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

Figure 1: We introduce the new 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 [46, 20] and natural language processing [50, 13] is a rich and active area of research. Specifically, there has been a lot of work on image captioning [52, 28, 54, 34, 2] and the related task of dense captioning [28, 27, 55, 58, 29, 32]. 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 [58]. 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 effectively 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 [49, 25, 12, 5]. Methods utilizing 3D volumetric grids have achieved impressive performance [22, 23, 31, 37, 15]. At the same time, methods operating on point clouds serve as an alternative and also achieve impressive results. For instance, Qi et al. [42] use a Hough voting scheme to aggregate points and generate object proposals while using a PointNet++ [44] backbone. Following this work, Qi et al. [43] 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 [52, 54, 14, 28, 34, 2, 26, 47]. Attention based captioning over grid regions [54, 34] and over detected objects [2, 35] 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, 57, 56] 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. [27] is more 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. [55] include the global image feature as context to the captioning input. Kim et al. [29] 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 gained great attention in image domain, such as image captioning [52, 54, 14, 28, 34, 2, 26, 47], dense captioning [27, 55, 29], text-to-image generation [45, 48, 18], visual grounding [24, 36, 59], 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 recent 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 one scan from ScanNet consisting of the geometry coordinates and additional point features capturing the visual appearance and the height from 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

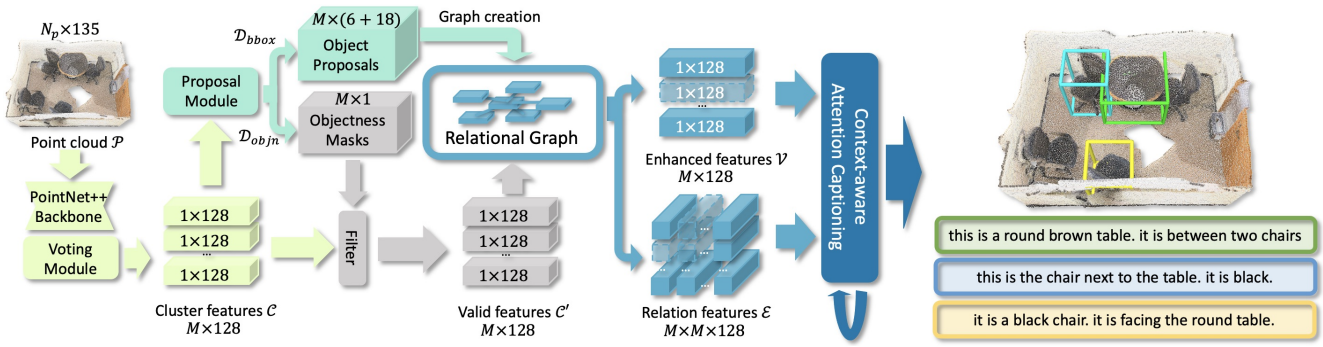


Figure 2: Scan2Cap takes as input a point cloud to generate the cluster features $\mathcal{C}$ for the proposal module, using a backbone following PointNet++ [44] and a voting module similar to Qi et al. [42]. The proposal module predicts the object proposals $\mathcal{D}_{\text{bbox}}$ as well as the 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 the each object proposal using the enhanced features and the relation features.

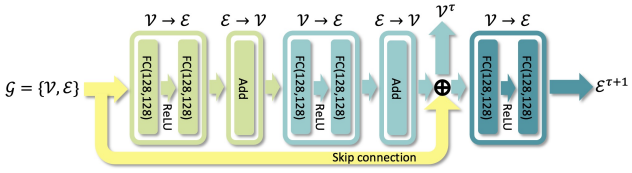
cloud as additional features. The image features are extracted using a pre-trained ENet [39]. Following Qi et al. [42], we also append the height of the point from the ground to the new point features. As a result, we represent the final point cloud data as $\mathcal{P} = \{(p_i, f_i)\} \in \mathcal{R}^{N_P \times 135}$, 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

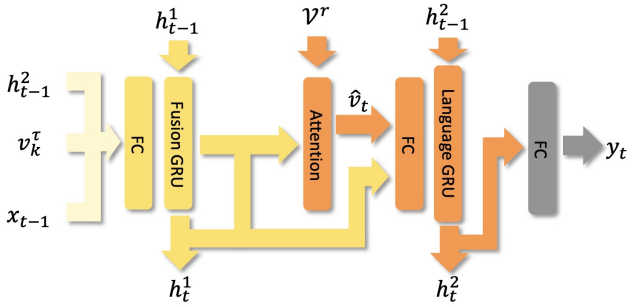
As the first step in our network, we detect all probable objects in the given point cloud with the back-projected multi-view image features discussed in 4.1. To construct our detection module, we adapt the PointNet++ [44] backbone and the voting module in VoteNet [42] to aggregate all object candidates 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



(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, 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 $\mathcal{V}^r$.

