Looking Outside the Box to Ground Language in 3D Scenes

Ayush Jain[†1], Nikolaos Gkanatsios[†1], Ishita Mediratta[§1,2], Katerina Fragkiadaki[1]
[1]Carnegie Mellon University, [2]Facebook AI Research
{ayushj2, ngkanats}@andrew.cmu.edu, ishitamed@fb.com, katef@cs.cmu.edu

Abstract

Existing language grounding models often use object proposal bottlenecks: a pre-trained detector proposes objects in the scene and the model learns to select the answer from these box proposals, without attending to the original image or 3D point cloud. Object detectors are typically trained on a fixed vocabulary of objects and attributes that is often too restrictive for open-domain language grounding, where an utterance may refer to visual entities at various levels of abstraction, such as a chair, the leg of a chair, or the tip of the front leg of a chair. We propose a model for grounding language in 3D scenes that bypasses box proposal bottlenecks with three main innovations: i) Iterative attention across the language stream, the point cloud feature stream and 3D box proposals. ii) Transformer decoders with non-parametric entity queries that decode 3D boxes for object and part referentials. iii) Joint supervision from 3D object annotations and language grounding annotations, by treating object detection as grounding of referential utterances comprised of a list of candidate category labels. These innovations result in significant quantitative gains (up to +9% absolute improvement on the SR3D benchmark) over previous approaches on popular 3D language grounding benchmarks. We ablate each of our innovations to show its contribution to the performance of the model. When applied on language grounding on 2D images with minor changes, it performs on par with the state-of-the-art while converges in half of the GPU time.

1. Introduction

Language grounding is the task of localizing objects in a scene that are mentioned in a language utterance. Recent approaches [11,13,19,24,26], both in the 3D and the 2D domain, rely on object proposal bottlenecks: a pre-trained detector proposes objects in the scene and the model is trained to select the answer from these box proposals. The original point cloud or image input are discarded upon extraction of the object proposals. This is problematic as an utterance may require information from or refer to entities at different levels of granularity, such as a chair or the leg of a chair or a stain on the leg of the chair. A generic object detector typically fails to propose all relevant entities for the given utterance bottom-up; there are simply too many entities to be proposed and is computationally infeasible to do so. Small, occluded, or rare objects are hard to detect without task-driven guidance. Consider Figure 1; we can easily miss the clock on the shelf unless someone draws our attention to it. Indeed, the 2D Faster-RCNN detector [45] trained on 1601 Visual Genome classes misses the object of interest. The quality of the pre-trained detector has been found to be crucial for the final performance on the downstream tasks [55]. The particularity of the visual domain is that relevant entities come at different levels of spatial abstraction which makes task-independent tokenization hard.

Recently, MDETR [25] proposed an architecture for language grounding in 2D images using end-to-end attention between the language and visual streams without box proposal bottlenecks. Instead, an image is tokenized into a set of feature vectors, each capturing both appearance features and spatial information from image's x, y coordinate. The visual feature vectors are concatenated with word embedding vectors and the sequence of tokens from both modalities passes through multiple layers of self-attention. A set of parametric query vectors cross-attend to this sequence of encoded tokens to decode all objects mentioned in the caption. MDETR excels in language-modulated detection in 2D images and achieves big leaps in performance over previous box-bottlenecked methods. Its design does not exploit pre-trained detectors to tokenize the image, rather, it learns to detect objects from scratch. It achieves this using additional intermediate supervision in terms of box annotations for *all* objects mentioned in an utterance, not just the referential object box.

We propose a model for grounding language in 3D and 2D scenes that uses box proposal streams as an additional tokenization of the visual input scene, that can inform decisions of the grounding model without constraining the box

[†] Equal contribution, order decided by `np.random.rand`
[§] Work done during an internship at CMU

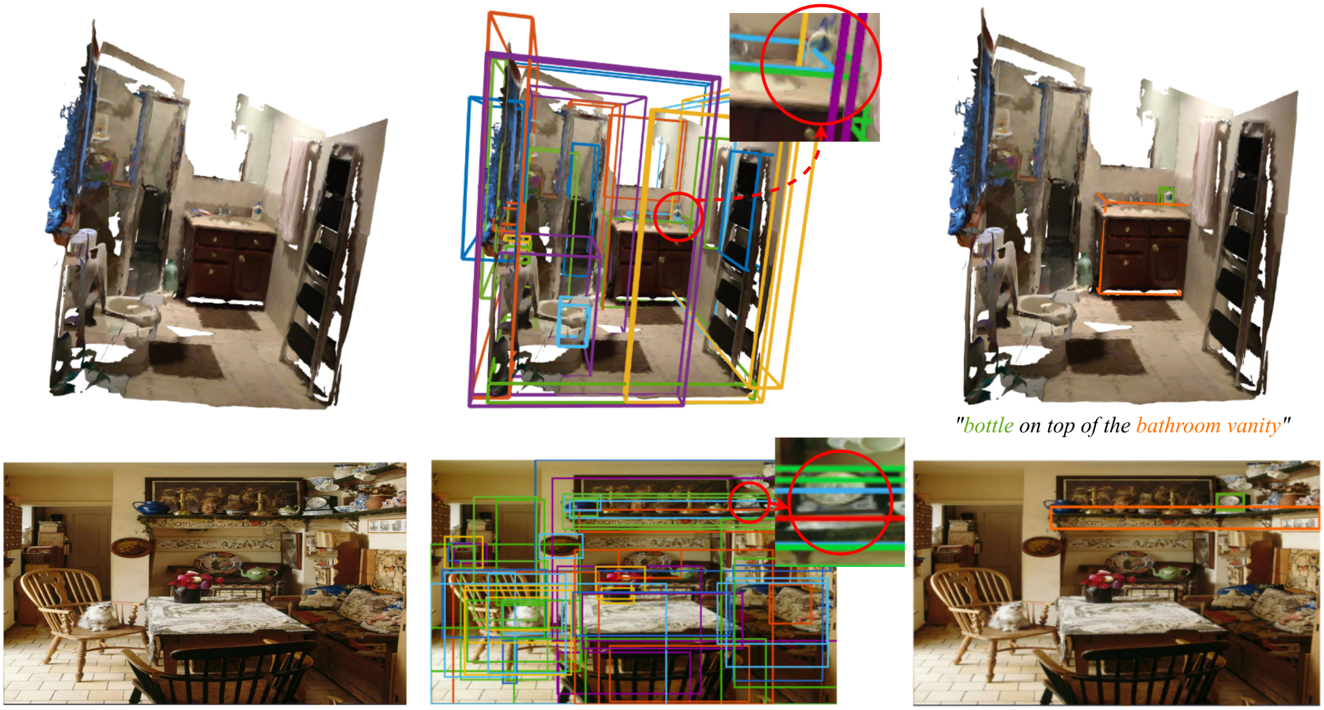

"bottle on top of the bathroom vanity"
"clock placed on top of the upper shelf"

Figure 1. **Outside-the-box language grounding in 3D (*top*) and in 2D (*bottom*).** Boxes detected by state-of-the-art object detectors (2D Faster-RCNN detector [45] trained on 1601 Visual Genome classes and 3D Group-Free detector [35] trained on 485 ScanNet classes) often fail to localize the object of interest (clock, bottle). Our model locates the relevant objects in both 3D and 2D scenes by attending across image/point cloud, language and box proposal streams, using non-parametric queries to decode all relevant objects and combining object detection supervision with language grounding supervision through detection phrases.

inferences. We call our model BEAUTY-DETR, as it uses both box proposals, obtained by a general purpose detector "bottom-up" , i.e., without any language-guided modulation, and "top-down" guidance from the language utterance, to localize the relevant objects in the scene. The work of [27] distinguishes between bottom-up attention drawn to salient parts of the scene, and top-down attention guided by the task of the agent, and can be guided by language, and our model is a computational implementation of both types of attention. It builds upon the d-DETR model of [58] for decoding fused features into relevant object boxes through non-parametric queries proposed by the point cloud features in 3D or pixel features in 2D. BEAUTY-DETR uses tripartite attention across the language stream, the pixel stream and a set of object proposals obtained by a pre-trained detector to predict the object(s) referenced in the utterance. Our losses build upon MDETR [25]: besides the standard box regression losses, we use contrastive losses between utterance snippets and box features to align boxes to noun phrases. We supervise BEAUTY-DETR through combined annotations for object detection and language grounding. Specifically, we convert box category annotations used typically for training object detectors [8, 32] into language grounding annotations, where the corresponding utterance is generated synthetically via a random shuffle of object category labels from the detector's vocabulary (Figure 2). We call this a detection phrase. BEAUTY-DETR is tasked to discard the negative labels in the detection phrase (assign them to no boxes) and localize the ones that exist in the point cloud/image with the correct box.

