [63, 73, 11] or tackle zero-shot settings for unseen nouns [56].

However, all these methods operate on 2D image datasets [47, 28, 71]. A recent dataset [41] integrates RGB-D images but lacks the complete 3D context beyond a single image. Qi et al. [52] study referring expressions in an embodied setting, where semantic annotations are projected from 3D to 2D bounding boxes on images observed by an agent. Our contribution is to lift NLP tasks to 3D by introducing the first large-scale effort that couples free-form descriptions to objects in 3D scans. Tab. 1 summarizes the difference between our ScanRefer dataset and existing 2D datasets.

Object Detection in 3D. Recent work on 3D object detection on volumetric grids [20, 19, 32, 42, 13] has been applied to several 3D RGB-D datasets [58, 9, 4]. As an alternative to regular grids, point-based methods, such as PointNet [50] or PointNet++ [51], have been used as backbones for 3D detection and/or object instance segmentation [65, 14]. Recently, Qi et al. [49] introduced VoteNet, a 3D object detection method for point clouds based on Hough Voting [21]. Our approach extracts geometric features in a similar fashion, but backprojects 2D feature information since the color signal is useful for describing 3D objects with natural language.

3D Vision and Language. Vision and language research is gaining popularity in image domains (e.g., image captioning [26, 59, 64, 38], image-text matching [15, 30, 34, 24, 16], and text-to-image generation [53, 16, 57]), but there is little work on vision and language in 3D. Chen et al. [7] learn a joint embedding of 3D shapes from ShapeNet [5] and corresponding natural language descriptions. Achlioptas et al. [3] disambiguate between different objects using language. Recent work has started to investigate grounding of language to 3D by identifying

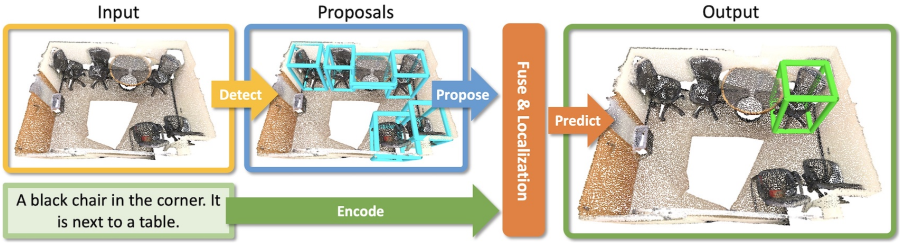


Fig. 2: Our task: ScanRefer takes as input a 3D scene point cloud and a description of an object in the scene, and predicts the object bounding box.

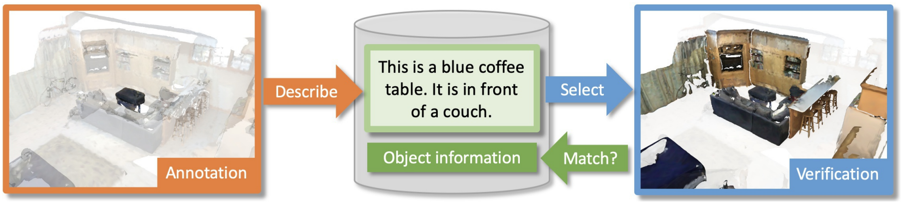


Fig. 3: Our data collection pipeline. The annotator writes a description for the focused object in the scene. Then, a verifier selects the objects that match the description. The selected object is compared with the target object to check that it can be uniquely identified by the description.

3D bounding boxes of target objects for simple arrangements of primitive shapes of different colors [48]. Instead of focusing on isolated objects, we consider large 3D RGB-D reconstructions that are typical in semantic 3D scene understanding. A closely related work by Kong et al. [31] studied the problem of coreference in text description of single-view RGB-D images of scenes, where they aimed to connect noun phrases in a scene description to 3D bounding boxes of objects. Concurrent with this work, Achlioptas et al. [2] introduces a new dataset and task that focuses on disambiguating objects from the same category with known localizations.

3 Task

We introduce the task of object localization in 3D scenes using natural language (Fig. 2). The input is a 3D scene and free-form text describing an object in the scene. The scene is represented as a point cloud with additional features such as colors and normals for each point. The goal is to predict the 3D bounding box of the object that matches the input description.

4 Dataset

The ScanRefer dataset is based on ScanNet [9] which is composed of 1,613 RGB-D scans taken in 806 unique indoor environments. We provide 5 descriptions for

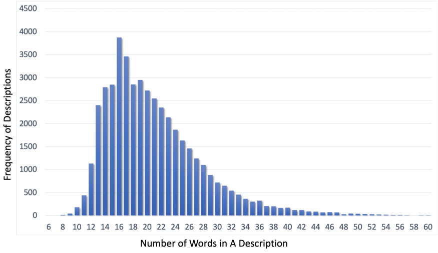
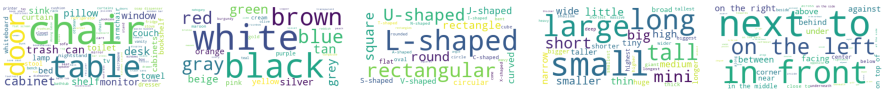


Fig. 4: Description lengths

Number of descriptions	51,583
Number of scenes	800
Number of objects	11,046
Number of objects per scene	13.81
Number of descriptions per scene	64.48
Number of descriptions per object	4.67
Size of vocabulary	4,197
Average length of descriptions	20.27

Table 2: ScanRefer dataset statistics.



(a) (b) (c) (d) (e)

Fig. 5: Word clouds of terms for (a) object names (b) colors (c) shapes (d) sizes, and (e) spatial relations for the ScanRefer dataset. Bigger fonts indicate more frequent terms in the descriptions.

each object in each scene, focusing on complete coverage of all objects that are present in the reconstruction. Here, we summarize the annotation process and statistics of our dataset (see supplement for more details).

4.1 Data Collection

We deploy a web-based annotation interface on Amazon Mechanical Turk (AMT) to collect object descriptions in the ScanNet scenes. The annotation pipeline consists of two stages: i) description collection, and ii) verification (Fig. 3). From each scene, we select objects to annotate by restricting to indoor furniture categories and excluding structural objects such as “Floor” and “Wall”. We manually check the selected objects are recognizable and filter out objects with reconstructions that are too incomplete or hard to identify.

Annotation The 3D web-based UI shows each object in context. The workers see all objects other than the target object faded out and a set of captured image frames to compensate for incomplete details in the reconstructions. The initial viewpoint is random but includes the target object. Camera controls allow for adjusting the camera view to better examine the target object. We ask the annotator to describe the appearance of the target and its spatial location relative to other objects. To ensure the descriptions are informative, we require the annotator to provide at least two full sentences. We batch and randomize the tasks so that each object is described by five different workers.

Verification We recruit trained workers (students) to verify that the descriptions are discriminative and correct. Verifiers are shown the 3D scene and a description, and are asked to select the objects (potentially multiple) in the

1. There is a brown wooden chair placed right against the wall.
2. This is a triangular shape table. The table is near the armchair.
3. The little nightstand. The nightstand is on the right of the bed.
4. This is a short trash can. It is in front of a taller trash can.
5. The couch is the biggest one below the picture. The couch has three seats and is brown.
6. This is a gray desk chair. This chair is the last one on the side closest to the open door.
7. The kitchen counter is covering the lower cabinets. The kitchen counter is under the upper cabinets that are mounted above.
8. This is a round bar stool. It is third from the wall.

Table 3: Examples from our dataset illustrating different types of phrases such as attributes (1-8) and parts (5), comparatives (4), superlatives (5), intra-class spatial relations (6), inter-class spatial relations (7) and ordinal numbers (8).

scene that match the description. Descriptions that result in the wrong object or multiple objects are filtered out. Verifiers also correct spelling and wording issues in the description when necessary. We filter out 2,823 invalid descriptions that do not match the target objects and fix writing issues for 2,129 descriptions.

4.2 Dataset Statistics

We collected 51,583 descriptions for 800 ScanNet scenes[2]. On average, there are 13.81 objects, 64.48 descriptions per scene, and 4.67 descriptions per object after filtering (see Tab. 2 for basic statistics, Tab. 3 for sample descriptions, and Fig. 4 for the distribution of the description lengths). The descriptions are complex and diverse, covering over 250 types of common indoor objects, and exhibiting interesting linguistic phenomena. Due to the complexity of the descriptions, one of the key challenges of our task is to determine what parts of the description describe the target object, and what parts describe neighboring objects. Among those descriptions, 41,034 mention object attributes such as color, shape, size, etc. We find that many people use spatial language (98.7%), color (74.7%), and shape terms (64.9%). In contrast, only 14.2% of the descriptions convey size information. Fig 5 shows commonly used object names and attributes. Tab. 3 shows interesting expressions, including comparatives ("taller") and superlatives ("the biggest one"), as well as phrases involving ordinals such as "third from the wall". Overall, there are 672 and 2,734 descriptions with comparative and superlative phrases. We provide more detailed statistics in the supplement.

5 Method

Our architecture consists of two main modules: 1) detection & encoding; 2) fusion & localization (Fig. 6). The detection & encoding module encodes the input point cloud and description, and outputs the object proposals and the language embedding, which are fed into the fusion module to mask out invalid

[2] 6 scenes are excluded since they do not contain any objects to describe

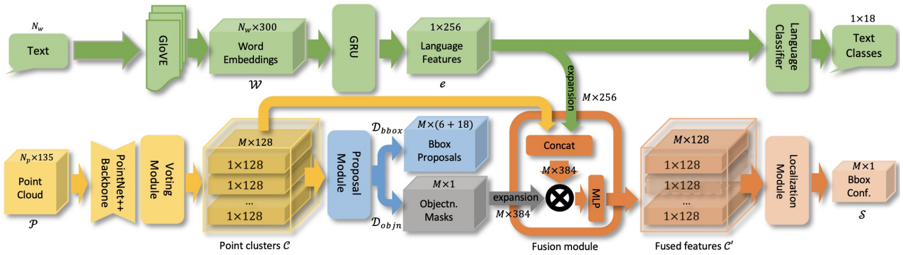


Fig. 6: ScanRefer architecture: The PointNet++ [51] backbone takes as input a point cloud and aggregates it to high-level point feature maps, which are then clustered and fused as object proposals by a voting module similar to Qi et al. [49]. Object proposals are masked by the objectness predictions, and then fused with the sentence embedding of the input descriptions, which is obtained by a GloVE [45] + GRU [8] embedding. In addition, an extra language-to-object classifier serves as a proxy loss. We apply a softmax function in the localization module to output the confidence scores for the object proposals.

object proposals and produce the fused features. Finally, the object proposal with the highest confidence predicted by the localization module is chosen as the final output.

5.1 Data Representations

Point clouds We randomly sample N_P vertices of one scan from ScanNet as the input point cloud $\mathcal{P} = \{(p_i, f_i)\}$, where $p_i \in \mathcal{R}^3$ represents the point coordinates in 3D space and f_i stands for additional point features such as colors and normals. Note that the point coordinates p_i provides only geometrical information and does not contain other visual information such as color and texture. Since descriptions of objects do refer to attributes such as color and texture, we incorporate visual appearance by adapting the feature projection scheme in Dai et al. [10] to project multi-view image features $v_i \in \mathcal{R}^{128}$ to the point cloud. The image features are extracted using a pre-trained ENet [44]. Following Qi et al. [49], we also append the height of the point from the ground and normals to the new point features $f'_i \in \mathcal{R}^{135}$. The final point cloud data is prepared offline as $\mathcal{P}' = \{(p_i, f'_i)\} \in \mathcal{R}^{N_P \times 135}$. We set N_P to $40,000$ in our experiments.
Descriptions We tokenize the input description with SpaCy [18] and the N_W tokens to 300-dimensional word embedding vectors $\mathcal{W} = \{w_j\} \in \mathcal{R}^{N_W \times 300}$ using pretrained GloVE word embeddings [45].

5.2 Network Architecture

Our method takes as input the preprocessed point cloud $\mathcal{P}'$ and the word embedding sequence $\mathcal{W}$ representing the input description and outputs the 3D bounding box for the proposal which is most likely referred to by the input description. Conceptually, our localization pipeline consists of the following four stages: detection, encoding, fusion and localization.

Detection As the first step in our network, we detect all probable objects in the given point cloud. To construct our detection module, we adapt the PointNet++ [51] backbone and the voting module in Qi et al. [49] to process the point cloud input and aggregate all object candidates to individual clusters. The output from the voting module is a set of point clusters $\mathcal{C} \in \mathcal{R}^{M \times 128}$ representing all object proposals with enriched point features, where M is the upper bound of the number of proposals. Next, the proposal module takes in the point clusters and processes those clusters to predict the objectness mask $\mathcal{D}_{\text{objn}} \in \mathcal{R}^{M \times 1}$ and the axis-aligned bounding boxes $\mathcal{D}_{\text{bbox}} \in \mathcal{R}^{M \times (6+18)}$ for all M proposals, where each $\mathcal{D}^i_{\text{bbox}} = (c_x, c_y, c_z, r_x, r_y, r_z, l)$ consists of the box center c, the box lengths r and a vector $l \in \mathcal{R}^{18}$ representing the semantic predictions.

Encoding The sequences of word embedding vectors of the input description are fed into a GRU cell [8] to aggregate the textual information. We take the final hidden state $e \in \mathcal{R}^{256}$ of the GRU cell as the final language embedding.

Fusion The outputs from the previous detection and encoding modules are fed into the fusion module (orange block in Fig. 6, see supplemental for details) to integrate the point features together with the language embeddings. Specifically, each feature vector $c_i \in \mathcal{R}^{128}$ in the point cluster $\mathcal{C}$ is concatenated with the language embedding $e \in \mathcal{R}^{256}$ as the extended feature vector, which is then masked by the predicted objectness mask $\mathcal{D}^i_{\text{objn}} \in \{0, 1\}$ and fused by a multi-layer perceptron as the the final fused cluster features $C' = \{c'_i\} \in \mathcal{R}^{M \times 128}$.

Localization The localization module aims to predict which of the proposed bounding boxes corresponds to the description. Point clusters with fused cluster features $\mathcal{C}' = \{c'_i\}$ are processed by a single layer perceptron to produce the raw scores of how likely each box is the target box. We use a softmax function to squash all the raw scores into the interval of $[0, 1]$ as the localization confidences $S = \{s_i\} \in \mathcal{R}^{M \times 1}$ for the proposed M bounding boxes.

5.3 Loss Function

Localization loss For the predicted localization confidence $s_i \in [0, 1]$ for object proposal $\mathcal{D}^i_{\text{bbox}}$, the target label is represented as $t_i \in \{0, 1\}$. Following the strategy of Yang et al. [68], we set the label t_j for the j^{th} box that has the highest IoU score with the ground truth box as 1 and others as 0. We then use a cross-entropy loss as the localization loss $\mathcal{L}_{\text{loc}} = -\sum_{i=1}^{M} t_i \log(s_i)$.

Object detection loss We use the same detection loss $\mathcal{L}_{det}$ as introduced in Qi et al. [49] for object proposals $\mathcal{D}^i_{\text{bbox}}$ and $\mathcal{D}^i_{\text{objn}}$: $\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{vote-reg}} + 0.5\mathcal{L}_{\text{objn-cls}} + \mathcal{L}_{\text{box}} + 0.1\mathcal{L}_{\text{sem-cls}}$, where $\mathcal{L}_{\text{vote-reg}}$, $\mathcal{L}_{\text{objn-cls}}$, $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{sem-cls}}$ represent the vote regression loss (defined in Qi et al. [49]), the objectness binary classification loss, box regression loss and the semantic classification loss for the 18 ScanNet benchmark classes, respectively. We ignore the bounding box orientations in our task and simplify $\mathcal{L}_{\text{box}}$ as $\mathcal{L}_{\text{box}} = \mathcal{L}_{\text{center-reg}} + 0.1\mathcal{L}_{\text{size-cls}} + \mathcal{L}_{\text{size-reg}}$, where

$\mathcal{L}_{\text{center-reg}}$, $\mathcal{L}_{\text{size-cls}}$ and $\mathcal{L}_{\text{size-reg}}$ are used for regressing the box center, classifying the box size and regressing the box size, respectively. We refer readers to Qi et al. [49] for more details.

Language to object classification loss To further supervise the training, we include an object classification loss based on the input description. We consider the 18 ScanNet benchmark classes (excluding the label "Floor" and "Wall"). The language to object classification loss $\mathcal{L}_{\text{cls}}$ is a multi-class cross-entropy loss.

Final loss The final loss is a linear combination of the localization loss, object detection loss and the language to object classification loss: $\mathcal{L} = \alpha\mathcal{L}_{\text{loc}} + \beta\mathcal{L}_{\text{det}} + \gamma\mathcal{L}_{\text{cls}}$, where α, β and γ are the weights for the individual loss terms. After fine-tuning on the validation split, we set those weights to 0.1, 10, and 1 in our experiments to ensure the loss terms are roughly of the same magnitude.

5.4 Training and Inference

Training During training, the detection and encoding modules propose object candidates as point clusters, which are then fed into the fusion and localization modules to fuse the features from the previous module and predict the final bounding boxes. We train the detection backbone end-to-end with the detection loss. In the localization module, we use a softmax function to compress the raw scores to $[0, 1]$. The higher the predicted confidence is, the more likely the proposal will be chosen as output. To filter out invalid object proposals, we use the predicted objectness mask to ensure that only positive proposals are taken into account. We set the maximum number of proposals M to 256 in practice.

Inference Since there can be overlapping detections, we apply a non-maximum suppression module to suppress those overlapping proposals in the inference step. The remaining object proposals are fed into the localization module to predict the final score for each proposal. The number of object proposals is less than the upper bound M in the training step.

Implementation Details We implement our architecture using PyTorch and train the model end-to-end using ADAM [29] with a learning rate of 1e$-$3. We train the model for roughly $130,000$ iterations until convergence. To avoid overfitting, we set the weight decay factor to 1e$-$5 and apply data augmentations to our training data. For point clouds, we apply rotation about all three axes by a random angle in $[-5°, 5°]$ and randomly translate the point cloud within 0.5 meters in all directions. We rotate around all axes (not just up), since the ground alignment in ScanNet is imperfect.

6 Experiments

Train/Val/Test Split. Following the official ScanNet [9] split, we split our data into train/val/test sets with 36,665, 9,508 and 5,410 samples respectively, ensuring disjoint scenes for each split. Results and analysis are conducted on the val split (except for results in Tab. 4 bottom). The test set is hidden and will be reserved for the ScanRefer benchmark.

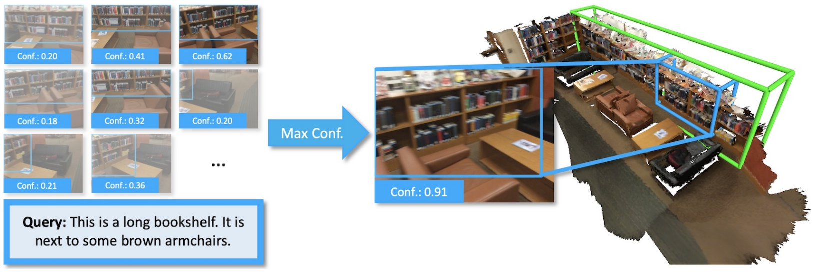


Fig. 7: Object localization in an image using a 2D grounding method and back-projecting the result to the 3D scene (blue box) vs. directly localizing in the 3D scene (green box). Grounding in 2D images suffers from the limited view of a single frame, which results in inaccurate 3D bounding boxes.

Metric. To evaluate the performance of our method, we measure the thresholded accuracy where the positive predictions have higher intersection over union (IoU) with the ground truths than the thresholds. Similar to work with 2D images, we use Acc@kIoU as our metric, where the threshold value k for IoU is set to 0.25 and 0.5 in our experiments.

Baselines. We design several baselines by 1) evaluating our language localization module on ground truth bounding boxes, 2) adapting 3D object detectors, and 3) adapting 2D referring methods to 3D using back-projection.

OracleCatRand & OracleRefer: To examine the difficulty of our task, we use an oracle with ground truth bounding boxes of objects, and predict the box by simply selecting a random box that matches the object category (OracleCatRand) or our trained fusion and localization modules (OracleRefer).

VoteNetRand & VoteNetBest: From the predicted object proposals of the VoteNet backbone [49], we select one of the bounding box proposals, either by selecting a box randomly with the correct semantic class label (VoteNetRand) or the best matching box given the ground truth (VoteNetBest). VoteNetBest provides an upper bound on how well the object detection component works for our task, while VoteNetRand provides a measure of whether additional information beyond the semantic label is required.

SCRC & One-stage: 2D image baselines for referring expression comprehension by extending SCRC [23] and One-stage [68] to 3D using back-projection. Since 2D referring expression methods operate on a single image frame, we construct a 2D training set by using the recorded camera pose associated with each annotation to retrieve the frame from the scan video with the closest camera pose. At inference time, we sample frames from the scans (using every 20th frame) and predict the target 2D bounding boxes in each frame. We then select the 2D bounding box with the highest confidence score from the bounding box candidates and project it to 3D using the depth map for that frame (see Fig. 7).

Ours: We compare our full end-to-end model against using a pretrained VoteNet backbone with a trained GRU [8] for selecting a matching bounding box.

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
OracleCatRand (GT boxes + RandCat)	100.00	100.00	18.09	17.84	29.99	29.76
OracleRefer (GT boxes + GRU)	74.09	73.55	32.57	32.00	40.63	40.06
VoteNetRand (VoteNet[49] + RandCat)	34.34	19.35	5.73	2.81	10.00	5.28
VoteNetBest (VoteNet[49] + Best)	88.85	85.50	46.63	46.42	55.10	54.33
SCRC [23] + backproj	24.03	9.22	17.77	5.97	18.70	6.45
One-stage [68] + backproj	29.32	22.82	18.72	6.49	20.38	9.04
Ours (VoteNet[48] + GRU)	**77.33**	51.73	30.43	19.46	39.52	25.72
Ours (end-to-end)	76.33	**53.51**	**32.73**	**21.11**	**41.19**	**27.40**
Test results (ScanRefer benchmark)						
OracleRefer (GT boxes + GRU)	72.37	71.84	31.81	31.26	39.69	39.13
VoteNetBest (VoteNet[49] + Best)	86.78	83.85	45.54	45.33	53.82	53.07
Ours (VoteNet[48] + GRU)	**72.55**	**47.24**	32.90	19.16	41.79	25.45
Ours (end-to-end)	71.06	46.66	**35.17**	**20.92**	**43.22**	**26.69**

Table 4: Comparison of localization results obtained by our ScanRefer and baseline models. We measure percentage of predictions whose IoU with the ground truth boxes are greater than 0.25 and 0.5. We also report scores on “unique” and “multiple” subsets; unique means that there is only a single object of its class in the scene. We outperform all baselines by a significant margin.

6.1 Task Difficulty

To understand how informative the input description is beyond capturing the object category, we analyze the performance of the methods on “unique” and “multiple” subsets with 1,875 and 7,663 samples from val split, respectively. The “unique” subset contains samples where only one unique object from a certain category matches the description, while the “multiple” subset contains ambiguous cases where there are multiple objects of the same category. For instance, if there is only one refrigerator in a scene, it is sufficient to identify that the sentence refers to a refrigerator. In contrast, if there are multiple objects of the same category in a scene (e.g., chair), the full description must be taken into account. From the OracleCatRand baseline, we see that information from the description, other than the object category, is necessary to disambiguate between multiple objects (see Tab. 4 Acc@0.5IoU multiple). From the OracleRefer baseline, we see that using our fused language module, we are able to improve beyond over selecting a random object of the same category (multiple Acc@0.5IoU increases from 17.84% to 32.00%), but we often fail to identify the correct object category (unique Acc@0.5IoU drops from 100.0% to 73.55%).

