Spatiality-guided Transformer for 3D Dense Captioning on Point Clouds

Heng Wang, **Chaoyi Zhang**, **Jianhui Yu** and **Weidong Cai**
School of Computer Science, University of Sydney, Australia
{heng.wang, chaoyi.zhang, jianhui.yu, tom.cai}@sydney.edu.au

Abstract

Dense captioning in 3D point clouds is an emerging vision-and-language task involving object-level 3D scene understanding. Apart from coarse semantic class prediction and bounding box regression as in traditional 3D object detection, 3D dense captioning aims at producing a further and finer instance-level label of natural language description on visual appearance and spatial relations for each scene object of interest. To detect and describe objects in a scene, following the spirit of neural machine translation, we propose a transformer-based encoder-decoder architecture, namely SpaCap3D, to transform objects into descriptions, where we especially investigate the relative spatiality of objects in 3D scenes and design a spatiality-guided encoder via a token-to-token spatial relation learning objective and an object-centric decoder for precise and spatiality-enhanced object caption generation. Evaluated on two benchmark datasets, ScanRefer and ReferIt3D, our proposed SpaCap3D outperforms the baseline method Scan2Cap by 4.94% and 9.61% in CIDEr@0.5IoU, respectively. Our project page with source code and supplementary files is available at https://SpaCap3D.github.io/.

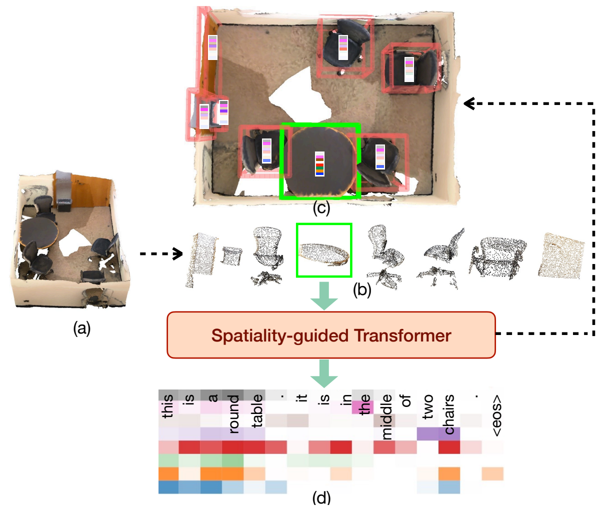


Figure 1: Dense captioning for a target point-cloud object. Target and its surrounding objects are marked in green and red, respectively. (a) Point-based scene input. (b) Detected vision tokens. (c) Neighbor-to-target contribution visualization in our encoder. (d) Target-to-word contribution visualization in our decoder. Detailed explanations including the color scheme used for attention heads, can be found in the supplementary.

1 Introduction

With continuous advance of deep learning in both computer vision and natural language processing, a variety of multimodal studies in these two areas have gained increasingly active attention [Uppal *et al.*, 2022]. Dense captioning, as first introduced in image domain [Johnson *et al.*, 2016], is a representative task among them to describe every salient pixel-formed area with a sequence of words. Just as many other multimodal tasks, the scope of conventional dense captioning research is mainly restricted to 2D space [Yang *et al.*, 2017; Yin *et al.*, 2019; Li *et al.*, 2019b; Kim *et al.*, 2019]. In recent past, with the popularity of 3D point-based scene data collection and application, 3D scene understanding and analysis have become feasible and prominent [Qi *et al.*, 2018; Wald *et al.*, 2020; Zhang *et al.*, 2021; Zhao *et al.*, 2021]. Also, two newly introduced dense annotation datasets tailored for 3D indoor scenes [Dai *et al.*, 2017], ScanRefer [Chen *et al.*, 2020] and ReferIt3D [Achlioptas *et al.*, 2020], create more opportunities in 3D multimodal research. Facilitated by these datasets and pioneering point cloud processing techniques [Qi *et al.*, 2017; Qi *et al.*, 2019], dense captioning has been recently lifted from 2D to 3D [Chen *et al.*, 2021] to localize and describe each object in a 3D point cloud scene, which is beneficial for applications such as robotics manipulation, augmented reality, and autonomous driving.

In real world, human descriptions of an object or instructions to navigate a robot always involve good understanding and capturing of relative spatiality in 3D space [Landau and Jackendoff, 1993; Skubic *et al.*, 2004]. In 3D dense captioning datasets, spatial language (*above*, *under*, *left*, *right*, *in front of*, *behind*, etc.) could be ubiquitous, taking up 98.7% and 90.5% in ScanRefer and ReferIt3D respectively according to their dataset statistics. However, such critical 3D spatiality has not been well explored in previous work [Chen *et al.*, 2021]. Also, the sequential training strategy in their adopted RNN-based captioner could make it prohibitively long to reach convergence. In contrast, attention mechanism in prevalent Transformer [Vaswani *et al.*, 2017] is capable of not only long-range relationship learning but also efficient parallel training. However, relation learning in transformer-based architectures depends only on the final task objective and lacks an explicit guidance on the relation, which could make it hard to precisely learn how 3D spatially-related an object is with respect to another one in our task.