We test BEAUTY-DETR on the 3D benchmarks of [1,5] and 2D benchmarks of [28,53]. In 3D environments, we report significant performance boosts over all prior methods, and also over ablative models that employ MDETR's design choices, i.e. us parametric object queries, do not consider a box proposal stream and do not use detection phrases during training. In 2D images, our model obtains competitive performance with MDETR on RefCOCO and RefCOCO+, while requires less than half of the GPU training time.

In summary, our contributions are: i) Extending language-modulated detection ideas of the state-of-the-art MDETR model to the 3D grounding domain and producing a language grounding model that works in both 3D and 2D. ii) The incorporation of state-of-the-art non-parametric query vectors for decoding detection boxes from detectors to language grounding. iii) The use of box proposal streams as part of the tokenization of the visual input. iv) Converting multi-object box annotations into language grounding annotations with detection phrases. v) Extensive quantitative results in 3D and 2D across multiple benchmarks, and extensive ablations to quantify each contribution of our model. We will make our code publicly available.

2. Related work

Object detection with transformers Object detection is a classic computer vision task where a closed set of object category labels is considered and the detection model is tasked to localize *all* instances of the these object categories. While earlier architectures rely on box proposal and classification heads over convolutional variants of image encoders [16, 33, 44], DETR [4] uses transformer architectures where a set of object query vectors attend to the scene and to one another and eventually decode objects. The recent model of d(eformable)-DETR [58] proposes to use deformable attention, a locally adaptive kernel that is predicted directly in each pixel location without attention to other pixel locations, thus saving the quadratic cost of pixel-to-pixel attention, by noting that content-based attention does not contribute significantly in performance [57]. The works of [35] and [41] extend transformer encoders and detector heads to 3D point cloud input.

Referential grounding Referential object grounding [28], the task of localizing the object(s) referenced in a language utterance, was introduced to handle the limitation of generic object detectors to reference visual entities relevant for a task yet absent from a general vocabulary. In close inspection, object annotations of a particular category can be treated as language grounding annotations where the referential utterance is a single word, namely, the category label itself, and this is precisely exploited by our model for co-training. For ease of exposition, we group existing models into three broad categories based on whether they pursue or not a generic, task-independent, visual tokenization of the scene: i) Models that tokenize the visual scene into discrete sets of entities using generic pre-trained high vocabulary object detectors [11, 13, 19, 24, 26]. Upon tokenization of the visual stream, many recent approaches use large-scale transformer models to fuse information across both vision and language modalities to localize referent objects or answer questions about an image or point cloud [7, 36, 50]. Instead of transformer layers, neural-symbolic approaches [39, 52] use programs of neural modules that are applied on the extracted visual tokens and their color and shape descriptors. These latter models have been mainly applied in simple domains, such as CLEVR [22] or CLEVRER [51], where the computational graph to answer a question or find an object is well-defined and an accurate tokenization of the scene can be obtained with existing object detectors. In all of these models, the original image is discarded upon extraction of the object proposals, i.e., the visual tokens. ii) Models that do not tokenize the visual scene but rather apply operations directly on pixels to extract relevant information, either end-to-end [38,49] MDETR is one of the latest products of this line of work with state-of-the-art performance across multiple 2D grounding datasets. iii) Modular network architectures [3,6,18,23] that process the input image as directed by the inferred computational graph, that may involve applying detectors or masking operations in specific locations to cast selective attention.

Our method attends directly on the visual stream (points/pixels) and thus does not have a detection bottleneck. At the same time, it does not discard the easy-to-detect proposals from an object detector.

3D Language Grounding has only recently gained popularity [1, 5]. Approaches in this category resemble their 2D counterparts, but use encoders suitable for point cloud input, such as PointNet++ [43].

All these methods are object proposal bottlenecked and their pipeline can be decomposed into three main steps. i) Representation of object boxes as point features [50], segmentation masks [54] or pure spatial/categorical features [47]. ii) Encoding of language using word embeddings [47, 50] and/or scene graphs [12]. iii) Fusion of the two modalities and scoring of each proposal using graph networks [20] or Transformers [50]. Most of these works also employ domain-specific design choices by explicitly encoding pairwise relationships [15, 20, 54] or by relying to heuristics, such as restricting attention [54, 56] and ignoring input modalities [47].

Due to the difficulty of detecting objects in 3D point clouds, popular benchmarks [1] evaluate using ground-truth object boxes. Our model is the first to evaluate using detected object boxes as opposed to oracle ones. Co-training of our model under both object detection and referential grounding objectives gives a significant boost over training for grounding alone. In addition, our design is not domain-specific and our model can work for both 3D and 2D scenes. Lastly, while previous 3D approaches score a set of proposals to return only one answer matching the "root" noun phrase, our model predicts boxes and corresponding spans and can thus output all mentioned objects.

3. Method

Given a referential language utterance, e.g., "find the plant that is on top of the end table" and a visual scene, which can be a 3D point cloud or a 2D image, our model is tasked to localize all objects mentioned in the utterance. Therefore, in the previous example, we expect one box for the "plant" and one for the "end table". The architecture of BEAUTY-DETR is depicted in Figure 2. The model

