Supplementary Material for "Explore and Tell: Embodied Visual Captioning in 3D Environments"

Anwen Hu[1], Shizhe Chen[2], Liang Zhang[1], Qin Jin[1]*
[1]School of Information, Renmin University of China
[2]INRIA
{anwenhu,zhangliang00,qjin}@ruc.edu.cn
shizhe.chen@inria.fr

We present more details about dataset construction and statistic in Appendix A. Details about the in-domain object detector, Template-based Captioner and implementation of CaBOT can be found in Appendix B. The captioning ablation with predicted trajectory is presented in Appendix C. The detailed ablation study about how to leverage instance recognition knowledge from the object detector is shown in Appendix D. More qualitative results generated by CaBOT can be found in Appendix E.

A. Dataset Details

A.1. Scene Simulation

The details of the three steps for scene simulation are introduced below.

Instance Resizing. The size of 3D models loaded directly from Kurbic is not consistent with human common sense. For example, a 'cup' instance can be as big as a 'table' instance. To ensure scenes are more realistic, we design heuristic resizing rules for different categories of 3D models. For example, 'bed' instances are resized to guarantee that the length is about 2.0m; 'skateboard' instances are resized to a length close to 0.6m; the length of 'bookshelf' instances is about 1.0m and their height should be lower than 2.0m.

Instance Selection. Our indoor environment is organized by arranging small objects around big furniture such as beds, tables, etc. To simulate such indoor environment, we set one big furniture namely base instance, and multiple relatively smaller objects namely placing instances. We manually divide the categories of instances into base categories and placing categories. For each scene, a base category and multiple placing categories are firstly selected. Due to the long tail distribution of different categories, simply sampling categories according to the number of instances under the category can lead to scenes with low diversity. Therefore, we empirically set a upper bound for sampling probability of each category when performing the category selection. Once the categories of a scene are determined, we randomly select an instance under each category.

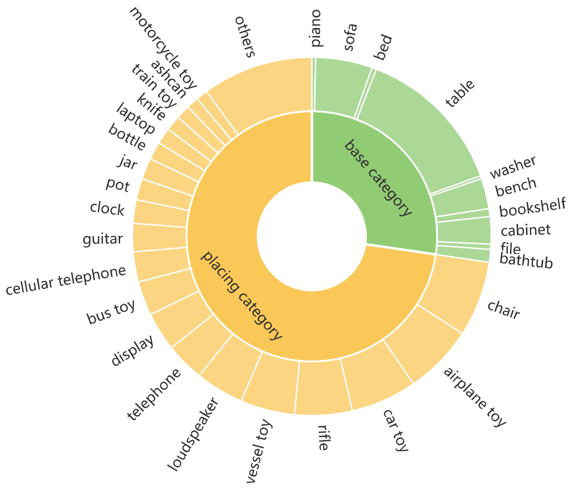


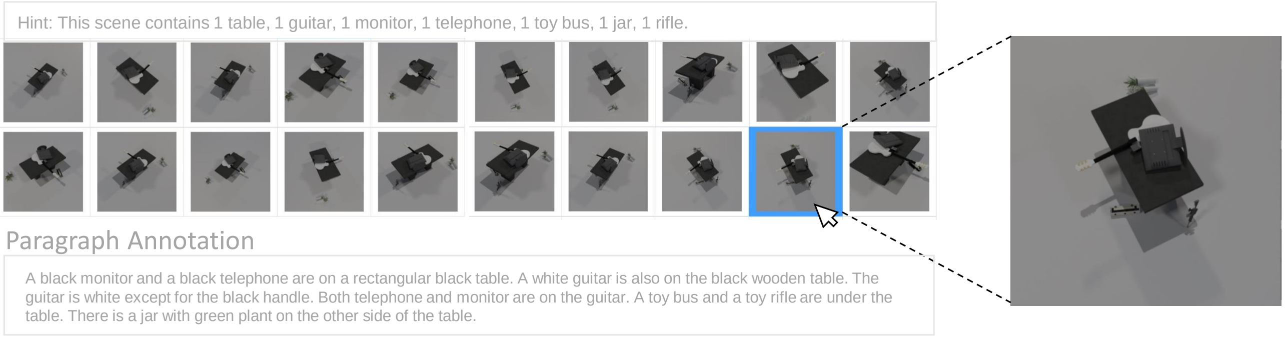
Figure 1. Category distribution of instances.