To bridge the gap, in this work, we conduct careful *relative spatiality modeling* to represent 3D spatial relations and propose a spatiality-guided Transformer for 3D dense captioning. Building upon a detection backbone which decomposes the input 3D scene into a set of object candidates (i.e., tokens), we propose **SpaCap3D**, a transformer-based encoder-decoder architecture, which consists of a spatiality-guided encoder where the relation learning among tokens is additionally supervised with a *token-to-token spatial relationship* guidance whose labels are on-the-fly generated main-axis spatial relations based on our relative spatiality modeling and an *object-centric* decoder to transform each spatiality-enhanced vision token into a description, as shown in Figure 1. With faster training and efficient usage of data at hand, our proposed method exceeds Scan2Cap [Chen *et al.*, 2021] by 4.94% and 9.61% in CIDEr@0.5IoU on ScanRefer [Chen *et al.*, 2020] and Nr3D from ReferIt3D [Achlioptas *et al.*, 2020], respectively. To iterate, our contributions are three-fold:

- We propose a token-to-token spatial relation learning objective with relative spatiality modeling to guide the encoding of main-axis spatial relations for better representation of 3D scene objects.
- An integrated and efficient transformer-based encoder-decoder architecture, SpaCap3D, is proposed for 3D dense captioning, consisting of a spatiality-guided encoder and an object-centric decoder.
- We achieve a new state-of-the-art performance on ScanRefer and Nr3D from ReferIt3D over previous work.

2 Related Work

2.1 Dense Captioning: from 2D to 3D

Following [Johnson *et al.*, 2016], current state-of-the-art 2D dense captioning methods use a region proposal network to detect salient regions and extract their CNN-based features as representation where a RNN captioner is applied to generate phrases or sentences. [Yang *et al.*, 2017] attached a late-fusion context feature extractor LSTM with a captioner one to emphasize contextual cues. [Yin *et al.*, 2019] and [Li *et al.*, 2019b] proposed to consider not only the global context but also the neighboring and the target-guided object context, respectively. In [Kim *et al.*, 2019], a sub-pred-obj relationship was learnt via a triple-stream network.

As 2D image is a projection of 3D world without depth dimension, spatial relations expressed in 2D dense captioning are usually implicit and ambiguous. To directly tackle 3D world, Scan2Cap [Chen *et al.*, 2021] proposed 3D dense captioning on point cloud data. In Scan2Cap, relations among object proposals are learnt through a message passing network where only angular deviation relations whose labels [Avetisyan *et al.*, 2019] (i.e., transformation matrices) are hard to collect and incomplete are taken into consideration, while captions are generated by RNN-based decoder following 2D dense captioning methods, which is time-consuming in training. Compared to Scan2Cap, our work focuses on more common spatial relations, and the relation labels are easy to obtain for all objects during training as our label generation process only requires access to the bounding box information (i.e., box center and size). In addition, fast parallel-training in transformer-based architectures guarantees the efficiency of our method.

2.2 Transformers in Image Captioning

Although Transformers in dense captioning have not been explored to the best of our knowledge, there are a string of works in the related image captioning area. To learn better region representations, encoders in existing work were incorporated with learnable prior knowledge [Cornia *et al.*, 2020], geometric weight learnt from geometry features [Herdade *et al.*, 2019], region and attribute representations [Li *et al.*, 2019a], inter- and intra-layer global representations [Ji *et al.*, 2021], or proposal- and grid-level features [Luo *et al.*, 2021]. Their decoders focus on how to learn the implicit relationship among region proposals so that a general and overall image-level caption can be generated. Yet, dense captioning in 3D world involves more diversities and degrees of freedom in object arrangements and it emphasizes finer and denser object-level descriptions, which captures more interactions between an object and its surrounding environment. To tackle these challenges, we use location-aware positional encoding to encode global position and a spatiality-guided encoder with token-to-token spatial relation learning objective to learn relative 3D spatial structures, while our object-centric decoder transforms each spatiality-enhanced object visual representation into a description.

3 Method

We present our spatiality-guided Transformer as SpaCap3D for 3D dense captioning in Figure 2. We first use a detector to decompose input scene into object proposals which we refer to as vision tokens, and then feed them into a spatiality-guided encoder for token-to-token relative 3D spatiality learning. Lastly, a shared object-centric decoder is conditioned on each spatiality-enhanced object vision token to describe them individually.

3.1 3D Object Detection

For an input point cloud of size $N \times (3 + K)$ including a 3-dim coordinate and extra K-dim features such as colors, normals, and multi-view features for each point, we first apply an object detector to generate M object candidates which are input tokens to later components. To make fair comparisons with existing work, we deploy the same seminal detector VoteNet [Qi *et al.*, 2019] with PointNet++ [Qi *et al.*, 2017] as feature aggregation backbone to produce initial object features $\mathcal{X} \in \mathbb{R}^{M \times C}$. We also keep the vote cluster center coor-