6.2 Quantitative Analysis

We evaluate the performance of our model against baselines on the val and the hidden test split of ScanRefer which serves as the ScanRefer benchmark (see Tab. 4). Note that for all results using Ours and VoteNet for object proposal,

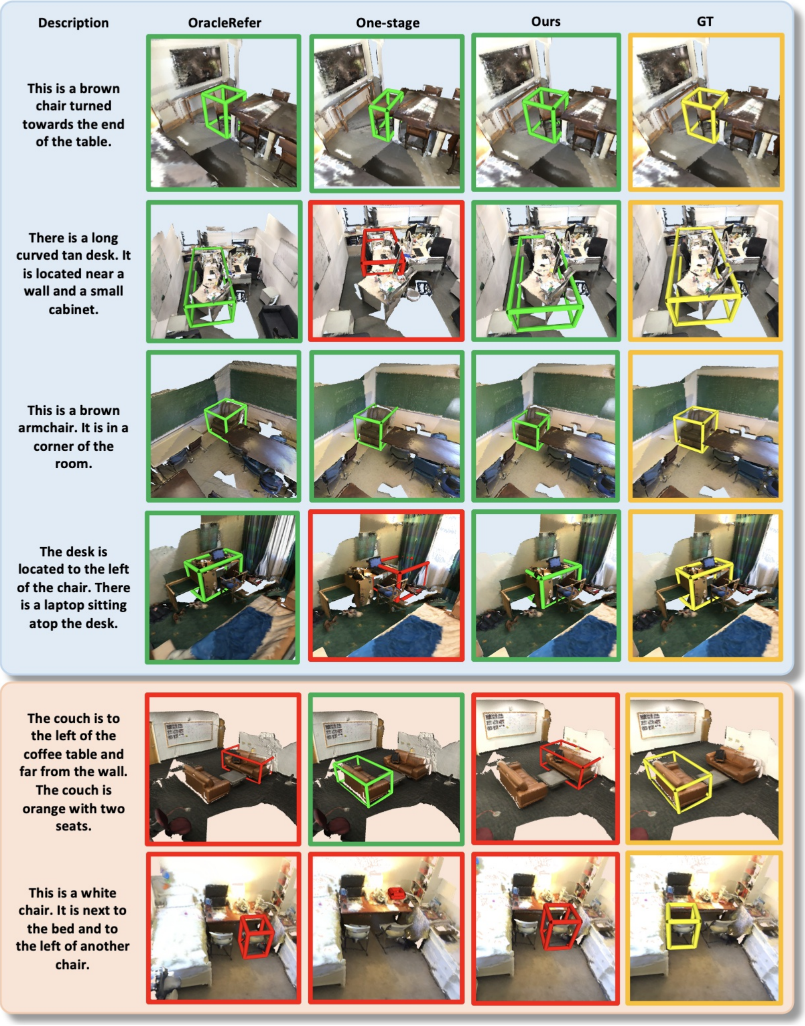

Description
OracleRefer
One-stage
Ours
GT
This is a brown chair turned towards the end of the table.
There is a long curved tan desk. It is located near a wall and a small cabinet.
This is a brown armchair. It is in a corner of the room.
The desk is located to the left of the chair. There is a laptop sitting atop the desk.
The couch is to the left of the coffee table and far from the wall. The couch is orange with two seats.
This is a white chair. It is next to the bed and to the left of another chair.

Fig. 8: Qualitative results from baseline methods and ScanRefer. Predicted boxes are marked green if they have an IoU score higher than 0.5, otherwise they are marked red. We show examples where our method produced good predictions (blue block) as well as failure cases (orange block). Image best viewed in color.

we take the average of 5 differently seeded subsamplings (of seed points and vote points) during inference (see supplemental for more details on experimental variance). Training the detection backbone jointly with the localization module (end-to-end) leads to a better performance when compared to the model trained separately (VoteNet[49] + GRU). However, as the accuracy gap between VoteNetBest and ours (end-to-end) indicates, there is still room for improving

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Ours (xyz)	63.98	43.57	29.28	18.99	36.01	23.76
Ours (xyz+rgb)	63.24	41.78	30.06	19.23	36.50	23.61
Ours (xyz+rgb+normals)	64.63	43.65	31.89	20.77	38.24	25.21
Ours (xyz+multiview)	77.20	52.69	32.08	19.86	40.84	26.23
Ours (xyz+multiview+normals)	**78.22**	52.38	33.61	20.77	**42.27**	26.90
Ours (xyz+lobjcls)	64.31	44.04	30.77	19.44	37.28	24.22
Ours (xyz+rgb+lobjcls)	65.00	43.31	30.63	19.75	37.30	24.32
Ours (xyz+rgb+normals+lobjcls)	67.64	46.19	32.06	**21.26**	38.97	26.10
Ours (xyz+multiview+lobjcls)	76.00	50.40	**34.05**	20.73	42.19	26.50
Ours (xyz+multiview+normals+lobjcls)	76.33	**53.51**	32.73	21.11	41.19	**27.40**

Table 5: Ablation study with different features. We measure the percentages of predictions whose IoU with the ground truth boxes are greater than 0.25 and 0.5. Unique means that there is only a single object of its class in the scene.

the match between language inputs and the visual signals. For the val split, we also include additional experiments on the 2D baselines and a comparison with VoteNetRand. With just category information, VoteNetRand is able to perform relatively well on the "unique" subset, but has trouble identifying the correct object in the "multiple" case. However, the gap between the VoteNetRand and OracleCatRand for the "unique" case shows that 3D object detection still need to be improved. Our method is able to improve over the bounding box predictions from VoteNetRand, and leverages additional information in the description to differentiate between ambiguous objects. It adapts better to the 3D context compared to the 2D methods (SCRC and One-stage) which is limited by the view of a single frame (see Fig. 7 and Fig. 8).

6.3 Qualitative Analysis

Fig. 8 shows results produced by OracleRefer, One-stage, and our method. The successful localization cases in the green boxes show our architecture can handle the semantic correlation between the scene contexts and the textual descriptions. In contrast, even provided with a pool of ground truth proposals, OracleRefer sometimes still fails to predict correct bounding boxes, while One-stage is limited by the single view and hence cannot produce accurate bounding boxes in 3D space. The failure case of OracleRefer suggests that our fusion & localization module can still be improved. Some failure cases of our method are displayed in the orange block in Fig. 8, indicating that our architecture cannot handle all spatial relations to distinguish between ambiguous objects.

6.4 Ablation Studies

We conduct an ablation study on our model to examine what components and point cloud features contribute to the performance (see Tab. 5).

Does a language-based object classifier help? To show the effectiveness of the extra supervision on input descriptions, we conduct an experiment with the language to object classifier (+lobjcls) and without. Architectures with a language to object classifier outperform ones without it. This indicates that it is helpful to predict the category of the target object based on the input description.
Do colors help? We compare our method trained with the geometry and multi-view image features (xyz+multiview+lobjcls) with a model trained with only geometry (xyz+lobjcls) and one trained with RGB values from the reconstructed meshes (xyz+rgb+lobjcls). ScanRefer trained with geometry and pre-processed multi-view image features outperforms the other two models. The performance of models with color information are higher than those that use only geometry.
Do other features help? We include normals from the ScanNet meshes to the input point cloud features and compare performance against networks trained without them. The additional 3D information improves performance. Our architecture trained with geometry, multi-view features, and normals (xyz+multiview+normals+lobjcls) achieves the best performance among all ablations.

7 Conclusion

In this work, we introduce the task of localizing a target object in a 3D point cloud using natural language descriptions. We collect the ScanReferdataset which contains 51,583 unique descriptions for 11,046 objects from 800 ScanNet [9] scenes. We propose an end-to-end method for localizing an object with a free-formed description as reference, which first proposes point clusters of interest and then matches them to the embeddings of the input sentence. Our architecture is capable of learning the semantic similarities of the given contexts and regressing the bounding boxes for the target objects. Overall, we hope that our new dataset and method will enable future research in the 3D visual language field.

Acknowledgements

We would like to thank the expert annotators Josefina Manieu Seguel and Rinu Shaji Mariam, all anonymous workers on Amazon Mechanical Turk and the student volunteers (Akshit Sharma, Yue Ruan, Ali Gholami, Yasaman Etesam, Leon Kochiev, Sonia Raychaudhuri) at Simon Fraser University for their efforts in building the ScanRefer dataset, and Akshit Sharma for helping with statistics and figures. This work is funded by Google (AugmentedPerception), the ERC Starting Grant Scan2CAD (804724), and a Google Faculty Award. We would also like to thank the support of the TUM-IAS Rudolf Mößbauer and Hans Fischer Fellowships (Focus Group Visual Computing), as well as the the German Research Foundation (DFG) under the Grant *Making Machine Learning on Static and Dynamic 3D Data Practical.* Angel X. Chang is supported by the Canada CIFAR AI Chair program. Finally, we thank Angela Dai for the video voice-over.

Bibliography

[1] Acharya, M., Jariwala, K., Kanan, C.: VQD: Visual query detection in natural scenes. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) (2019)

[2] Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)

[3] Achlioptas, P., Fan, J., Hawkins, R.X., Goodman, N.D., Guibas, L.J.: ShapeGlot: Learning language for shape differentiation. In: Proc. International Conference on Computer Vision (ICCV) (2019)

[4] Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: Proceedings of the International Conference on 3D Vision (3DV) (2017)

[5] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)

[6] Chen, D.J., Jia, S., Lo, Y.C., Chen, H.T., Liu, T.L.: See-through-text grouping for referring image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[7] Chen, K., Choy, C.B., Savva, M., Chang, A.X., Funkhouser, T., Savarese, S.: Text2Shape: Generating shapes from natural language by learning joint embeddings. In: Proc. Asian Conference on Computer Vision (ACCV) (2018)

[8] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)

[9] Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR) (2017)

[10] Dai, A., Nießner, M.: 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In: Proceedings of the European Conference on Computer Vision (2018)

[11] Datta, S., Sikka, K., Roy, A., Ahuja, K., Parikh, D., Divakaran, A.: Align2Ground: Weakly supervised phrase grounding guided by image-caption alignment. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[12] Dogan, P., Sigal, L., Gross, M.: Neural sequential phrase grounding (SeqGROUND). In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

[13] Elich, C., Engelmann, F., Schult, J., Kontogianni, T., Leibe, B.: 3D-BEVIS: Birds-eye-view instance segmentation. arXiv preprint arXiv:1904.02199 (2019)

[14] Engelmann, F., Kontogianni, T., Leibe, B.: Dilated point convolutions: On the receptive field of point convolutions. arXiv preprint arXiv:1907.12046 (2019)

[15] Feng, F., Wang, X., Li, R.: Cross-modal retrieval with correspondence autoencoder. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 7–16. ACM (2014)

[16] Gu, J., Cai, J., Joty, S.R., Niu, L., Wang, G.: Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7181–7189 (2018)

[17] Hong, R., Liu, D., Mo, X., He, X., Zhang, H.: Learning to compose and reason with language tree structures for visual grounding. IEEE transactions on pattern analysis and machine intelligence (2019)

[18] Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017), to appear

[19] Hou, J., Dai, A., Nießner, M.: 3D-SIC: 3D semantic instance completion for RGB-D scans. arXiv preprint arXiv:1904.12012 (2019)

[20] Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

[21] Hough, P.V.: Machine analysis of bubble chamber pictures. In: Conf. Proc. vol. 590914, pp. 554–558 (1959)

[22] Hu, R., Rohrbach, M., Darrell, T.: Segmentation from natural language expressions. In: European Conference on Computer Vision. pp. 108–124. Springer (2016)

[23] Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4555–4564 (2016)

[24] Huang, Y., Wang, W., Wang, L.: Instance-aware image and sentence matching with selective multimodal LSTM. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2310–2318 (2017)

[25] Huang, Y., Wu, Q., Song, C., Wang, L.: Learning semantic concepts and order for image and sentence matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6163–6171 (2018)

[26] Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2015)

[27] Karpathy, A., Joulin, A., Fei-Fei, L.: Deep fragment embeddings for bidirectional image sentence mapping. In: Advances in neural information processing systems (2014)

[28] Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: ReferItGame: Referring to objects in photographs of natural scenes. In: Proceedings of the 2014

conference on empirical methods in natural language processing (EMNLP). pp. 787–798 (2014)

[29] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[30] Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014)

[31] Kong, C., Lin, D., Bansal, M., Urtasun, R., Fidler, S.: What are you talking about? text-to-image coreference. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3558–3565 (2014)

[32] Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3D instance segmentation via multi-task metric learning. arXiv preprint arXiv:1906.08650 (2019)

[33] Li, R., Li, K., Kuo, Y.C., Shu, M., Qi, X., Shen, X., Jia, J.: Referring image segmentation via recurrent refinement networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)

[34] Li, S., Xiao, T., Li, H., Yang, W., Wang, X.: Identity-aware textual-visual matching with latent co-attention. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1890–1899 (2017)

[35] Liu, C., Lin, Z., Shen, X., Yang, J., Lu, X., Yuille, A.: Recurrent multimodal interaction for referring image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2017)

[36] Liu, D., Zhang, H., Wu, F., Zha, Z.J.: Learning to assemble neural module tree networks for visual grounding. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[37] Liu, X., Wang, Z., Shao, J., Wang, X., Li, H.: Improving referring expression grounding with cross-modal attention-guided erasing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

[38] Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 375–383 (2017)

[39] Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)

[40] Margffoy-Tuay, E., Pérez, J.C., Botero, E., Arbeláez, P.: Dynamic multimodal instance segmentation guided by natural language queries. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)

[41] Mauceri, C., Palmer, M., Heckman, C.: SUN-Spot: An RGB-D dataset with spatial referring expressions. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)

[42] Narita, G., Seno, T., Ishikawa, T., Kaji, Y.: PanopticFusion: Online volumetric semantic mapping at the level of stuff and things. arXiv preprint arXiv:1903.01177 (2019)

[43] Nguyen, A., Do, T.T., Reid, I., Caldwell, D.G., Tsagarakis, N.G.: Object captioning and retrieval with natural language. arXiv preprint arXiv:1803.06152 (2018)

[44] Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)

[45] Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (2014)

[46] Plummer, B.A., Kordas, P., Hadi Kiapour, M., Zheng, S., Piramuthu, R., Lazebnik, S.: Conditional image-text embedding networks. In: Proceedings of the European Conference on Computer Vision (2018)

[47] Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision (2015)

[48] Prabhudesai, M., Tung, H.Y.F., Javed, S.A., Sieb, M., Harley, A.W., Fragkiadaki, K.: Embodied language grounding with implicit 3D visual feature representations. arXiv preprint arXiv:1910.01210 (2019)

[49] Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3D object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[50] Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)

[51] Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems (2017)

[52] Qi, Y., Wu, Q., Anderson, P., Liu, M., Shen, C., Hengel, A.v.d.: REVERIE: Remote embodied visual referring expression in real indoor environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)

[53] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396 (2016)

[54] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems (2015)

[55] Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B.: Grounding of textual phrases in images by reconstruction. In: European Conference on Computer Vision. pp. 817–834 (2016)

[56] Sadhu, A., Chen, K., Nevatia, R.: Zero-shot grounding of objects from natural language queries. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[57] Sharma, S., Suhubdy, D., Michalski, V., Kahou, S.E., Bengio, Y.: ChatPainter: Improving text to image generation using dialogue. arXiv preprint arXiv:1802.08216 (2018)

[58] Song, S., Lichtenberg, S.P., Xiao, J.: SUN RGB-D: A RGB-D scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 567–576 (2015)

[59] Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3156–3164 (2015)

[60] Wang, L., Li, Y., Huang, J., Lazebnik, S.: Learning two-branch neural networks for image-text matching tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)

[61] Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5005–5013 (2016)

[62] Wang, P., Wu, Q., Cao, J., Shen, C., Gao, L., Hengel, A.v.d.: Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

[63] Xiao, F., Sigal, L., Jae Lee, Y.: Weakly-supervised visual grounding of phrases with linguistic structures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)

[64] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)

[65] Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N.: Learning object bounding boxes for 3D instance segmentation on point clouds. arXiv preprint arXiv:1906.01140 (2019)

[66] Yang, S., Li, G., Yu, Y.: Cross-modal relationship inference for grounding referring expressions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

[67] Yang, S., Li, G., Yu, Y.: Dynamic graph attention for referring expression comprehension. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[68] Yang, Z., Gong, B., Wang, L., Huang, W., Yu, D., Luo, J.: A fast and accurate one-stage approach to visual grounding. In: Proceedings of the IEEE International Conference on Computer Vision (2019)

[69] Ye, L., Rochan, M., Liu, Z., Wang, Y.: Cross-modal self-attention network for referring image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

[70] Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: MAttNet: Modular attention network for referring expression comprehension. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)

[71] Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: European Conference on Computer Vision. pp. 69–85. Springer (2016)

[72] Yu, L., Tan, H., Bansal, M., Berg, T.L.: A joint speaker-listener-reinforcer model for referring expressions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)

[73] Zhao, F., Li, J., Zhao, J., Feng, J.: Weakly supervised phrase localization with multi-scale anchored transformer network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)

[74] Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: European conference on computer vision. pp. 391–405. Springer (2014)

Supplementary Material

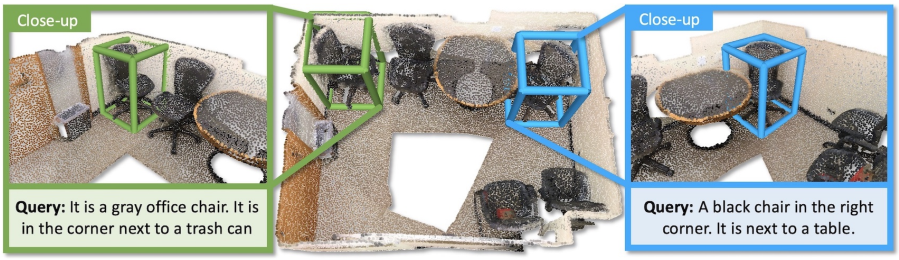

Close-up
Query: It is a gray office chair. It is in the corner next to a trash can
Close-up
Query: A black chair in the right corner. It is next to a table.

Fig. 9: ScanRefer localizes objects in a scene given a language description as input. In many cases, including this example, there are multiple objects from the same category in a single scene which makes the problem challenging and interesting at the same time.

In this supplementary material, we provide addition details on the data collection and statistic of the ScanRefer dataset (Section A); we also provide implementation details of our localization network (Section B), as well as additional quantitative (Section C) and qualitative comparisons (Section D).

A Dataset

A.1 Statistics

We present the distribution of categories of the ScanRefer dataset in Fig. 10. ScanRefer provides a large coverage of furniture (e.g., chair, table, cabinet, bed, etc.) in indoor environments with various sizes, colors, materials, and locations. We use the same category names as in the original ScanNet dataset [9]. In total, we annotate 11,046 objects from 265 categories from ScanNet [9]. Following the ScanNet voxel labeling task, we aggregate these finer-grained categories into 17 coarse categories and group the remaining object types into "Others" for a total of 18 object categories that we use to train the language-based object classifier.

Fig. 11 shows the distribution of finer-grained objects in the category "Others". For each of the 18 coarse categories, Fig. 12 shows the average and maximum number of objects for that category in a scene in which an object of that category appears. For instance, for scenes that contains a bed, the average number of beds is 1.22 and the maximum is 3.

We also provide detailed statistics in our training and validation splits in Tab. 6. To further address the difficulty of our task, we present additional details about the "unique" and "multiple" subsets in Tab. 7. The "unique" subset consists of cases where there is just one unique object of that category (from the 18 ScanNet classes), in the scene. In these cases, the object can be localized

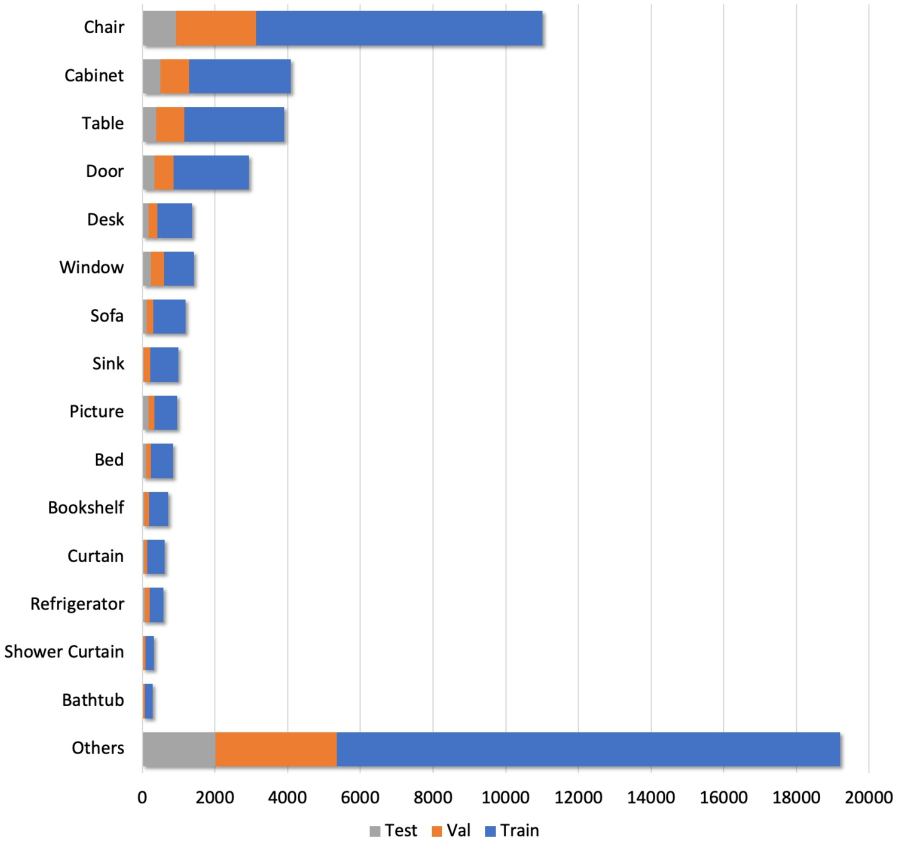


Fig. 10: Distribution of categories of objects in the ScanRefer dataset with annotated language descriptions.

	Train	Val	Test	Total
Number of descriptions	36,665	9,508	5,410	51,583
Number of scenes	562	141	97	800
Number of objects	7,875	2,068	1,103	11,046
Number of objects per scene	14.01	14.67	11.37	14.14
Number of descriptions per scene	65.24	67.43	55.77	65.68
Number of descriptions per object	4.66	4.60	4.90	4.64

Table 6: ScanRefer dataset statistics on Train and Val splits.

(assuming perfect object detection) just by identifying the semantic class of the target object from the description (e.g., localizing the table in the scene Fig. 9). The “multiple” subset refers to cases where there are multiple objects of the

Number of objects per scene	Unique	Multiple	Overall
total	3.00	11.81	14.14
same category as the target object	1.00	4.96	2.98

Table 7: Average number of objects (per scene) for the "Unique" and "Multiple" subsets of the ScanRefer dataset. Assuming ground truth bounding boxes, there are on average 14 different objects for to disambiguate between. For the "Multiple" subset, there are on average 5 objects to disambiguate between even if we could match the semantic class perfectly.

same category as the target object in the scene, thus requiring disambiguation between multiple objects of the same time (e.g., localizing a specific chair in the scene in Fig. 9). As shown in Tab. 7, since there are on average more objects of the same category as the target object in the "multiple" subset than in the "unique", it is more challenging to correctly localize the target object in the "multiple" subset.

A.2 Collection Details

In this section, we provide more details of the data annotation and verification processes of ScanRefer. The data collection took place over one month and involved 1,929 AMT workers. Together, the description collection and verification took around 4,984 man hours in total.

Annotation We deploy our web-based annotation application on Amazon Mechanical Turk (AMT) to collect object descriptions in the reconstructed RGB-D scans, as shown in Fig. 13a. To ensure that the initial descriptions are written in proper English, we restrict the workers to be from the United States, the United Kingdom, Canada, and Australia. The workers are asked to finish a batch of 5 description tasks within a time limit of 2 hours once the assignment is accepted on AMT. To ensure the descriptions are diverse and linguistically rich, we require that each description consists of at least two sentences. Before the annotation task begins, the AMT workers are also presented with the instructions shown in Fig. 13b. We request that the workers provide the following information in the descriptions:

- The appearance of the object such as shape, color, material and so on.
- The location of that object in the scene, e.g., "the chair is in the center of this room".
- The relative position to other objects in the scene, for instance, "this chair is the second one from the left".