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:

$$\mathcal{V} \to \mathcal{E} : g^{\tau+1}_{i,j} = f^\tau([g^\tau_i, g^\tau_j - g^\tau_i]) \quad (1)$$

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 [41] 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 $\mathcal{V}^\tau = \{v_i^\tau\} \in \mathcal{R}^{M\times 128}$ 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 $\mathcal{V}^r = \{v_1^r, ..., v_k^\tau, ..., v_M^r\}$. Intuitively, the attention module will attend to the neighbor objects and their associated relations with the current object. We define the intermediate attention distribution $\alpha_t \in \mathcal{R}^{M\times 128}$ 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 $W_a \in \mathcal{R}^{128\times 1}$, $W_v \in \mathcal{R}^{128\times 128}$, $W_h \in \mathcal{R}^{512\times 128}$ are learnable parameters. $\mathbb{1}_h \in \mathcal{R}^{M\times 1}$ and $\mathbb{1}_a \in \mathcal{R}^{1\times 128}$ are identity matrices. Finally, the attention module outputs aggregated context vector $\hat{v}_t = \sum_{i=1}^{M} \mathcal{V}_i^r \odot \alpha_{ti}$ to represent the attended object and corresponding 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. [42] 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. [42]), 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. [42] 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 [54, 52, 28].

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 [40] and train end-to-end using ADAM [30] 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 [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 [51] and BLEU [38], 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 [51], BLEU-4 [38], METEOR [4] and ROUGE [33], 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.

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 [52]. In addition to [52], 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.

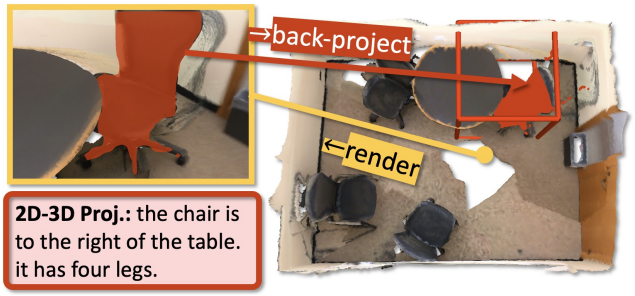


Figure 4: In 2D-3D Proj, we first generate a description for each detected object in a rendered viewpoint. Then we back-project the object mask to the 3D space to evaluate the caption with our proposed caption evaluation metric.

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* [42]** Similar to *OracleRetr3D*, but we use the features of the 3D bounding boxes detected using a pre-trained VoteNet [42].

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. [52]. 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

	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 [42]	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 [51], 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 2: Comparison of 3D dense captioning results obtained by ours 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.

scans using a pre-trained VoteNet [42], 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. [52].

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. 8, 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.

5.2. Qualitative Analysis

We see from Fig. 5 that the captions retrieved by OracleRetr2D hallucinates objects that are not there, while Oracle2Cap2D provides inaccurate captions that fails to capture correct local context. In contrast, the captions from Oracle2Cap3D is longer and capture relationships with the surrounding objects, such as the "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 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 randomly selected 100 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 in local environment. As demonstrated in Tab. 3, directly captioning objects in

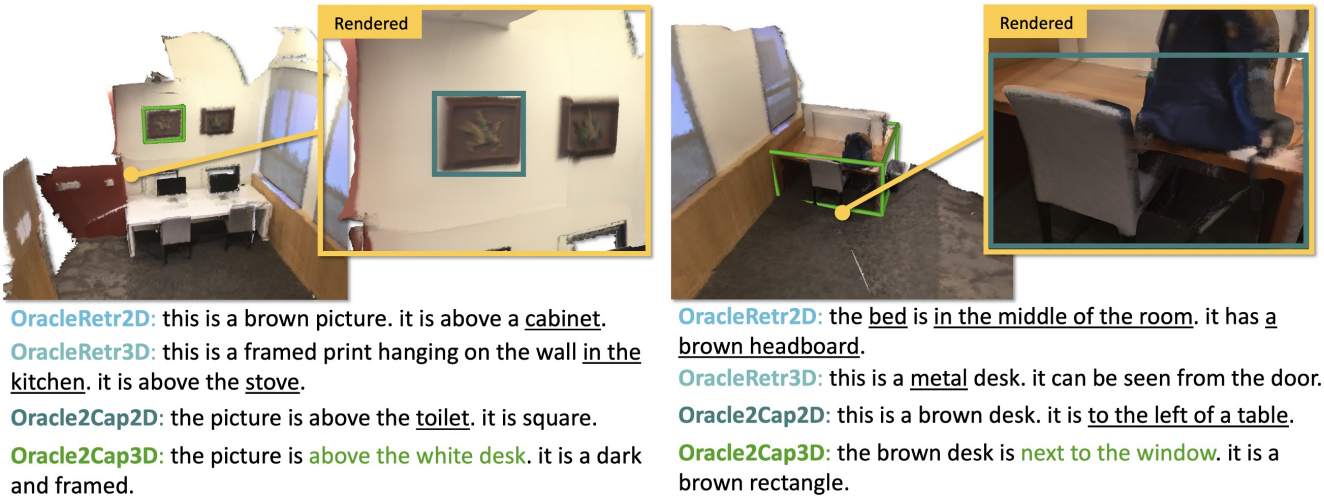


Figure 5: Qualitative results from skylines with GT input with inaccurate parts of the generated caption underscored. Captioning in 3D benefits from the richness of 3D context, while captioning with 2D information fails to capture the details of the local physical environment. Best viewed in color.