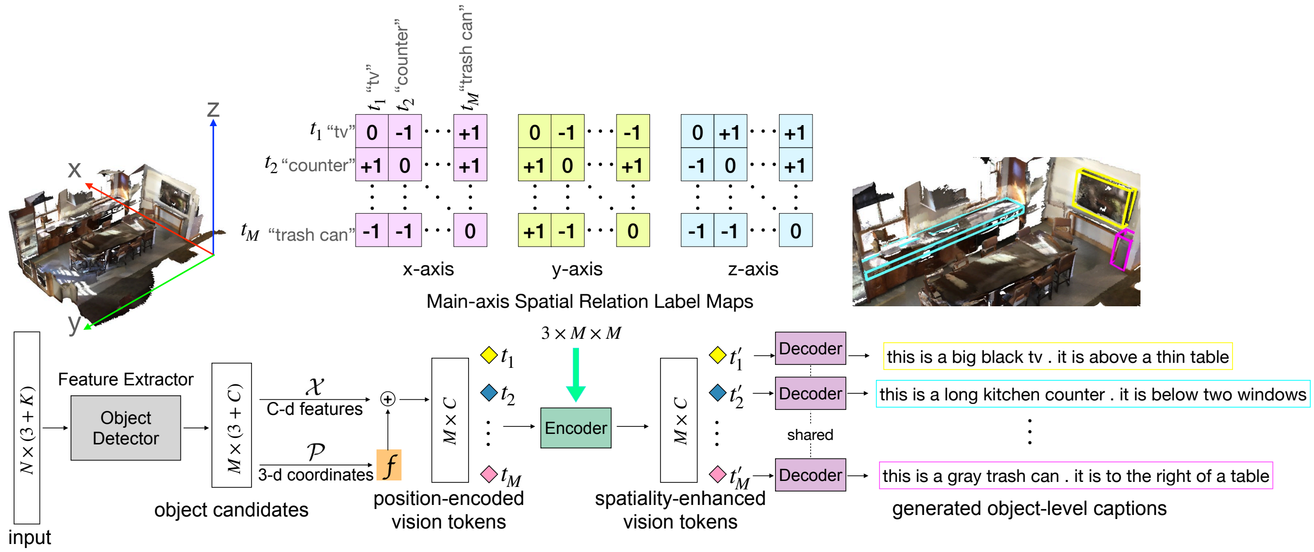


Figure 2: The overview of our proposed method SpaCap3D for spatiality-guided 3D dense captioning. The encoder-decoder framework consists of an object detector to generate object proposals (i.e., tokens), a learnable function f to project coordinates, a token-to-token spatial relation-guided encoder to incorporate relative 3D spatiality into tokens, and a shared object-centric decoder to generate per-object descriptions.

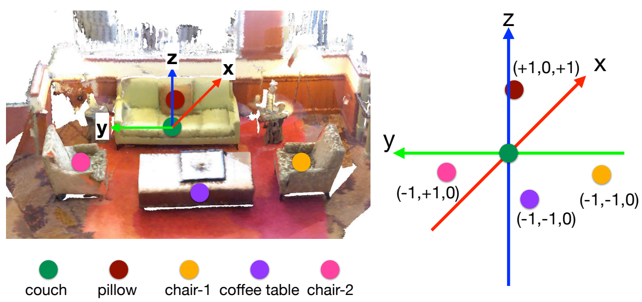
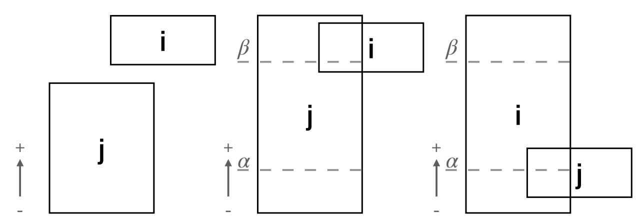


Figure 3: An example of our 3D spatiality modeling of main-axis spatial relations. With respect to the couch, chair-2 is in the negative half x-axis, positive half y-axis, and on the same floor along z-axis, hence its spatial relation to couch is represented as (-1, +1, 0). As spatial relation is relative, the relation of couch to chair-2 is expressed reversely as (+1, -1, 0).



(a) Absolute. (b) Covered. (c) Covering.

Figure 4: Three cases when an object o_i is to the positive direction of another object o_j along x-/y- axis. Top view. The arrow points to the positive direction. α and β are the lower and upper area limits, respectively.

dinates $\mathcal{P} \in \mathbb{R}^{M\times 3}$ from its proposal module as global location information for later positional encoding.

3.2 Token-to-Token Spatial Relationship Learning

To generate spatiality-enhanced captions, we carefully conduct a relative spatiality modeling from which spatial relations among tokens can be formulated and learnt through a token-to-token (T2T) spatial relationship learning objective.

Relative Spatiality Modeling

We first introduce how we model the relative spatiality in 3D scenes. We construct a local 3D coordinate system (right-handed) with an object itself as the origin center, and the relation of a surrounding object with respect to the center object can be represented as a (λ_x, λ_y, λ_z)-triplet where each entry $\lambda \in \{+1, -1, 0\}$ indicates which half axis the surrounding object sits along different axes (+1 for positive, -1 for negative, and 0 for same position), as illustrated in Figure 3. Note we only consider coarse direction such as positive or negative and ignore the exact displacement along axes to decomplexify the modeling. Specifically, according to the intersection of two objects, the definition of a positive relation can vary. Before discussing it, we introduce the notations we use in the following. $\square_i$ indicates the bounding box of an object o_i. And $\square^k_{i:\wedge/\vee/|}$ denotes the parameters of the bounding box along k-axis where $k \in \{x, y, z\}$, i.e., $\wedge$ for minimum value, $\vee$ for maximum value, and $|$ for side length. As relations along z-axis involve different heights while those along x-/y- axis are grounded to the same floor level, we discuss the criteria of being positive for them separately.