Instance Placing. In addition to the category of instances, the position of instances is another crucial factor for scene diversity. It is difficult and tedious to manually design rules to place each instance. In this work, we perform physic simulation with the open-source PyBullet physics engine to place instances automatically. Concretely, the base instance is first placed in the center of the scene. The rest instances are then placed at 0.5m higher above the base instance and fall simultaneously due to gravity. Because of the collision between instances, the final orientations and locations of instances vary a lot across scenes.

A.2. Annotation Details

The ET-Cap dataset contains good viewpoints and informative descriptions for each scene. Fig. 2 presents the an-

*Corresponding Author.

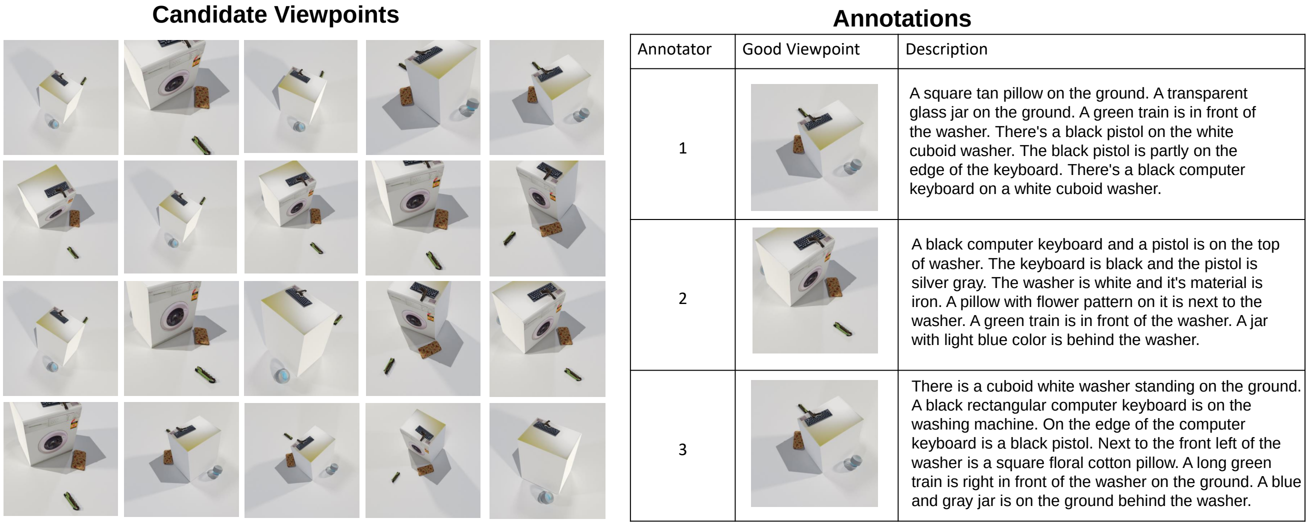


Hint: This scene contains 1 table, 1 guitar, 1 monitor, 1 telephone, 1 toy bus, 1 jar, 1 rifle.

Paragraph Annotation

A black monitor and a black telephone are on a rectangular black table. A white guitar is also on the black wooden table. The guitar is white except for the black handle. Both telephone and monitor are on the guitar. A toy bus and a toy rifle are under the table. There is a jar with green plant on the other side of the table.

Figure 2. The annotation interface of viewpoint selection and paragraph annotation. The blue box denotes the selected viewpoint by the annotator. Each image can be enlarged to see instance details.