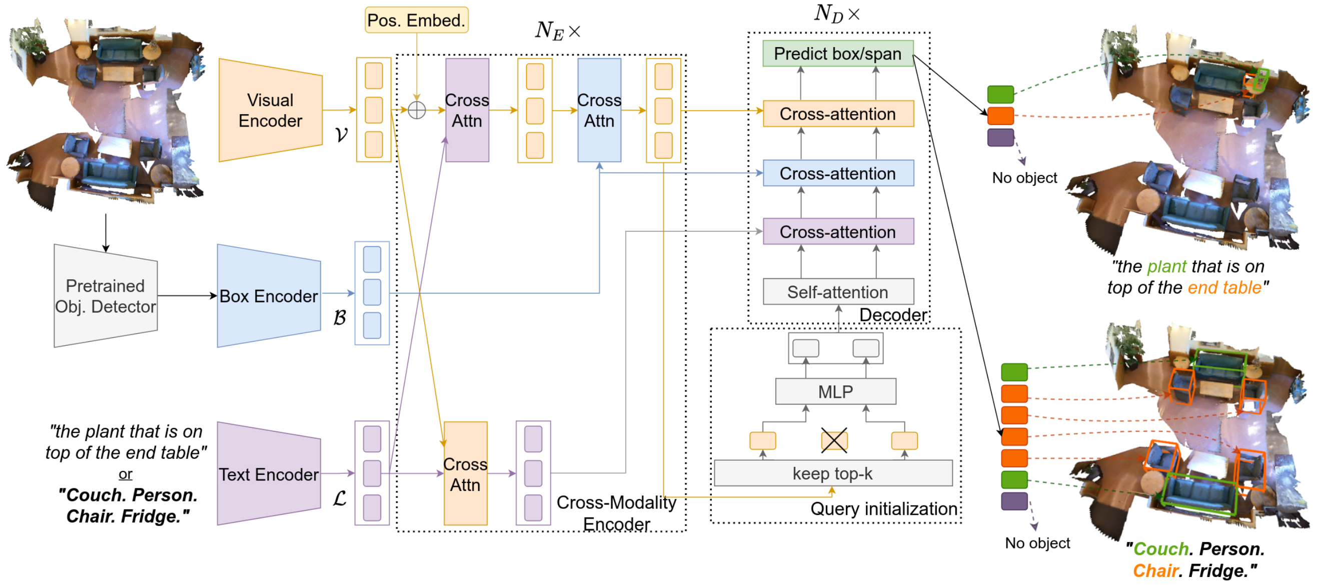


Figure 2. **BEAUTY-DETR architecture.** Given a 3D scene and a language referential expression, our goal is to localize in the scene all the objects mentioned in the utterance. The visual scene and the utterance are encoded into a sequence of tokens each using visual and language encoders. A pre-trained object detector extracts object box proposals that are featurized using their spatial and categorical information. At each encoder layer, visual and language tokens cross-attend and then the visual tokens attend to the detected boxes. At the end of the encoder, visual tokens are mapped to confidence scores and high-scoring tokens instantiate query vectors. The query vectors after layers of self and cross attention eventually predicts a bounding box for an object and a span in the language utterance that the box refers to. We show in bold detection phrases used to co-train BEAUTY-DETR alongside the standard referential utterances.

encodes a language utterance, a 3D point cloud or a 2D image, as well as a set of detected object box proposals, into separate sequences of tokens, and uses cross-attention layers to fuse information across them. After encoding, high scoring visual tokens are decoded to object boxes and are aligned to the corresponding word tokens in the utterance. The alignment of visual to language tokens is supervised by contrastive and span prediction losses inspired by MDETR. We present architecture details in Section 3.1 and training objectives in Section 3.2.

3.1. Architecture

Within-modality encoding In 2D, we encode an RGB image using a pre-trained ResNet101 backbone [17]. The 2D visual features are added with 2D Fourier positional encodings, same as in [21, 58]. These are standard sinusoidal embeddings, as introduced in [48], but computed in the x and y dimension separately and then concatenated. In 3D, we encode a 3D point cloud using a PointNet++ backbone [43]. The 3D visual features are added with a learnable 3D positional encoding, same as [35]: we pass the coordinates of the points through a small multilayer perceptron (MLP). In both cases, the resulting visual features are flattened to form a sequence of visual tokens, $\mathcal{V} \in \mathbb{R}^{n_v \times c_v}$, where n_v is the number of visual tokens and c_v is the number of visual feature channels.

The input visual scene is fed to a general purpose detector to obtain a set of object box proposals. Following prior literature, we use Faster-RCNN [45] for RGB images, pre-trained on a vocabulary of 1601 object categories on Visual Genome [29], and Group-Free detector [35] for 3D point clouds pre-trained on a vocabulary of 485 object categories in ScanNet [8]. The detected 2D and 3D box proposals that surpass a detection threshold (0.50 in 2D and 0.25 in 3D) are encoded using a box encoder by mapping their spatial coordinates and categorical class information to an embedding vector each, and concatenated to form an object token. Appearance information of box visual features can also be considered during box encoding. We discuss ablations of box encoders in the supplementary file. Let $\mathcal{O} \in \mathbb{R}^{n_o \times c_o}$ denote the object token sequence.

The words of the input utterance are encoded using a pre-trained RoBERTa [34] backbone, a carefully optimized version of BERT [10] pre-trained for masked token prediction. This maps the utterance to a sequence of word tokens $\mathcal{L} \in \mathbb{R}^{n_\ell \times c_\ell}$.

All visual, word and box tokens are mapped using (different per modality) MLPs to same-length feature vectors.

Cross-modality Encoder The three modalities interact through a sequence of N_E multi-modality encoding layers comprised of self- and cross-attention operations [36]. In each encoding layer, visual and language tokens cross-attend to one another and are updated using standard key-value attention. Then, the resulting language-conditioned visual tokens attend to the object tokens. In 2D images, we find it beneficial to have self-attention layers in the language and image streams using attention and deformable attention, respectively. These self-attention operations do not help in the 3D domain where the encoding layers only include cross-attention updates. We hypothesize this is due to the much smaller number of training examples available in the 3D language grounding datasets, in comparison to 2D.

Decoder The contextualized visual tokens from the last multi-modality encoding layer are used to predict confidence scores, one per token. The top-K highest scoring tokens are each fed into an MLP to predict a vector which stands for an *object query*, i.e. a vector that will decode a box center and size relative to the location of the corresponding visual token. We call these *non-parametric queries* and they have been used before in 2D object detection by d-DETR [58] and 3D object detection by the group-free detector [35]. They are predicted by visual tokens from the current scene, as opposed to *parametric queries* used in [4] that correspond to a learned set of vectors shared across all scenes. Positional encodings of the predicted box are used as positional embeddings of object query vectors. The object query vectors are updated in a residual manner through N_D decoder layers. In each decoder layer, we employ four types of attention operations. First, the object queries self-attend to one another to contextually refine their estimates. Second, they attend to the contextualized word embeddings to condition on the utterance. Next, they attend to the object proposals and then in the image or point cloud features. This order of cross-attention operations allows the queries to be guided by language, select or discard the existing box proposals, and then condition on these high-objectness areas to explore the scene as needed. At the end of each decoding layer, there is a prediction head that predicts a box center displacement, height and width vector, and a token span for each object query that localizes the corresponding object box and aligns it with the language input. The positional embeddings of this predicted box is used as object query positional embeddings for the next decoder layers, while the object query itself is just residually updated.

3.2. Supervision

Language grounding models have effectively combined supervision across multiple referential, caption description and question answering tasks, which is an important factor of their success. Notable examples are VilBERT [37] and 12in1 [37] methods. Object detection annotations have not been considered yet during such co-training. Yet, object detection is an instance of referential language grounding in which the utterance is a single word, namely, the object category label.

3.2.1 Co-training with detection phrases

We cast object detection as the grounding of referential utterances comprised of a sequence of object category labels, as shown in Figure 2. Specifically, given the detector's vocabulary of object category labels, we randomly sample a fixed number of them—some appear in the visual scene and some do not—and generate synthetic utterances by sequencing the sampled category labels, e.g., *"Couch. Person. Chair. Fridge."*, we call them detection phrases. We treat these utterances as referential expressions to be grounded: the task is to localize all object instances of the category labels mentioned in the utterance if they appear in the scene. The sampling of negative labels category labels (labels for which there are no instances present) operates as negative training: when presented with a caption that erroneously mentions an object, the model is trained to match the wrong labels to no object. Details on this negative training can be found in the supplementary. While MDETR [25] had partially considered negative training, they only use one category label at a time, and not with the motivation of improving performance of the grounding model through densifying supervision. In our case, we found supervision from detection phrases to be important, especially in the 3D domain, due to lack of referential grounding annotations.

