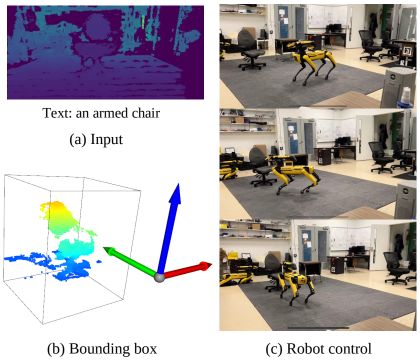
RobotFetchMeIt Final Report: Language-Guided 3D Object Detection

Rao Fu, Yiwen Chen, Zichuan Wang, Xinyu Liu
Department of Computer Science, Brown University
115 Waterman St, Providence, RI 02912
{rao_fu,yiwen_chen,zichuan_wang,xinyu_liu}@brown.edu
https://github.com/GuardianWang/RobotFetchMeIt

Abstract

We present a language-guided 3D object detection model that detects the 3D bounding box of an object that matches the text description. Different from other 3D object detection models that predict both 3D location and object class, our method uses a bounding box proposer to predict 3D bounding box candidates. Then, the shape generator will generate features that correspond to the shape description. We design a HyperNetwork as the bounding box scorer, which predicts the probability of an object within a bounding box matching the shape description. In this way, our model can perform language-guided fine-grained 3D object detection. We evaluated the performance of each module as well as the overall success rate through a Boston Dynamics Spot robot in the real world.



(a) Input

(b) Bounding box

(c) Robot control

Figure 1. A Spot robot detects and moves to the object that matches the text description. (a) Model input consists of a depth image and a text description. (b) Detected bounding box with an object that matches the text description. (c) Spot moves to the object.

1. Introduction

Research interest in Human-Robot Interaction (HRI) has increased recently in the academic community [3]. Robots can provide necessary assistance in our daily life. For example, assistive robots act as companions for the elderly and people with disabilities; unmanned vehicles can assist humans in doing scientific research and rescue tasks in the various environment; human-like robots [8] have been built for a better and more natural HRI experience. We believe the ability to accurately detect and classify objects is fundamental to various downstream tasks.

In this paper, we introduce a language-guided 3D object detection model. Compared with 2D detection networks, 3D data provides abundant spatial and semantic information. Typical 3D object detection models are hard to generalize to new classes or fine-grained classes because adding a new class requires expanding the dimension of the output layer and re-training the whole network. To solve this problem, we propose a language-guided shape generation module to extract the deep feature of text input. We also design a HyperNetwork as the bounding box scorer, which is to determine the probability of the point cloud within a bounding box matching the text description.

The input format of our model is simple. Our model only requires as input a depth image and a text description about the desired shape and class of an object. Our model doesn't require any RGB image, so it is robust against lighting conditions. Also, the texture of an object won't affect the performance of our model.

The key contributions of our work are as follows:

- We design a language-guided 3D object detection model that only takes in a depth image and a text description as input.
- We design a HyperNetwork as the bounding box scorer, which outputs the probability of an object matching the language description.
- We apply the model to Spot and perform real-world experiments to evaluate our model.