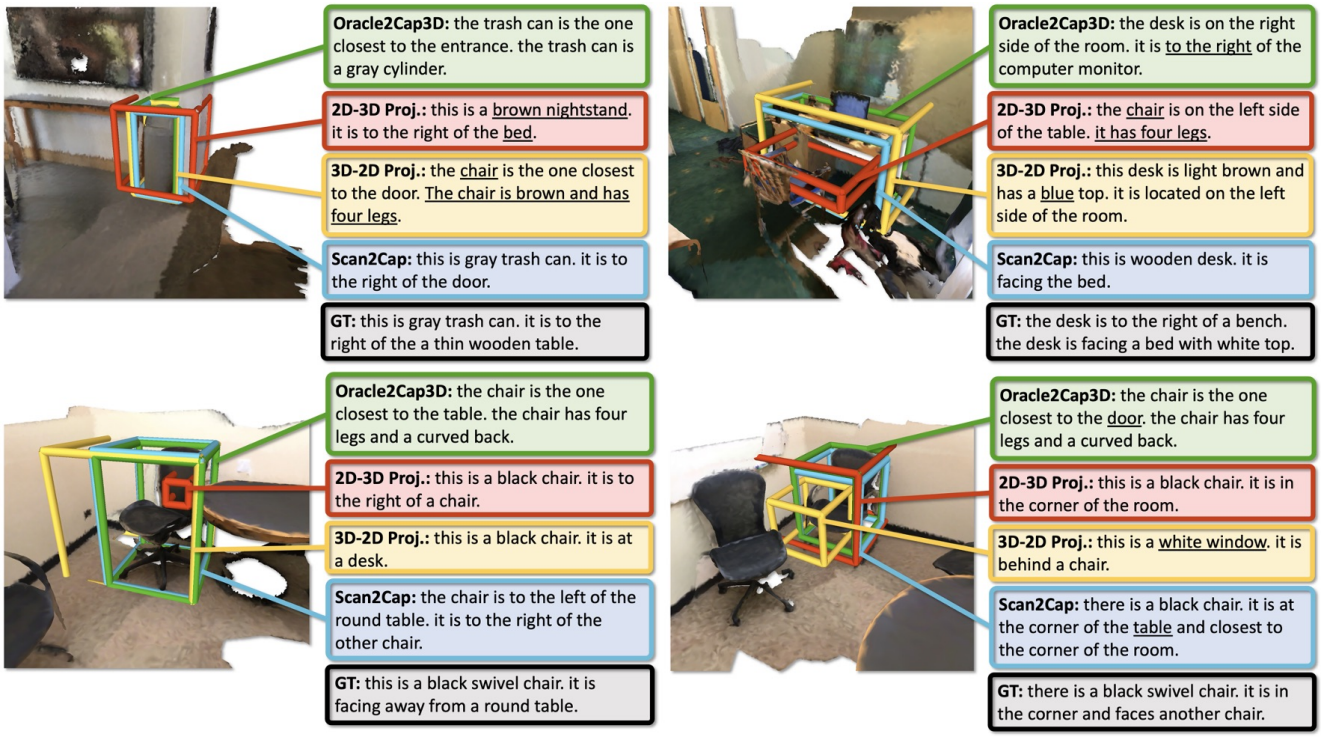


Figure 6: Qualitative results from baseline methods and Scan2Cap with inaccurate parts of the generated caption underscored. Scan2Cap produces good bounding boxes with descriptions for the target appearance and their relational interactions with objects nearby. In contrast, the baselines suffers from poor bounding box predictions or limited view and produces less informative captions. Best viewed in color.

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.

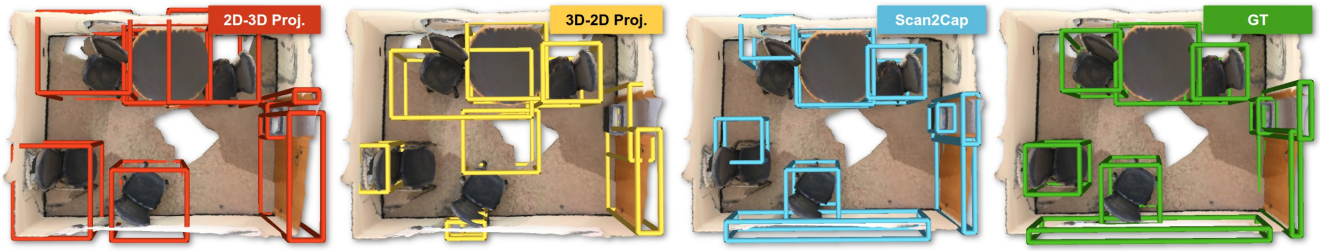


Figure 7: Comparison of object detections of baseline methods and Scan2Cap. 2D-3D Proj. suffers from the detection performance gap between image and 3D space. Scan2Cap produces better bounding boxes compared to 3D-2D Proj. due to the end-to-end fine-tuning.

	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 [42] 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. The higher the better.