Training losses We supervise the outputs of all prediction heads in each layer of the decoder. Following DETR [4], we use Hungarian matching to assign a subset of object queries to the ground-truth objects based on the intersection-over-union (IoU) and label matching between predicted and ground-truth boxes. For the queries that are matched to a ground-truth box, we use the L1 regression loss and generalized IoU (gIoU) loss [46] for the bounding box predictions. We align detected object boxes to spans in the input utterance using the two MDETR [25] objectives: i) Soft token prediction for each object query that corresponds to a softmax over 256 word positions, each one corresponding to a token in the input utterance, where each query is supervised to predict a uniform distribution over all token positions that correspond to the object it is matched. ii) Contrastive matching between query embedding and word embedding vectors that ensures that the inner product of the ground-truth word-box pair embeddings is higher than the inner product of non-corresponding word-box pairs. The query vectors that are not matched upon Hungarian match-

Method	SR3D		NR3D	ScanRefer (Val. Set)
	Acc. (Det)	Acc. (GT)	Acc. (Det)	Acc. (Det)
ReferIt3DNet [1]	$27.7^{\dagger}$	39.8	$24.0^{\dagger}$	26.4
ScanRefer [5]	-	-	-	35.5
TGNN [20]	-	45.0	-	37.4
InstanceRefer [54]	$31.5^{\ddagger}$	48.0	$29.9^{\ddagger}$	40.2
FFL-3DOG [12]	-	-	-	41.3
LanguageRefer [47]	$39.5^{\dagger}$	56.0	$28.6^{\dagger}$	-
3DVG-Transformer [56]	-	51.4	-	45.9
TransRefer3D [15]	-	57.4	-	-
SAT-2D [50]*	$35.4^{\dagger}$	57.9	$31.7^{\dagger}$	44.5
BEAUTY-DETR (ours)	**48.5**	**60.4**	**34.1**	**46.4**

Table 1. **Results on language grounding in 3D point clouds.** We evaluate top-1 accuracy using ground-truth *(GT)* or detected *(Det)* boxes under 0.25 threshold. * denotes method uses extra 2D image features. † denotes evaluation with detected boxes using the authors' code and checkpoints. ‡ denotes re-training using the authors' code. For [56], we compare against their 3D-only version.

ing with any ground-truth object box are set to predict "no span" and they take part in the contrastive losses as negatives. We ask the model to decode not only the "target" referent object, but all object mentions in the utterance, when such annotations are available. This provides denser supervision than supervising the target referent alone.

Model	Accuracy
BEAUTY-DETR	**48.5**
w/o attention on points	41.9
w/o attention on box stream	46.7
w/o co-training with detection phrases	44.5
with parametric object queries	33.8

Table 2. **Ablation of design choices for BEAUTY-DETR on SR3D validation set**. We remove/add one component every time.

	Overall		Detected		Missed		
	Acc.	Recall	Acc.	Recall	Acc.	Recall	Epochs
BEAUTY-DETR	48.5	82.5	62.9	95.8	16.1	52.8	30
w/o Attention on points	41.9	69.2	60.5	100.0	0.0	0.0	20
w/o Attention on box stream	46.7	81.7	57.5	93.1	22.4	56.4	70

Table 3. **Performance Analysis on SR3D.** Accuracy on SR3D for our model and ablative variants depending on whether the detector did (3rd column) or failed (4th column) to detect the target. We mention the number of training epochs needed for each model to converge to optimal performance on the validation set.

3.3. Implementation Details

In 3D, the input point cloud is encoded with PointNet++ [43] using the same hyperparameters as in [35], pre-trained on ScanNet [8]. We use the last layer's features, resulting in 1024 visual tokens. In the decoder, the object queries are formed from the 256 most confident visual tokens. We set $N_E = 3$ with no self-attention layers, $N_D = 6$. All attention layers are implemented using standard self-/cross-attention. In 2D, we encode the image using a pre-trained ResNet-101. We set $N_E = 6$ and $N_D = 6$. All attention layers to the visual stream are implemented with deformable attention [58], attention to other the language stream or detected boxes is the standard attention of [36,48]. More implementation details are included in supplementary.

4. Experiments

We test BEAUTY-DETR on language grounding in 3D and 2D scenes. Our experiments aim to answer the following questions: **(i)** How does BEAUTY-DETR perform compared to the state-of-the-art in 3D and 2D grounding of referential expressions? **(ii)** How do different components of our model affect performance, for example, the attention on the object proposal stream, the inclusion of detection phrases and the employment of non-parametric queries?

4.1. Results on 3D language grounding benchmarks

For 3D language grounding, we test BEAUTY-DETR on SR3D/NR3D [1] and ScanRefer [5] benchmarks. All three benchmarks contain pairs of 3D point clouds of indoor scenes from ScanNet [8] and corresponding language referential expressions, and the task is to localize the objects referenced in the utterance. The utterances in SR3D are shorter and synthetic, e.g. "Choose the couch that is underneath the picture", while utterances in NR3D and ScanRefer contain natural utterances that are longer and noisier, e.g. "From the set of chairs against the wall, the chair farthest from the red wall, in the group of chairs that is closer to the red wall". For fair comparison against previous methods, we separately train BEAUTY-DETR on each of SR3D, NR3D and ScanRefer, extended with ScanNet detection phrase grounding. SR3D provides annotations for all objects mentioned in the utterance, so during training we supervise localization of all objects mentioned.

We compare BEAUTY-DETR to other state-of-the-art 3D language grounding approaches in Table 1. All previous models that have been tested in SR3D or NR3D bench-

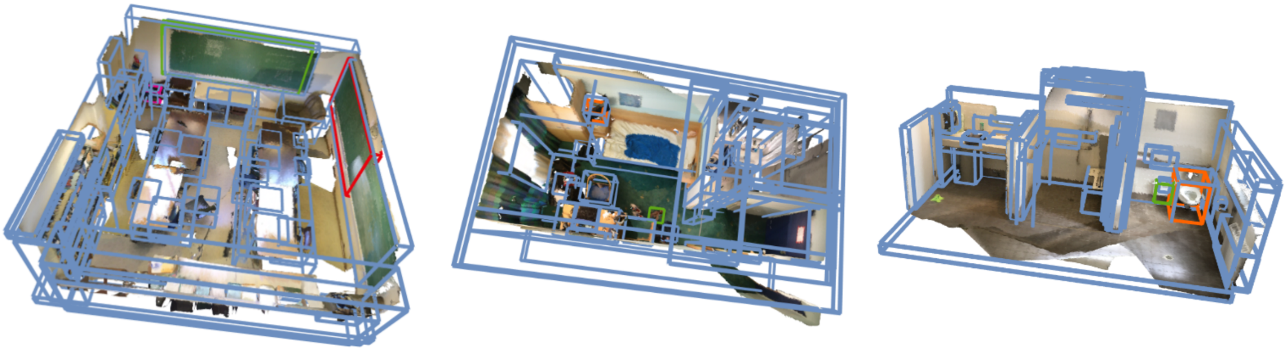

"facing the front of the trash can, pick the blackboard that is to the right of it"
(a)
"find the shoes in front of the tv"
(b)
"select the dustbin next to toilet"
(c)

Figure 3. **Qualitative results of BEAUTY-DETR in the SR3D benchmark.** Predictions for the target are shown in green and for other mentioned objects in orange. The detected proposals appear in blue. (a) The variant without box stream (red box) fails to exploit the information given by the detector, but BEAUTY-DETR succeeds. (b) The detector misses the "shoes" any variant which only look at boxes (and not visual features) fails. (c) The detector is successful in finding the "dustbin", still BEAUTY-DETR refines the box to get a more accurate bounding box.

marks are box-bottlenecked models that use *ground-truth* 3D object boxes (without category labels) and learn to select one of the them as the answer. We thus consider two evaluation setups: i) *Det*: where we re-train previous models using their publicly available code and provide the same 3D object proposals we use in BEAUTY-DETR , obtained by Group-Free object detector trained to detect 485 categories in ScanNet (Section *Det* in Table-1). ii) *GT*, where we use ground-truth 3D object boxes for our model and baseline (denoted section *GT* in Table 1) on SR3D to compare against prior work directly. We use top-1 accuracy metric, which measures the percentage of times we can find the target box with an IoU higher than 0.25.

Under all different protocols, our model outperforms existing approaches by a large margin, including the recent SAT-2D [50] that uses additional 2D image features during training. BEAUTY-DETR does not use 2D image features, but it can be easily extended to do so. The margins are larger on the *Det* setup, since competing models are box-bottlenecked and thus fail when the referenced object is not detected. In NR3D and ScanRefer the gains for our model are smaller in comparison to SR3D since language is very complex and the language hints are harder to interpret to improve localization of object referents. We show qualitative results in Figure 3. For more qualitative results of our model and baselines, please check the supplementary file.