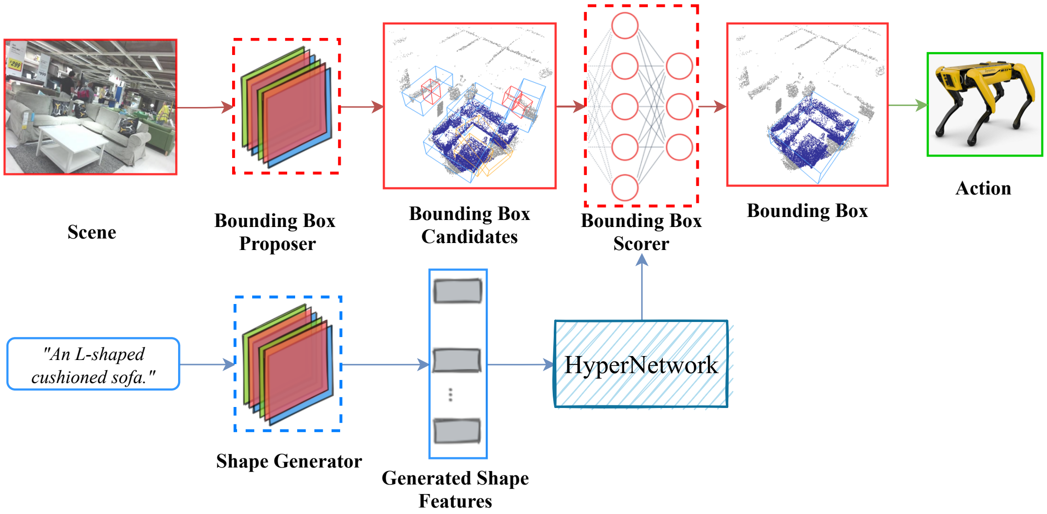


Figure 2. Model structure. Our model consists of a bounding box proposer, a shape generator, and a bounding box scorer. The bounding box proposer takes as input a depth image (for better visualization we use a color image) and outputs multiple candidate 3D bounding boxes. The shape generator generates features that correspond to the input shape description. We apply a HyperNetwork as the bounding box scorer, which predicts the probability of an object within a bounding box matching the shape description. Finally, our robot will move towards the target object.

2. Related Work

3D object detection. 3D object detection predicts oriented 3D bounding boxes from point clouds. PointNet [11] applies MLP to each point feature and uses max pooling to get the global feature of a point cloud. PointNet++ [12] proposes to first hierarchically sample and group points before applying PointNet. Built on top of PointNet++, VoteNet [10] puts forward deep hough voting to cluster bounding box centroids. ImVoteNet [9] boosters 3D object detection by integrating 2D RGB image votes. Group-Free 3D [7] replaces hand-crafted grouping scheme with the attention technique used by Transformers. Considering simplicity, performance and training speed, we use VoteNet as our bounding box proposer.

Language-guided shape generation. Previous work falls into two categories: sequence-to-sequence generation and CLIP-conditional generative models. For sequence-to-sequence generation, previous works [1, 2] use sequence-to-sequence models to match the text and shape modalities. They construct text-shape datasets of tens of thousand of text-shape pairs and then train sequence-to-sequence models to match the text features with the corresponding shape features. CLIP-Forge [13] is a representative work of CLIP-conditional generative models. It uses a normalizing flow model to map the distribution of the text-image joint-embedding space to the shape embedding space and shows some successful cases of zero-shot text-conditioned shape generation. In this paper, we use a CLIP-conditional generative model.

HyperNetworks. HyperNetworks proposes to use one network to generate the weights for another network [4]. This method is especially useful for guiding the result of an existing network using another feature as input. Sitzmann et al [14]. proposed to use such structure in the scenario of 3D-structured neural scene representations. By using shape class as an input feature for the HyperNetwork module, they generate the weight for the light-field network, thus inherently ensuring the multi-view consistency of the reconstructed representation across the same shape class. This paper is inspired by such an idea and uses text-shape as an input to the hyper module to determine the weight of the classification network applied to extracted point clouds.

3. Approach

The design of our model is illustrated in Fig 2. The model consists of 3 major modules: the 3D object detection module for bounding box proposal, the language-guided shape generation module for text and shape feature extraction, and the HyperNetwork for bounding box scoring.

3.1. 3D object detection

We use VoteNet [10] as our 3D bounding box proposer. VoteNet takes as input N ($N = 20000$) 3D points and predicts the parameters of K oriented bounding boxes. VoteNet is composed of three phases: feature learning, voting, and object proposal. First, PointNet++ [12] is applied to learn deep features of the point cloud and M seed points will be selected. Then, each point $p \in \mathbb{R}^3$ will vote the bounding box centroid $c \in \mathbb{R}^3$ by learning an offset $\Delta p \in \mathbb{R}^3$, where $c = p + \Delta p$. Finally, the M voted centroids will be clustered into K groups, and 7 parameters (center, radius, rotation along z-axis) of an oriented bounding box will be predicted for each group.

3.2. Language-guided shape generation

We use a CLIP-Conditioned shape generation model similar to CLIP-Forge [13]. We add another mapping network to make sure the consistency between the CLIP-text features and CLIP-images features, which will further improve the text-shape correspondences. The inference pipeline comprises three components: (1) *CLIP Text Encoder* and *Text Mapping network* that extract domain-specific text feature t'; (2) *Flow Model* that generates generates shape features $\{e_i\}$ given domain-specific text feature t'; (3) **Shape Decoder** that reconstruct 3D shapes $\{S_i\}$ from shape features $\{e_i\}$. The three components are trained separately during the training time. The Mapping Network has a Text Mapping Network $M_T()$ and an Image Mapping Network $M_I()$. They may the CLIP text feature t and CLIP image feature i to domain-specific text features t' and image features i' respectively:$M_T(t) = t', M_I(i) = i'$. We propose to use contrastive learning loss to train the mapping networks: $L_{contrast} = contrast(i', t')$. For the training details of the *Flow Model* and the **Shape Decoder**, please refer to CLIP-Forge [13].