Same floor. Depending on how two objects overlap with each other, we categorize the criteria of o_i being positive w.r.t. o_j along x-/y- axis into three cases as illustrated in Figure 4. (a) **Absolute positive**: when the overlapping area does not exceed the side length of o_j and both the bottom and up of o_i are above those of o_j: $(\square^k_{i:\wedge} \geq \square^k_{j:\wedge}) \cap (\square^k_{i:\vee} > \square^k_{j:\vee})$ (b) **Covered positive**: when the overlapping area equals the side length of o_i itself, i.e, o_i is completely covered by o_j, and it resides at the upper area of o_j: $(\square^k_{i:\wedge} > \square^k_{j:\wedge} + \alpha \times \square^k_{j:|}) \cap (\square^k_{j:\wedge} + \beta \times \square^k_{j:|} < \square^k_{i:\vee} \leq \square^k_{j:\vee})$ (c) **Covering positive**: this is the reverse situation of condition (b) when the overlapping area equals the side length of o_j instead, and o_j lags at o_i's lower area: $(\square^k_{i:\wedge} < \square^k_{j:\wedge} < \square^k_{i:\wedge} + \alpha \times \square^k_{i:|}) \cap (\square^k_{j:\vee} < \square^k_{i:\wedge} + \beta \times \square^k_{i:|})$. We additionally define two objects are at the same position when the bottom and up of one object are within a certain tolerance ϵ from those of the other one: $(| \square^k_{i:\vee} - \square^k_{j:\vee} | \leq \epsilon) \cap (| \square^k_{i:\wedge} - \square^k_{j:\wedge} | \leq \epsilon)$. The lower and upper area limits α, β, and tolerance ϵ are empirically set as 0.3, 0.7, and $0.1 \times \square^k_{j:|}$, respectively.

Various heights. We define o_i is at positive direction to o_j when o_i is over the lower area of o_j: $\square^z_{i:\wedge} \geq \square^z_{j:\wedge} + \alpha \times \square^z_{j:|}$.

Positional Encoding

Before spatial relation learning, as shown in Figure 2, we apply a learnable function $f(\cdot)$ to each vote cluster center $p \in \mathcal{P}$ to incorporate the global location information into each token. Specifically, $f(\cdot)$ is defined as:

$$f(p) = (\sigma(BN(pW_1)))W_2, \tag{1}$$

where $W_1 \in \mathbb{R}^{3 \times C}$ and $W_2 \in \mathbb{R}^{C \times C}$ are two linear transformations to project 3-dim geometric features into the same high dimensional space as general features $\mathcal{X}$. We use ReLU as the activation function σ and a Batch Normalization (BN) layer to adjust the feature distribution. The input tokens $T = \{t_1, t_2, ..., t_M\} \in \mathbb{R}^{M \times C}$ for our encoder are then created by adding the newly projected C-dim global geometry into $\mathcal{X}$.

Spatial Relation Learning

When learning a token-to-token spatial relationship, we aim to capture the corresponding object-to-object relation. For a target token t_i and its neighboring token t_j, we select their ground truth objects o_i and o_j as the ones with the nearest centers to their predicted centers. We can generate three main-axis spatial relation label maps, $\mathcal{R}^x$, $\mathcal{R}^y$, and $\mathcal{R}^z$, for the M tokens based on their ground truth objects' relations, as per the criteria defined above. Label entries $\{r^x_{i,j}, r^y_{i,j}, r^z_{i,j}\} \in \{+1, -1, 0\}$ define how the object o_i represented by token t_i is in positive/negative/same direction along x-, y-, and z-axis to another object o_j represented by t_j, respectively. In a standard transformer, the encoder is composed of n repetitions of a multi-head self-attention (MSA) layer and a feed-forward network (FFN). A normalization and residual link (AddNorm) is applied for each layer. The updated token t'_i after attention mechanism is defined as the summation of $\omega_{i,j} t_j$ for $j = 0, 1, ..., M$ where $\omega_{i,j}$ represents the attention coefficient of t_i to t_j. In other words, the updated token is comprised of different contribution of its neighboring tokens. To encode such contribution with relative 3D spatiality information, we apply our relation prediction head (RPH) to each contribution $\omega_{i,j} t_j$. As illustrated in Figure 5,

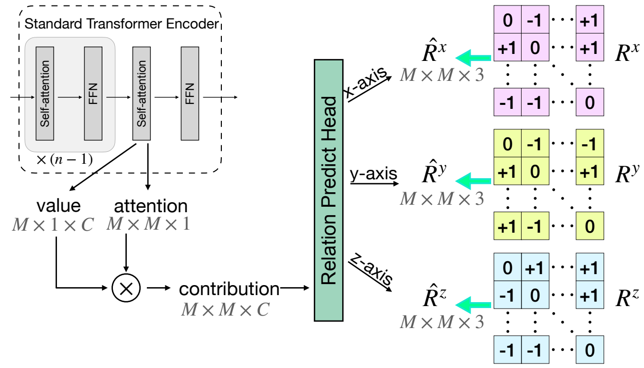


Figure 5: Detailed encoder architecture. AddNorm is omitted for brevity.