Verification After collecting the descriptions from AMT, we do a quick inspection of the descriptions and manually filter and reject obvious bad descriptions

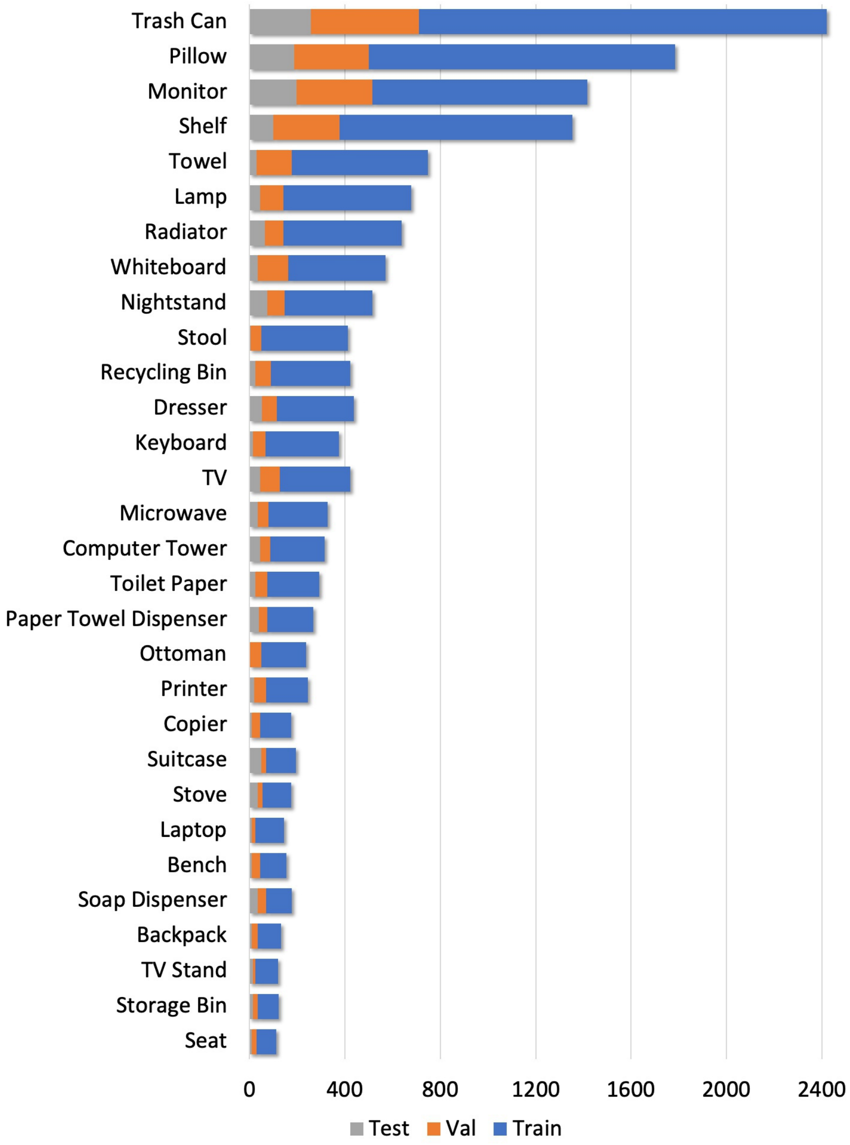


Fig. 11: Distribution of the top 30 categories in the "Others" category of the Train/Val/Test splits of the ScanRefer dataset (sorted in descending order according to the number of objects in the Train split).

before we start the verification process. We then verify the collected object descriptions by recruiting trained students to perform the verification task on our WebGL-based application, as shown in Fig. 14a. To ensure that the descriptions

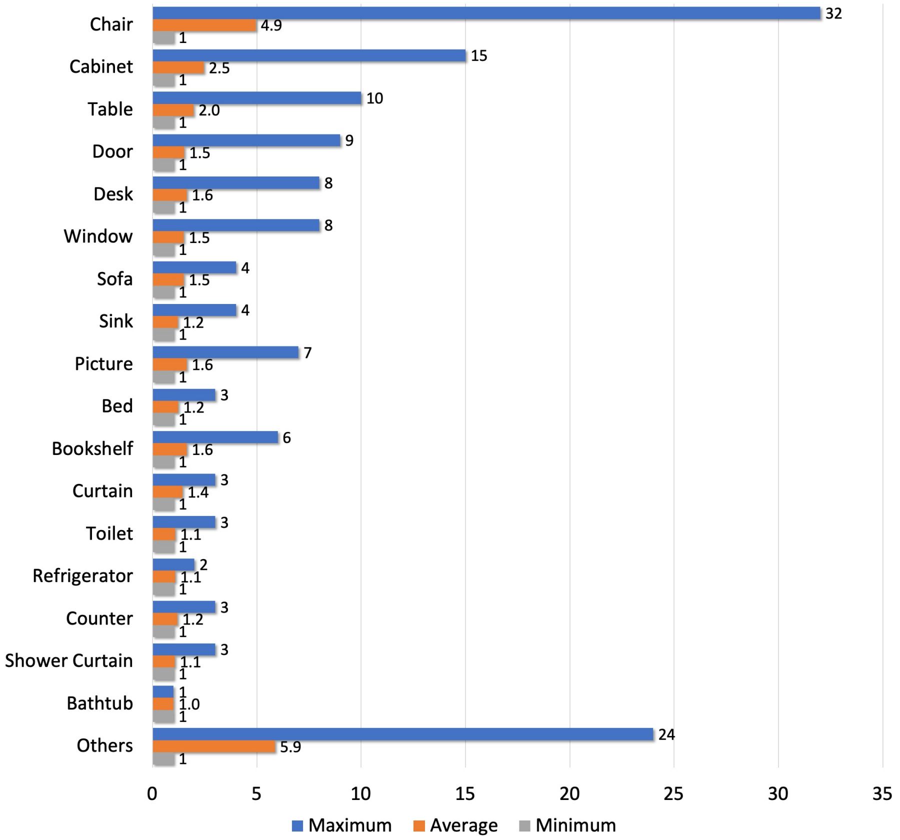


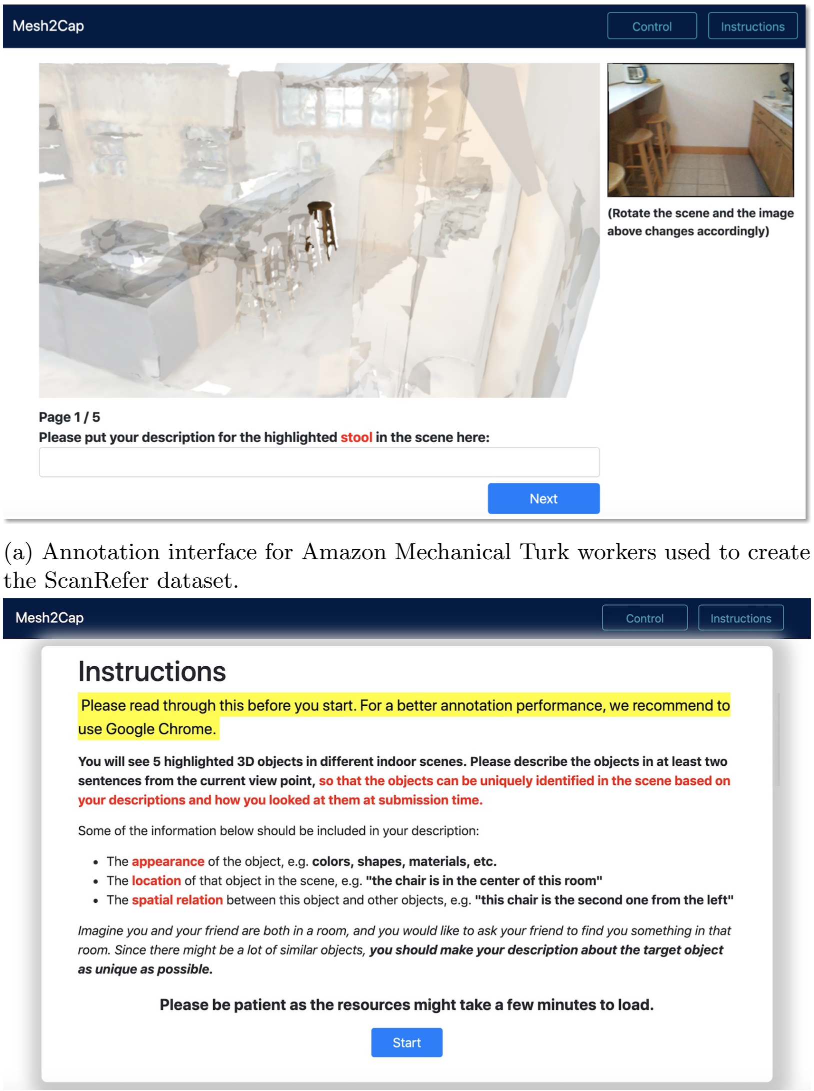
Fig. 12: Average and maximum numbers of objects in each category per scene in the ScanRefer dataset. For each category, we only consider scenes that contains the corresponding objects.

provided are discriminative (e.g., can pick out which one of the chairs is being described), the verifiers are asked to select the objects in the scene that match the descriptions the best. The verifiers are also asked to fix any spelling and wording issues, e.g., “hair” instead of “chair”, and submit the corrected descriptions to our database. To guide the trained verifiers, we provide the verification instructions as shown in Fig. 14b.

B Additional Implementation Details

B.1 Fusion Module

Fig. 15 shows the feature fusion process in our localization pipeline. Concretely, the fusion module first concatenates the point clusters $C = c_i \in \mathcal{R}^{M \times 128}$ and



(Rotate the scene and the image above changes accordingly)

Page 1 / 5

Please put your description for the highlighted stool in the scene here:

(a) Annotation interface for Amazon Mechanical Turk workers used to create the ScanRefer dataset.

Instructions

Please read through this before you start. For a better annotation performance, we recommend to use Google Chrome.

You will see 5 highlighted 3D objects in different indoor scenes. Please describe the objects in at least two sentences from the current view point, so that the objects can be uniquely identified in the scene based on your descriptions and how you looked at them at submission time.

Some of the information below should be included in your description:

- The **appearance** of the object, e.g. **colors, shapes, materials, etc.**
- The **location** of that object in the scene, e.g. **"the chair is in the center of this room"**
- The **spatial relation** between this object and other objects, e.g. **"this chair is the second one from the left"**

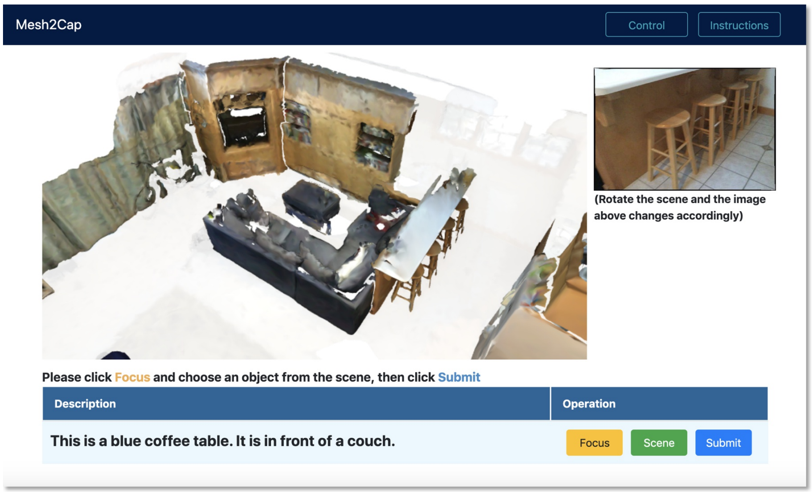
Imagine you and your friend are both in a room, and you would like to ask your friend to find you something in that room. Since there might be a lot of similar objects, ***you should make your description about the target object as unique as possible.***

Please be patient as the resources might take a few minutes to load.

(b) Annotation instructions shown to the Amazon Mechanical Turk workers.

Fig. 13: (a) Our web-based annotation interface: annotators are requested to describe a batch of 5 target objects. The viewpoint can be adjusted by the user while the image on the right is chosen based on the camera view. (b) Screenshot of the instructions for the Amazon Mechanical Turk workers before providing descriptions for objects.

expanded language embedding $E = e' \in \mathcal{R}^{M \times 256}$, then multiply the expanded

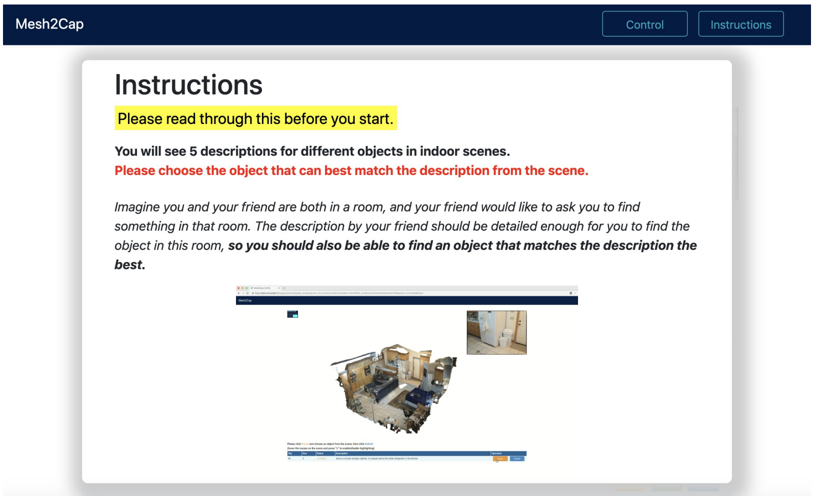


(Rotate the scene and the image above changes accordingly)

Please click Focus and choose an object from the scene, then click Submit

Description	Operation
This is a blue coffee table. It is in front of a couch.	Focus Scene Submit

(a) Verification interface used by trained student verifiers in order to verify each annotation done earlier by the annotation Amazon Mechanical Turk workers.



Instructions

Please read through this before you start.

You will see 5 descriptions for different objects in indoor scenes.
Please choose the object that can best match the description from the scene.

Imagine you and your friend are both in a room, and your friend would like to ask you to find something in that room. The description by your friend should be detailed enough for you to find the object in this room, ***so you should also be able to find an object that matches the description the best.***

(b) Verification instructions shown to the trained student verifiers.

Fig. 14: (a) Our web-based verification interface: verifiers are asked to select objects that match the provided descriptions from the collection step. The ambiguous descriptions, which can be used to match multiple objects in the scene, are excluded from the final dataset. (b) Screenshot of the instructions that the trained verifiers have to go through before starting the verification.

objectness mask $D'_{objn} \in \mathcal{R}^{M\times 384}$ to filter out invalid object proposals. A multi-

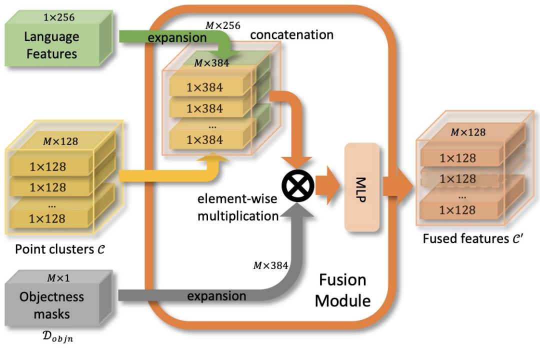


Fig. 15: The fusion module takes as input the aggregated point clusters, the language embeddings, and the predicted objectness masks. It first concatenates the point clusters with the expanded language features as the raw fused features, of which the invalid ones will be masked out by the predicted objectness masks. Finally, a multi-layer perceptron takes in the raw fused features and outputs the final fused multimodal point features.

	cab.	bed	chair	sofa	tabl.	door	wind.	bkshf.	pic.	cntr.	desk	curt.	fridg.	showr.	toil.	sink	bath.	others	mAP
[a]	4.77	85.51	64.42	72.74	30.39	11.17	6.62	17.32	0.35	2.16	35.79	7.80	16.69	16.96	76.74	16.77	69.57	5.68	30.08
[b]	9.93	**88.43**	67.12	69.44	**39.76**	12.20	5.11	20.27	0.02	9.27	41.52	16.10	**30.79**	5.77	77.32	14.93	61.02	7.82	32.05
[c]	7.01	88.01	67.13	73.69	32.87	12.36	9.01	17.61	0.31	9.27	44.78	16.25	20.29	3.55	76.50	12.33	72.24	8.08	31.74
[d]	11.16	87.20	**70.58**	75.17	36.76	11.47	6.72	13.40	1.09	7.08	48.38	11.64	19.96	4.29	85.29	**18.20**	72.83	**10.74**	32.89
[e]	7.22	87.72	67.24	72.42	33.66	11.55	8.80	20.16	0.14	**9.82**	46.07	15.91	22.48	2.67	77.82	13.17	68.14	8.01	31.83
[f]	**12.74**	83.91	69.94	72.17	36.11	**13.38**	8.42	17.52	**1.99**	6.58	**46.65**	**17.65**	24.04	**31.30**	75.99	10.31	61.92	9.78	**33.36**
[g]	10.53	84.00	63.48	**75.27**	30.62	7.78	8.45	18.08	1.18	5.47	39.27	10.14	18.83	8.93	69.99	9.36	**75.59**	7.97	30.27
[h]	11.11	85.63	67.81	71.04	34.96	9.54	6.22	16.37	1.67	6.28	36.07	12.93	17.40	7.46	68.74	11.77	65.69	7.71	29.91
[i]	10.72	86.71	69.86	72.77	32.60	16.33	8.16	19.64	1.14	7.08	42.21	14.31	22.99	6.92	**86.09**	8.06	65.51	8.79	32.22
[j]	9.76	87.93	65.93	72.59	31.60	9.48	9.05	**23.86**	0.37	6.69	42.22	13.86	21.42	16.35	80.41	12.30	57.80	7.40	31.61
[k]	8.92	88.20	70.37	73.93	32.89	10.54	**9.21**	14.05	0.48	6.91	44.74	6.54	17.76	27.64	81.18	12.86	62.40	9.06	32.09

Table 8: Object detection results measured using mean average precision (mAP) at IOU of 0.5 for the 18 difference classes for [a] VoteNet [49], [b] Ours (xyz), [c] Ours (xyz+rgb), [d] Ours (xyz+rgb+normals), [e] Ours (xyz+multiview), [f] Ours (xyz+multiview+normals), [g] Ours (xyz+lobjcls), [h] Ours (xyz+rgb+lobjcls), [i] Ours (xyz+rgb+normals+lobjcls), [j] Ours (xyz+multiview+lobjcls), [k] Ours (xyz+multiview+normals+lobjcls). Training with point normals (compare rows [d,f] to rows [c,e]) and multiview features (compare rows [e,f] to rows [c,d]) clearly leads to better performance. As expected, models with the language-based object classifier (rows [g-k]) does not results in better object detection compared to models without such a module (rows [b-f]).

layer perceptron maps the filtered feature maps into the final fused features $\mathcal{C}' \in \mathcal{R}^{M\times128}$ as the output of the fusion module.

C Additional quantitative analysis

C.1 Object Detection Results

In order to evaluate the 3D object detection, we conduct ablations of our architecture with different point cloud features as well as ablating the inclusion of the language-based object classifier (see Tab. 8). We also compare against the the object detection results of VoteNet [49]. We use the mean average precision (mAP) thresholded by IoU value 0.5 as our evaluation metric and examine the object detection results for different object categories. We exclude structural objects such as "Floor" and "Wall". We group all categories which are not in the ScanNet benchmark categories [9] including "Otherfurnitures", "Otherstructure", and "Otherprop" into the "Others" category in our evaluation. Note that the "Others" category in our evaluation includes additional types of objects, such as "Pillow" and "Keyboard", with respect to those in the "Otherfurniture" category of the ScanNet benchmark.

While our 3D object detector is robust in identifying and separating out instances of large objects that are typically placed away from walls (e.g., bed, chair, sofa, toilet, bathtub), it is not as reliable at identifying instances of flat objects (e.g., picture, window, door) and objects with unclear instance boundaries (e.g., cabinet, shelving) and smaller objects (e.g., sink, others). Overall, our best 3D object detector only achieves a mAP of 33%, suggesting that improving 3D object detection, especially better instance detection for the "other" category, is a key challenge in our task of localizing objects in 3D using natural language.

As shown in Tab. 8, including point normals as extra point features (rows [d,f]) in training increases the detection results when compared to the models trained without the normals (rows [c,e]). Also, training with extracted high-level color features from the multi-view images (rows [e,f]) also produces better detection results compared with the results from models trained with just the raw RGB values (rows [c,d]). Note that networks equipped with the language-based object classifier (rows [g-k]) fail to produce better detection results compared to the ones without the extra language classifier module (rows [b-f]). This behavior is expected as the description provides additional information which helps to differentiate between objects of the same category; but it has no information for helping with object detection.

C.2 Training and Evaluation Variance

Since there is a random sampling of 40,000 points from the original point cloud in the VoteNet [49] detection backbone, we conduct experiments to measure the training and evaluation variance across multiple runs. As shown in Tab. 9 and Tab. 10, due to random sampling, there is a stddev of 0.30 across training runs and a stddev of 0.37 across evaluation runs. For more reliable results, we average the results of 5 evaluation runs with different random seeds when using VoteNet.

random seed	unique Acc@0.5	multiple Acc@0.5	overall Acc@0.5
2	46.83	20.57	25.66
4	47.96	19.45	24.98
8	45.96	20.05	25.07
standard deviation	0.82	0.46	0.30
mean	46.92	20.02	25.23

Table 9: Variance between training runs. We train our model (xyz+rgb+lobjcls) with three different random seeds (2, 4, 8) and evaluate the trained model using a fixed random seed 42. We have a training stddev of 0.30.

random seed	unique Acc@0.5	multiple Acc@0.5	overall Acc@0.5
42	48.89	22.24	27.40
2	49.28	22.05	27.34
4	48.68	21.56	26.82
8	48.29	21.99	27.09
16	50.35	21.42	27.03
32	49.55	21.75	27.14
64	49.61	22.25	27.56
128	49.28	21.57	26.95
256	49.88	21.98	27.39
512	47.29	21.99	28.12
standard deviation	0.87	0.29	0.37
mean	49.11	21.88	27.28

Table 10: Variance between evaluation runs due to the random sampling of points in the VoteNet [49]. We train our model (xyz+multiview+normal+lobjcls) with the a fixed random seed of 42 and evaluate the trained model using 10 different random seeds as shown in the first column. We have a evaluation stddev of 0.37.

C.3 Additional Ablation Study

In Tab. 11, we examine what happens when we feed different language inputs into our pipeline.

Does our method really learn from the full descriptions? To evaluate the impact of information from the full descriptions versus just the identification of the type of object to locate, we compare using the full description as input versus using the semantic label or the object name as the input. For example, for a target object "trash can" with the description *This is a short trash can. It is in front of a taller trash can.*, we input "trash can" as the object name and "others" as the semantic label (see Sec. A.1 for list of semantic classes). The results in Tab. 11 show that using the full descriptions improves the localization performance compared to using just the semantic labels as input. Comparing the

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Ours (semantic labels)	61.60	39.04	28.26	18.98	34.72	21.88
Ours (object names)	70.53	44.69	32.34	20.33	39.75	25.05
Ours (first sentences)	73.52	46.60	33.71	21.20	41.44	26.12
Ours (whole descriptions)	**76.33**	**53.51**	**32.73**	**21.11**	**41.19**	**27.40**

Table 11: Ablation study with different input lengths. We measure the percentages of predictions whose IoU with the ground truth boxes are greater than 0.25 and 0.5. Unique means that there is only a single object of its class in the scene. Obviously, the richer information the descriptions contain, the better our localization pipeline performs.

performance of using semantic labels and object names, we see that inputting the semantic labels helps with the performance in the “unique” scenarios where there is only one object from a certain category, but suffers in the “multiple” scenarios where more information is needed to distinguish between objects that are grouped into the same broad category (e.g., “trash can” and “laptop” would both be categorized as “other”, and “armchair” would provide more information than just the coarse semantic label “chair”).

Are the first sentences enough for the task? Since we deliberately collect at least two sentences as descriptions for the objects to ensure the richness of information, we also conduct experiments to show that the full description (with potentially multiple sentences) result in better performance than using only the first sentences. As Tab. 11 shows, the model trained on longer descriptions performs better than the one trained just on the first sentences.

D Additional Qualitative Analysis

We present additional examples of localization results by our method and the baselines for further qualitative analysis.

Qualitative results comparing VoteNet [49]+GRU and VoteNetBest with out method We show more qualitative results in Fig. 16 to display the difference in performance between these three methods. As shown in the first column in Fig. 16, using a pretrained VoteNet [49] detection backbone provides reasonable bounding box around objects, but still performs slightly worse than our method where we train the detection backbone and localization module in an end-to-end fashion (see the third column ”ours”).