	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU
VoteNet [42]+GRU [9]	34.31	21.42	20.13	41.33
VoteNet [42]+CAC	36.15	21.58	20.65	41.78
VoteNet [42]+RG+CAC	**39.08**	**23.32**	**21.97**	**44.78**

Table 5: Ablation study with different components in our method: VoteNet [42] + GRU [9], which is similar to "show and tell" [52]; 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.

Does context-aware attention captioning help? We compare our model with the basic description generation component (GRU) introduced in Vinyals et al. [52] 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 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 the 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 new 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 the descriptions for them in natural language, which is able to address the 3D localization and describing problem at the same time. We apply an attention-based captioning pipeline equipped with a message passing network to generate descriptive tokens while referring to the related components in the local context. Our architecture can effectively localize and describe the 3D objects in the scene and it also outperforms the 2D-based dense captioning methods on the 3D dense description generation task by a big margin. Overall, we hope that our work will enable future research in the 3D visual language field.

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, 12, 15

[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, 13

[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, 12, 13, 15

[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, 12, 16

[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, 12, 15

[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, 12, 13

[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, 12, 13, 15, 16

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 12

[22] 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

[23] 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

[24] 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

[25] 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

[26] 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 Vision (ECCV)*, pages 499–515, 2018. 2

[27] 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

[28] 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

[29] 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

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[31] 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

[32] 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

[33] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 5, 13

[34] 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

[35] 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

[36] 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

[37] 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

[38] 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, 13

[39] 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

[40] 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, 12

[41] 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

[42] 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, 8, 15, 16

[43] 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

[44] 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

[45] 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

[46] 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

[47] 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

[48] 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

[49] 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

[50] 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

[51] 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, 13

[52] 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, 6, 8, 12, 15, 16

[53] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 12

[54] 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

[55] 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

[56] 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

[57] 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

[58] 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

[59] 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

Supplementary Material

In the supplemental, we provide additional details on the 2D captioning experiments to explain the choice of 2D input and captioning method that we use (Sec. A). We also provide details about the 3d-to-2d projection (Sec. B), additional ablation studies (Sec. C.1) and qualitative examples (Sec. C.2) for our 3D experiments.

Figure 8: We compare how each input choice affects the performance of our 2D captioning experiments with oracle bounding boxes. We show the caption generated using show and tell (S&T) for the best matching frame selected from the video recording (A+M, bottom left), rendered annotated viewpoint (A+R, bottom right), and from the bird's eye view (BEV, top). The BEV provides a good overview of large objects, but can miss smaller objects such as trashcans placed underneath desks. The matched frame may not fully capture the object of interest or provide enough context for informative captions (see Tab. 6 for quantitative comparisons).

A. 2D experiments

A.1. Experimental setup

We conduct a series of experiments in 2D to select the input, captioning method, and visual features for our 2D baselines. We implement the models for the 2D experiments using PyTorch [40] and Detectron2 [53].

Choice of 2D input However, we find that it is often challenging to find a good matching frame (see Fig. 9), and using the rendered frames leads to better captioning performance (see Tab. 6) despite the rendering artifacts. Fig. 9 shows examples of viewpoints for which it is challenging to find a good matching frame from the video frames. We suspect that the poor performance of captioning with matched frames is due to the differences in viewpoints as well as the extremely limited field of view and motion blur found in the video frames. In addition, we also check the captioning performance from a bird-eye-view.

Description Generation in 2D (rendered vs matched vs BEV)							
VF	VP	DET	CAP	C	B-4	M	R
G	A+R	-	S&T	49.61	11.41	15.64	40.59
G + T	A+R	O	S&T	**59.12**	**12.73**	**16.61**	**41.32**
G	A+M	-	S&T	11.50	1.63	5.64	13.86
G + T	A+M	O	S&T	16.76	2.01	6.14	14.23
G	BEV	-	S&T	19.94	8.74	14.64	36.53
G + T	BEV	O	S&T	24.21	9.69	14.41	37.38
T + C	A+R	O	TD	**51.35**	**13.09**	**15.88**	**43.52**
G + T + C	A+R	O	TD	18.10	5.65	11.37	33.10
T + C	A+M	O	TD	12.77	1.58	5.84	15.42
G + T + C	A+M	O	TD	14.00	1.68	5.74	15.41

Table 6: We compare captions for oracle bounding boxes from annotated viewpoints with rendered (A+R), matched frames (A+M), and from the birds-eye-view (BEV) on the ScanRefer [7] validation split. We observe that the rendered frames consistently result in better captions for different features (global (G), with target object features (T), and context object features (C)) and captioning methods (show and tell (S&T) vs top-down attention (TD)).

Captioning method For selecting a 2D captioning method, we experiment with a simple model, show and tell (S&T [52]), as well as the popular bottom-up and top-down attention model (TD [2]), and a recent state-of-the-art captioning method, the meshed-memory transformer (M^2 [10]). The S&T [52] and TD [2] models are similar to the original ones, but we replace LSTM [21] with GRU [9] due to the small size of the ScanRefer [7] dataset. In addition to the captioning methods above, we also compare our method against the retrieval baselines (Retr).