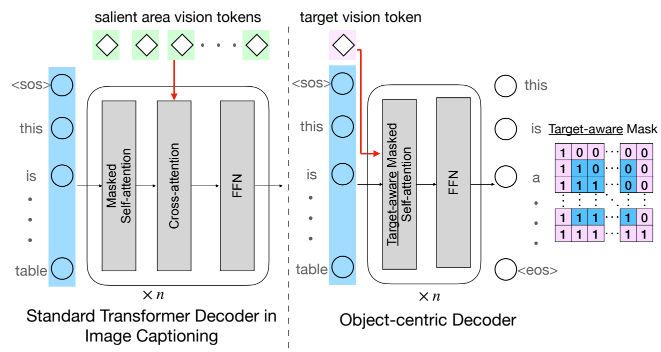


Figure 6: Detailed decoder architecture. Positional encoding and AddNorm for decoder are omitted for brevity.

the relation prediction happens at the last encoder block. We use a standard three-layer MLP with two ReLU activated C-dim hidden layers and a linear output layer. The output of the RPH is a 9-dim vector where each three represents the predicted relation along a main axis. The T2T spatial relation learning is hence guided by our relation loss as:

$$L_{relation} = \sum_{k \in \{x,y,z\}} L_{CE}(\hat{\mathcal{R}}^k, \mathcal{R}^k), \tag{2}$$

where L_{CE} denotes three-class cross-entropy loss.

3.3 Object-centric Decoder

In image captioning Transformers, the decoder consisting of n stacks of a masked MSA layer, a cross-attention layer with the output of encoder, and a FFN, attends all salient area vision tokens to conclude one sentence describing the whole image. On the other hand, in dense captioning, the target is each object. Therefore, we propose an object-centric decoder with target-aware masked self-attention layer to update each word token by attending both its previous words and the target vision token as depicted in Figure 6. Compared to the standard decoder, our design can fulfill the dense captioning task but in a more concise and efficient manner. More specifically, it would stack a target vision token mask (in pink) on the basis of the existing word token mask (in blue) and feed the target vision token as well as word tokens together into

Datasets	Methods	Input	C@0.5IoU	B-4@0.5IoU	M@0.5IoU	R@0.5IoU	mAP@0.5IoU
ScanRefer	Scan2Cap	xyz+normal+mv	39.08	23.32	21.97	44.78	32.21
	Ours-base	xyz	40.19	24.71	22.01	45.49	32.32
	Ours	xyz	42.53	25.02	22.22	45.65	34.44
	Ours	xyz+normal+rgb	42.76	**25.38**	**22.84**	**45.66**	35.55
	Ours	xyz+normal+mv	**44.02**	25.26	22.33	45.36	**36.64**
Nr3D/ReferIt3D	Scan2Cap	xyz+normal+mv	24.10	15.01	21.01	47.95	32.21
	Ours-base	xyz	31.06	17.94	22.03	49.63	30.65
	Ours	xyz	31.43	18.98	22.24	49.79	33.17
	Ours	xyz+normal+rgb	33.24	19.46	**22.61**	50.41	33.23
	Ours	xyz+normal+mv	**33.71**	**19.92**	22.61	**50.50**	**38.11**

Table 1: Quantitative comparison with SOTA methods on ScanRefer and Nr3D/ReferIt3D. Ours-base is the baseline variant of standard Transformer adapted for 3D dense captioning, where we use standard encoder with sinusoidal positional encoding and late-guide decoder. The input denotes various combinations of different information: *xyz* refers to points' coordinates. *normal* means the normal vector of each point. *rgb* uses color information and *mv* stands for pretrained 2D multi-view features.

the self-attention layer. Considering ours as an early-guide way to condition the decoder on the target object, we also implemented a late-guide variant and have it ablated in Table 2.

3.4 Learning Objective

We define our final loss as $L = \delta * L_{det} + L_{des} + \zeta * L_{relation}$, where $L_{relation}$ is our proposed T2T spatial relation learning objective defined in Equation 2. As for the object detection loss L_{det} and the description loss L_{des}, we follow Scan2Cap, and more details can be found in [Chen *et al.*, 2021]. δ and ζ are set as 10 and 0.1, respectively, to maintain similar magnitude of different losses.

4 Experiments

4.1 Datasets, Metrics, and Implementation Details

Datasets. We evaluate our proposed method on ScanRefer [Chen *et al.*, 2020] and Nr3D from ReferIt3D [Achlioptas *et al.*, 2020], both of which provide free-form human descriptions for objects in ScanNet [Dai *et al.*, 2017]. Same as Scan2Cap [Chen *et al.*, 2021], for ScanRefer/Nr3D, we train on 36,665/32,919 captions for 7,875/4,664 objects from 562/511 scenes and evaluate on 9,508/8,584 descriptions for 2,068/1,214 objects from 141/130 scenes.