More qualitative examples comparing OracleRefer and One-stage (with 2D to 3D backprojection) with our method To illustrate the difference in performance between the methods, we provide more qualitative results. We split the localization results into “unique” (Fig. 17) and “multiple” (Fig. 18 & Fig. 19) subsets. As shown in Fig. 17, for the “unique” subset, our method is able to identify and localize the object. In contrast, the 2D method (One-Stage), is able to

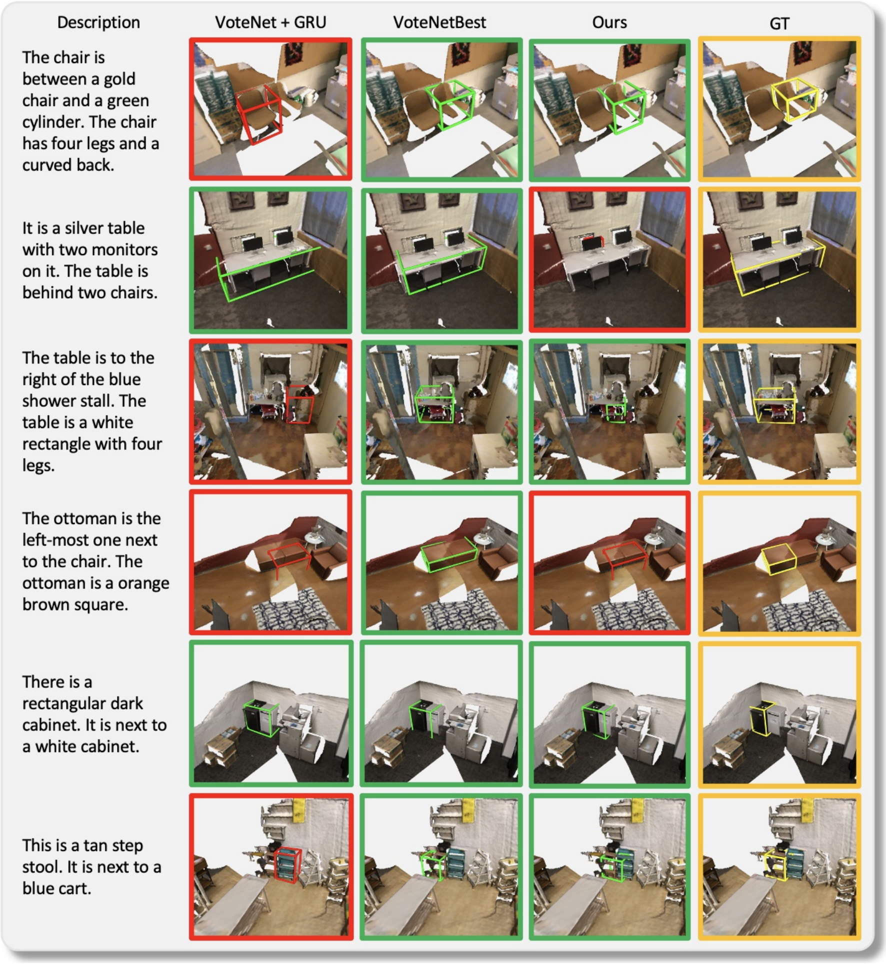
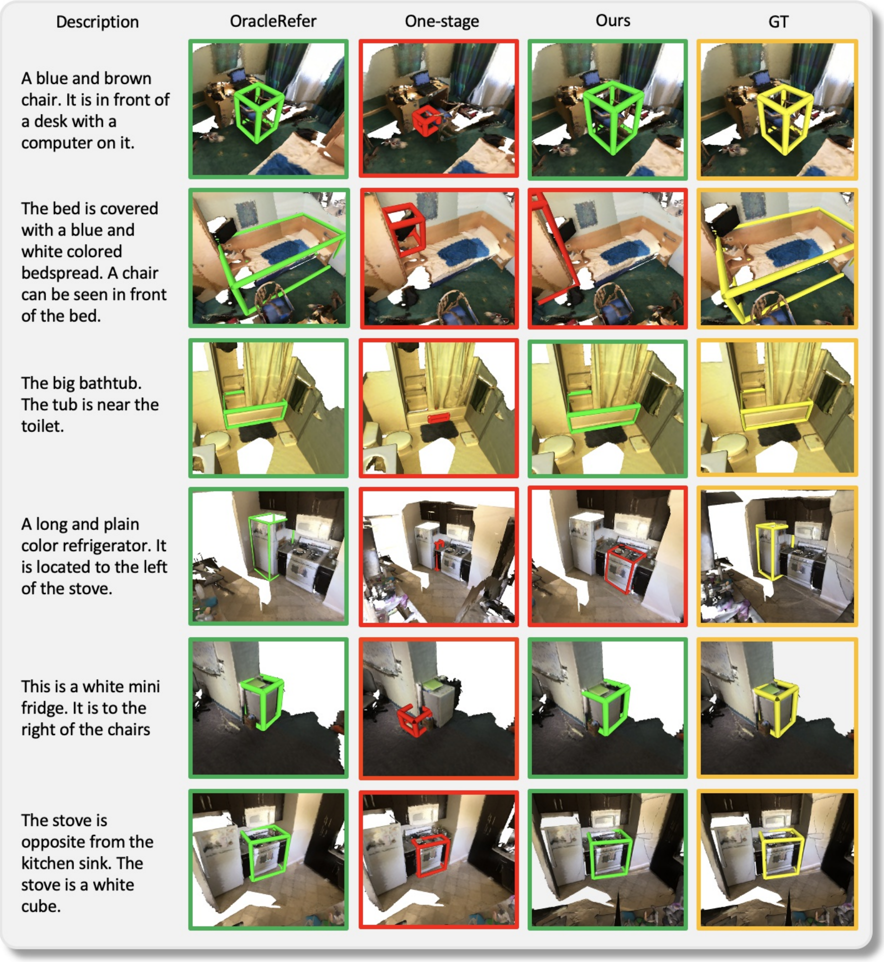

Description
VoteNet + GRU
VoteNetBest
Ours
GT
The chair is between a gold chair and a green cylinder. The chair has four legs and a curved back.
It is a silver table with two monitors on it. The table is behind two chairs.
The table is to the right of the blue shower stall. The table is a white rectangle with four legs.
The ottoman is the left-most one next to the chair. The ottoman is a orange brown square.
There is a rectangular dark cabinet. It is next to a white cabinet.
This is a tan step stool. It is next to a blue cart.

Fig. 16: Additional qualitative analysis comparing our method with VoteNet [49]+GRU and VoteNetBest.

identify the rough location of the object, but the backprojected 3D bounding box does not match the ground truth very well. For the "multiple" subset, there are challenging cases where our method fails to localize the target object. Fig. 18 and 19 show that our method is able to localize objects correctly (Fig. 18 rows 1,5, Fig. 19 rows 1-3,5-6) even when there are other objects of the same category in the scene. Our method is sometimes limited by the accuracy of the object detector, which tends to produce inaccurate bounding boxes for small objects such as pictures (Fig. 18 row 2). This indicates that the object detection can still be improved. Our method also has trouble disambiguating between objects based on spatial relations (Fig. 18 rows 3-4,6). For instance, for comparative



A blue and brown chair. It is in front of a desk with a computer on it.

The bed is covered with a blue and white colored bedspread. A chair can be seen in front of the bed.

The big bathtub. The tub is near the toilet.

A long and plain color refrigerator. It is located to the left of the stove.

This is a white mini fridge. It is to the right of the chairs

The stove is opposite from the kitchen sink. The stove is a white cube.

Fig. 17: Additional qualitative analysis in the "unique" scenarios where there is only one object from a certain category. Our method is capable of localizing the target object in a 3D indoor scene with the help of the free-form description.

phrases (e.g., "leftmost" or "rightmost") or counting (e.g., "the second one from the left"), the model fails to pick out the correct object (Fig. 18 rows 4).

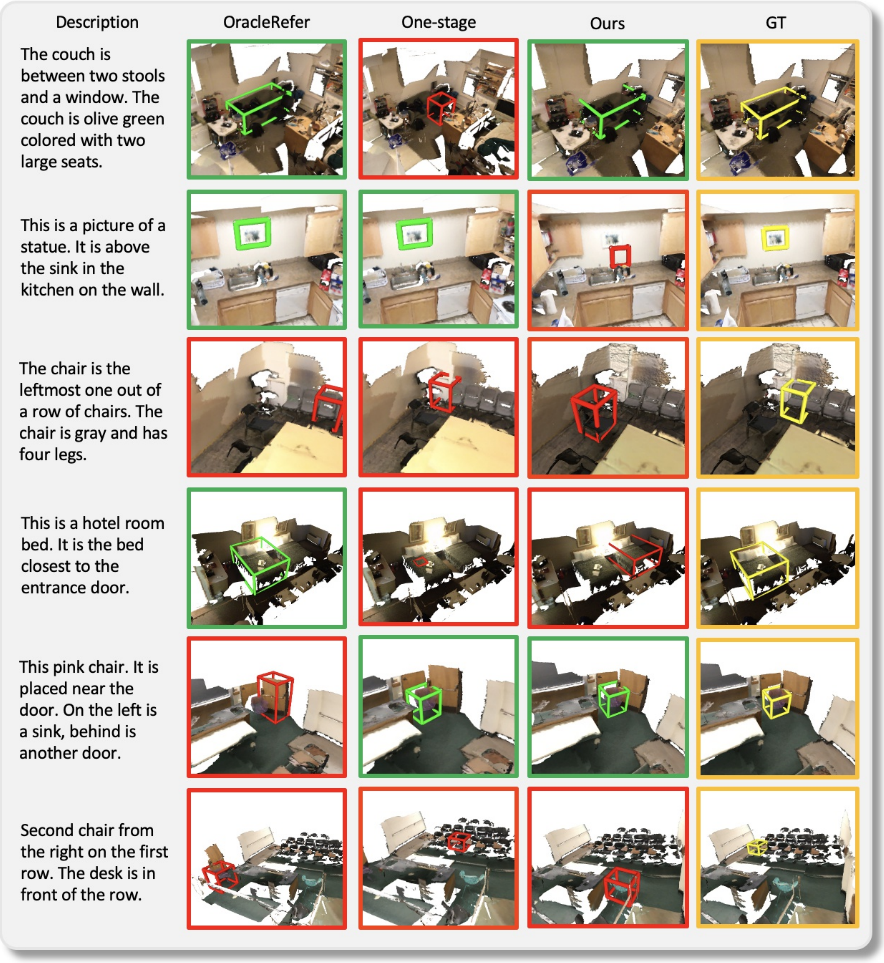

Description
OracleRefer
One-stage
Ours
GT
The couch is between two stools and a window. The couch is olive green colored with two large seats.
This is a picture of a statue. It is above the sink in the kitchen on the wall.
The chair is the leftmost one out of a row of chairs. The chair is gray and has four legs.
This is a hotel room bed. It is the bed closest to the entrance door.
This pink chair. It is placed near the door. On the left is a sink, behind is another door.
Second chair from the right on the first row. The desk is in front of the row.

Fig. 18: Additional qualitative analysis for the "multiple" subset where there are multiple objects with the same category as the target objects. While our methods can correctly localize the target object in some cases (rows 1,5), it often fails due to the limited accuracy of the object detector (row 2) or difficulty disambiguating between multiple instances (rows 3,4,6).

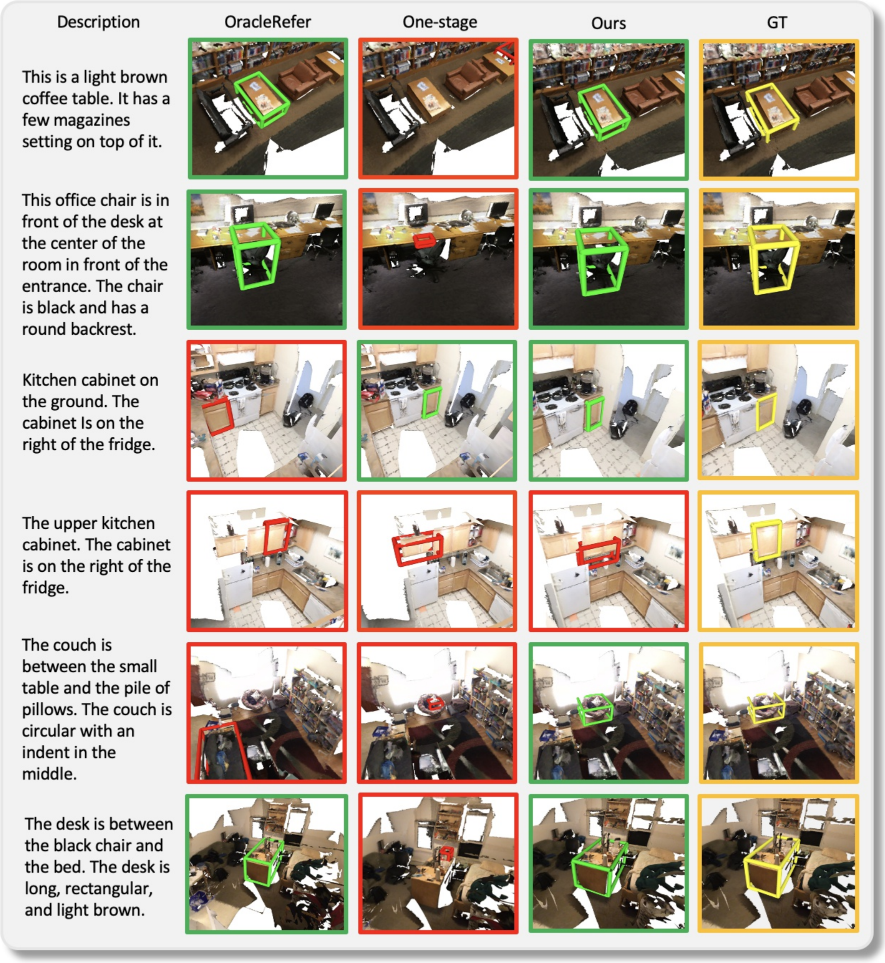

Description
OracleRefer
One-stage
Ours
GT
This is a light brown coffee table. It has a few magazines setting on top of it.
This office chair is in front of the desk at the center of the room in front of the entrance. The chair is black and has a round backrest.
Kitchen cabinet on the ground. The cabinet Is on the right of the fridge.
The upper kitchen cabinet. The cabinet is on the right of the fridge.
The couch is between the small table and the pile of pillows. The couch is circular with an indent in the middle.
The desk is between the black chair and the bed. The desk is long, rectangular, and light brown.

Fig. 19: Additional qualitative analysis for the "multiple" subset where there are multiple objects with the same category as the target objects. While our methods can correctly localize the target object in some cases (rows 1-3,5-6), it can fail due to the limited accuracy of the object detector and difficulty handling spatial relations (rows 4).

SPRINGER NATURE LICENSE TERMS AND CONDITIONS

May 11, 2023

This Agreement between Mr. Zhenyu Chen ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5545810431003
License date	May 11, 2023
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	ScanRefer: 3D Object Localization in RGB-D Scans Using Natural Language
Licensed Content Author	Dave Zhenyu Chen, Angel X. Chang, Matthias Nießner
Licensed Content Date	Jan 1, 2020
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	30 - 99
Author of this Springer Nature content	yes
Title	Grounding Natural Language to 3D Scenes
Institution name	Technical University of Munich
Expected presentation date	Dec 2023
Requestor Location	Mr. Zhenyu Chen Euckenstr. 27 Munich, Bayern 81369 Germany Attn: Mr. Zhenyu Chen
Billing Type	Invoice
Billing Address	Mr. Zhenyu Chen Euckenstr. 27 Munich, Germany 81369 Attn: Mr. Zhenyu Chen
Total	**0.00 EUR**

Terms and Conditions

Springer Nature Customer Service Centre GmbH Terms and Conditions

The following terms and conditions ("Terms and Conditions") together with the terms specified in your [RightsLink] constitute the License ("License") between you as Licensee and Springer Nature Customer Service Centre GmbH as Licensor. By clicking 'accept' and completing the transaction for your use of the material ("Licensed Material"), you confirm your acceptance of and obligation to be bound by these Terms and Conditions.

1. Grant and Scope of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, non-sublicensable, revocable, world-wide License to reproduce, distribute, communicate to the public, make available, broadcast, electronically transmit or create derivative works using the Licensed Material for the purpose(s) specified in your RightsLink Licence Details

only. Licenses are granted for the specific use requested in the order and for no other use, subject to these Terms and Conditions. You acknowledge and agree that the rights granted to you under this License do not include the right to modify, edit, translate, include in collective works, or create derivative works of the Licensed Material in whole or in part unless expressly stated in your RightsLink Licence Details. You may use the Licensed Material only as permitted under this Agreement and will not reproduce, distribute, display, perform, or otherwise use or exploit any Licensed Material in any way, in whole or in part, except as expressly permitted by this License.

1. 2. You may only use the Licensed Content in the manner and to the extent permitted by these Terms and Conditions, by your RightsLink Licence Details and by any applicable laws.

1. 3. A separate license may be required for any additional use of the Licensed Material, e.g. where a license has been purchased for print use only, separate permission must be obtained for electronic re-use. Similarly, a License is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the License.

1. 4. Any content within the Licensed Material that is owned by third parties is expressly excluded from the License.

1. 5. Rights for additional reuses such as custom editions, computer/mobile applications, film or TV reuses and/or any other derivative rights requests require additional permission and may be subject to an additional fee. Please apply to journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

2. Reservation of Rights

Licensor reserves all rights not expressly granted to you under this License. You acknowledge and agree that nothing in this License limits or restricts Licensor's rights in or use of the Licensed Material in any way. Neither this License, nor any act, omission, or statement by Licensor or you, conveys any ownership right to you in any Licensed Material, or to any element or portion thereof. As between Licensor and you, Licensor owns and retains all right, title, and interest in and to the Licensed Material subject to the license granted in Section 1.1. Your permission to use the Licensed Material is expressly conditioned on you not impairing Licensor's or the applicable copyright owner's rights in the Licensed Material in any way.

3. Restrictions on use

3. 1. Minor editing privileges are allowed for adaptations for stylistic purposes or formatting purposes provided such alterations do not alter the original meaning or intention of the Licensed Material and the new figure(s) are still accurate and representative of the Licensed Material. Any other changes including but not limited to, cropping, adapting, and/or omitting material that affect the meaning, intention or moral rights of the author(s) are strictly prohibited.

3. 2. You must not use any Licensed Material as part of any design or trademark.

3. 3. Licensed Material may be used in Open Access Publications (OAP), but any such reuse must include a clear acknowledgment of this permission visible at the same time as the figures/tables/illustration or abstract and which must indicate that the Licensed Material is not part of the governing OA license but has been reproduced with permission. This may be indicated according to any standard referencing system but must include at a minimum 'Book/Journal title, Author, Journal Name (if applicable), Volume (if applicable), Publisher, Year, reproduced with permission from SNCSC'.

4. STM Permission Guidelines

4. 1. An alternative scope of license may apply to signatories of the STM Permissions Guidelines ("STM PG") as amended from time to time and made available at https://www.stm-assoc.org/intellectual-property/permissions/permissions-guidelines/.

4. 2. For content reuse requests that qualify for permission under the STM PG, and which may be updated from time to time, the STM PG supersede the terms and conditions contained in this License.

4. 3. If a License has been granted under the STM PG, but the STM PG no longer apply at the time of publication, further permission must be sought from the Rightsholder. Contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

5. Duration of License

5. 1. Unless otherwise indicated on your License, a License is valid from the date of purchase ("License Date") until the end of the relevant period in the below table:

Reuse in a medical communications project	Reuse up to distribution or time period indicated in License
Reuse in a dissertation/thesis	Lifetime of thesis
Reuse in a journal/magazine	Lifetime of journal/magazine
Reuse in a book/textbook	Lifetime of edition
Reuse on a website	1 year unless otherwise specified in the License
Reuse in a presentation/slide kit/poster	Lifetime of presentation/slide kit/poster. Note: publication whether electronic or in print of presentation/slide kit/poster may require further permission.
Reuse in conference proceedings	Lifetime of conference proceedings
Reuse in an annual report	Lifetime of annual report
Reuse in training/CME materials	Reuse up to distribution or time period indicated in License
Reuse in newsmedia	Lifetime of newsmedia
Reuse in coursepack/classroom materials	Reuse up to distribution and/or time period indicated in license

6. Acknowledgement

6. 1. The Licensor's permission must be acknowledged next to the Licensed Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract and must be hyperlinked to the journal/book's homepage.

6. 2. Acknowledgement may be provided according to any standard referencing system and at a minimum should include "Author, Article/Book Title, Journal name/Book imprint, volume, page number, year, Springer Nature".

7. Reuse in a dissertation or thesis

7. 1. Where 'reuse in a dissertation/thesis' has been selected, the following terms apply: Print rights of the Version of Record are provided for; electronic rights for use only on institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/) and only up to what is required by the awarding institution.

7. 2. For theses published under an ISBN or ISSN, separate permission is required. Please contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

7. 3. Authors must properly cite the published manuscript in their thesis according to current citation standards and include the following acknowledgement: '*Reproduced with permission from Springer Nature*'.

8. License Fee

You must pay the fee set forth in the License Agreement (the "License Fees"). All amounts payable by you under this License are exclusive of any sales, use, withholding, value added or similar taxes, government fees or levies or other assessments. Collection and/or remittance of such taxes to the relevant tax authority shall be the responsibility of the party who has the legal obligation to do so.

9. Warranty

9. 1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. **You are solely responsible for ensuring that the material you wish to license is original to the Licensor and does not carry the copyright of another entity or third party (as credited in the published version).** If the credit line on any part of the Licensed Material indicates that it was reprinted or adapted with permission from another source, then you should seek additional permission from that source to reuse the material.

9. 2. EXCEPT FOR THE EXPRESS WARRANTY STATED HEREIN AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR PROVIDES THE LICENSED MATERIAL "AS IS" AND MAKES NO OTHER REPRESENTATION OR WARRANTY. LICENSOR EXPRESSLY DISCLAIMS ANY LIABILITY FOR ANY CLAIM ARISING FROM OR OUT OF THE CONTENT, INCLUDING BUT NOT LIMITED TO ANY ERRORS, INACCURACIES, OMISSIONS, OR DEFECTS CONTAINED THEREIN, AND ANY IMPLIED OR EXPRESS WARRANTY AS TO MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR EXEMPLARY DAMAGES, HOWEVER CAUSED,

ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE LICENSED MATERIAL REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION APPLIES NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

10. Termination and Cancellation

10. 1. The License and all rights granted hereunder will continue until the end of the applicable period shown in Clause 5.1 above. Thereafter, this license will be terminated and all rights granted hereunder will cease.

10. 2. Licensor reserves the right to terminate the License in the event that payment is not received in full or if you breach the terms of this License.

11. General

11. 1. The License and the rights and obligations of the parties hereto shall be construed, interpreted and determined in accordance with the laws of the Federal Republic of Germany without reference to the stipulations of the CISG (United Nations Convention on Contracts for the International Sale of Goods) or to Germany´s choice-of-law principle.

11. 2. The parties acknowledge and agree that any controversies and disputes arising out of this License shall be decided exclusively by the courts of or having jurisdiction for Heidelberg, Germany, as far as legally permissible.

11. 3. This License is solely for Licensor's and Licensee's benefit. It is not for the benefit of any other person or entity.

Questions? For questions on Copyright Clearance Center accounts or website issues please contact springernaturesupport@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777. For questions on Springer Nature licensing please visit https://www.springernature.com/gp/partners/rights-permissions-third-party-distribution

Other Conditions:

Version 1.4 - Dec 2022

Questions? E-mail us at customercare@copyright.com.

Scan2Cap: Context-aware Dense Captioning in RGB-D Scans

Dave Zhenyu Chen[1] Ali Gholami[2] Matthias Nießner[1] Angel X. Chang[2]
[1]Technical University of Munich [2]Simon Fraser University
https://daveredrum.github.io/Scan2Cap/

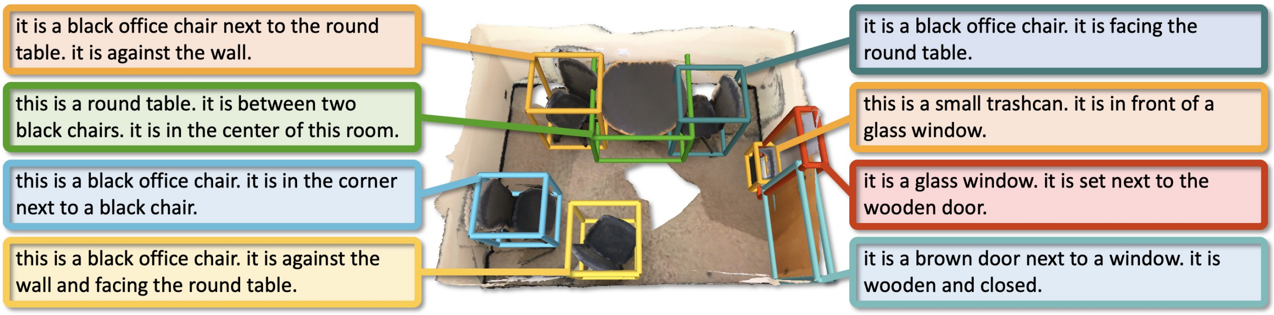

it is a black office chair next to the round table. it is against the wall.
this is a round table. it is between two black chairs. it is in the center of this room.
this is a black office chair. it is in the corner next to a black chair.
this is a black office chair. it is against the wall and facing the round table.
it is a black office chair. it is facing the round table.
this is a small trashcan. it is in front of a glass window.
it is a glass window. it is set next to the wooden door.
it is a brown door next to a window. it is wooden and closed.

Figure 1: We introduce the task of dense captioning in RGB-D scans with a model that can densely localize objects in a 3D scene and describe them using natural language in a single forward pass.

Abstract

*We introduce the task of dense captioning in 3D scans from commodity RGB-D sensors. As input, we assume a point cloud of a 3D scene; the expected output is the bounding boxes along with the descriptions for the underlying objects. To address the 3D object detection and description problems, we propose Scan2Cap, an end-to-end trained method, to detect objects in the input scene and describe them in natural language. We use an attention mechanism that generates descriptive tokens while referring to the related components in the local context. To reflect object relations (i.e. relative spatial relations) in the generated captions, we use a message passing graph module to facilitate learning object relation features. Our method can effectively localize and describe 3D objects in scenes from the ScanRefer dataset, outperforming 2D baseline methods by a significant margin (**27.61% CiDEr@0.5IoU** improvement).*

1. Introduction

The intersection of visual scene understanding [45, 20] and natural language processing [49, 13] is a rich and active area of research. Specifically, there has been a lot of work on image captioning [51, 27, 52, 33, 2] and the related task of dense captioning [27, 26, 53, 56, 28, 31]. In dense captioning, individual objects are localized in an image and each object is described using natural language. So far, dense captioning work has operated purely on 2D visual data, most commonly single-view images that are limited by the field of view. Images are inherently viewpoint specific and scale agnostic, and fail to capture the physical extent of 3D objects (i.e. the actual size of the objects) and their locations in the environment.