4.1.1 Ablative analysis

We ablate all our design choices on SR3D for 3D BEAUTY-DETR in Table 2. First, we significantly outperform (by 6.6%) an object-bottlenecked variant, analogous to ViLBERT [36], which does not attend on points directly (w/o Attention on points). Removing attention on the box stream also causes an absolute 1.8% drop. Furthermore, co-training with object detection utterances contributes 4% in performance (from 44.5% to 48.5%). Notably, even without that supervision, our model still largely outperforms all previous approaches under the *Det* setup. Lastly, if we replace the task-dependent non-parametric object queries with the scene-independent parametric ones that MDETR [25] uses, we observe a vast drop by 14.7%.

To further investigate the contribution of each attention stream, in Table 3 we measure the recall of each model, as the percentage of times any detected box is successful, and we report results for both the cases when the detector is successful as well as when it is not. We find that 30.8% of the times the detector misses the ground-truth boxes and any box-bottlenecked model that does not attend to points will for sure fail. On the contrary, our model can still work in 16.1% of the cases where an object detector fails, as also shown in Figure 3b. Compared to the variant without attention on the box stream, BEAUTY-DETR achieves better performance while converging in less than half epochs.

4.2. Results on 2D language grounding benchmarks

For 2D language grounding, we test BEAUTY-DETR on referring expression datasets RefCOCO [28] and RefCOCO+ [53]. Similar to 3D datasets, the task is to localise the object referred by the sentence. We first pre-train on combined grounding annotations from Flickr30k [42], referring expression datasets [28, 40, 53], Visual Genome [29] and detection phrases from the MS-COCO object detection dataset [32]. During pre-training the task is to detect all instances of objects mentioned in the sentence. For instance,

	RefCOCO			RefCOCO+			Training	Training
Method	val	testA	testB	val	testA	testB	Epochs	GPU Hours
UNITER_L [7]	81.4	87.0	74.2	75.9	81.5	66.7	-	-
VILLA_L [14]	82.4	87.5	74.8	76.2	81.5	66.8	-	-
MDETR [25]	86.8	**89.6**	81.4	79.5	**84.1**	70.6	40 + 5	5480
BEAUTY-DETR (ours)	**87.9**	88.1	**83.1**	**79.8**	80.2	**70.9**	**11 + 5**	**2524**

Table 4. **Results on language grounding in 2D RefCOCO and RefCOCO+ Datasets on accuracy metric using standard val/testA/testB splits**. All training times are computed using same V100 GPU machines. Training epochs are written as $x + y$ where x = number of pre-training epochs and y = number of fine-tuning epochs. All reported results use ResNet101 backbone for fair comparison.

Model	Accuracy
BEAUTY-DETR w/o co-training with detection phrases	77.0
BEAUTY-DETR w/o box stream and w/o detection phrases	76.3
BEAUTY-DETR with parametric queries w/o co-training with detection phrases	74.2
BEAUTY-DETR	**79.4**

Table 5. **Ablation for BEAUTY-DETR on the RefCOCO validation set**.

if the sentence is "Clock placed on top of the upper shelf" the model needs to predict a bounding box around the shelf and the clock placed on the shelf (and not other clocks). Then we finetune for 5 epochs for RefCOCO and 3 for RefCOCO+ where the task is to only detect the root object i.e. a bounding box around the clock. MDETR uses an identical pretrain-then-finetune scheme but without supervision from detection phrases.

On RefCOCO and RefCOCO+, we report top-1 accuracy on the standard val/testA/testB split. The results in Table 4 indicate that our model trains two times faster than MDETR while getting comparable performance. This computational gain comes mostly from deformable attention. Deformable attention applies only on the visual stream and cannot be employed by MDETR, where the visual and language tokens are concatenated in a single stream. We show more results on Flickr30k [42], as well as qualitative results, in supplementary.

Since pre-training is computationally expensive due to the size of the combined datasets, we do our ablations on RefCOCO without pre-training in Table 5 and use the best design choices for pre-training. Consistent with 3D, removing detection sentences results in an accuracy drop of 2.4%. Removing attention to the box stream results in further drop of 0.7% in accuracy. Using parametric queries achieves 74.2%, resulting in a drop of 2.8% accuracy. However, different than the 3D case, the difference in performance between the parametric and non-parametric queries, and using v/s not using box stream is not that pronounced.

5. Limitations

Our work has three main limitations. Firstly, while we tackle the issues on the visual side by removing object bottlenecks, it remains unclear how to understand the language stream better beyond just naively encoding the language by BERT like models. Thus even when provided with perfect ground truth boxes, the model fails to achieve perfect performance due to failure in re-ranking the box proposals. Secondly, the training time and resources required to pre-train on 2D domain still remains quite expensive, thus preventing us from doing detailed ablation study on all the design choices like we perform in 3D domain. Finally, although BEAUTY-DETR is able to ground objects that the detector misses, still its performance is significantly worse when the detector fails, as shown in Table 3. Ideally, a grounding model should be robust to these cases. In our future work we aim to tackle these limitations.

6. Conclusion

We present BEAUTY-DETR, a model for referential grounding in 3D and 2D scenes that attends to object proposals, language and pixel streams to localize objects mentioned in language utterances. BEAUTY-DETR builds upon the 2D grounding model of MDETR [25], extends it to the 3D grounding domain and enhances it with an additional box proposal tokenization stream, non-parametric query heads for decoding objects and supervision through detection phrases. The performance of our model in 2D datasets closely matches or surpasses MDETR, while in the 3D domain much outperforms the naive MDETR-equivalent implementation as shown in extensive ablations. Moreover, it much surpasses the state-of-the-art in multiple 3D language grounding benchmarks. BEAUTY-DETR is also the first model in 3D referential grounding that operates on the realistic setup of not having access to oracle object proposals, but rather detects them from the input 3D point cloud.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. ReferIt3D: Neural Listen-

ers for Fine-Grained 3D Object Identification in Real-World Scenes. In Proc. ECCV, 2020. 2, 3, 6

[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 Proc. CVPR, 2018. 11

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to Compose Neural Networks for Question Answering. In Proc. NAACL, 2016. 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In Proc. ECCV, 2020. 3, 5

[5] Dave Zhenyu Chen, Angel Chang, and Matthias Nießner. ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language. In Proc. ECCV, 2020. 2, 3, 6

[6] Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. Meta Module Network for Compositional Visual Reasoning. In Proc. WACV, 2021. 3

[7] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: UNiversal Image-TExt Representation Learning. In Proc. ECCV, 2020. 3, 8

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In Proc. CVPR, 2017. 2, 4, 6, 11

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In Proc. CVPR, 2009. 11

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proc. NAACL, 2019. 4

[11] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From Captions to Visual Concepts and Back. In Proc. CVPR, 2015. 1, 3

[12] Mingtao Feng, Zhen Li, Qi Li, Liang Zhang, XiangDong Zhang, Guangming Zhu, Hui Zhang, Yaonan Wang, and Ajmal Mian. Free-form Description Guided 3D Visual Graph Network for Object Grounding in Point Cloud. In Proc. ICCV, 2021. 3, 6

[13] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In Proc. EMNLP, 2016. 1, 3

[14] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-Scale Adversarial Training for Vision-and-Language Representation Learning. In Proc. NeurIPS, 2020. 8

[15] Dailan He, Yusheng Zhao, Junyu Luo, Tianrui Hui, Shaofei Huang, Aixi Zhang, and Si Liu. TransRefer3D: Entity-and-Relation Aware Transformer for Fine-Grained 3D Visual Grounding. In Proc. ACMMM, 2021. 3, 6

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In Proc. ICCV, 2017. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In Proc. CVPR, 2016. 4, 11

[18] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to Reason: End-to-End Module Networks for Visual Question Answering. In Proc. ICCV, 2017. 3

[19] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling Relationships in Referential Expressions with Compositional Modular Networks. In Proc. CVPR, 2017. 1, 3

[20] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-Guided Graph Neural Networks for Referring 3D Instance Segmentation. In Proc. AAAI, 2021. 3, 6