Visual features For visual features, we experiment with using the global visual features for the entire image (G), features from just the target object (T), and features from the context objects (C). For object-based features, we rely on object bounding boxes that are either oracle (O), detected using a 2D object detector (2DM), or back-projected from 3D (3DV). For our 2D detection, We use Mask R-CNN [20] with a pre-trained ResNet-101 [19] as our backbone and then fine-tune it on the ScanRefer training split using rendered viewpoints.

A.2. Results

In this section we evaluate our instance segmentation and captioning methods in 2D.

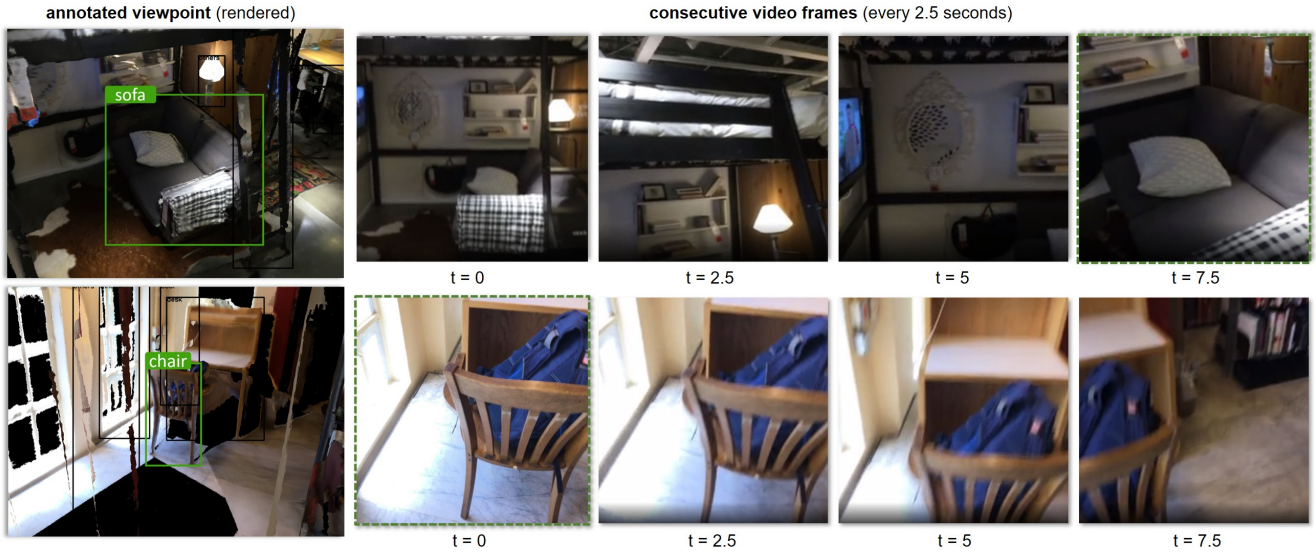


Figure 9: Examples of difficult to match viewpoints, with the rendered frame for the annotated viewpoint on the left, and sample frames from the video on the right (selected matched frame shown with dashed borders). The bounding box for the target object is shown in green. Due to a lack of video recording coverage, it is often impossible to match the exact viewpoint camera direction and origin. Frames from the video recording suffers from motion blur and have a view that is too close up, and missing contextual objects.

	bath.	bed	bkshf.	cab.	chair	cntr.	curt.	desk	door	others	pic.	fridg.	showr.	sink	sofa	tabl.	toil.	wind.	mAP	mAP50	mAP75
DET	12.84	37.66	20.33	16.09	32.39	18.63	16.21	14.47	14.55	20.98	24.72	17.30	18.90	19.73	29.91	28.71	58.22	16.09	23.21	36.01	24.45
SEG	9.74	23.61	1.38	15.25	27.97	7.53	12.82	6.95	11.79	19.66	23.74	18.12	17.91	20.03	25.86	28.23	56.72	9.62	18.72	32.01	19.37

Table 7: 2D object detection (DET) and instance segmentation (SEG) results on the ScanRefer [7] validation split. Reported values for each object category is the *mAP* at IoU $= 0.50 : .05 : 0.95$ (averaged over 10 IoU thresholds). *mAP* is the class averaged precision at IoU $= 0.50 : .05 : 0.95$ (averaged over 10 IoU thresholds). *mAP50* is the class averaged precision at IoU $= 0.50$. *mAP75* is the class averaged precision at IoU $= 0.75$. We use a Mask R-CNN [20] with a pre-trained ResNet-101 [19] backbone and fine-tune it on the ScanRefer [7] training split.

A.2.1 Object detection and instance segmentation

We evaluate the model performance on object detection and instance segmentation via *mAP* (mean average precision). Tab. 7 demonstrates our object detection and instance segmentation results.

A.2.2 Captioning

We evaluate the captions generated for 2D inputs using the well-established CiDEr [51], BLEU-4 [38], METEOR [4] and ROUGE [33], abbreviated as C, B-4, M, R, respectively. Tab. 8 shows our captioning experiment results and Fig. 10 shows examples from the different methods. Note that the captioning metrics reported here are not comparable to dense captioning metrics reported in the main paper, as these does not take into account the IoU, and we evaluate the predicted caption against the ground truth caption for each respective viewpoint.