Metrics. We benchmark the performances on both detection and captioning perspectives. For detection, we use the mean average precision thresholded by 0.5 IoU score (mAP@0.5). For captioning, we employ m@0.5IoU where only the prediction whose IoU is larger than 0.5 will be considered [Chen *et al.*, 2021]. The captioning metric m can be the one especially designed for image captioning such as CIDEr (C) [Vedantam *et al.*, 2015], or those focusing more on machine translation or on text summarizing such as BLEU-4 (B-4) [Papineni *et al.*, 2002], METEOR (M) [Banerjee and Lavie, 2005], and ROUGE (R) [Lin, 2004].

Implementation Details. To make fair comparisons, we use the same training and testing protocols as Scan2Cap. Following Scan2Cap, we set the input number of points N as 40,000 and the number of object proposal M as 256. The output feature dimension C from object detector is 128. For Transformer, we set the number of encoder and decoder blocks n as 6 and the number of heads in multi-head attentions as 8. The dimensionality of input and output of each layer is 128 except that for the inner-layer of feed-forward networks as 2048. We implement AddNorm as pre-LN where outputs are first normalized and then added with inputs. Keeping the length of descriptions within 30 tokens and marking words not appearing in GloVE [Pennington *et al.*, 2014] as unknown, we learn the word embedding from scratch encoded with sinusoidal positional encoding. We implement[1] our proposed model in PyTorch [Paszke *et al.*, 2019] and train end-to-end with ADAM [Kingma and Ba, 2015] in a learning rate of 1×10^{-3}. The detection backbone is fine-tuned together with the end-to-end training from the pretrained VoteNet [Qi *et al.*, 2019] model provided by Scan2Cap. To avoid overfitting, we apply the same weight decay factor 1×10^{-5} and the same data augmentation as Scan2Cap. We firstly randomly flip the input point cloud along the YZ- and XZ-plane and then rotate along x-, y-, and z-axis by a random angle within $[-5°, 5°]$. We finally translate the point cloud along all axes by a random distance within $[-0.5, 0.5]$ meters. All experiments were trained on a single GeForce RTX 2080Ti GPU with a batch size of 8 samples for 50 epochs, while the model is checked and saved when it reaches the best CIDEr@0.5IoU on val split every 2000 iterations. Training with our proposed framework takes around 33 and 29 hours for ScanRefer and Nr3D/ReferIt3D, respectively. During inference, we use non-maximum suppression to remove overlapping proposals and only keep those proposals whose IoUs with ground truth bounding boxes are larger than 0.5. With a batch size of 8, the inference time including evaluation with the four captioning metrics, for ScanRefer and Nr3D/ReferIt3D, is 83 and 75 seconds, respectively. We implement the attention visualization shown in Figure 1 and the supplementary based on [Vig, 2019].

4.2 Quantitative Comparison

Table 1 presents the quantitative comparison with SOTA methods, showing our proposed SpaCap3D with xyz-input