[21] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General Perception with Iterative Attention. In Proc. ICML, 2021. 4

[22] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In Proc. CVPR, 2017. 3

[23] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Fei-Fei Li, C. Lawrence Zitnick, and Ross B. Girshick. Inferring and Executing Programs for Visual Reasoning. In Proc. ICCV, 2017. 3

[24] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In Proc. CVPR, 2016. 1, 3

[25] Aishwarya Kamath, Mannat Singh, Yann André LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. MDETR - Modulated Detection for End-to-End Multi-Modal Understanding. In Proc. ICCV, 2021. 1, 2, 5, 7, 8, 13

[26] Andrej Karpathy and Li Fei-Fei. Deep Visual-semantic Alignments for Generating Image Descriptions. In Proc. CVPR, 2015. 1, 3

[27] Fumi Katsuki and Christos Constantinidis. Bottom-up and Top-down Attention: Different Processes and Overlapping Neural Systems. The Neuroscientist, 20(5), 2014. 2

[28] Sahar Kazemzadeh, Vicente Ordonez, Marc André Matten, and Tamara L. Berg. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In Proc. EMNLP, 2014. 2, 3, 7

[29] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. International Journal of Computer Vision, 123, 2016. 4, 7, 11

[30] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. ArXiv, abs/1908.03557, 2019. 12, 13

[31] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In Proc. ICCV, 2017. 11

[32] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In Proc. ECCV, 2014. 2, 7

[33] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In Proc. ECCV, 2016. 3

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv, abs/1907.11692, 2019. 4

[35] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-Free 3D Object Detection via Transformers. In Proc. ICCV, 2021. 2, 3, 4, 5, 6, 11

[36] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In Proc. NeurIPS, 2019. 3, 5, 6, 7, 11

[37] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-Task Vision and Language Representation Learning. In Proc. CVPR, 2020. 5

[38] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical Question-Image Co-Attention for Visual Question Answering. In Proc. NIPS, 2016. 3

[39] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In Proc. ICLR, 2019. 3

[40] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana-Maria Camburu, Alan Loddon Yuille, and Kevin P. Murphy. Generation and Comprehension of Unambiguous Object Descriptions. In Proc. CVPR, 2016. 7

[41] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In Proc. ICCV, 2021. 3

[42] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In Proc. ICCV, 2015. 7, 8, 12

[43] Charles Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proc. NIPS, 2017. 3, 4, 6, 11

[44] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In Proc. CVPR, 2016. 3

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In Proc. NIPS, 2015. 1, 2, 4

[46] Seyed Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proc. CVPR, 2019. 5

[47] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. LanguageRefer: Spatial-Language Model for 3D Visual Grounding. In Proc. CoRL, 2021. 3, 6

[48] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In Proc. NIPS, 2017. 4, 6, 11

[49] Ke Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proc. ICML, 2015. 3

[50] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. SAT: 2D Semantics Assisted Training for 3D Visual Grounding. In Proc. ICCV, 2021. 3, 6, 7

[51] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: CoLlision Events for Video REpresentation and Reasoning. In Proc. ICLR, 2020. 3

[52] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In NeurIPS, 2018. 3

[53] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling Context in Referring Expressions. In Proc. ECCV, 2016. 2, 7

[54] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Zhen Li, and Shuguang Cui. InstanceRefer: Cooperative Holistic Understanding for Visual Grounding on Point Clouds through Instance Multi-level Contextual Referring. In Proc. ICCV, 2021. 3, 6

[55] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting Visual Representations in Vision-Language Models. In Proc. CVPR, 2021. 1

[56] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3DVG-Transformer: Relation Modeling for Visual Grounding on Point Clouds. In Proc. ICCV, 2021. 3, 6

[57] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Ching-Feng Lin, and Jifeng Dai. An Empirical Study of Spatial Attention Mechanisms in Deep Networks. In Proc. ICCV, 2019. 3

[58] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In Proc. ICLR, 2021. 2, 3, 4, 5, 6, 11

A. Appendix

A.1. Implementation details

We report here architecture choices as well as training hyperparameters. We implement BEAUTY-DETR in PyTorch. For the 3D version, the point cloud is encoded with PointNet++ [43] using the same hyperparameters as in [35], pre-trained on ScanNet [8]. We use the last layer's features, resulting in 1024 visual tokens. In the cross-modality encoder, instead of allowing the visual features to attend to the box features, we directly concatenated the box features to the input point cloud. Specifically, for all the points that lie inside a box, we concatenate this box's features directly to their point features (xyz and color). If a point lies inside multiple boxes, we randomly sample one box's features. Points that do not lie inside inside any box are padded with zeros. This is computationally cheaper than cross-attending visual features to box features and works well in 3D since the objects do not intersect. In 2D, however, it does not work well since the objects and thus their boxes overlap a lot and hence usually a pixel falls inside multiple boxes. We ablate more on this fusion in A.3. In the decoder, the queries are formed from the top 256 most confident visual tokens. To compute this confidence score, each visual token is fed to an MLP to give a scalar value. We supervise these values using Focal Loss [31]. Specifically, since each visual token corresponds to a point with known coordinates, we associate visual tokens to ground-truth object centers and keep the 4 closest points to each center. We consider these matched points as positives, i.e. here points with high ground-truth objectness. The same scoring method is employed in [35]. We set $N_E = 3$ with no self-attention layers, $N_D = 6$. All attention layers are implemented using standard self-/cross-attention [36, 48].

For the 2D version, the image is encoded using ResNet-101 [17] pretrained on ImageNet [9]. We use multi-scale features as in [58]. The feature maps of the different scales are flattened and concatenated in the spatial dimension, leading to 17821 visual tokens. The feature dimension of each token is 256. To obtain the box proposals, we use the detector of [2] trained on 1601 classes of Visual Genome [29]. The detected boxes are encoded using their spatial and categorical features. Specifically, we compute the 2D Fourier features of each box and feed them to an MLP, then we concatenate this vector with a learnable semantic class embedding and feed to another MLP to obtain the box embeddings. To form queries, we rank visual tokens based on their confidence score and keep the 300 most confidence ones. This confidence layer is supervised using Focal Loss [31]: we assign a positive objectness scores to every point that lies inside a ground-truth answer box. We set $N_E = 6$ and $N_D = 6$. All attention layers to the visual stream are implemented with deformable attention [58], attention to either the language stream or detected boxes is the standard attention of [36, 48].

For the 3D model, we use a learning rate of $1e{-}5$ for RoBERTa and $1e{-}4$ for all other layers. We are able to fit a batch size of 6 on a single GPU of 12GB. Under these conditions, each epoch takes around 3 hours. For the 2D model, we use a learning rate of $1e{-}6$ for Resnet101 visual encoder, $5e{-}6$ for RoBERTa text encoder and $1e{-}5$ for rest of the layers. We pre-train on 64 V100s with a batch size of 1, and finetune on RefCOCO/RefCOCO+ with a batch size of 2 on 16 V100s. The total training time is included in the respective tables. We will release pre-trained checkpoints for both 3D and 2D models.

A.2. Negative training with detection phrases

We devise object detection as language grounding of an utterance formed by concatenating a sequence of category labels, e.g. “Chair. Dining table. Bed. Plant. Sofa.”. The task is again to i) detect the mentioned objects in the scene, i.e. return bounding boxes of their instances, and ii) associate each localized box to a span, i.e. an object category in the utterance.

To form these detection phrases, one solution could be to concatenate all object classes into a long utterance. However, this can be impractical if the domain-vocabulary is “open”, or, in practice, very large (485 classes in ScanNet, 1600 in Visual Genome and so on). Instead, assuming that we have object annotations, we sample out of the positive labels that are annotated for a scene and a number of negative ones, corresponding to class names that do not appear in the scene. Having negative classes in the detection phrases helps the precision of the model, as it learns not to fire for every noun phrase that appears in an utterance. More specifically, the text-query contrastive losses described in the main paper push the negative class' text representation away from the query representation of existing objects.