Surprisingly, we find that the simple baseline of S&T outperforms other methods such as the top-down attention (TD) and meshed-memory transformer (M^2) on CiDEr and METEOR. We suspect that this is partly due to the limited amount of training data (MSCOCO has 113,287 training images with five captions each while ScanRefer has only 36,665 descriptions in the train split). Thus, for our 2D-based baselines in the main paper, we chose to use S&T with features from the global image and the target object.

B. 3D to 2D projection details

In order to caption the objects in the images using 3D detected information, we estimate the camera viewpoints from the 3D bounding boxes and project the 3D bounding boxes to the rendered single-view images for captioning. We show the example in Fig. 11.

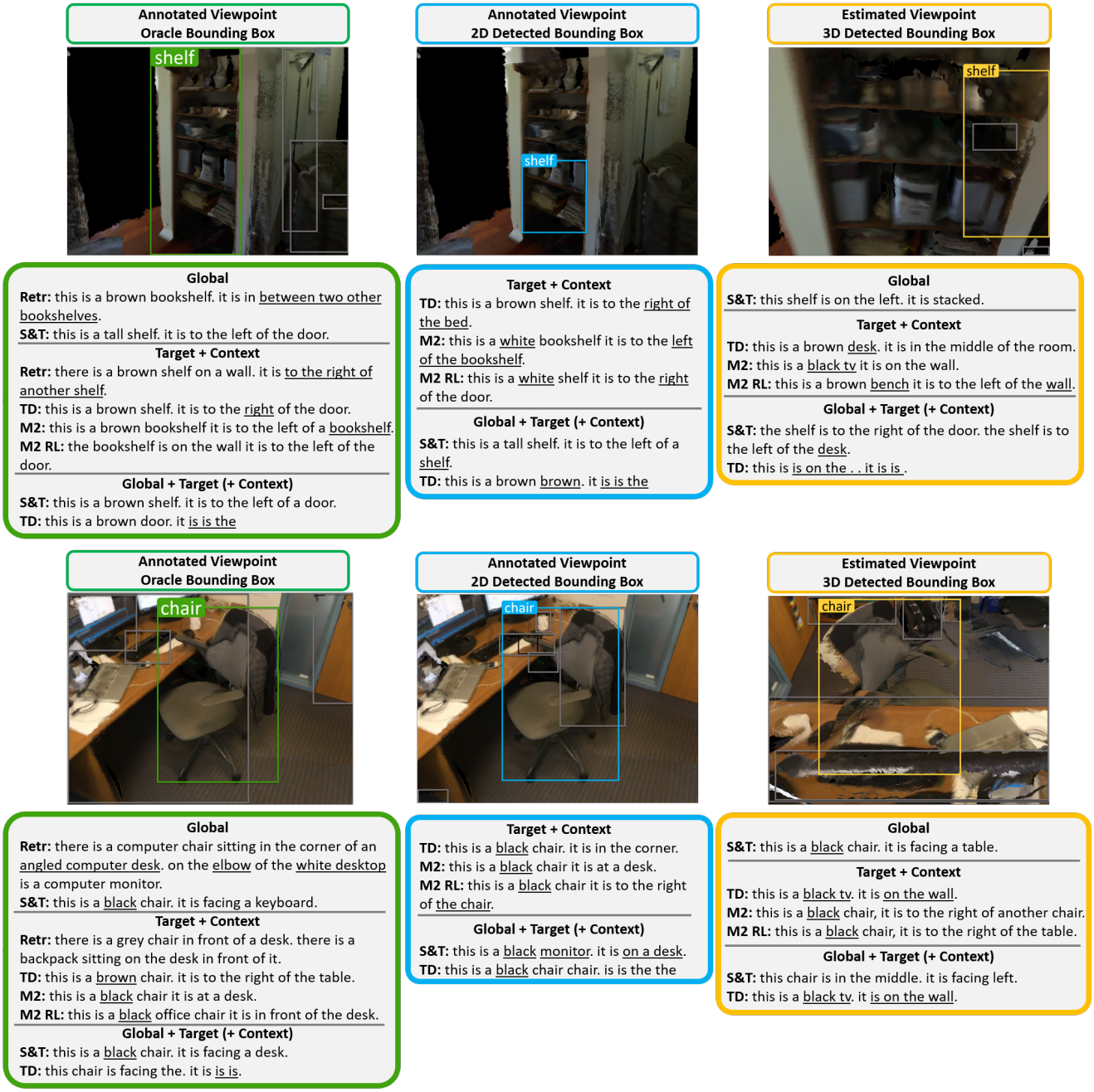


Figure 10: Examples of captions generated from 2D rendered frames with oracle bounding boxes (O-left), detected boxes from Mask-RCNN (2DM-middle), and projected bounding boxes from 3D to 2D (3DV-right). The oracle and Mask-RCNN predictions are from the annotated viewpoint, while the 3D to 2D projection is using an estimated viewpoint. The bounding box for the target object is shown in color, while the bounding box for the context objects are in gray. Inaccurate parts of the caption are underscored.