In this work, we introduce the new task of dense captioning in 3D scenes. We aim to jointly localize and describe each object in a 3D scene. We show that leveraging the 3D information of an object such as actual object size or object location results in more accurate descriptions.

Apart from the 2D constraints in images, even seminal work on dense captioning suffers from *aperture* issues [56]. Object relations are often neglected while describing scene objects, which makes the task more challenging. We address this problem with a graph-based attentive captioning architecture that jointly learns object features and object relation features on the instance level, and generates descriptive tokens. Specifically, our proposed method (referred to as Scan2Cap) consists of two critical components: 1) *Relational Graph* facilitates learning the object features and object relation features using a message passing neural network; 2) *Context-aware Attention Captioning* generates the descriptive tokens while attending to the object and object relation features. In summary, our contribution is fourfold:

- We introduce the 3D dense captioning task to densely detect and describe 3D objects in RGB-D scans.

- We propose a novel message passing graph module that facilitates learning of the 3D object features and 3D object relation features.
- We propose an end-to-end trained method that can take 3D object features and 3D object relation features into account when describing the 3D object in a single forward pass.
- We show that our method outperforms 2D-3D back-projected results of 2D captioning baselines by a significant margin (**27.61%**).

2. Related work

2.1. 3D Object Detection

There are many methods for 3D object detection on 3D RGB-D datasets [48, 24, 12, 5]. Methods utilizing 3D volumetric grids have achieved impressive performance [21, 22, 30, 36, 15]. At the same time, methods operating on point clouds serve as an alternative and also achieve impressive results. For instance, Qi et al. [41] use a Hough voting scheme to aggregate points and generate object proposals while using a PointNet++ [43] backbone. Following this work, Qi et al. [42] recently proposed a pipeline to jointly perform voting in both point clouds and associated images. Our method builds on these works as we utilize the same backbone for processing the input geometry; however, we back-project multi-view image features to point clouds to leverage the original RGB input, since appearance is critical for accurately describing the target objects in the scene.

2.2. Image Captioning

Image captioning has attracted a great deal of interest [51, 52, 14, 27, 33, 2, 25, 46]. Attention based captioning over grid regions [52, 33] and over detected objects [2, 34] allows focusing on specific image regions while captioning. One recent trend is the attempt to capture relationships between objects using attention and graph neural networks [16, 55, 54] or transformers [10]. We build on these ideas to propose a 3D captioning network with graphs that capture object relations in 3D.

The dense captioning task introduced by Johnson et al. [26] is closely related to our task. This task is a variant of image captioning where captions are generated for all detected objects. While achieving impressive results, this method does not consider the context outside of the salient image regions. To tackle this issue, Yang et al. [53] include the global image feature as context to the captioning input. Kim et al. [28] explicitly model the relations between detected regions in the image. Due to the limited view of a single image, prior work on 2D images could not capture the large context available in 3D environments. In contrast, we focus on decomposing the input 3D scene and capturing the appearance and spatial information of the objects in the 3D environment.

2.3. 3D Vision and Language

While the joint field of vision and language has received much attention in the image domain, in tasks such as image captioning [51, 52, 14, 27, 33, 2, 25, 46], dense captioning [26, 53, 28], text-to-image generation [44, 47, 18], visual grounding [23, 35, 57], vision and language in 3D is still not well-explored. Chen et al. [8] introduces a dataset which consists of descriptions for ShapeNet [6] objects, enabling text-to-shape generation and shape captioning. On the scene level, Chen et al. [7] propose a dataset for localizing object in ScanNet [12] scenes using natural language expressions. Concurrently, Achlioptas et al. [1] propose another dataset for distinguishing fine-grained objects in ScanNet scenes using natural language queries. This work enables research on connecting natural language to 3D environments, and inspires our work to densely localize and describe 3D objects with respect to the scene context.

3. Task

We introduce the task of dense captioning in 3D scenes. The input for this task is a point cloud of a scene, consisting of the object geometries as well as several additional point features such as RGB values and normal vectors. The expected output is the object bounding boxes for the underlying instances in the scene and their corresponding natural language descriptions.

4. Method

We propose an end-to-end architecture on the input point clouds to address the 3D dense description generation task. Our architecture consists of the following main components: 1) detection backbone; 2) relational graph; 3) context-aware attention captioning. As Fig. 2 shows, our network takes a point cloud as input, and generates a set of 3D object proposals using the detection module. A relational graph module then enhances object features using contextual cues and provides object relation features. Finally, a context-aware attention module generates descriptions from the enhanced object and relation features.

4.1. Data Representation

As input to the detection module, we assume a point cloud $\mathcal{P}$ of a scan from ScanNet consisting of the geometry coordinates and additional point features capturing the visual appearance and the height from the ground. To obtain the extended visual point features, we follow Chen et al. [7] and adapt the feature projection scheme of Dai and Nießner [11] to back-project multi-view image features to the point

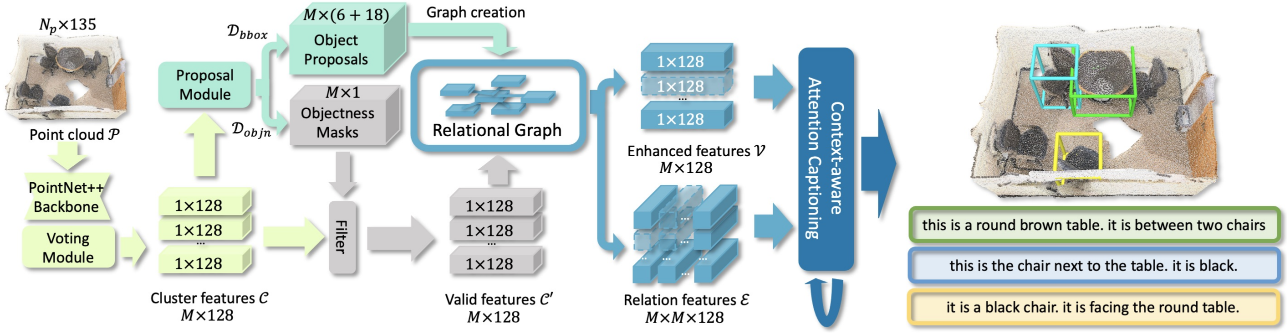

N_p×135
Point cloud $\mathcal{P}$
PointNet++ Backbone
Voting Module
Proposal Module
$\mathcal{D}_{bbox}$
$\mathcal{D}_{objn}$
M×(6 + 18)
Object Proposals
M×1
Objectness Masks
1×128
1×128
1×128
Cluster features $\mathcal{C}$
M×128
Filter
Graph creation
Relational Graph
1×128
1×128
1×128
Valid features $\mathcal{C}'$
M×128
1×128
1×128
1×128
Enhanced features $\mathcal{V}$
M×128
Relation features $\mathcal{E}$
M×M×128
Context-aware Attention Captioning
this is a round brown table. it is between two chairs
this is the chair next to the table. it is black.
it is a black chair. it is facing the round table.

Figure 2: Scan2Cap takes as input a point cloud to generate the cluster features $\mathcal{C}$ for the proposal module, using a backbone following PointNet++ [43] and a voting module similar to Qi et al. [41]. The proposal module predicts the object proposals $\mathcal{D}_{\text{bbox}}$ as well as the objectness masks $\mathcal{D}_{\text{objn}}$, which are later used for filtering the cluster features as the valid features $\mathcal{C}^{'}$. A graph is then constructed using the object proposals and the valid cluster features. The relational graph module takes in the graph and outputs the enhanced object features $\mathcal{V}$ and the relation features $\mathcal{C}^{'}$. As the last step, the context-aware attention captioning module, inspired by Anderson et al. [2], generates descriptive tokens for each object proposal using the enhanced features and the relation features.

cloud as additional features. The image features are extracted using a pre-trained ENet [38]. Following Qi et al. [41], we also append the height of the point from the ground to the new point features. As a result, we represent the final point cloud data as $\mathcal{P} = \{(p_i, f_i)\} \in \mathcal{R}^{N_P \times 135}$, where $p_i \in \mathcal{R}^3, i = 1, ..., N_P$ are the coordinates and $f_i \in \mathcal{R}^{132}$ are the additional features.

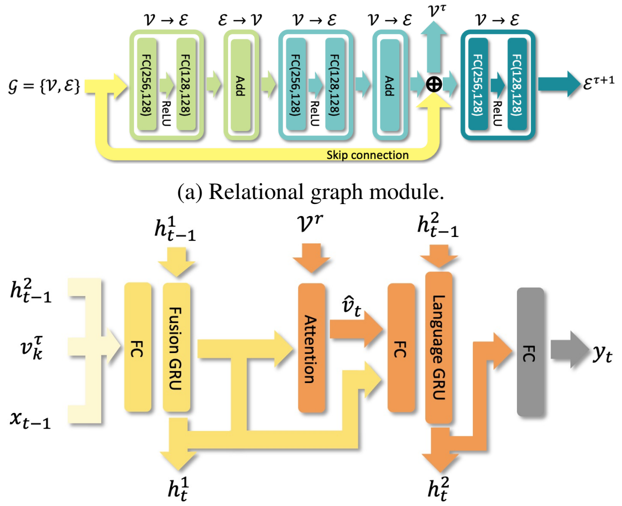
4.2. Detection Backbone

As the first step in our network, we detect all probable objects in the given point cloud with the back-projected multi-view image features discussed in 4.1. To construct our detection module, we adapt the PointNet++ [43] backbone and the voting module in VoteNet [41] to aggregate all object candidates to individual clusters. The output from the voting module is a set of point clusters $\mathcal{C} \in \mathcal{R}^{M \times 128}$ representing all object proposals with enriched point features, where M is the upper bound of the number of proposals. Next, the proposal module takes in the point clusters to predict the objectness mask $\mathcal{D}_{\text{objn}} \in \mathcal{R}^{M \times 1}$ and the axis-aligned bounding boxes $\mathcal{D}_{\text{bbox}} \in \mathcal{R}^{M \times (6+18)}$ for all M proposals, where each $\mathcal{D}^i_{\text{bbox}} = (c_x, c_y, c_z, r_x, r_y, r_z, l)$ consists of the box center c, the box lengths r and a vector $l \in \mathcal{R}^{18}$ representing the semantic predictions.

4.3. Relational Graph

Describing the object in the scene often involves its appearance and spatial location with respect to nearby objects. Therefore, we propose a relational graph module equipped with a message passing network to enhance the object features and extract the object relation features. We create a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where we treat the object proposals as nodes in the graph and relationship between objects as edges. For the edges, we consider only the nearest K objects surrounding each object. We use standard neural message passing [17] where the message passing at graph step τ is defined as follows:

$$\mathcal{V} \rightarrow \mathcal{E} : g^{\tau+1}_{i,j} = f^{\tau}([g^{\tau}_i, g^{\tau}_j - g^{\tau}_i]) \tag{1}$$


$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
$\mathcal{V} \rightarrow \mathcal{E}$
FC(256,128)
ReLU
FC(128,128)
$\mathcal{E} \rightarrow \mathcal{V}$
Add
$\mathcal{V} \rightarrow \mathcal{E}$
FC(256,128)
ReLU
FC(128,128)
$\mathcal{V} \rightarrow \mathcal{E}$
Add
$\mathcal{V}^{\tau}$
⊕
$\mathcal{V} \rightarrow \mathcal{E}$
FC(256,128)
ReLU
FC(128,128)
$\mathcal{E}^{\tau+1}$
Skip connection

(a) Relational graph module.

(b) Context-aware attention captioning module.

Figure 3: (a) Context enhancement module takes in the scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and produces the enhanced object features $\mathcal{V}^{\tau}$ and object relation features $\mathcal{E}^{\tau+1}$; (b) At time step t, the context-aware captioning module takes in the enhanced features v^{τ}_k of the target object and generates the next token y_t with the help of attention mechanism on the attention context features $\mathcal{V}^r$.

where $g_i^\tau \in \mathcal{R}^{128}$ and $g_j^\tau \in \mathcal{R}^{128}$ are the features of nodes i and j at graph step τ. $g_{i,j}^{\tau+1} \in \mathcal{R}^{128}$ denotes the message between nodes i and j at the next graph step $\tau+1$. $[\cdot,\cdot]$ concatenates two vectors. $f^\tau(\cdot)$ is a learnable non-linear function, which is in practice set as an MLP. The aggregated node features from messages after every message passing step is defined as $\mathcal{E} \rightarrow \mathcal{V} : g_i^{\tau+1} = \sum_{k=1}^{K} g_{i,k}^\tau$. We take the node features $\mathcal{V}^\tau$ in the last graph step τ as the output enhanced object features. We append an additional message passing layer after the last graph step and use the learned message $\mathcal{E}^{\tau+1}$ as the output object relation features. An MLP is attached to the output message passing layer to predict the angular deviations between two objects. We illustrate the relational graph module in Fig. 3a.

4.4. Context-aware Attention Captioning

Inspired by Anderson et al. [2], we design a context aware attention captioning module which takes both the enhanced object features and object relation features and generates the caption one token at a time, as shown in Fig. 3b.

Fusion GRU. At time-step t of caption generation, we first concatenate three vectors as the fused input feature u_{t-1}^1: GRU hidden state from time-step $t-1$ denoted as $h_{t-1}^2 \in R^{512}$, enhanced object feature $v_k^\tau \in R^{128}$ of the k^{th} object and GloVE [40] embedding of the token generated at $t-1$ denoted as $x_t = W_e y_{t-1} \in R^{300}$. The Fusion GRU handles the fused input feature u_{t-1}^1 and delivers the hidden state h_t^1 to the attention module.

Attention module. Unlike the attention module in Anderson et al. [2] which only considers object features, we include both the enhanced object features $\mathcal{V}^\tau = \{v_i^\tau\} \in \mathcal{R}^{M\times128}$ as well as the object relation features $e_{k,j} \in \mathcal{R}^{128}$. We add each object relation feature $e_{k,j}$ between the object k and its neighbor j to the corresponding enhanced object feature v_j of the j^{th} object as the final attention context feature set $\mathcal{V}^r = \{v_1^r, ..., v_k^\tau, ..., v_M^r\}$. Intuitively, the attention module will attend to the neighbor objects and their associated relations with the current object. We define the intermediate attention distribution $\alpha_t \in \mathcal{R}^{M\times128}$ over the context features as:

$$\alpha_t = \text{softmax}((\mathcal{V}^r W_v + \mathbb{1}_h h_{t-1}^{1T} W_h) W_a) \mathbb{1}_a \qquad (2)$$

where $W_a \in \mathcal{R}^{128\times1}$, $W_v \in \mathcal{R}^{128\times128}$, $W_h \in \mathcal{R}^{512\times128}$ are learnable parameters. $\mathbb{1}_h \in \mathcal{R}^{M\times1}$ and $\mathbb{1}_a \in \mathcal{R}^{1\times128}$ are identity matrices. Finally, the attention module outputs the aggregated context vector $\hat{v}_t = \sum_{i=1}^{M} \mathcal{V}_i^r \odot \alpha_{ti}$ to represent the attended object and inter-object relation.

Language GRU. We then concatenate the hidden state h_{t-1}^1 of the Fusion GRU in last time step and the aggregated context vector $\hat{v}_t$, and process them with a MLP as the fused feature u_t^2. The language GRU takes in the fused input u_t^2 and delivers the hidden state h_t^2 to the output MLP to predict token y_t at the current time step t.

4.5. Training Objective

Object detection loss. We use the same detection loss $\mathcal{L}_{det}$ as introduced in Qi et al. [41] for object proposals $\mathcal{D}_{\text{bbox}}$ and $\mathcal{D}_{\text{objn}}$: $\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{vote-reg}} + 0.5\mathcal{L}_{\text{objn-cls}} + \mathcal{L}_{\text{box}} + 0.1\mathcal{L}_{\text{sem-cls}}$, where $\mathcal{L}_{\text{vote-reg}}$, $\mathcal{L}_{\text{objn-cls}}$, $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{sem-cls}}$ represent the vote regression loss (defined in Qi et al. [41]), the objectness binary classification loss, box regression loss and the semantic classification loss for the 18 ScanNet benchmark classes, respectively. We ignore the bounding box orientations in our task and simplify $\mathcal{L}_{\text{box}}$ as $\mathcal{L}_{\text{box}} = \mathcal{L}_{\text{center-reg}} + 0.1\mathcal{L}_{\text{size-cls}} + \mathcal{L}_{\text{size-reg}}$, where $\mathcal{L}_{\text{center-reg}}$, $\mathcal{L}_{\text{size-cls}}$ and $\mathcal{L}_{\text{size-reg}}$ are used for regressing the box center, classifying the box size and regressing the box size, respectively. We refer readers to Qi et al. [41] for more details.

Relative orientation loss. To stabilize the learning process of the relational graph module, we apply a relative orientation loss $\mathcal{L}_{\text{ad}}$ on the message passing network as a proxy loss. We discretize the output angular deviations ranges from 0° to 180° into 6 classes, and use a cross entropy loss as our classification loss. We construct the ground truth labels using the transformation matrices of the aligned CAD models in Scan2CAD [3], and mask out objects not provided in Scan2CAD in the loss function.

Description loss. The main objective loss constrains the description generation. We apply a conventional cross entropy loss function $\mathcal{L}_{\text{des}}$ on the generated token probabilities, as in previous work [52, 51, 27].

Final loss. We combine all three loss terms in a linear manner as our final loss function:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{det}} + \beta\mathcal{L}_{\text{ad}} + \gamma\mathcal{L}_{\text{des}} \qquad (3)$$

where α, β and γ are the weights for the individual loss terms. After fine-tuning on the validation split, we set those weights to $\alpha = 10$, $\beta = 1$, and $\gamma = 0.1$ in our experiments to ensure the loss terms are roughly of the same magnitude.

4.6. Training and Inference

In our experiments, we randomly select 40,000 points from ScanNet mesh vertices. During training, we set the upper bound of the number of object proposals as $M = 256$. We only use the unmasked predictions corresponding to the provided objects in Scan2CAD for minimizing the relative orientation loss, as stated in 4.5. To optimize the description loss, we select the generated description of the object proposal with the largest IoU with the ground truth bounding box. During inference, we apply a non-maximum suppression module to suppress overlapping proposals.

4.7. Implementation Details

We implement our architecture using PyTorch [39] and train end-to-end using ADAM [29] with a learning rate of

$1\mathrm{e}{-3}$. We train the model for $90,000$ iterations until convergence. To avoid overfitting, we set the weight decay factor to $1\mathrm{e}{-5}$ and apply data augmentation to our training data. Following ScanRefer [7], the point cloud is rotated by a random angle in $[-5°, 5°]$ about all three axes and randomly translated within 0.5 meters in all directions. Since the ground alignment in ScanNet is imperfect, the rotation is around all axes (not just up). We truncate descriptions longer than 30 tokens and add SOS and EOS tokens to indicate the start and end of the description.

5. Experiments

Dataset. We use the ScanRefer [7] dataset which consists of 51,583 descriptions for 11,046 objects in 800 ScanNet [12] scenes. The descriptions contain information about the appearance of the objects (e.g. "this is a black wooden chair"), and the spatial relations between the annotated object and nearby objects (e.g. "the chair is placed at the end of the long dining table right before the TV on the wall").

Train&val splits. Following the official ScanRefer [7] benchmark split, we divide our data into train/val sets with 36,665 and 9,508 samples respectively, ensuring disjoint scenes for each split. Results and analysis are conducted on the val split, as the hidden test set is not officially available.

Metrics. To jointly measure the quality of the generated description and the detected bounding boxes, we evaluate the descriptions by combining standard image captioning metrics such as CiDEr [50] and BLEU [37], with Intersection-over-Union (IoU) scores between predicted bounding boxes and the target bounding boxes. We define our combined metrics as $m@k\text{IoU} = \frac{1}{N}\sum_{i=0}^{N} m_i u_i$, where $u_i \in \{0, 1\}$ is set to 1 if the IoU score for the i^{th} box is greater than k, otherwise 0. We use m to represent the captioning metrics CiDEr [50], BLEU-4 [37], METEOR [4] and ROUGE [32], abbreviated as C, B-4, M, R, respectively. N is the number of ground truth or detected object bounding boxes. We use mean average precision (mAP) thresholded by IoU as the object detection metric.

Skylines with ground truth input. To examine the upper limit of our proposed 3D dense captioning task, we use the ground truth (GT) object bounding boxes for generating object descriptions using our method and retrieval based approaches. We compare the performance of captioning in 3D with existing 2D-based captioning methods. For our 2D-based baselines, we generate descriptions for the 2D renders of the reconstructed ScanNet [12] scenes using the recorded viewpoints in ScanRefer [7].

Oracle2Cap3D We use ground truth 3D object bounding box features instead of detection backbone predictions to generate object descriptions. The relational graph and context-aware attention captioning module learn and generate corresponding captioning for each object. We use the same hyper-parameters with the Scan2Cap experiment.

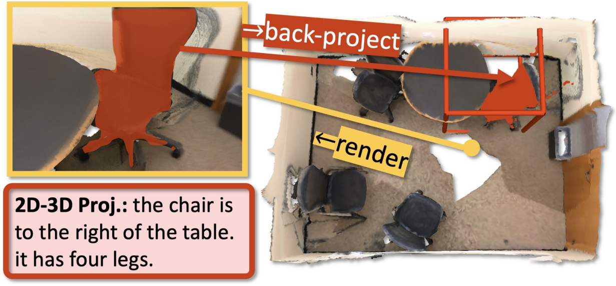

→back-project
←render
2D-3D Proj.: the chair is to the right of the table. it has four legs.

Figure 4: In 2D-3D Proj, we first generate a description for each detected object in a rendered viewpoint. Then we back-project the object mask to the 3D space to evaluate the caption with our proposed caption evaluation metric.

OracleRetr3D We use the ground truth 3D object bounding box features in the val split to obtain the description for the most similar object features in the train split.

Oracle2Cap2D We first concatenate the global image and target object features and feed it to a caption generation method similar to [51]. In addition to [51], we try a memory augmented meshed transformer [10]. Surprisingly, the former performs better (see supplementary for details). We suspect that this performance gap is due to noisy 2D input and the size of our dataset, which does not allow for training complex methods (e.g. transformers) to their maximum potential. The target object bounding boxes are extracted using rendered ground truth instance masks and their features are extracted using a pre-trained ResNet-101 [19].

OracleRetr2D Similar to *OracleRetr3D*, use ground truth 2D object bounding box features in the val split to retrieve the description from the most similar train split object.

Baselines. We design experiments that leverage the detected object information in the input for description generation. Additionally, we show how existing 2D-based captioning methods perform in our newly proposed task.

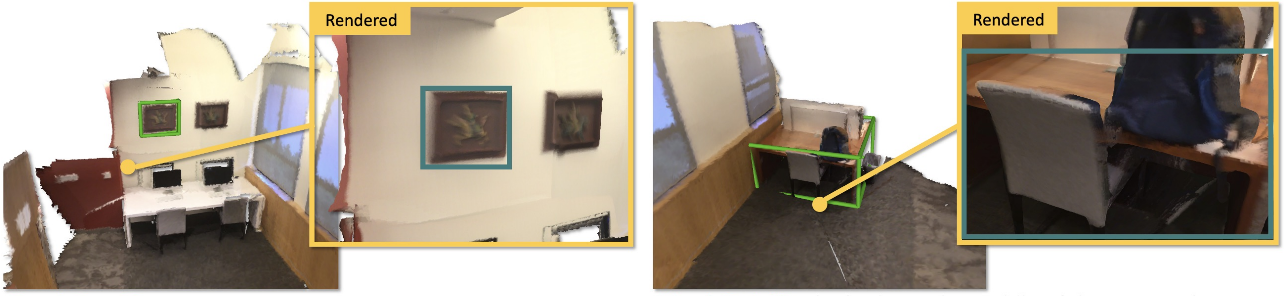
***VoteNetRetr* [41]** Similar to *OracleRetr3D*, but we use the features of the 3D bounding boxes detected using a pre-trained VoteNet [41].

2D-3D Proj We first detect the object bounding boxes in rendered images using a pre-trained Mask R-CNN [20] with a ResNet-101 [19] backbone, then feed the 2D object bounding box features to our description generation module similar to Vinyals et al. [51]. We evaluate the generated captions in 3D by back-projecting the 2D masks to 3D using inverse camera extrinsics (see Fig. 4).

3D-2D Proj We first detect the object bounding boxes in scans using a pre-trained VoteNet [41], then project the bounding boxes to the rendered images. The 2D bounding box features are fed to our captioning module which uses the same decoding scheme as in Vinyals et al. [51].