MDETR also considers an object detection evaluation. However, there are two noticeable differences. First, they use only single-category utterances, e.g. “Dog.”. This category can be either positive (appears in the annotations) or negative (does not appear in the annotations), according to a sampling ratio. Opposite to that, our detection phrases are longer, consisting of multiple object categories, both positive and negative. Second, MDETR employs these sentences after pre-training, to train and evaluate their model as an object detector. Instead, we mix detection phrases through the training, leading to considerable quantitative gains in both 3D and 2D.

Lastly, although the ratio r of positive to negative classes that appear in a detection phrase is a hyperparameter, we report results only for $r = 2$ and sample at most 8 positive classes. We leave tuning of this hyperparameter for future research.

Model	Accuracy
box features only	44.2
box features and logits	43.1
box features and class embeddings	**48.5**

Table 6. **Ablation for BEAUTY-DETR on the SR3D validation set**. We compare between different box encoding choices.

Model	Accuracy
separate stream	45.7
concatenated to point cloud	**48.5**

Table 7. **Ablation for BEAUTY-DETR on the SR3D validation set**. We compare two fusion techniques between the visual and box stream in the encoder.

A.3. Additional ablations on 3D

We first ablate on how to encode the bounding box stream. We consider three options: i) bounding box features alone, ii) bounding box features and soft logits obtained from the detector, iii) bounding box features and class embeddings, which is the approach we use in BEAUTY-DETR . In all cases, the layers used to encode the boxes are the same, as described in the main paper. To encode logits, we apply softmax and a linear layer that map the 485-d vectors to 32-d. Then we use this vector as the "class embedding", identical to how we handle class embeddings in case iii. The comparison is shown in Table 6. Combining box features and class embeddings gives the best performance. The model that uses logits underperforms, possibly because the predicted logits for training and testing come from different distributions: the detector is overconfident in the training set (giving more peaky distributions) but less confident on the test set (resulting in more smoothed distributions).

We also ablate on how to attend to the box stream in the encoder phase. We experiment with a) having boxes as a separate stream and allowing visual tokens to cross-attend to it; or b) append directly to every point in the cloud the features of a box that contain it, padding with zeros for points that do not lie inside any box, as described in A.1, which is what we use in the 3D version of BEAUTY-DETR . Appending features to the point cloud works better, but attending to a separate stream of boxes still largely outperforms the highest-performing competitor in the literature (LanguageRefer with 39.5%). In our 2D implementation, appending box features to pixels did not work, probably because of significant overlap between multiple object proposals.

A.4. Additional Results on 2D Language Grounding

We test BEAUTY-DETR on Flickr30k entities dataset [42]. Given an utterance about an image, the task is to predict bounding boxes for all the objects mentioned. Note that unlike the referential grounding task we show in the main paper and is evaluated on RefCOCO/RefCOCO+, the task here is to find *all* objects mentioned in the utterance and not only the root object. Following MDETR, we directly evaluate our pre-trained model on Flickr30k without any further fine-tuning. For evaluation, we follow the ANY-BOX protocol [30] and evaluate our performance in terms of Recall metric on standard val and test splits. The results are shown in Table-8. We achieve results comparable with state-of-the-art MDETR model while converging in less than half the number of GPU hours.

A.5. More qualitative results

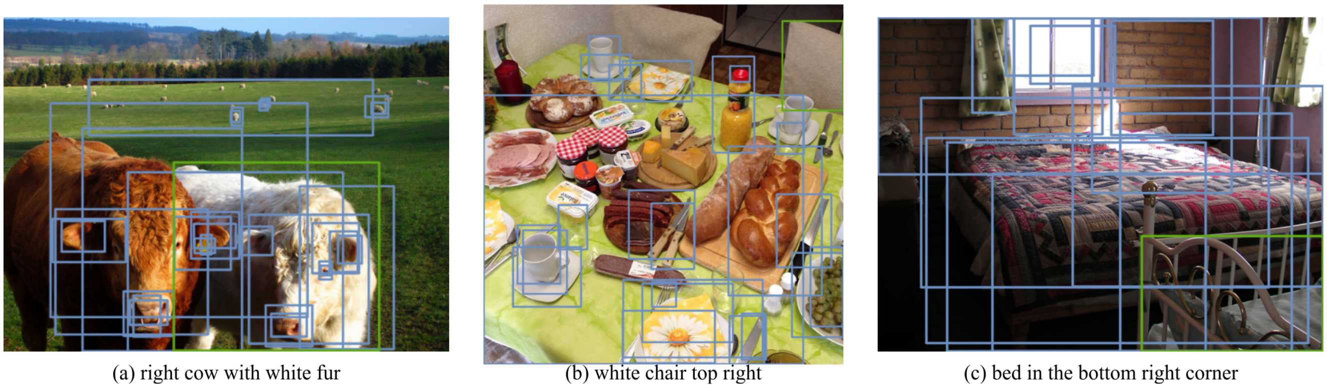
We show qualitative results of the 2D version of BEAUTY-DETR on RefCOCO in Figure 4. We also show failure cases on SR3D in Figure 5. More qualitative results on other datasets are shown in Figures 6, 7, 8.

A.6. Ethical Impact

The datasets used in this study, especially the 2D ones, have some harmful biases associated to themselves. There is a dire need to address and mitigate these biases as much as possible. Hence, we advise the readers to proceed with caution when using our model in a production pipeline.

Method	Val			Test			Training	Training
	R@1	R@5	R@10	R@1	R@5	R@10	Epochs	GPU hours
VisualBERT [30]	70.4	84.5	86.3	71.3	85.0	86.5	-	-
MDETR [25]	**82.5**	**92.9**	**94.9**	**83.4**	**93.5**	**95.3**	40	5480
BEAUTY-DETR (ours)	80.4	90.0	92.1	81.0	90.9	92.9	**11**	**2464**

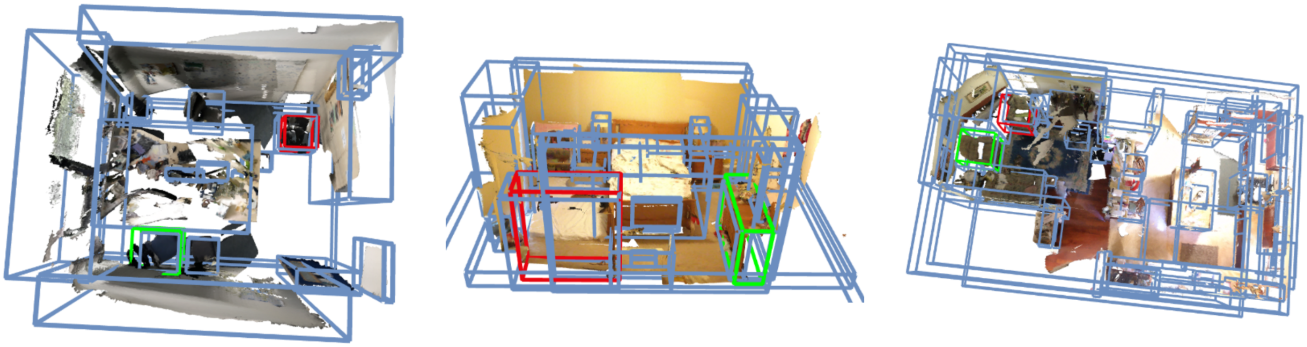
Table 8. **Results on language grounding in Flickr30k 2D images** using Recall@k metric and computational efficiency. All training times are computed using same V100 GPU machines.