Viewpoint estimation from 3D detections. We take several heuristics into account to estimate the viewpoints for the detected 3D boxes. To start with, we compute the average distance between the target objects and the recorded viewpoints (1.97 meters). Then, assuming the camera height as 1.70 meters, we compute the horizontal distance

Description Generation in 2D (Rendered Viewpoints)							
VF	VP	DET	CAP	C	B-4	M	R
G	A	-	Retr	12.07	4.58	11.50	29.37
G	A	-	S&T	49.61	11.41	15.64	40.59
T	A	O	Retr	23.00	7.28	13.44	33.82
T + C	A	O	TD	51.35	**13.09**	15.88	**43.52**
T + C	A	O	M^2	34.72	7.13	12.69	33.60
T + C	A	O	M^2 RL	42.77	9.03	14.34	36.27
G + T	A	O	S&T	**59.12**	12.73	**16.61**	41.32
G + T + C	A	O	TD	18.10	5.65	11.37	33.10
T + C	A	2DM	TD	35.65	**11.00**	14.30	**40.70**
T + C	A	2DM	M^2	31.02	7.19	12.28	33.22
T + C	A	2DM	M^2 RL	35.91	8.52	13.53	35.33
G + T	A	2DM	S&T	**41.44**	10.95	**15.08**	39.04
G + T + C	A	2DM	TD	14.84	4.95	10.85	31.52
G	E	-	S&T	28.52	24.03	18.92	47.76
T + C	E	3DV	TD	28.25	**30.11**	18.9	**52.14**
T + C	E	3DV	M^2	11.44	19.67	14.23	40.42
T + C	E	3DV	M^2 RL	11.83	24.79	15.47	42.69
G + T	E	3DV	S&T	**31.48**	25.35	**19.09**	47.06
G + T + C	E	3DV	TD	9.66	9.68	13.14	38.38

Table 8: Results of caption generation with rendered viewpoints on the ScanRefer [7] validation split. Captioning metrics are calculated by comparing the generated caption against the reference caption corresponding to the annotated viewpoint. VF is the input visual feature which can include the full image (G), context objects (C), and/or target object (T). VP is the viewpoint that can be annotated (A), estimated (E), or bird's eye viewpoint (BEV). DET is the object bounding box which can be the ground truth box (O), Mask R-CNN [20] detected in 2D (2DM) or back-projected VoteNet [42] detection in 3D (3DV). CAP is the captioning method which can be cosine retrieval (Retr), Show and tell (S&T) [52], Top-down attention [2] (TD), Meshed memory transformer [10] without and with self-critical optimization respectively (M^2) and (M^2 RL). Since S&T with global and target object features (G+T) gives the best CiDEr score, we select it as the 2D captioning method for the main paper.

between the target objects and the viewpoints (0.99 meter). We randomly pick the points on the circle with the horizontal radius 0.99 meters to the target objects. We repeat the random selection process until the selected viewpoints are inside the scenes and the target objects are visible in the view.

Projecting 3D detections to the estimated views. We derive the camera extrinsics from the estimated viewpoints as we assume the cameras are always targeting at the center of the 3D bounding boxes. We keep the camera intrinsic as in ScanNet. Then, we use these camera parameters to render the single-view images for the 3D scans. The 3D bounding boxes are then projected into the image space as the targets and contexts for generating captions.

C. Additional 3D captioning results

C.1. Additional quantitative analysis

Do other 3D features help? We include colors and normals from the ScanNet meshes to the input point cloud features and compare performance against networks trained without them. As displayed in Tab. 9, our architecture trained with geometry, multi-view features and normal vectors (xyz+multiview+normal) achieves the best performance among all ablations. This matches the feature ablation from ScanRefer [7].

C.2. Additional qualitative analysis

Do graph and attention help with captioning? We compare our model (VoteNet+RG+CAC) with the basic description generation component (VoteNet+GRU) introduced in Vinyals et al. [52] and our model equipped only with the context-aware attention captioning (VoteNet+CAC). As shown in Fig. 12, though all three methods produce good bounding boxes (IoU$>$0.5), VoteNet+GRU makes mistakes when describing the target objects. VoteNet+CAC refers to the target and the objects nearby in the scene, but still fails to correctly reveal the relative spatial relationships. In contrast, VoteNet+RG+CAC can properly handle the interplay of describing the target appearance and the relative spatial relationships in the local environment.