	Captioning	Detection	C@0.25IoU	B-4@0.25IoU	M@0.25IoU	R@0.25IoU	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	mAP@0.5IoU
2D-3D Proj.	2D	Mask R-CNN	18.29	10.27	16.67	33.63	8.31	2.31	12.54	25.93	10.50
3D-2D Proj.	2D	VoteNet	19.73	17.86	19.83	40.68	11.47	8.56	15.73	31.65	31.83
VoteNetRetr [41]	3D	VoteNet	15.12	18.09	19.93	38.99	10.18	13.38	17.14	33.22	31.83
Ours	3D	VoteNet	**56.82**	**34.18**	**26.29**	**55.27**	**39.08**	**23.32**	**21.97**	**44.78**	**32.21**

Table 1: Comparison of 3D dense captioning results obtained by Scan2Cap and other baseline methods. We average the scores of the conventional captioning metrics, e.g. CiDEr [50], with the percentage of the predicted bounding boxes whose IoU with the ground truth are greater than 0.25 and 0.5. Our method outperforms all baselines with a remarkable margin.



OracleRetr2D: this is a brown picture. it is above a cabinet.

OracleRetr3D: this is a framed print hanging on the wall in the kitchen. it is above the stove.

Oracle2Cap2D: the picture is above the toilet. it is square.

Oracle2Cap3D: the picture is above the white desk. it is a dark and framed.

OracleRetr2D: the bed is in the middle of the room. it has a brown headboard.

OracleRetr3D: this is a metal desk. it can be seen from the door.

Oracle2Cap2D: this is a brown desk. it is to the left of a table.

Oracle2Cap3D: the brown desk is next to the window. it is a brown rectangle.

Figure 5: Qualitative results from skylines with GT input with inaccurate parts of the generated caption underscored. Captioning in 3D benefits from the richness of 3D context, while captioning with 2D information fails to capture the details of the local physical environment. Best viewed in color.

	Cap	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU
OracleRetr2D	2D	20.51	20.17	23.76	50.98
Oracle2Cap2D	2D	58.44	37.05	28.59	61.35
OracleRetr3D	3D	33.03	23.36	25.80	52.99
Oracle2Cap3D	3D	**67.95**	**41.49**	**29.23**	**63.66**

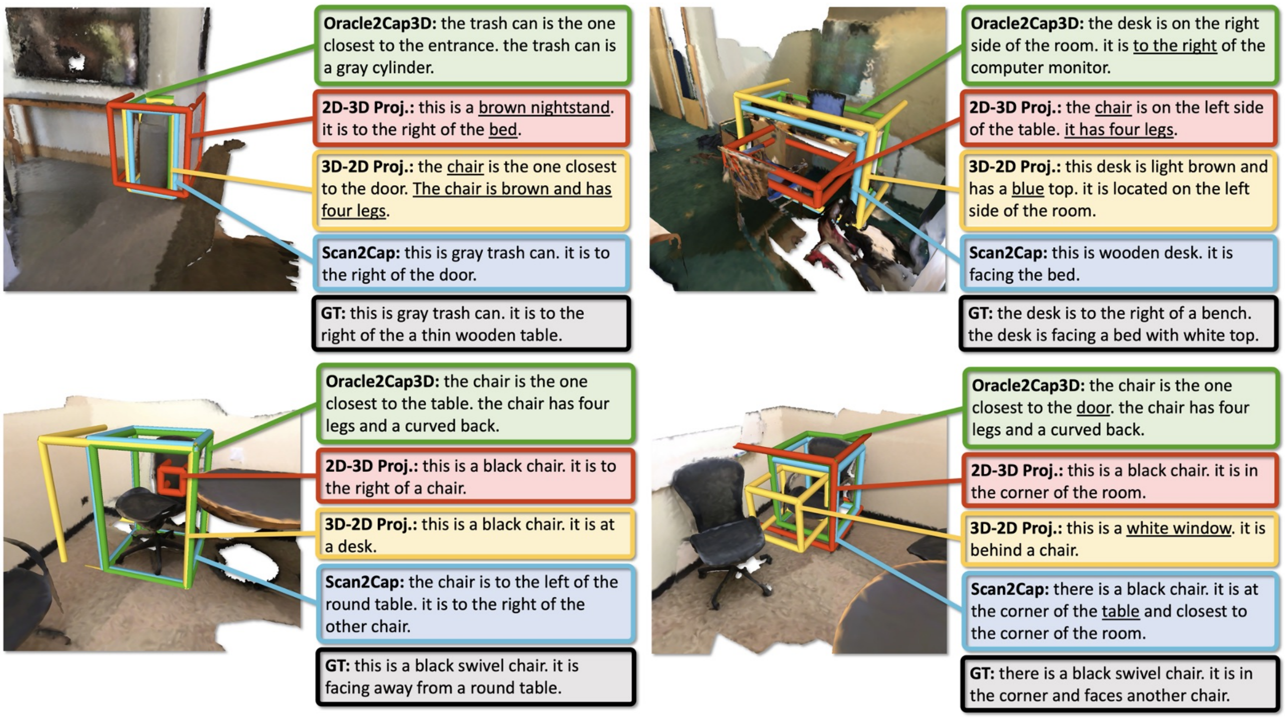
Table 2: Comparison of 3D dense captioning results obtained by our method and other baseline methods with GT detections. We average the scores of the conventional captioning metrics with the percentage of the predicted bounding boxes whose IoU with the ground truth are greater than 0.5. Our method with GT bounding boxes outperforms all variants with a remarkable margin.

	Cap	Acc (Category)	Acc (Attribute)	Acc (Relation)
Oracle2Cap2D	2D	69.00	67.42	37.00
Oracle2Cap3D	3D	85.15 (**+16.15**)	72.22 (**+4.80**)	76.24 (**+39.24**)
Ours	3D	84.16 (**+15.16**)	64.21 (-3.21)	69.00 (**+32.00**)

Table 3: Manual analysis of the generated captions obtained by skyline methods with GT input and ours. We measure the accuracy of three different aspects (object categories, appearance attributes and spatial relations) in the generated captions. Compared to captioning in 2D, captioning directly in 3D better capture these aspects in descriptions, especially for describing spatial relations in the local environment.

5.1. Quantitative Analysis

We compare our method with the baseline methods on the official val split of ScanRefer [7]. As there is no direct prior work on this newly proposed task, we divide description generation into: 1) generating the object bounding boxes and descriptions in 2D input, and back-projecting the bounding boxes to 3D using camera parameters; 2) directly generating object bounding boxes with descriptions in 3D space. As shown in Tab. 1, describing the detected objects in 3D results in a big performance boost compared to the back-projected 2D approach (39.08% compared to 11.47% on C@0.5IoU). When using ground truth, descriptions generated with 3D object bounding boxes (*Oracle2Cap3D*) effectively outperform their counterparts that use 2D object bounding box information (*Oracle2Cap2D*), as shown in Tab. 2. The performance gap between our method and *Oracle2Cap3D* indicates that the detection backbone can be further improved as a potential future work.



Oracle2Cap3D: the trash can is the one closest to the entrance. the trash can is a gray cylinder.

2D-3D Proj.: this is a brown nightstand. it is to the right of the bed.

3D-2D Proj.: the chair is the one closest to the door. The chair is brown and has four legs.

Scan2Cap: this is gray trash can. it is to the right of the door.

GT: this is gray trash can. it is to the right of the a thin wooden table.

Oracle2Cap3D: the desk is on the right side of the room. it is to the right of the computer monitor.

2D-3D Proj.: the chair is on the left side of the table. it has four legs.

3D-2D Proj.: this desk is light brown and has a blue top. it is located on the left side of the room.

Scan2Cap: this is wooden desk. it is facing the bed.

GT: the desk is to the right of a bench. the desk is facing a bed with white top.

Oracle2Cap3D: the chair is the one closest to the table. the chair has four legs and a curved back.

2D-3D Proj.: this is a black chair. it is to the right of a chair.

3D-2D Proj.: this is a black chair. it is at a desk.

Scan2Cap: the chair is to the left of the round table. it is to the right of the other chair.

GT: this is a black swivel chair. it is facing away from a round table.

Oracle2Cap3D: the chair is the one closest to the door. the chair has four legs and a curved back.

2D-3D Proj.: this is a black chair. it is in the corner of the room.

3D-2D Proj.: this is a white window. it is behind a chair.

Scan2Cap: there is a black chair. it is at the corner of the table and closest to the corner of the room.

GT: there is a black swivel chair. it is in the corner and faces another chair.

Figure 6: Qualitative results from baseline methods and Scan2Cap with inaccurate parts of the generated caption underscored. Scan2Cap produces good bounding boxes with descriptions for the target appearance and their relational interactions with objects nearby. In contrast, the baselines suffers from poor bounding box predictions or limited view and produces less informative captions. Best viewed in color.

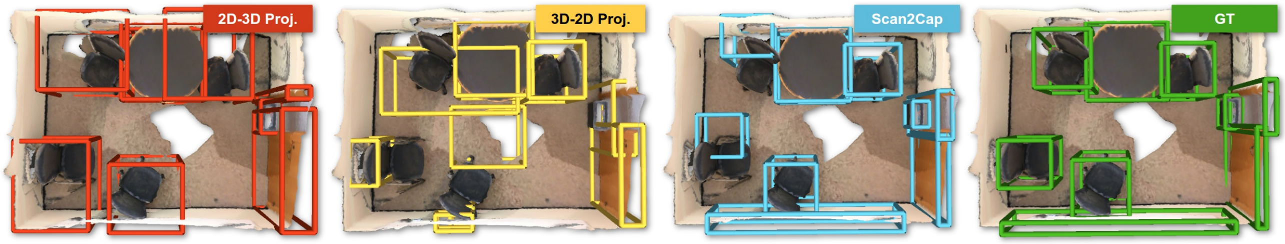


Figure 7: Comparison of object detections of baseline methods and Scan2Cap. 2D-3D Proj. suffers from the detection performance gap between image and 3D space. Scan2Cap produces better bounding boxes compared to 3D-2D Proj. due to the end-to-end fine-tuning.

	C@0.25IoU	B-4@0.25IoU	M@0.25IoU	R@0.25IoU	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	mAP@0.5IoU
Ours (fixed VoteNet)	56.20	**35.14**	26.14	**55.71**	33.87	20.11	20.48	42.33	31.83
Ours (end-to-end)	**56.82**	34.18	**26.29**	55.27	**39.08**	**23.32**	**21.97**	**44.78**	**32.21**

Table 4: Ablation study with a fixed pre-trained VoteNet [41] and an end-to-end fine-tuned VoteNet. We compute standard captioning metrics with respect to the percentage of the predicted bounding box whose IoU with the ground truth are greater than 0.25 and 0.5. Higher values are better.

5.2. Qualitative Analysis

We see from Fig. 5 that the captions retrieved by OracleRetr2D hallucinate objects that are not there, while Oracle2Cap2D provides inaccurate captions that fail to capture correct local context. In contrast, the captions from Oracle2Cap3D are longer and capture relationships with the surrounding objects, such as "above the white desk" and "next to the window". Fig. 6 show the qualitative results of Oracle2Cap3D, 2D-3D Proj, 3D-2D Proj and our method (Scan2Cap). Leveraging the end-to-end training, Scan2Cap

	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU
VoteNet [41]+GRU [9]	34.31	21.42	20.13	41.33
VoteNet [41]+CAC	36.15	21.58	20.65	41.78
VoteNet [41]+RG+CAC	**39.08**	**23.32**	**21.97**	**44.78**

Table 5: Ablation study with different components in our method: VoteNet [41] + GRU [9], which is similar to "show and tell" [51]; VoteNet + Context-aware Attention Captioning (CAC); VoteNet + Relational Graph (RG) + Context-aware Attention Captioning (CAC), namely Scan2Cap. We compute standard captioning metrics with respect to the percentage of the predicted bounding boxes whose IoU with the ground truth are greater than 0.5. The higher the better. Clearly, our method with attention mechanism and graph module is shown to be effective.

is able to predict better object bounding boxes compared to the baseline methods (see Fig. 6 top row). Aside from the improved quality of object bounding boxes, descriptions generated by our method are richer when describing the relations between objects (see second row of Fig. 6).

Provided with the ground truth object information, Oracle2Cap3D can include even more details in the descriptions. However, there are mistakes with the local surroundings (see the sample in the right column in Fig. 6), indicating there is still room for improvement. In contrast, image-based 2D-3D Proj. suffers from limitations of the 2D input and fails to produce good bounding boxes with detailed descriptions. Compared to our method, 3D-2D Proj. fails to predict good bounding boxes because of the lack of a fine-tuned detection backbone, as shown in Fig. 7.

5.3. Analysis and Ablations

Is it better to caption in 3D or 2D? One question we want to study is whether it is better to caption in 3D or 2D. Therefore, we conduct a manual analysis on 100 randomly selected descriptions generated by Oracle2Cap2D, Oracle2Cap3D and our method. In this analysis, we manually check if those descriptions correctly capture three important aspects for indoor objects: object categories, appearance attributes and spatial relations. As demonstrated in Tab. 3, directly captioning objects in 3D captures those aspects more accurately when comparing Oracle2Cap3D with Oracle2Cap2D, especially for describing the spatial relations. However, the accuracy drop on object attributes from Oracle2Cap2D to our method (-3.21%) shows the detection backbone can still be improved.

Does context-aware attention captioning help? We compare our model with the basic description generation component (GRU) introduced in Vinyals et al. [51] and our model with the context-aware attention captioning (CAC) as discussed in Sec. 4.4. The model equipped with the context-aware captioning module outperforms its counterpart without attention mechanism on all metrics (see the first row vs. the second row in Tab. 5).

Does the relational graph help? We evaluate the performance of our method against our model without the proposed relational graph (RG) and/or the context-aware attention captioning (CAC). As shown in Tab. 5, our model equipped with the context enhancement module (third row) outperforms all other ablations.

Does end-to-end training help? We show in Tab. 4 the effectiveness of fine-tuning the pretrained VoteNet end-to-end with the description generation objective. We observe that end-to-end training of the network allows for gradient updates from our relative orientation loss and description generation loss that compensate for detection errors. While the fine-tuned VoteNet detection backbone delivers similar detection results, its performance on describing objects outperforms its fixed ablation by a big margin on all more demanding metrics (see columns for metrics m@0.5IoU in Tab. 4).

6. Conclusion

In this work, we introduce the task of dense description generation in RGB-D scans. We propose an end-to-end trained architecture to localize the 3D objects in the input point cloud and generate descriptions for them in natural language. Thus, we address the 3D localization and description generation problems at the same time. We apply an attention-based captioning pipeline equipped with a message passing network to generate descriptive tokens while referring to related components in the local context. Our architecture effectively localizes and describes 3D objects, outperforming 2D-based dense captioning methods on the 3D dense description generation task by a large margin. Nevertheless, our method struggles to capture complex relations like ordinal counting. For instance, our method only predicts "the round chair next to another wooden chair", while the ground truth "the *third* round chair from the wall" reveals more fine-grained spatial relations, indicating possibilities for improvement. Overall, we hope that our work will enable future research in 3D vision and language.

Acknowledgements

This work is funded by Google (AugmentedPerception), the ERC Starting Grant Scan2CAD (804724), and a Google Faculty Award. We would also like to thank the support of the TUM-IAS Rudolf Mößbauer and Hans Fischer Fellowships (Focus Group Visual Computing), as well as the the German Research Foundation (DFG) under the Grant *Making Machine Learning on Static and Dynamic 3D Data Practical*. Angel X. Chang is supported by the Canada CIFAR AI Chair program, and Ali Gholami is funded through an NSERC Discovery Grant.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018. 1, 2, 3, 4

[3] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. Scan2CAD: Learning CAD model alignment in RGB-D scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 4

[4] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 5

[5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *Proceedings of the International Conference on 3D Vision (3DV)*. 2

[6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[7] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. *16th European Conference on Computer Vision (ECCV)*, 2020. 2, 5, 6

[8] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2Shape: Generating shapes from natural language by learning joint embeddings. In *Proc. Asian Conference on Computer Vision (ACCV)*, 2018. 2

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 8

[10] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020. 2, 5

[11] Angela Dai and Matthias Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018. 2

[12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 2, 5

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[14] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 2

[15] Cathrin Elich, Francis Engelmann, Jonas Schult, Theodora Kontogianni, and Bastian Leibe. 3D-BEVIS: Birds-eye-view instance segmentation. *arXiv preprint arXiv:1904.02199*, 2019. 2

[16] Lizhao Gao, Bo Wang, and Wenmin Wang. Image captioning with scene-graph based semantic concepts. In *Proceedings of the International Conference on Machine Learning and Computing*, pages 225–229, 2018. 2

[17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the International Conference on Machine Learning*, 2017. 3

[18] Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7181–7189, 2018. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 5

[21] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019. 2

[22] Ji Hou, Angela Dai, and Matthias Nießner. RevealNet: Seeing behind objects in RGB-D scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2098–2107, 2020. 2

[23] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, 2016. 2

[24] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. SceneNN: A scene meshes dataset with annotations. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 92–101. IEEE, 2016. 2

[25] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vi-*

sion (ECCV), pages 499–515, 2018. 2

[26] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016. 1, 2

[27] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 1, 2, 4

[28] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Dense relational captioning: Triple-stream networks for relationship-based captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6271–6280, 2019. 1, 2

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[30] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3D instance segmentation via multi-task metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9256–9266, 2019. 2

[31] Xiangyang Li, Shuqiang Jiang, and Jungong Han. Learning object context for dense captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8650–8657, 2019. 1

[32] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 5

[33] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383, 2017. 1, 2

[34] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7219–7228, 2018. 2

[35] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 2

[36] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. PanopticFusion: Online volumetric semantic mapping at the level of stuff and things. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. 2

[37] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 5

[38] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 3

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035. 2019. 4

[40] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 4

[41] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 2, 3, 4, 5, 6, 7, 8

[42] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. ImVoteNet: Boosting 3D object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4404–4413, 2020. 2

[43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2, 3

[44] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 2

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[46] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017. 2

[47] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. ChatPainter: Improving text to image generation using dialogue. *arXiv preprint arXiv:1802.08216*, 2018. 2

[48] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 2

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1

[50] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 5, 6

[51] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer*

vision and pattern recognition, pages 3156–3164, 2015. 1, 2, 4, 5, 8

[52] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 1, 2, 4

[53] Linjie Yang, Kevin Tang, Jianchao Yang, and Li-Jia Li. Dense captioning with joint inference and visual context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2193–2202, 2017. 1, 2

[54] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019. 2

[55] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699, 2018. 2

[56] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, and Jing Shao. Context and attribute grounded dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6241–6250, 2019. 1

[57] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. MAttNet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

CCC RightsLink

Home | Help | Live Chat | Zhenyu Chen

IEEE Requesting permission to reuse content from an IEEE publication

Scan2Cap: Context-aware Dense Captioning in RGB-D Scans

Conference Proceedings:
2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Author: Dave Zhenyu Chen; Ali Gholami; Matthias Nießner; Angel X. Chang

Publisher: IEEE

Date: 20-25 June 2021

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

D^{3}Net: A Unified Speaker-Listener Architecture for 3D Dense Captioning and Visual Grounding

Dave Zhenyu Chen[1] Qirui Wu[2] Matthias Nießner[1] Angel X. Chang[2]
[1]Technical University of Munich [2]Simon Fraser University
https://daveredrum.github.io/D3Net/

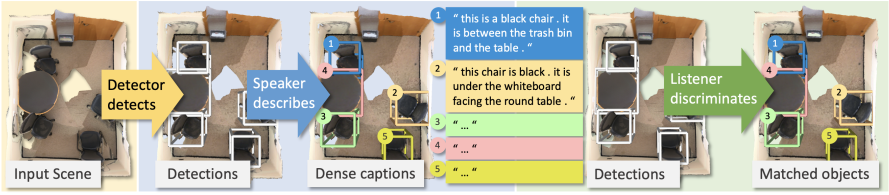

Input Scene
Detector detects
Detections
Speaker describes
1
2
3
4
5
Dense captions
" this is a black chair . it is between the trash bin and the table . "
" this chair is black . it is under the whiteboard facing the round table . "
" ... "
" ... "
" ... "
Detections
Listener discriminates
Matched objects

Fig. 1: We introduce D^{3}Net, an end-to-end neural speaker-listener architecture that can **d**etect, **d**escribe and **d**iscriminate. D^{3}Net also enables semi-supervised training on ScanNet data with partially annotated descriptions.

Abstract. Recent work on dense captioning and visual grounding in 3D have achieved impressive results. Despite developments in both areas, the limited amount of available 3D vision-language data causes overfitting issues for 3D visual grounding and 3D dense captioning methods. Also, how to discriminatively describe objects in complex 3D environments is not fully studied yet. To address these challenges, we present D^{3}Net, an end-to-end neural speaker-listener architecture that can **d**etect, **d**escribe and **d**iscriminate. Our D^{3}Net unifies dense captioning and visual grounding in 3D in a self-critical manner. This self-critical property of D^{3}Net encourages generation of discriminative object captions and enables semi-supervised training on scan data with partially annotated descriptions. Our method outperforms SOTA methods in both tasks on the ScanRefer dataset, surpassing the SOTA 3D dense captioning method by a significant margin.

1 Introduction

Recently, there has been increasing interest in bridging 3D visual scene understanding [41, 18, 19, 5, 11, 22, 46] and natural language processing [48, 13, 4, 34, 55]. The task of 3D visual grounding [6, 59, 60] localizes 3D objects described by natural language queries. 3D dense captioning proposed by Chen et al. [7] is

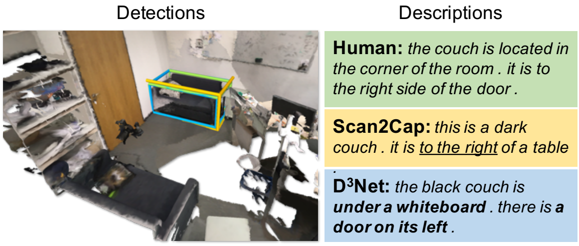

Detections
Descriptions
Human: the couch is located in the corner of the room . it is to the right side of the door .
Scan2Cap: this is a dark couch . it is to the right of a table
D3Net: the black couch is under a whiteboard . there is a door on its left .

Fig. 2: Prior work [7] struggle to produce discriminative object captions. Also, captions often appear to be template-based. In contrast, our D^{3}Net generates discriminative object captions.

the reverse task where we generate descriptions for 3D objects in RGB-D scans. Both tasks enable applications such as assistive robots and natural language control in AR/VR systems.

However, existing work on 3D visual grounding [6, 1, 59, 23, 60] and dense captioning [7, 58] treats the two problems as separate, with *detect-then-discriminate* or *detect-then-describe* being the common strategies for tackling the two tasks. Separating the two complementary tasks hinders holistic 3D scene understanding where the ultimate goal is to create models that can infer: 1) what are the objects; 2) how to describe each object; 3) what object is being referred to through natural language. The disadvantages of having separated strategies are twofold. First, the detect-then-describe strategy often struggles to describe target objects in a discriminative way. In Fig. 2, the generated descriptions from Scan2Cap [7] fail to uniquely describe the target objects, especially in scenes with several similar objects. Second, existing 3D visual grounding methods [6, 60] in the detect-then-discriminate strategy suffer from severe overfitting issue, partly due to the small amount of 3D vision-language data [6, 1] which is limited compared to counterpart 2D datasets such as MSCOCO [32].

To address these issues, we propose an end-to-end self-critical solution, D^{3}Net, to enable discriminability in dense caption generation and utilize the generated captions improve localization. Relevant work in image captioning [36, 33] tackles similar issues where the generated captions are indiscriminative and repetitive by explicitly reinforcing discriminative caption generation with an image retrieval loss. Inspired by this scheme, we introduce a speaker-listener strategy, where the captioning module "speaks" about the 3D objects, while the localization module "listens" and finds the targets. Our proposed speaker-listener architecture can **d**etect, **d**escribe and **d**iscriminate, as illustrated in Fig. 1. The key idea is to reinforce the speaker to generate discriminative descriptions so that the listener can better localize the described targets given those descriptions.

This approach brings another benefit. Since the speaker-listener architecture self-critically generates and discriminates descriptions, we can train on scenes without any object descriptions. We see further improvements in 3D dense captioning and 3D visual grounding performance when using this additional data alongside annotated scenes. This can allow for semi-supervised training on RGB-D scans beyond the ScanNet dataset. To summarize, our contributions are:

- We introduce a unified speaker-listener architecture to generate discriminative object descriptions in RGB-D scans. Our architecture allows for a semi-supervised training scheme that can alleviate data shortage in the 3D vision-language field.
- We study how the different components impact performance and find that having a strong detector is essential, and that by jointly optimizing the detector, speaker, and listener we can improve detection as well as 3D dense captioning and visual grounding.
- We show that our method outperforms the state-of-the-art for both 3D dense captioning and 3D visual grounding method by a significant margin.