[1]https://github.com/heng-hw/SpaCap3D

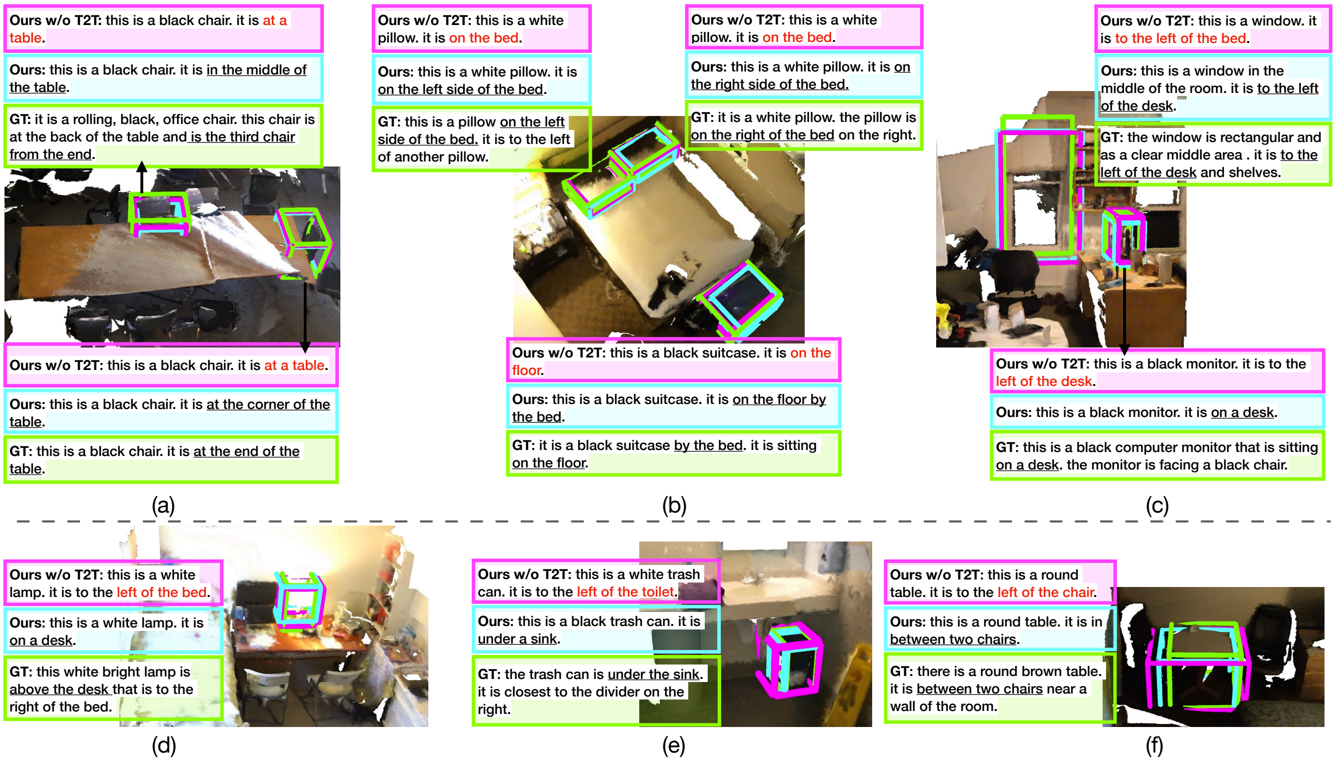


Figure 7: Visualization for our method with and without token-to-token (T2T) spatial relation guidance. Caption boxes share the same color with detection bounding boxes for ground truth (green), ours w/ T2T (blue), and ours w/o T2T (pink). Imprecise parts of sentences produced by ours w/o T2T guidance are marked in red, and correctly expressed spatial relations predicted by T2T-guided method are underscored.

outperforms not only the baseline Scan2Cap but also the standard Transformer (ours-base) in all metrics. It is worth noting that SpaCap3D manages to exceed Scan2Cap, even when SpaCap3D only takes simple coordinates as inputs whereas Scan2Cap uses much richer pretrained multi-view features. To be comparable with Scan2Cap's input setting, we also provide a variant of our proposed method with color and normal features added and it achieves better results as expected. The inclusion of multi-view features can further boost the performance and we point out that it requires 6 more hours to train compared to its simpler counterpart (i.e., xyz+normal+rgb).

The average forward pass time per training batch is around 0.5s and 0.2s, and the average per-batch inference time is around 11.7s and 2.3s, respectively for Scan2Cap and our SpaCap3D (taking the same xyz-input), which demonstrates our efficiency.

4.3 Qualitative Analysis

To visualize the importance of relative spatiality, we display some detect-and-describe results of the proposed method with and without relative 3D spatiality learning in Figure 7. If the T2T guidance is discarded, the generated descriptions could lack unique spatial relations and tend to be general as shown in Figure 7 (a) and (b). While our spatiality-guided Transformer distinguishes two chairs at a table (*"middle"* and *"corner"*) and two pillows on the bed (*"left"* and *"right"*) by their spatiality, the method without such guidance could collapse into generic expressions lacking specific spatial relations. Also in Figure 7 (b), our proposed method with T2T is capable of describing more relations for the suitcase compared to the one without T2T guidance (*"on the floor by the bed"* vs. *"on the floor"*). Figure 7 (c), (d), and (e) show cases when T2T guidance boosts correct spatial relation prediction. We also emphasize Figure 7 (f) where a table is in between two chairs. Instead of just describing the relation between the table and one chair, the T2T-guided method considers the existence of both chairs and generates more thoughtful expression. More results are displayed in Supplementary Figure 3.

4.4 Ablation Study

Component Analysis

We investigate components of our proposed architecture, the late-guide and early-guide decoder, attention-based encoder with vote center-based positional encoding, and token-to-token spatial relation learning (T2T), in Table 2. Model A and B adopt the decoder alone and the outcome that Model B achieves 3.56% and 1.17% improvement over Model A on captioning and detection respectively demonstrates the superiority of our proposed early-guide decoder. Based on Model B, Model C uses an attention-based encoder to learn the long-range dependency among object proposals, which leads to a detection performance increase by 1.31%. With the guidance of our T2T spatial relation learning objective, the encoder functions better as can be seen in the results from Model D which performs the best in both captioning and detection.

Model	Decoder		Encoder	T2T	C@0.5IoU *(captioning)*	mAP@0.5IoU *(detection)*
	late-guide	early-guide				
A	✓				37.80	30.97
B		✓			41.36	32.14
C		✓	✓		41.14	33.45
D		✓	✓	✓	**42.53**	**34.44**

Table 2: Ablation study on different components of our proposed method. T2T denotes the token-to-token spatial relation learning objective.

Positional encoding		C@0.5IoU
non-learnable	none	39.41
	sinusoidal	39.44
learnable	random	42.29
	box center	42.49
	box center*	40.04
	vote center	**42.53**

Table 3: Ablation study on choices of encoder's positional encoding. box center∗ indicates concatenation of box center and size.

Positional Encoding Analysis

To verify the choice of learnable vote center-based positional encoding for encoder, we elaborate on different ways in Table 3. The non-learnable sinusoidal method in standard Transformer has slightly better effect over the one without any positional encoding, showing the necessity of such encoding in our task. For learnable encoding, we compare with random one used in 2D object detection Transformer [Carion *et al.*, 2020] and box center-based one adopted in 3D object detection Transformer [Liu *et al.*, 2021]. More details of these learnable positional encoding methods can be found in the supplementary file. We find that random learnable positional encoding can boost the performance compared to non-learnable ones and the incorporation of object position information can further advance the performance. Among all learnable encoding ways, our vote center-based one achieves the best results.