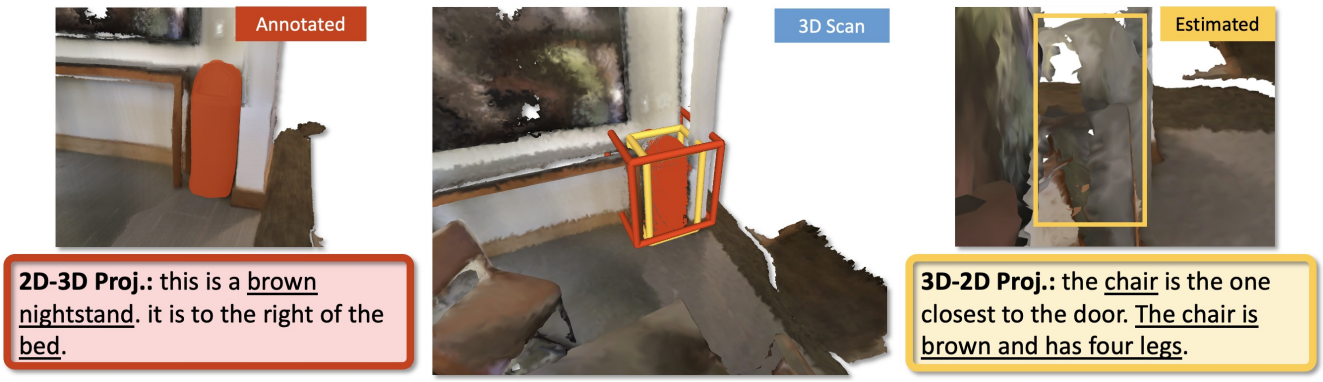


Figure 11: Comparison of generated captions based on 2D-3D and 3D-2D projected detections (2D-3D Proj. and 3D-2D Proj respectively). In 2D-3D Proj., we first detect object mask in the rendered annotated viewpoints using Mask R-CNN [20] (as shown in the red box on the left), and generate the caption for the detected object. While in 3D-2D Proj., we first detect object bounding boxes in 3D using VoteNet [42], then estimate a viewpoint for the detected 3D bounding box, and we back-project the detected bounding box to 2D. We then generate the caption based on the estimated viewpoint and the back-projected bounding box (see the yellow box on the right).

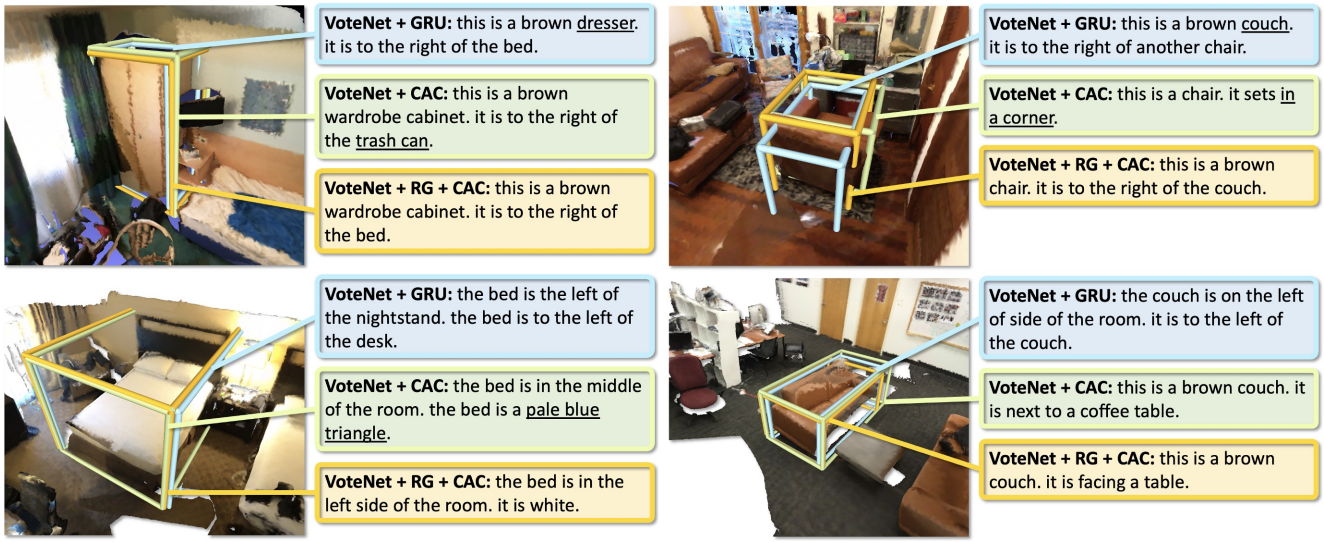


Figure 12: Ablation study with different components in our method: VoteNet [42] + GRU [9], which is similar to "show and tell" Vinyals et al. [52]; VoteNet + Context-aware Attention Captioning (CAC); VoteNet + Relational Graph (RG) + Context-aware Attention Captioning (CAC), namely Scan2Cap. We underscore the inaccurate aspects in the descriptions. Image best viewed in color.

	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 (xyz)	47.21	29.41	24.89	50.74	32.94	20.63	21.10	41.58	27.45
Ours (xyz+rgb)	49.36	32.88	25.52	54.20	33.41	21.61	**22.12**	43.61	27.52
Ours (xyz+rgb+normal)	53.73	**34.25**	26.14	54.95	35.20	22.36	21.44	43.57	29.13
Ours (xyz+multiview)	54.94	32.73	25.90	53.51	36.89	21.77	21.39	42.83	31.43
Ours (xyz+multiview+normal)	**56.82**	34.18	**26.29**	**55.27**	**39.08**	**23.32**	21.97	**44.78**	**32.21**

Table 9: Ablation study with different features. 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. The higher the better.