2 Related Work

Vision and language in 3D. Recently, there has been growing interest in grounding language to 3D data [8, 2, 6, 1, 52, 44, 47]. Chen et al. [6] and Achlioptas et al. [1] introduce two complementary datasets consisting of descriptions of real-world 3D objects from ScanNet [11] reconstructions, named ScanRefer and ReferIt3D, respectively. ScanRefer proposes the joint task of detecting and localizing objects in a 3D scan based on a textual description, while ReferIt3D is focused on distinguishing 3D objects from the same semantic class given ground-truth bounding boxes. Yuan et al. [59] localize objects by decomposing input queries into fine-grained aspects, and use PointGroup [25] as their visual backbone. However, the frozen detection backbone is not fine-tuned together with the localization module. Zhao et al. [60] propose a transformer-based architecture with a VoteNet [41] backbone to handle multimodal contexts during localization. Despite the improved matching module, their work still suffers from poor quality detections due to the weak 3D detector. We show that fine-tuning an improved 3D detector is essential to getting good predictions and good localization performance. Chen et al. [7] introduce the task of densely detecting and captioning objects in RGB-D scans. Recently, Yuan et al. [58] aggregate the 2D features to point cloud to generate faithful object descriptions. Although their methods can effectively detect objects and generate captions w.r.t. their attributes, the quality of the bounding boxes and the discriminability of the captions are inadequate. Our method explicitly handles the discriminability of the generated captions through a self-critical speaker-listener architecture, resulting in the state-of-the-art performance in both 3D dense captioning and 3D visual grounding tasks.

Generating captions in images. Image captioning has attracted a great deal of interest [50, 53, 14, 28, 35, 3, 26, 43, 45]. Recent work [36, 33] suggest that traditional encoder-decoder-based image captioning methods suffer from the discriminability issues. Luo et al. [36] propose an additional image retrieval branch to reinforce discriminative caption generation. Liu et al. [33] propose a reinforcement learning method to train not only on annotated web images, but also images without any paired captions. In contrast to generating captions for the entire image, in the dense captioning task we densely generate captions for each detected object in the input image [27, 54, 30]. Although such methods are effective for generating captions in 2D images, directly applying such training techniques on 3D dense captioning can lead to unsatisfactory results, since the captions involve 3D geometric relationships. In contrast, we work directly on 3D scene input dealing with object attributes as well as 3D spatial relationships.

Grounding referential expressions in images. There has been tremendous progress in the task of grounding referential expressions in images, also known as visual grounding [29, 40, 38, 21, 56, 20]. Given an image and a natural language text query as input, the target object is either localized by a bounding box [21, 56], or a segmentation mask [20]. These methods have achieved great success in the image domain. However, they are not designed to deal with 3D geometry inputs and handle complex 3D spatial relationships. Our proposed method directly decomposes the 3D input data with a sparse convolutional detection backbone, which produces accurate object proposals as well as semantically rich features.

Speaker-listener models for grounding. The speaker-listener model is a popular architecture for pragmatic language understanding, where a line of research explores how the context and communicative goals affect the linguistics [10, 16]. Recent work use neural speaker-listener architectures to tackle referring expression generation [38, 57, 37], vision-language navigation [15], and shape differentiation [2]. Mao et al. [38] construct a CNN-LSTM architecture optimized by a softmax loss to directly discriminate the generated referential expressions. There is no separate neural listener module compared with our method. Luo and Shakhnarovich [37] and Yu et al. [57] introduce a LSTM-based neural listener in the speaker-listener pipeline, but generating the referential expression is not directly supervised via the listener model, but rather trained via a proxy objective. In contrast, our method directly optimizes the Transformer-based neural listener for the visual grounding task by discriminating the generated object captions without any proxy training objective. Similarly, Achlioptas et al. [2] includes a pretrained and frozen listener in the training objective, while ours enables joint end-to-end optimization for both the speaker and listener via policy gradient algorithm. We experimentally show our method to be effective for semi-supervised learning in the two 3D vision-language tasks.

3 Method

D^{3}Net has three components: a 3D object detector, the speaker (captioning) module, and the listener (localization) module. Fig. 3 shows the overall architecture and training flow. The point clouds are fed into the detector to predict object proposals. The speaker takes object proposals as input to produce captions. To increase caption discriminability, we match these captions with object proposals via the listener. Caption quality is measured by the CIDEr [49] scores and the listener loss, which are back-propagated via REINFORCE [51] as re-

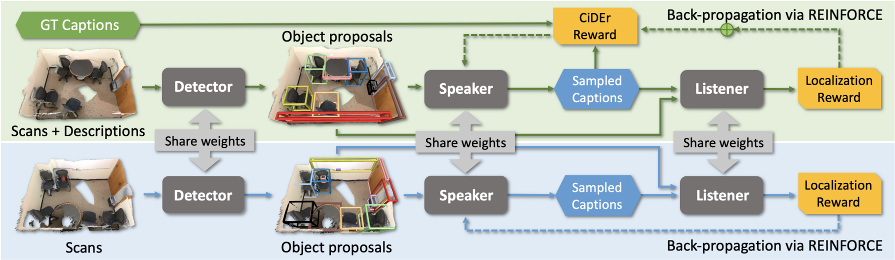


Fig. 3: D^{3}Net architecture. We input point clouds into the *detector* to predict object proposals. Then, those proposals are fed into the speaker to generate captions that *describes* each object. To *discriminate* the object described by each caption, the listener matches the generated captions with object proposals. The captioning and localization results are back-propagated via REINFORCE [51] as rewards through the dashed lines. D^{3}Net also enables end-to-end training on point clouds with no GT object descriptions (bottom blue block).

wards to the speaker. Our architecture can handle scenes without ground-truth (GT) object descriptions by reinforcing the speaker with the listener loss only.

3.1 Modules

Detector. We use PointGroup [25] as our detector module. PointGroup is a relatively simple model for 3D instance segmentation that achieves competitive performance on the ScanNet benchmark. We use ENet to augment the point clouds with multi-view features, following Dai and Nießner [12]. PointGroup uses a U-Net architecture with a SparseConvNet backbone to encode point features, cluster the points, and uses ScoreNet, another U-Net structure, to score each cluster. We take the cluster features after ScoreNet as the encoded object features. We refer readers to the original paper [25] for more details. The object bounding boxes are determined by taking the minimum and maximum points in the point clusters, and are produced as final outputs of our detector module.

Speaker. We base our speaker on the dense captioning method introduced by Chen et al. [7]. Our speaker module has two submodules: 1) a relational graph module, which is responsible for learning object-to-object spatial location relationships; 2) a context-aware attention captioning module, which attentively generates descriptive tokens with respect to the object attributes as well as the object-to-object spatial relationships.

Listener. For the listener, we follow the architecture introduced by Chen et al. [6] but replace the multi-modal fusion module with the transformer-based multimodal fusion module of Zhao et al. [60]. Our listener module has two submodules: 1) a language encoding module with a GRU cell; 2) a transformer-based multimodal fusion module similar to Zhao et al. [60], which attends to elements in the input query descriptions and the detected object proposals. As in Chen et al.

[6], we also incorporate a language object classifier to discriminate the semantics of the target objects in the input query descriptions.

3.2 Training Objective

The three modules are designed to be trained in an end-to-end fashion (see Figure 3). In this section, we describe the loss for each module, and how they are combined for the overall loss.

Detection loss. We use the instance segmentation loss introduced in PointGroup [25] to train the 3D backbone. The detection loss is composed of four parts: $L_{\text{det}} = L_{\text{sem}} + L_{\text{o\_reg}} + L_{\text{o\_dir}} + L_{\text{c\_score}}$. L_{sem} is a cross-entropy loss supervising semantic label prediction for each point. $L_{\text{o\_reg}}$ is a L_1 regression loss constraining the learned point offsets belonging to the same cluster. $L_{\text{o\_dir}}$ constrains the direction of predicted offset vectors, defined as the means of minus cosine similarities. It helps regress precise offsets, particularly for boundary points of large-size objects, since these points are relatively far from the instance centroids. $L_{\text{c\_score}}$ is another binary cross-entropy loss supervising the predicted objectness scores.

Listener loss. The listener loss is composed of a localization loss L_{loc} and a language-based object classification loss L_{lobjcls}. To obtain the localization loss L_{loc}, we first require a target bounding box. We use the detected bounding box with the highest IoU with the GT bounding box as the target bounding box. Then, a cross-entropy loss L_{loc} is applied to supervise the matching score prediction. In the end-to-end training scenario, the detected bounding boxes associated with the generated descriptions from the speaker are treated as the target bounding boxes. The language object classification loss is a cross-entropy loss L_{lobjcls} to supervise the classification based on the input description. The target classes are consistent with the ScanNet 18 classes, excluding structural objects such as "floor" and "wall".

Speaker loss using MLE training objective. The speaker loss is a standard captioning loss from maximum likelihood estimation (MLE). During training, provided with a pair of GT bounding box and the associated GT description, we optimize the description associated with the predicted bounding box which has the highest IoU score with the current GT bounding box. We first treat the description generation task as a sequence prediction task, factorized as: $L_{\text{spk-XE}}(\theta) = -\sum_{t=1}^{T} \log p(\hat{c_t}|\hat{c_1}, ..., c_t\hat{}_{-1}; I, \theta)$, where $\hat{c_t}$ denotes the generated token at step t; I and θ represent the visual signal and model parameter, respectively. The token $\hat{c_t}$ is sampled from the probability distribution over the pre-defined vocabulary. The generation process is performed by greedy decoding or beam search in an autoregressive manner, and we use the argmax function to sample each token.

Joint loss using REINFORCE training objective. We use REINFORCE to train the detector-speaker-listener jointly. We first describe the enhanced speaker-loss, $L_{\text{spk-R}}$ that is trained using reinforcement learning to produce discriminative captions. We then describe the overall loss used in end-to-end training. Following prior work [36, 33, 42, 17, 57, 43], generating descriptions is treated as a reinforcement learning task. In the setting of reinforcement learning, the speaker module is treated as the "agent", while the previously generated words and the input visual signal I are the "environment". At step t, generating word $\hat{c_t}$ by the speaker module is deemed as the "action" taken with the policy p_θ, which is defined by the speaker module parameters θ. Specifically, with the generated description $\hat{C} = \{c_1, ..., c_T\}$, the objective is to maximize the reward function $R(\hat{C}, I)$. We apply the "REINFORCE with baseline" algorithm following Rennie et al. [43] to reduce the variance of this loss function, where a baseline reward $R(C^*, I)$ of the description C^* independent of $\hat{C}$ is introduced. We apply beam search to sample descriptions and choose the greedily decoded descriptions as the baseline. The simplified policy gradient is:

$$L_{\text{spk-R}}(\theta) \approx -(R(\hat{C}, I) - R(C^*, I)) \sum_{t=1}^{T} \text{log}p(\hat{c_t}|I, \theta) \tag{1}$$

Rewards. As the word-level sampling through the argmax function is non-differentiable, the subsequent listener loss cannot be directly back-propagated through the speaker module. A workaround is to use the gumbel softmax reparametrization trick [24]. Following the training scheme of Liu et al. [33] and Luo et al. [36], the listener loss can be inserted into the REINFORCE reward function to increase the discriminability of generated referential descriptions. Specifically, given the localization loss L_{loc} and the language object classification loss L_{lobjcls}, the reward function $R(\hat{C})$ is the weighted sum of the CIDEr score of the sampled description and the listener-related losses:

$$R(\hat{C}, I) = R^{\text{CIDEr}}(\hat{C}, I) - \alpha[L_{\text{loc}}(\hat{C}) + \beta L_{\text{lobjcls}}(\hat{C})] \tag{2}$$

where α and β are the weights balancing the CIDEr reward and the listener rewards. We empirically set them to 0.1 and 1 in our experiments, respectively. To stabilize the training, the reward related to the baseline description $R(C^*)$ should be formulated analogously. Note that there should be no gradient calculation and back-propagation for the baseline C^*. For scenes with no GT descriptions provided, the CIDEr reward is cancelled in the reward function, which in this case becomes $R(\hat{C}, I) = -\alpha[L_{\text{loc}}(\hat{C}) + \beta L_{\text{lobjcls}}(\hat{C})]$.

Relative orientation loss. Following Chen et al. [7], we adopt the relative orientation loss on the message passing module as a proxy loss. The object-to-object relative orientations ranging from 0° to 180° are discretized into 6 classes. We apply a simple cross-entropy loss L_{ori} to supervise the relative orientation predictions.

Overall loss. We combine loss terms in our end-to-end joint training objective as: $L = L_{\text{det}} + L_{\text{spk-R}} + 0.3L_{\text{ori}}$.

3.3 Training

We use a stage-wise training strategy for stable training. We first pretrain the detector backbone on all training scans in ScanNet via the detector loss L_{det}.

We then train the dense captioning pipeline with the pretrained detector and a newly initialized speaker end-to-end via the detector loss and the speaker MLE loss $L_{\text{spk-XE}}$. After the speaker MLE loss converges, we train the visual grounding pipeline with the fine-tuned frozen detector and the listener via the listener loss L_{loc}. Finally, we fine-tune the entire speaker-listener architecture with the overall loss L.

3.4 Inference

During inference, we use the detector and the speaker to do 3D dense captioning and the listener to do visual grounding. The detector first produces object proposals, and the speaker generates a description for each object proposal. We take the minimum and maximum coordinates in the predicted object instance masks to construct the bounding boxes. For the object proposals that are assigned to the same ground truth, we keep only the one with the highest IoU with the GT bounding box. When evaluating the detector itself, the non-maximum suppression is applied.

4 Experiments

4.1 Dataset

We use the ScanRefer [6] dataset consisting of around 51k descriptions for over 11k objects in 800 ScanNet [11] scans. The descriptions include information about the appearance of the objects, as well as the object-to-object spatial relationships. We follow the official split from the ScanRefer benchmark for training and validation. We report our visual grounding results on the validation split and benchmark results on the hidden test set[1]. Our dense captioning results are on the validation split due to the lack of the test grounding truth. We also conduct experiments on the ReferIt3D dataset [1] (please see the supplemental).

4.2 Semi-supervised Training with Extra Data

As the scans in ScanRefer dataset are only a subset of scans in ScanNet, we extend the training set by including all re-scans of the same scenes for semi-supervised training. Unlike the scans in ScanRefer, these re-scans do not have per object descriptions. We can control how much extra data to use by randomly sampling (with replacement) from the set of re-scans. We experiment with augmenting our data with 0.1 to 1 times the amount of annotated data as extra data. During training, we randomly select detected objects in the sampled extra scans for subsequent dense captioning and visual grounding. For the complete 'extra' scenario, we use a comparable amount (1x) of extra data as the annotated data in ScanRefer.

[1] http://kaldir.vc.in.tum.de/scanrefer_benchmark

Table 1: Quantitative results on 3D dense captioning and object detection. As in Chen et al. [7], we average the conventional captioning evaluation metrics with the percentage of the predicted bounding boxes whose IoU with the GTs are higher than 0.5. Our speaker model outperforms the baseline Scan2Cap without training via REINFORCE, while training with CIDEr reward further boosts the dense captioning performance. We also showcase the effectiveness of training with additional scans with no description annotations. Our speaker-listener architecture trained with 1x extra data achieves the best performance.

	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	mAP@0.5
Scan2Cap [7]	39.08	23.32	21.97	44.78	32.21
X-Trans2Cap [58]	43.87	25.05	22.46	44.97	35.31
Ours (MLE)	46.07	30.29	24.35	51.67	50.93
Ours (CIDEr)	57.88	32.64	24.86	52.26	51.01
Ours (CIDEr+fixed loc.)	58.93	33.36	25.12	52.62	51.04
Ours (CIDEr+loc.)	61.30	34.50	25.25	52.80	52.07
Ours (CIDEr+loc.+lobjcls.)	61.50	35.05	25.48	53.31	52.58
Ours (w/ 0.1x extra data)	61.91	35.03	25.38	53.25	52.64
Ours (w/ 0.5x extra data)	62.36	35.54	25.43	53.67	53.17
Ours (w/ 1x extra data)	**62.64**	**35.68**	**25.72**	**53.90**	**53.95**

4.3 Implementation Details

We implement the PointGroup backbone using the Minkowski Engine [9] (see supplement). For the backbone, we train using Adam [31] with a learning rate of 2e-3, on the ScanNet train split with batch size 4 for 140k iterations, until convergence. For data augmentation, we follow Jiang et al. [25], randomly applying jitter, mirroring about the YZ-plane, and rotation about the Z axis (up-axis) to each point cloud scene. We then use the Adam optimizer with learning rate 1e-3 to train the detector and the listener on the ScanRefer dataset with batch size 4 for 60k iterations, until convergence. Each scan is paired with 8 descriptions (i.e. 4 scans and 32 descriptions per batch iteration). Then, we combine the trained detector with the newly initialized speaker on the ScanRefer dataset for the 3D dense captioning task, where the weights of the detector are frozen. We again use Adam with learning rate 1e-3, with the training process converging within 14k iterations. All our experiments are conducted on a RTX 3090, and all neural modules are implemented using PyTorch [39].

4.4 Quantitative Results

3D dense captioning and detection Tab. 1 compares our 3D dense captioning and object detection results against the baseline methods Scan2Cap [7] and X-Trans2Cap [58]. Leveraging the improved PointGroup based detector, our speaker model trained with the conventional MLE objective (Ours (MLE)) outperforms Scan2Cap and X-Trans2Cap by a large margin in all metrics. As expected, training with the CIDEr reward (Ours (CIDEr)) significantly improves the CIDEr score. We note that other captioning metrics are also improved, but

Table 2: Quantitative results on 3D visual grounding. We adapt the evaluation setting as in Chen et al. [6]. "Unique" means there is only one object belongs to a specific class in the scene, while "multiple" represents the cases where more than one object from a specific class can be found in the scene. Clearly, our base visual grounding network outperforms all baselines even before being put into the speaker-listener architecture. After the speaker-listener fine-tuning, our method achieves the state-of-the-art performance on the ScanRefer validation set and the public benchmark. Note that 3DVG-Trans+ is an unpublished extension of 3DVG-Trans [60] which appears only on the public benchmark.

	Val Acc@0.5IoU			Test Acc@0.5IoU		
	Unique	Multiple	Overall	Unique	Multiple	Overall
ScanRefer [6]	53.51	21.11	27.40	43.53	20.97	26.03
TGNN [23]	56.80	23.18	29.70	58.90	25.30	32.80
InstanceRefer [59]	66.83	24.77	32.93	66.69	26.88	35.80
3DVG-Trans [60]	60.64	28.42	34.67	55.15	29.33	35.12
3DVG-Trans+ [60]	-	-	-	57.87	**31.02**	37.04
Ours (w/o fine-tuning)	70.35	27.11	35.58	65.79	27.26	35.90
Ours	**72.04**	**30.05**	**37.87**	**68.43**	30.74	**39.19**

the detection mAP@0.5 remains similar. Training with object localization reward (Ours (CIDEr+loc.)) improves both captioning and detection further due to the improved discriminability during description generation. Note that if we use a frozen pretrained listener (Ours (CIDEr+fixed loc.)), the improvement is not as significant as when we allow the listener weights to be fine-tuned (Ours (CIDEr+loc.)). Our full model with the full listener reward incorporates an additional language object classification loss (Ours (CIDEr+loc.+lobjcls.)) and further improves the performance for both tasks.

Does additional data help? As our method allow for training the listener with scans without language data, we investigate the effectiveness of training with additional ScanNet data that have not been annotated with descriptions. We vary the amount of extra scan data (without descriptions) from 0.1x to 1x of fully annotated data and train our full model with CIDEr and full listener reward (loc.+lobjcls.). Our results (last three rows of Tab. 1), show that our semi-supervised training strategy can leverage the extra data to improve both dense captioning and object detection.

3D visual grounding Tab. 2 compares our results against prior 3D visual grounding methods ScanRefer [6], TGNN [23], InstanceRefer [59] and 3DVG-Transformer [60], and 3DVG-Trans+, an unpublished extension. Our method trained only with the detection loss and the listener loss ("Ours w/o fine-tuning"), i.e. without the speaker-listener setting, outperforms all the previous methods in the "Unique" and "Overall" scenarios. We find the improved fusion module together with the improved detector is sufficient to outperform 3DVG-Trans. Due to the improved detector, our method can distinguish objects in the "Unique" case, where the semantic labels play an important role. Meanwhile, 3DVG-Trans [60] still outperforms our base listener when discriminating objects

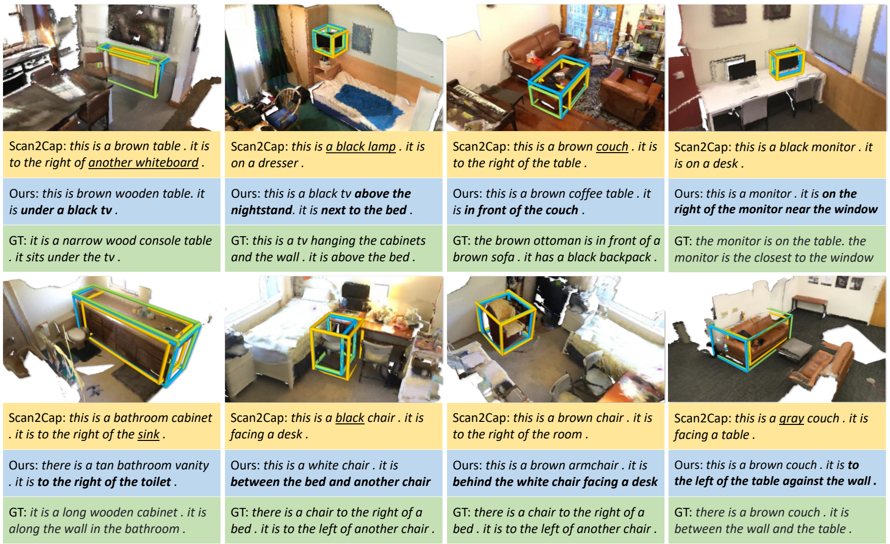

Scan2Cap: this is a brown table . it is to the right of another whiteboard .
Ours: this is brown wooden table. it is under a black tv .
GT: it is a narrow wood console table . it sits under the tv .
Scan2Cap: this is a black lamp . it is on a dresser .
Ours: this is a black tv above the nightstand. it is next to the bed .
GT: this is a tv hanging the cabinets and the wall . it is above the bed .
Scan2Cap: this is a brown couch . it is to the right of the table .
Ours: this is a brown coffee table . it is in front of the couch .
GT: the brown ottoman is in front of a brown sofa . it has a black backpack .
Scan2Cap: this is a black monitor . it is on a desk .
Ours: this is a monitor . it is on the right of the monitor near the window
GT: the monitor is on the table. the monitor is the closest to the window
Scan2Cap: this is a bathroom cabinet . it is to the right of the sink .
Ours: there is a tan bathroom vanity . it is to the right of the toilet .
GT: it is a long wooden cabinet . it is along the wall in the bathroom .
Scan2Cap: this is a black chair . it is facing a desk .
Ours: this is a white chair . it is between the bed and another chair
GT: there is a chair to the right of a bed . it is to the left of another chair .
Scan2Cap: this is a brown chair . it is to the right of the room .
Ours: this is a brown armchair . it is behind the white chair facing a desk
GT: there is a chair to the right of a bed . it is to the left of another chair .
Scan2Cap: this is a gray couch . it is facing a table .
Ours: this is a brown couch . it is to the left of the table against the wall .
GT: there is a brown couch . it is between the wall and the table .

Fig. 4: Qualitative results in 3D dense captioning task from Scan2Cap [7] and our method. We underline the inaccurate words and mark the spatially discriminative phrases in bold. Our method qualitatively outperforms Scan2Cap in producing better object bounding boxes and more discriminative descriptions.

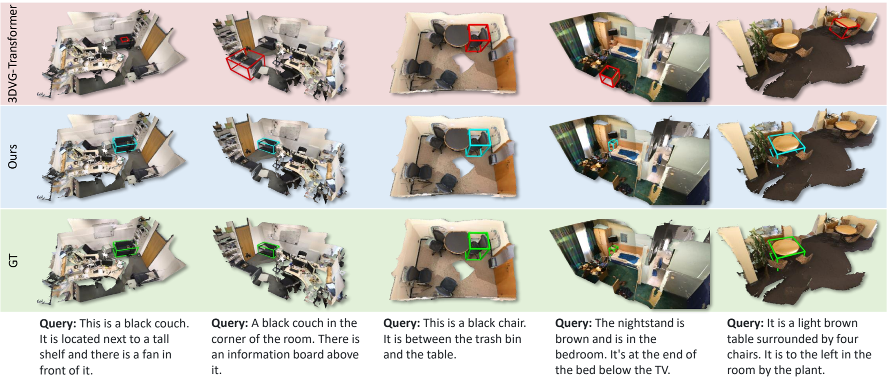
from the same class ("Multiple" case). Our end-to-end speaker-listener (last row) outperforms all previous method including 3DVG-Trans.

4.5 Qualitative Analysis

3D dense captioning Fig. 4 compares our results with object captions from Scan2Cap [7]. Descriptions generated by Scan2Cap cannot uniquely identify the target object in the input scenes (see the yellow block on the bottom right). Also, Scan2Cap produces inaccurate object bounding boxes, which affects the quality of object captions (see the yellow block on the top left). Compared to captions from Scan2Cap, our method produces more discriminative object captions that specifies more spatial relations (see bolded phrases in the blue blocks).
3D visual grounding Fig. 5 compares our results with 3DVG-Transformer [60]. Though 3DVG-Transformer is able to pick the correct object, it suffers from poor object detections and is constrained by the performance of the VoteNet-based detection backbone (see the first column). Our method is capable of selecting the queried objects while also predicting more accurate object bounding boxes.