(a) right cow with white fur

(b) white chair top right

(c) bed in the bottom right corner

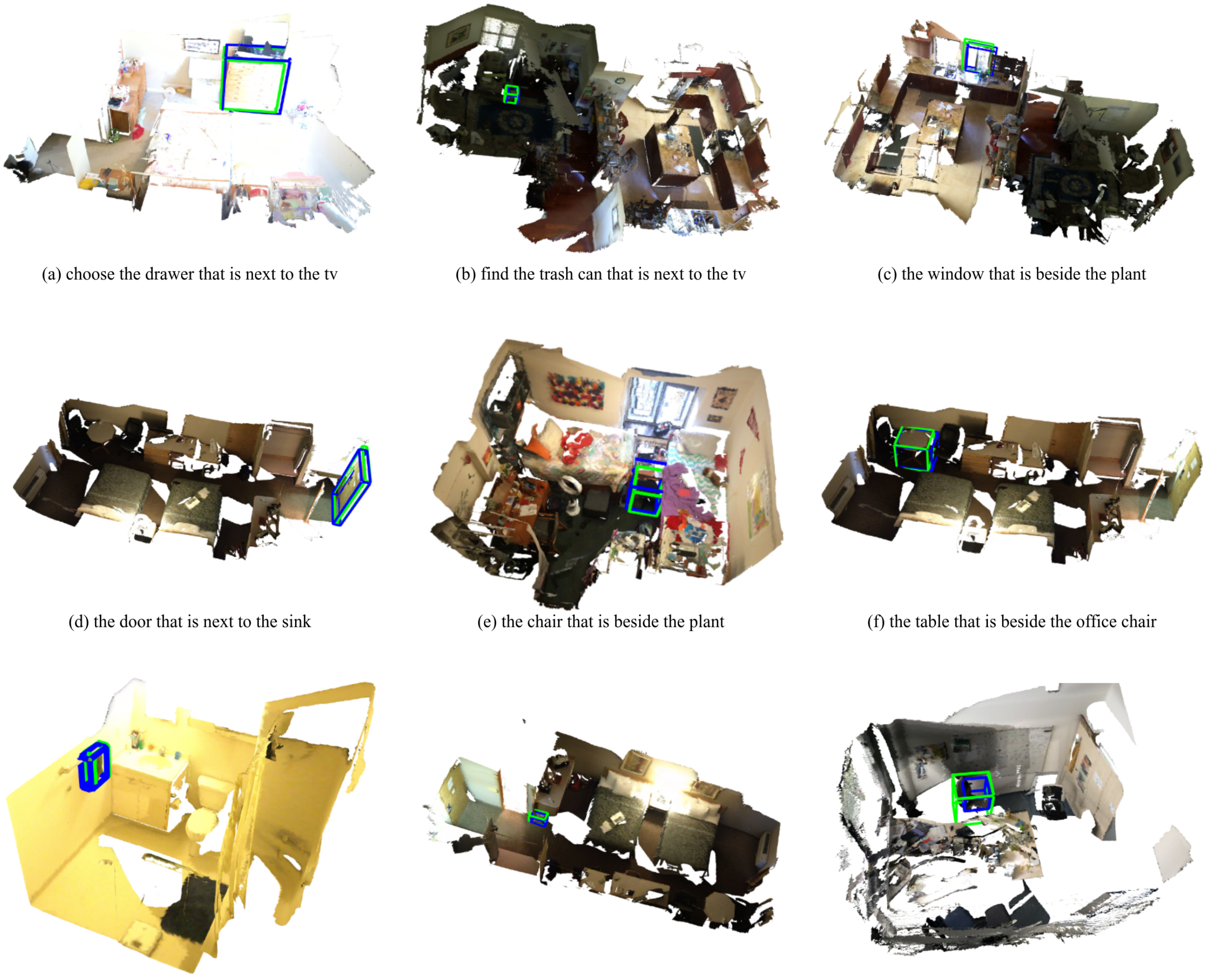
Figure 4. **Qualitative results of BEAUTY-DETR on RefCOCO.** The detector's proposals are shown in blue, our model's prediction in green. BEAUTY-DETR can predict boxes that the detector misses, e.g. in (b), the chair is missed by the detector so none of the previous detection-bottlenecked approaches could ground this phrase. In (a) and (c) the detector succeeds with low IoU but BEAUTY-DETR is able to predict a tight box around the referent object.


(a) the office chair that is beside the chair

(b) find the dresser that is next to the trash can

(c) find the armchair that is next to the table

Figure 5. Failure cases of BEAUTY-DETR on SR3D. Our predictions with red, ground-truth with green. Even if the box is there, still our model can fail, proving that ranking the correct boxes over other proposals remains a hard problem.



(a) choose the drawer that is next to the tv

(b) find the trash can that is next to the tv

(c) the window that is beside the plant

(d) the door that is next to the sink

(e) the chair that is beside the plant

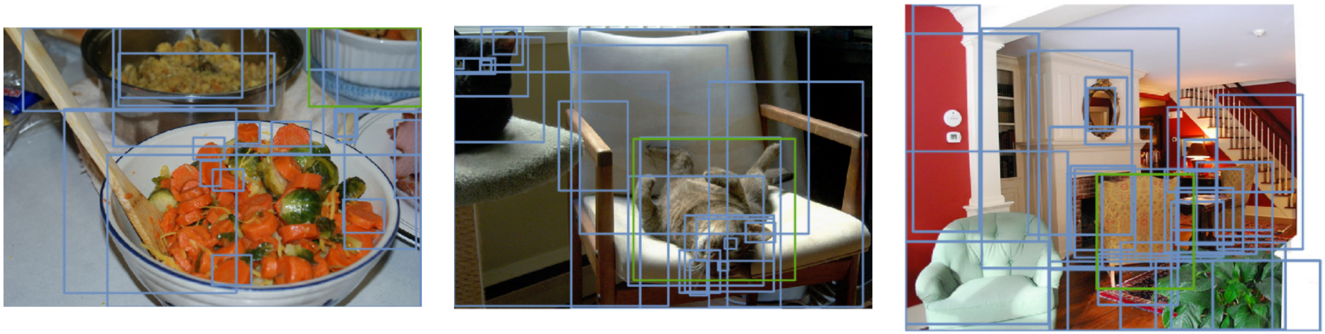
(f) the table that is beside the office chair

(g) choose the towel that is beside the toilet paper

(h) find the trash can that is next to the mirror

(i) office chair next to the bulletin board

Figure 6. More qualitative results of BEAUTY-DETR on SR3D. Our predictions are shown in blue, ground-truth in green.

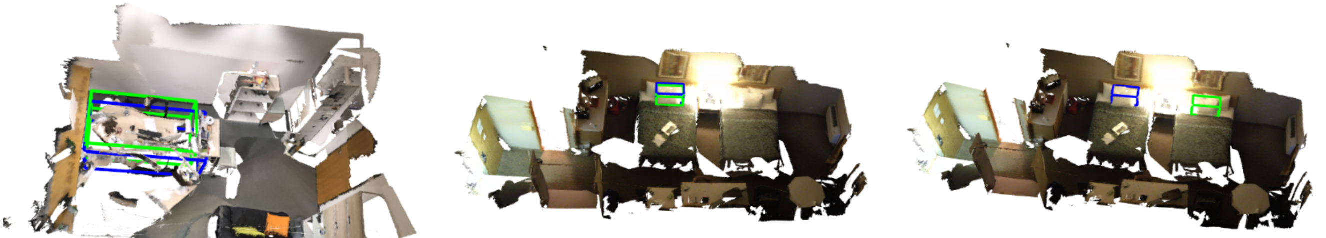


(a) dish in top right corner

(b) cat lying upside down

(c) brown chair on left near the fireplace

Figure 7. More qualitative results of BEAUTY-DETR on RefCOCO. Our predictions are shown in green, detected boxes in blue.



(a) choose the desk on the opposite side of the couch

(b) Facing the beds you want the front pillow on the left bed.

(c) Standing at the foot of the bed looking at the head on the bed. It is the pillow in the from on the bed on the right.

Figure 8. Qualitative results of BEAUTY-DETR on NR3D. Our predictions are shown blue, ground-truth in green. The language of NR3D is more complex and the utterances are longer. Case (c) is a failure case.