3.3. HyperNetworks

To decide whether the detected point cloud cluster is the shape we are looking for given the text input, we use a simple PointNet [11] classification network. In PointNet, we classify each point cloud cluster generated in Sec 3.1 as either "belongs to the target class" or "does not belong to the target class". To make this classification network analyze points differently according to the text input, we use the input text to produce the weight used in PointNet.

Input text is encoded to a feature vector **z** by the text-encoder described in Sec 3.1. **z** is fed to the hyper module as an input vector. The HyperNetwork learns to analyze input text features and outputs weights that can be used by the PointNet classification network by jointly training the two.

3.4. Spot motion plan

We use the Spot SDK[1] for motion control. In SUN-RGBD [15], right-hand side is $+x$ and inward is $+y$. In the robot coordinates, when docking, Spot faces $+x$ and the left-hand side is $+y$. Because the depth cameras on both sides of the Spot have a much better quality than the front depth camera, we use the right camera to capture the depth map of the scene. Once we get the 3D bounding box parameters, we will first transform them into Spot coordinates, then we will leave 0.5m as a safety distance. Next, Spot will move to the target location.

4. Experiments

4.1. 3D object detection

To choose the most suitable 3D detection backbone, we trained VoteNet [10], ImVoteNet [9] and Group-Free 3D [7] on the SUN RGB-D [15] training set and tested their performance on the SUN RGB-D validation set. Results are shown in Tab 1 and Tab 2. Furthermore, in Tab 3 we compared the scalability and usability of these models. Because we don't need the classification result of the bounding box proposer, recall is more important than precision. Also, VoteNet takes much less time to train than ImVoteNet. Most importantly, VoteNet doesn't require RGB images as input, and the Spot robot we are using doesn't have an RGB camera on its right-hand side body. Considering these factors, VoteNet is more suitable for our application.

Table 1. 3D object detection results on SUN RGB-D [15] validation set. Evaluation metric is mAP@0.25.

method	bathtub	bed	bookshelf	chair	desk	sofa	table	toilet	dresser	night stand	mAP
VoteNet [10]	71.3	82.4	30.0	74.5	24.2	63.8	49.5	89.0	26.3	58.2	56.9
ImVoteNet [9]	74.3	87.3	**43.3**	76.5	29.0	**72.0**	51.5	90.5	**43.5**	**70.5**	**63.8**
Group-Free 3D [7]	**80.0**	**87.8**	32.5	**79.4**	**32.6**	70.0	**53.8**	**91.1**	36.0	66.7	63.0

Table 2. 3D object detection results on SUN RGB-D [15] validation set. Evaluation metric is mAR@0.25.

method	bathtub	bed	bookshelf	chair	desk	sofa	table	toilet	dresser	night stand	mAR
VoteNet [10]	86.5	93.9	69.6	87.2	**79.7**	88.6	84.3	**98.0**	81.5	87.0	85.6
ImVoteNet [9]	**88.5**	**95.4**	**78.3**	**87.8**	79.2	**92.5**	**86.3**	97.4	**86.1**	**91.7**	**88.3**

Table 3. Comparison between detection backbones.

method	RGB	size	training time	inference speed
VoteNet [10]	✗	**12MB**	**10hrs**	100fps
ImVoteNet [9]	✓	22MB	12hrs	7fps
Group-Free 3D [7]	✓	14MB	125hrs	**143fps**

4.2. Language model

We use the recall precision(**Shape RP@N**) metric in the image embedding space and shape embedding space to evaluate the consistency between the text description and the generated shapes. Our baseline method is CLIP-Forge [13]. As is shown in Tab 4, our method with the mapping network outperforms the baseline method.

4.3. Motion control

The demonstration of Spot motion control is shown in Fig 3. As we can see, at the ending positions, Spot is always facing the center of the object. Besides, there exists

[1] https://github.com/boston-dynamics/spot-sdk

Table 4. Quantitative comparison with other methods.

DataSet	Methods	Image		Shape		
		RP@1	RP@10	RP@1	RP@10	FID
Text2Shape [2]	Baseline	4.12	18.41	0.12	1.93	21.50
	Ours	**8.36**	**30.96**	**1.20**	**7.07**	**18.77**

some safety distance between Spot the object. The result indicates that our detection backbone can accurately predict the 3D bounding box. Also, the design of safety distance is important to prevent Spot from crashing into the object.