4.6 Analysis and Ablation Studies

Does better detection backbone help? From Tab. 1, we see that using a better detector can significant improve performance. We further examine the ef-



Query: This is a black couch. It is located next to a tall shelf and there is a fan in front of it.

Query: A black couch in the corner of the room. There is an information board above it.

Query: This is a black chair. It is between the trash bin and the table.

Query: The nightstand is brown and is in the bedroom. It's at the end of the bed below the TV.

Query: It is a light brown table surrounded by four chairs. It is to the left in the room by the plant.

Fig. 5: 3D visual grounding results using 3DVG-Transformer [60] and our method. 3DVG-Transformer fails to accurately predict object bounding boxes, while our method produces accurate bounding boxes and correctly distinguishes target objects from distractors.

fect of using different detection backbones (VoteNet and PointGroup) compared to GT bounding boxes in Tab. 3. For each detection backbone, we use four variants of our method: the models trained without the joint speaker-listener architecture, and the speaker-listener architecture trained with CIDEr reward, listener reward and extra ScanNet data. The results with GT boxes show the effectiveness of our speaker-listener architecture, when detections are perfect. The large improvement from VoteNet [41] to PointGroup [25] show the importance of a better detection backbone. The gap between GT and VoteNet/PointGroup shows there is room for further improvement.

Are the generated descriptions more discriminative? To check whether the speaker-listener architecture generates more discriminative descriptions, we conduct an automatic evaluation via a reverse task. In this task, we feed the generated descriptions and GT bounding boxes into a pretrained neural listener model similar to Zhao et al. [60]. The predicted visual grounding results are evaluated in the same way as in our 3D visual grounding experiments. Higher grounding accuracy indicates better discrimination, especially in the "Multiple" case. Results (Tab. 4) show that our speaker-listener architecture generates more discriminative descriptions compared to Scan2Cap [7]. The discrimination is further improved when training with extra ScanNet data. To disentangle the affect of imperfectly predicted bounding boxes, we also train and evaluate our method with GT boxes (see last two rows in Tab. 4). We see that our semi-supervised speaker-listener architecture generates more discriminative descriptions.

Does the listener help with captioning? The third to the sixth rows in Tab. 1 measure the benefit of training the speaker together with the listener (Ours (CIDEr+loc.) and Ours (CIDEr+loc.+lobjcls.)) rather than training the speaker alone (Ours (CIDEr)). Training with the listener improves all captioning metrics. Also, training jointly with an unfrozen listener (Ours (CIDEr+loc.)

Table 3: Quantitative results on object detection, dense captioning and visual grounding in RGB-D scans. We train our method using different detection backbones as well as the ground truth bounding boxes. Our method trained with CIDEr and listener reward as well as the additional data outperforms the pretrained speaker and listener models.

							Unique	Multiple	Overall
Method	Detection	mAP@0.5	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	Acc@0.5IoU	Acc@0.5IoU	Acc@0.5IoU
Ours (MLE)	GT	100.00	71.41	42.95	29.67	64.93	88.45	36.46	46.03
Ours (CIDEr)	GT	100.00	94.80	47.92	30.80	**66.34**	-	-	-
Ours (CIDEr+lis.)	GT	100.00	95.62	47.65	**30.93**	66.31	89.76	36.85	47.14
Ours (CIDEr+lis.+extra)	GT	100.00	**96.31**	**48.20**	30.80	66.10	**89.86**	**40.66**	**48.17**
Ours (MLE)	VoteNet	32.21	39.08	23.32	21.97	**44.78**	56.41	21.11	27.95
Ours (CIDEr)	VoteNet	37.66	46.88	25.96	22.10	44.69	-	-	-
Ours (CIDEr+lis.)	VoteNet	38.03	47.32	24.76	21.66	43.62	57.90	20.73	28.03
Ours (CIDEr+lis.+extra)	VoteNet	**38.82**	**48.38**	**26.09**	**22.15**	44.74	**58.40**	**21.66**	**29.25**
Ours (MLE)	PointGroup	47.19	46.07	30.29	24.35	51.67	70.35	27.11	35.58
Ours (CIDEr)	PointGroup	52.44	57.88	32.64	24.86	52.26	-	-	-
Ours (CIDEr+lis.)	PointGroup	52.58	61.50	35.05	25.48	53.31	71.04	27.40	35.62
Ours (CIDEr+lis.+extra)	PointGroup	**53.95**	**62.64**	**35.68**	**25.72**	**53.90**	**72.04**	**30.05**	**37.87**

Table 4: We automatically evaluate the discriminability of the generated object descriptions. A pretrained neural listener similar to Zhao et al. [60] is fed with the GT object features and the descriptions generated by Scan2Cap [7] as well as our method. Higher grounding accuracy indicates better discriminability, especially in the "multiple" case. To alleviate noisy detections, the evaluation results on the descriptions generated from the GT object features are also presented. Our method generates more discriminative descriptions compared to Scan2Cap.

		Unique	Multiple	Overall
	detection	Acc@0.5IoU	Acc@0.5IoU	Acc@0.5IoU
Scan2Cap [7]	VN [41]	80.52	29.95	39.08
Ours (w/ CIDEr & lis.)	PG [25]	81.16	30.22	41.62
Ours (w/ CIDEr & lis. & extra)	PG [25]	**81.27**	**30.33**	**41.73**
Ours (w/ CIDEr & lis.)	GT	89.76	38.53	48.07
Ours (w/ CIDEr & lis. & extra)	GT	**90.29**	**40.66**	**49.71**

leads to a better performance when compared with the variant with a pretrained and frozen listener (Ours (CIDEr+fixed loc.), which is similar to Achlioptas et al. [2]. Additionally, as the detector is not only fine-tuned with the speaker but also with the listener, the additional supervision from the listener helps with the detection performance as well.

To analyze the quality of the generated object captions, we asked 5 students to perform a fine-grained manual analysis of the captions. Each student was presented with a batch of 100 randomly selected object captions with associated objects highlighted in the 3D scene. The student are then asked to indicate if the respective aspects were included and correctly described. The manual analysis results in Tab. 5 shows that our method generates more accurate descriptions compared to Scan2Cap. In particular, training with the listener and extra ScanNet data produces more accurate spatial relations in the descriptions. The results

Table 5: Manual analysis of captions generated by Scan2Cap [7] and variants of our method. We measure accuracy in three different aspects: object categories, appearance attributes and spatial relations. Our method generates more accurate descriptions in all aspects, especially for describing spatial relations.

	Acc (Category)	Acc (Attribute)	Acc (Relation)
Scan2Cap [7]	84.10	64.21	69.00
Ours (MLE)	88.00 (+3.84)	74.73 (+10.53)	69.00 (+0.00)
Ours (CIDEr)	88.89 (+4.73)	75.00 (+10.79)	68.00 (-1.00)
Ours (CIDEr+lis.)	90.91 (+6.75)	77.38 (+13.17)	75.00 (+6.00)
Ours (CIDEr+lis.+extra)	92.93 **(+8.77)**	80.95 **(+16.74)**	78.57 **(+9.57)**

of fine-grained manual analysis complements the automatic captioning evaluation metric. While metrics such as CIDEr captures the overall similarity of the generated sentences against the references, the accuracies in Tab. 5 measures the correctness of the decomposed visual attributes.

Does the speaker help with grounding? Tab. 2 compares grounding results between a pretrained listener (Ours w/o fine-tuning) and a fine-tuned speaker-listener model (Ours). Although the grounding performance drops in the “Unique” subset, the improvements in “Multiple” suggests better discriminability in tougher and ambiguous scenarios.

5 Conclusion

We present D^{3}Net, an end-to-end speaker-listener architecture that can **d**etect, **d**escribe and **d**iscriminate. Specifically, the speaker iteratively generates descriptive tokens given the object proposals detected by the detector, while the listener discriminates the object proposals in the scene with the generated captions. The self-discriminative property of D^{3}Net also enables semi-supervised training on ScanNet data without the annotated descriptions. Our method outperforms the previous SOTA methods in both tasks on ScanRefer, surpassing the previous SOTA 3D dense captioning method by a significant margin. Our architecture can serve as an initial step towards leveraging unannotated 3D data for language and 3D vision. Overall, we hope that our work will encourage more future research in 3D vision and language.

Acknowledgements

This work is funded by Google (AugmentedPerception), the ERC Starting Grant Scan2CAD (804724), and a Google Faculty Award. We would also like to thank the support of the TUM-IAS Rudolf Mößbauer and Hans Fischer Fellowships (Focus Group Visual Computing), as well as the the German Research Foundation (DFG) under the Grant *Making Machine Learning on Static and Dynamic 3D Data Practical*. This work is also supported in part by the Canada CIFAR AI Chair program and an NSERC Discovery Grant.

References

1. Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In: European Conference on Computer Vision, pp. 422–440, Springer (2020) 2, 3, 8
2. Achlioptas, P., Fan, J., Hawkins, R., Goodman, N., Guibas, L.J.: ShapeGlot: Learning language for shape differentiation. In: Proceedings of the IEEE international conference on computer vision (2019) 3, 4, 13
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6077–6086 (2018) 3
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020) 1
5. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: Proceedings of the International Conference on 3D Vision, pp. 667–676, IEEE (2017) 1
6. Chen, D.Z., Chang, A.X., Nießner, M.: ScanRefer: 3D object localization in RGB-D scans using natural language. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16, pp. 202–221, Springer (2020) 1, 2, 3, 5, 6, 8, 10
7. Chen, D.Z., Gholami, A., Nießner, M., Chang, A.X.: Scan2Cap: Context-aware dense captioning in RGB-D scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3193–3203 (2021) 1, 2, 3, 5, 7, 9, 11, 12, 13, 14
8. Chen, K., Choy, C.B., Savva, M., Chang, A.X., Funkhouser, T., Savarese, S.: Text2shape: Generating shapes from natural language by learning joint embeddings. In: Asian Conference on Computer Vision, pp. 100–116, Springer (2018) 3
9. Choy, C., Gwak, J., Savarese, S.: 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3075–3084 (2019) 9
10. Cole, P., Morgan, J.L.: Syntax and semantics. volume 3: Speech acts. Tijdschrift Voor Filosofie **39**(3) (1977) 4
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5828–5839 (2017) 1, 3, 8
12. Dai, A., Nießner, M.: 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 452–468 (2018) 5
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 1
14. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2625–2634 (2015) 3

15. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. In: Advances in Neural Information Processing Systems (2018) 4
16. Golland, D., Liang, P., Klein, D.: A game-theoretic approach to generating spatial descriptions. In: Proceedings of the 2010 conference on empirical methods in natural language processing, pp. 410–419 (2010) 4
17. Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T.: Generating visual explanations. In: European conference on computer vision, pp. 3–19, Springer (2016) 6
18. Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4421–4430 (2019) 1
19. Hou, J., Dai, A., Nießner, M.: RevealNet: Seeing behind objects in RGB-D scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2098–2107 (2020) 1
20. Hu, R., Rohrbach, M., Darrell, T.: Segmentation from natural language expressions. In: European Conference on Computer Vision, pp. 108–124, Springer (2016) 4
21. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4555–4564 (2016) 4
22. Hua, B.S., Pham, Q.H., Nguyen, D.T., Tran, M.K., Yu, L.F., Yeung, S.K.: Scenenn: A scene meshes dataset with annotations. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 92–101, IEEE (2016) 1
23. Huang, P.H., Lee, H.H., Chen, H.T., Liu, T.L.: Text-guided graph neural networks for referring 3D instance segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021) 2, 10
24. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016) 7
25. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: PointGroup: Dual-set point grouping for 3D instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4867–4876 (2020) 3, 5, 6, 9, 12, 13
26. Jiang, W., Ma, L., Jiang, Y.G., Liu, W., Zhang, T.: Recurrent fusion network for image captioning. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 499–515 (2018) 3
27. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4565–4574 (2016) 4
28. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3128–3137 (2015) 3
29. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 787–798 (2014) 4
30. Kim, D.J., Choi, J., Oh, T.H., Kweon, I.S.: Dense relational captioning: Triple-stream networks for relationship-based captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6271–6280 (2019) 4

31. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
32. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, pp. 740–755, Springer (2014) 2
33. Liu, X., Li, H., Shao, J., Chen, D., Wang, X.: Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 338–354 (2018) 2, 3, 6, 7
34. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019) 1
35. Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 375–383 (2017) 3
36. Luo, R., Price, B., Cohen, S., Shakhnarovich, G.: Discriminability objective for training descriptive captions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6964–6974 (2018) 2, 3, 6, 7
37. Luo, R., Shakhnarovich, G.: Comprehension-guided referring expressions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7102–7111 (2017) 4
38. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 11–20 (2016) 4
39. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035, Curran Associates, Inc. (2019) 9
40. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision, pp. 2641–2649 (2015) 4
41. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep Hough voting for 3D object detection in point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9277–9286 (2019) 1, 3, 12, 13
42. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732 (2015) 6
43. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7008–7024 (2017) 3, 6, 7
44. Roh, J., Desingh, K., Farhadi, A., Fox, D.: LanguageRefer: Spatial-language model for 3D visual grounding. In: Proceedings of the Conference on Robot Learning (2021) 3
45. Sidorov, O., Hu, R., Rohrbach, M., Singh, A.: Textcaps: a dataset for image captioning with reading comprehension. In: European Conference on Computer Vision, pp. 742–758, Springer (2020) 3
46. Song, S., Lichtenberg, S.P., Xiao, J.: SUN RGB-D: A RGB-D scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 567–576 (2015) 1

47. Thomason, J., Shridhar, M., Bisk, Y., Paxton, C., Zettlemoyer, L.: Language grounding with 3D objects. In: Proceedings of the Conference on Robot Learning (2021) 3
48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008 (2017) 1
49. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4566–4575 (2015) 4
50. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3156–3164 (2015) 3
51. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3), 229–256 (1992) 4, 5
52. Wu, X., Averbuch-Elor, H., Sun, J., Snavely, N.: Towers of babel: Combining images, language, and 3D geometry for learning multimodal vision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021) 3
53. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning, pp. 2048–2057 (2015) 3
54. Yang, L., Tang, K., Yang, J., Li, L.J.: Dense captioning with joint inference and visual context. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2193–2202 (2017) 4
55. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems **32** (2019) 1
56. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: MattNet: Modular attention network for referring expression comprehension. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1307–1315 (2018) 4
57. Yu, L., Tan, H., Bansal, M., Berg, T.L.: A joint speaker-listener-reinforcer model for referring expressions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7282–7290 (2017) 4, 6
58. Yuan, Z., Yan, X., Liao, Y., Guo, Y., Li, G., Li, Z., Cui, S.: X-Trans2Cap: Cross-modal knowledge transfer using transformer for 3D dense captioning (2022), arXiv:2203.00843 2, 3, 9
59. Yuan, Z., Yan, X., Liao, Y., Zhang, R., Wang, S., Li, Z., Cui, S.: 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, pp. 1791–1800 (2021) 1, 2, 3, 10
60. Zhao, L., Cai, D., Sheng, L., Xu, D.: 3DVG-Transformer: Relation modeling for visual grounding on point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2928–2937 (2021) 1, 2, 3, 5, 10, 11, 12, 13

SPRINGER NATURE LICENSE TERMS AND CONDITIONS

May 11, 2023

This Agreement between Mr. Zhenyu Chen ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5545810594583
License date	May 11, 2023
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	D ^{3} 3 Net: A Unified Speaker-Listener Architecture for 3D Dense Captioning and Visual Grounding
Licensed Content Author	Dave Zhenyu Chen, Qirui Wu, Matthias Nießner et al
Licensed Content Date	Jan 1, 2022
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	30 - 99
Author of this Springer Nature content	yes
Title	Grounding Natural Language to 3D Scenes
Institution name	Technical University of Munich
Expected presentation date	Dec 2023
Requestor Location	Mr. Zhenyu Chen Euckenstr. 27 Munich, Bayern 81369 Germany Attn: Mr. Zhenyu Chen
Billing Type	Invoice
Billing Address	Mr. Zhenyu Chen Euckenstr. 27 Munich, Germany 81369 Attn: Mr. Zhenyu Chen
Total	**0.00 EUR**
Terms and Conditions	

Springer Nature Customer Service Centre GmbH Terms and Conditions

The following terms and conditions ("Terms and Conditions") together with the terms specified in your [RightsLink] constitute the License ("License") between you as Licensee and Springer Nature Customer Service Centre GmbH as Licensor. By clicking 'accept' and completing the transaction for your use of the material ("Licensed Material"), you confirm your acceptance of and obligation to be bound by these Terms and Conditions.

1. Grant and Scope of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, non-sublicensable, revocable, world-wide License to reproduce, distribute, communicate to the public, make available, broadcast, electronically transmit or

create derivative works using the Licensed Material for the purpose(s) specified in your RightsLink Licence Details only. Licenses are granted for the specific use requested in the order and for no other use, subject to these Terms and Conditions. You acknowledge and agree that the rights granted to you under this License do not include the right to modify, edit, translate, include in collective works, or create derivative works of the Licensed Material in whole or in part unless expressly stated in your RightsLink Licence Details. You may use the Licensed Material only as permitted under this Agreement and will not reproduce, distribute, display, perform, or otherwise use or exploit any Licensed Material in any way, in whole or in part, except as expressly permitted by this License.

1. 2. You may only use the Licensed Content in the manner and to the extent permitted by these Terms and Conditions, by your RightsLink Licence Details and by any applicable laws.

1. 3. A separate license may be required for any additional use of the Licensed Material, e.g. where a license has been purchased for print use only, separate permission must be obtained for electronic re-use. Similarly, a License is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the License.

1. 4. Any content within the Licensed Material that is owned by third parties is expressly excluded from the License.

1. 5. Rights for additional reuses such as custom editions, computer/mobile applications, film or TV reuses and/or any other derivative rights requests require additional permission and may be subject to an additional fee. Please apply to journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

2. Reservation of Rights

Licensor reserves all rights not expressly granted to you under this License. You acknowledge and agree that nothing in this License limits or restricts Licensor's rights in or use of the Licensed Material in any way. Neither this License, nor any act, omission, or statement by Licensor or you, conveys any ownership right to you in any Licensed Material, or to any element or portion thereof. As between Licensor and you, Licensor owns and retains all right, title, and interest in and to the Licensed Material subject to the license granted in Section 1.1. Your permission to use the Licensed Material is expressly conditioned on you not impairing Licensor's or the applicable copyright owner's rights in the Licensed Material in any way.

3. Restrictions on use

3. 1. Minor editing privileges are allowed for adaptations for stylistic purposes or formatting purposes provided such alterations do not alter the original meaning or intention of the Licensed Material and the new figure(s) are still accurate and representative of the Licensed Material. Any other changes including but not limited to, cropping, adapting, and/or omitting material that affect the meaning, intention or moral rights of the author(s) are strictly prohibited.

3. 2. You must not use any Licensed Material as part of any design or trademark.

3. 3. Licensed Material may be used in Open Access Publications (OAP), but any such reuse must include a clear acknowledgment of this permission visible at the same time as the figures/tables/illustration or abstract and which must indicate that the Licensed Material is not part of the governing OA license but has been reproduced with permission. This may be indicated according to any standard referencing system but must include at a minimum 'Book/Journal title, Author, Journal Name (if applicable), Volume (if applicable), Publisher, Year, reproduced with permission from SNCSC'.

4. STM Permission Guidelines

4. 1. An alternative scope of license may apply to signatories of the STM Permissions Guidelines ("STM PG") as amended from time to time and made available at https://www.stm-assoc.org/intellectual-property/permissions/permissions-guidelines/.

4. 2. For content reuse requests that qualify for permission under the STM PG, and which may be updated from time to time, the STM PG supersede the terms and conditions contained in this License.

4. 3. If a License has been granted under the STM PG, but the STM PG no longer apply at the time of publication, further permission must be sought from the Rightsholder. Contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

5. Duration of License

5. 1. Unless otherwise indicated on your License, a License is valid from the date of purchase ("License Date") until the end of the relevant period in the below table:

Reuse in a medical communications project	Reuse up to distribution or time period indicated in License
Reuse in a dissertation/thesis	Lifetime of thesis
Reuse in a journal/magazine	Lifetime of journal/magazine
Reuse in a book/textbook	Lifetime of edition
Reuse on a website	1 year unless otherwise specified in the License
Reuse in a presentation/slide kit/poster	Lifetime of presentation/slide kit/poster. Note: publication whether electronic or in print of presentation/slide kit/poster may require further permission.
Reuse in conference proceedings	Lifetime of conference proceedings
Reuse in an annual report	Lifetime of annual report
Reuse in training/CME materials	Reuse up to distribution or time period indicated in License
Reuse in newsmedia	Lifetime of newsmedia
Reuse in coursepack/classroom materials	Reuse up to distribution and/or time period indicated in license

6. Acknowledgement

6. 1. The Licensor's permission must be acknowledged next to the Licensed Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract and must be hyperlinked to the journal/book's homepage.

6. 2. Acknowledgement may be provided according to any standard referencing system and at a minimum should include "Author, Article/Book Title, Journal name/Book imprint, volume, page number, year, Springer Nature".

7. Reuse in a dissertation or thesis

7. 1. Where 'reuse in a dissertation/thesis' has been selected, the following terms apply: Print rights of the Version of Record are provided for; electronic rights for use only on institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/) and only up to what is required by the awarding institution.

7. 2. For theses published under an ISBN or ISSN, separate permission is required. Please contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

7. 3. Authors must properly cite the published manuscript in their thesis according to current citation standards and include the following acknowledgement: '*Reproduced with permission from Springer Nature*'.

8. License Fee

You must pay the fee set forth in the License Agreement (the "License Fees"). All amounts payable by you under this License are exclusive of any sales, use, withholding, value added or similar taxes, government fees or levies or other assessments. Collection and/or remittance of such taxes to the relevant tax authority shall be the responsibility of the party who has the legal obligation to do so.

9. Warranty

9. 1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. **You are solely responsible for ensuring that the material you wish to license is original to the Licensor and does not carry the copyright of another entity or third party (as credited in the published version).** If the credit line on any part of the Licensed Material indicates that it was reprinted or adapted with permission from another source, then you should seek additional permission from that source to reuse the material.

9. 2. EXCEPT FOR THE EXPRESS WARRANTY STATED HEREIN AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR PROVIDES THE LICENSED MATERIAL "AS IS" AND MAKES NO OTHER REPRESENTATION OR WARRANTY. LICENSOR EXPRESSLY DISCLAIMS ANY LIABILITY FOR ANY CLAIM ARISING FROM OR OUT OF THE CONTENT, INCLUDING BUT NOT LIMITED TO ANY ERRORS, INACCURACIES, OMISSIONS, OR DEFECTS CONTAINED THEREIN, AND ANY IMPLIED OR EXPRESS

WARRANTY AS TO MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR EXEMPLARY DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE LICENSED MATERIAL REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION APPLIES NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

10. Termination and Cancellation

10. 1. The License and all rights granted hereunder will continue until the end of the applicable period shown in Clause 5.1 above. Thereafter, this license will be terminated and all rights granted hereunder will cease.

10. 2. Licensor reserves the right to terminate the License in the event that payment is not received in full or if you breach the terms of this License.

11. General

11. 1. The License and the rights and obligations of the parties hereto shall be construed, interpreted and determined in accordance with the laws of the Federal Republic of Germany without reference to the stipulations of the CISG (United Nations Convention on Contracts for the International Sale of Goods) or to Germany´s choice-of-law principle.

11. 2. The parties acknowledge and agree that any controversies and disputes arising out of this License shall be decided exclusively by the courts of or having jurisdiction for Heidelberg, Germany, as far as legally permissible.

11. 3. This License is solely for Licensor's and Licensee's benefit. It is not for the benefit of any other person or entity.

Questions? For questions on Copyright Clearance Center accounts or website issues please contact springernaturesupport@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777. For questions on Springer Nature licensing please visit https://www.springernature.com/gp/partners/rights-permissions-third-party-distribution

Other Conditions:

Version 1.4 - Dec 2022

Questions? E-mail us at customercare@copyright.com.

Acronyms

1D, 2D, 3D, ...	N spatial dimentions.
Adam	Adaptive Moment Estimation.
AR	Augmented Reality.
BCE	Binary Cross Entropy.
BN	Batch Normalization.
CE	Cross Entropy.
CNN	Convolutional Neural Network.
CPU	Central Processing Unit.
GPU	Graphical Processing Unit.
GRU	Gated Recurrent Unit.
LiDAR	Light Detection and Ranging.
LN	Layer Normalization.
LSTM	Long Short-term Memory.
MLP	Multi-Layer Perception.
MSE	Mean Squared Error.
NMS	Non Maximum Suppression.
ReLU	Rectified Linear Unit.
RNN	Recurrent Neural Network.
SDF	Signed Distance Field.
SGD	Stochastic Gradient Descent.
SIFT	Scale-Invariant Feature Transform.
SOTA	State-of-the-art.
SVM	Support Vector Machine.
VR	Virtual Reality.

List of Tables

List of Figures