5 Conclusion

In this work, we propose a new state-of-the-art framework dubbed as SpaCap3D for the newly emerging 3D dense captioning task. We propose to formulate object relations with relative 3D spatiality modeling, based on which we build a transformer-based architecture where a spatiality-guided encoder learns how objects interact with their surrounding environment in 3D spatiality via a token-to-token spatial relation learning guidance, and a shared object-centric decoder is conditioned on each spatiality-enhanced token to individually generate precise and unambiguous object-level captions. Experiments on two benchmark datasets show that our integrated framework outperforms the baseline method by a great deal in both accuracy and efficiency.

References

[Achlioptas *et al.*, 2020] 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 *ECCV*, pages 422–440. Springer, 2020.

[Avetisyan *et al.*, 2019] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2CAD: Learning CAD model alignment in RGB-D scans. In *CVPR*, pages 2614–2623, 2019.

[Banerjee and Lavie, 2005] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop: Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, 2005.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020.

[Chen *et al.*, 2020] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. In *ECCV*, pages 202–221. Springer, 2020.

[Chen *et al.*, 2021] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2Cap: Context-aware dense captioning in RGB-D scans. In *CVPR*, pages 3193–3203, 2021.

[Cornia *et al.*, 2020] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *CVPR*, pages 10578–10587, 2020.

[Dai *et al.*, 2017] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[Herdade *et al.*, 2019] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: transforming objects into words. In *NeurIPS*, pages 11137–11147, 2019.

[Ji *et al.*, 2021] Jiayi Ji, Yunpeng Luo, Xiaoshuai Sun, Fuhai Chen, Gen Luo, Yongjian Wu, Yue Gao, and Rongrong Ji. Improving image captioning by leveraging intra-and inter-layer global representation in transformer network. In *AAAI*, volume 35, pages 1655–1663, 2021.

[Johnson *et al.*, 2016] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. DenseCap: Fully convolutional localization networks for dense captioning. In *CVPR*, pages 4565–4574, 2016.

[Kim *et al.*, 2019] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Dense relational captioning: Triple-stream networks for relationship-based captioning. In *CVPR*, pages 6271–6280, 2019.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Landau and Jackendoff, 1993] Barbara Landau and Ray Jackendoff. “What” and “where” in spatial language and spatial cognition. *Behavioral and Brain Sciences*, 16(2):255–265, 1993.

[Li *et al.*, 2019a] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *ICCV*, pages 8928–8937, 2019.

[Li *et al.*, 2019b] Xiangyang Li, Shuqiang Jiang, and Jungong Han. Learning object context for dense captioning. In *AAAI*, volume 33, pages 8650–8657, 2019.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop: Text Summarization Branches Out*, pages 74–81, 2004.

[Liu *et al.*, 2021] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3D object detection via transformers. In *ICCV*, pages 2949–2958, 2021.

[Luo *et al.*, 2021] Yunpeng Luo, Jiayi Ji, Xiaoshuai Sun, Liujuan Cao, Yongjian Wu, Feiyue Huang, Chia-Wen Lin, and Rongrong Ji. Dual-level collaborative transformer for image captioning. In *AAAI*, volume 35, pages 2286–2293, 2021.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Qi *et al.*, 2017] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017.

[Qi *et al.*, 2018] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3D object detection from RGB-D data. In *CVPR*, pages 918–927, 2018.

[Qi *et al.*, 2019] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *ICCV*, pages 9277–9286, 2019.

[Skubic *et al.*, 2004] Marjorie Skubic, Dennis Perzanowski, Samuel Blisard, Alan Schultz, William Adams, Magda Bugajska, and Derek Brock. Spatial language for human-robot dialogs. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(2):154–167, 2004.

[Uppal *et al.*, 2022] Shagun Uppal, Sarthak Bhagat, Devamanyu Hazarika, Navonil Majumder, Soujanya Poria, Roger Zimmermann, and Amir Zadeh. Multimodal research in vision and language: A review of current and emerging trends. *Information Fusion*, 77:149–171, 2022.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[Vedantam *et al.*, 2015] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015.

[Vig, 2019] Jesse Vig. A multiscale visualization of attention in the transformer model. In *ACL: System Demonstrations*, pages 37–42, 2019.

[Wald *et al.*, 2020] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *CVPR*, pages 3961–3970, 2020.

[Yang *et al.*, 2017] Linjie Yang, Kevin Tang, Jianchao Yang, and Li-Jia Li. Dense captioning with joint inference and visual context. In *CVPR*, pages 2193–2202, 2017.

[Yin *et al.*, 2019] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, and Jing Shao. Context and attribute grounded dense captioning. In *CVPR*, pages 6241–6250, 2019.

[Zhang *et al.*, 2021] Chaoyi Zhang, Jianhui Yu, Yang Song, and Weidong Cai. Exploiting edge-oriented reasoning for 3D point-based scene graph analysis. In *CVPR*, pages 9705–9715, 2021.

[Zhao *et al.*, 2021] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3DVG-Transformer: Relation modeling for visual grounding on point clouds. In *ICCV*, pages 2928–2937, 2021.