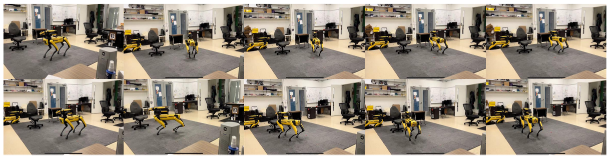


Figure 3. Motion control of Spot. Target location is inferred by 3D bounding box center and radius. Top: Spot goes to the desk on the right. Bottom: Spot goes to the chair on the left.

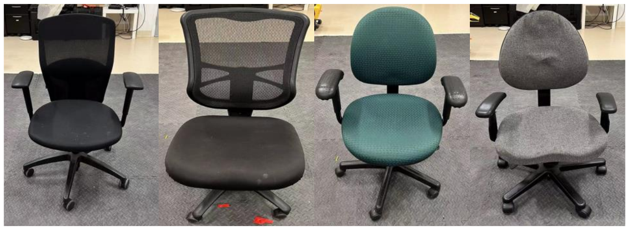
4.4. Real-world test

To test the generalization ability of our model, we conducted real-world tests on 4 different office chairs. In Fig 4 we show the office chairs we are using and corresponding suitable descriptions. In our test settings, Spot can see 2 chairs each time. One chair matches the text description and the other doesn't. For each description, we run the model 10 times. Each time the orientation and location of chairs are changed randomly. We also randomly change the chair that doesn't match the description. The precision and recall of each text are shown in Tab 5.

As the result indicates, our model performs the best on chairs with a cuboid back, but the worst on round backs. Our explanation for this phenomenon is that the round-back chairs we are using don't have a perfect round back. From the last two columns of Fig 4 we can see that some parts of the boundary are straight, which may lead the hypernetwork to misclassify such chairs.

Table 5. Real-world test on 4 chairs and 3 text descriptions.

text	precision	recall
a tall office swivel chair	0.471	0.600
a chair with a round back	0.077	0.100
a chair with a cuboid back	0.765	0.900



a tall office swivel chair; a chair with a cuboid back

a chair with a cuboid back

a chair with a round back

a chair with a round back

Figure 4. Office chairs and suitable text descriptions.

5. Conclusion and future work

In this work, we propose a language-guided 3D object detection model that only takes as input a depth image and a text description. We apply a HyperNetwork as the bounding box scorer, which outputs the probability of an object matching the language description.

One significant limitation is that our model cannot deal with color-related descriptions. We may need RGB images as input and train the shape generation model with color features. Another aspect to improve is scene exploration. If the target object is not in the current view, the robot will need to explore the scene with POMDP [5] algorithms.

6. Division of work

Rao Fu: Investigate 3D object detection. Train and improve Group-Free 3D for object detection. Implement language-conditioned shape feature generation. Implement object point clouds exaction from a point cloud scene using VoteNet. Canonicalize the extracted point cloud, which could be used in HyperNetwork. Design HyperNetwork.

Yiwen Chen: Implemented and trained HyperNetwork. Test shape detector performance on synthetic data and natural language input. Helped integrating separate modules into one workable application.

Zichuan Wang: Trained Votenet on 4 GPUs. Researched on a 3D simulation environment AI2Thor [6]. Integrated all sections into the final demo code, including 3D object detection, language-guided bounding box scorer, and robot control. Visualized point clouds with Open3D [16].

Xinyu Liu: Problem formulation and scoping: defined system input and output, decided the object detection model architecture based on results from other group members, defined simulated and real-robot training and test environments. Robot perception and control: implemented robot navigation. Reviewed decision-makers used in visual navigation literature (future work).

References

[1] Panos Achlioptas, Judy Fan, X.D. Robert Hawkins, D. Noah Goodman, and J. Leonidas Guibas. ShapeGlot: Learning language for shape differentiation. *CoRR*, abs/1905.02925, 2019. 2

[2] 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 *Asian conference on computer vision*, pages 100–116. Springer, 2018. 2, 4

[3] Michael A Goodrich and Alan C Schultz. *Human-robot interaction: a survey*. Now Publishers Inc, 2008. 1

[4] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 2

[5] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998. 4

[6] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 4

[7] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021. 2, 3

[8] Jaana Parviainen and Mark Coeckelbergh. The political choreography of the sophia robot: beyond robot rights and citizenship to political performances for the social robotics market. *AI & society*, 36(3):715–724, 2021. 1

[9] 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, 3

[10] 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/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 2, 3

[11] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017. 2, 3

[12] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[13] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *arXiv preprint arXiv:2110.02624*, 2021. 2, 3

[14] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[15] 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. 3

[16] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 4