Candidate Viewpoints

Annotations

Annotator	Good Viewpoint	Description
1		A square tan pillow on the ground. A transparent glass jar on the ground. A green train is in front of the washer. There's a black pistol on the white cuboid washer. The black pistol is partly on the edge of the keyboard. There's a black computer keyboard on a white cuboid washer.
2		A black computer keyboard and a pistol is on the top of washer. The keyboard is black and the pistol is silver gray. The washer is white and it's material is iron. A pillow with flower pattern on it is next to the washer. A green train is in front of the washer. A jar with light blue color is behind the washer.
3		There is a cuboid white washer standing on the ground. A black rectangular computer keyboard is on the washing machine. On the edge of the computer keyboard is a black pistol. Next to the front left of the washer is a square floral cotton pillow. A long green train is right in front of the washer on the ground. A blue and gray jar is on the ground behind the washer.

Figure 3. An example of good viewpoint selection and paragraph annotation by three annotators.

notation interface. For each scene, the annotator is provided with 20 images rendered from candidate viewpoints and a hint about the category of instances in the scene. The hint is a template `This scene contains [number`$_1$`] [category`$_1$`], [number`$_2$`] [category`$_2$`], ..., [number`$_k$`] [category`$_k$`].`” filled with instance categories and counts. After selecting a good image, annotators should write a paragraph description at the bottom of the annotation interface. As shown in Fig. 3, each scene is annotated by three workers. They may choose the same viewpoints but write different descriptions.

In total, we hired 21 adults (8 males and 13 females) from 9 different cities to do the annotation. The annotators are proficient in English. It takes 2 months and costs about 5,700 dollars to complete annotating ET-Cap.

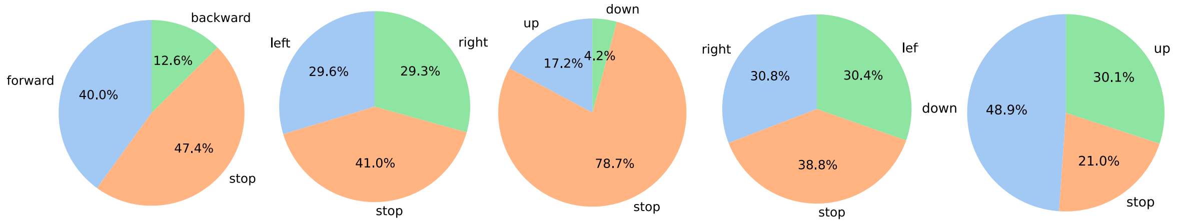
A.3. Dataset Statistics

As shown in Fig. 1, instance categories are diverse overall. The ‘table’ instance is relatively more than other categories of instances because it is the most common indoor furniture. Fig. 4 shows the direction distribution of each action type in ground-truth trajectories.

B. Method Details

B.1. In-domain Object Detector Details

To study whether the object detection ability contributes to Embodied Captioning task, we train an in-domain DETR[1] model and leverage its parameters in our model, as mentioned in Sec. 4.1 and Sec. 4.2. To train the in-domain detector, we construct extra 4,398 scenes like ET-Cap and render 87,960 images with object category and bounding boxes annotation given by Kubric simulator. The model is trained for 46 epochs and achieves 39.4 mAP on a test set, which includes 150 scenes with 3,000 images.



(a) forward-backward move (b) left-right move (c) up-down move (d) heading rotate (e) elevation rotate

Figure 4. Distribution of forward-backward move actions (a), left-right move actions (b), up-down move actions (c), heading rotate actions (d) and elevation rotate actions (e).

Table 1. Captioning ablation study with trajectories predicted by the CaBOT navigator. 'single' means only utilizing the end view for region-level cross-attention, and 'multi' means using the mean view of all observations.

	Region-level CrossAtt	Trajectory-Level CrossAtt	Init Backbone	BLEU4	METEOR	ROUGE-L	CIDEr	SPICE	BLEU4^l	METEOR^l	ROUGE-L^l	CIDEr^l	SPICE^l
r1	single	×	×	24.71	20.31	44.37	27.14	13.66	18.58	18.20	37.96	23.22	11.70
r2	multi	×	×	24.70	22.06	43.38	28.50	16.06	16.87	19.37	37.10	24.34	13.74
r3	×	✓	×	25.40	23.16	44.35	36.98	17.60	18.37	20.47	37.92	31.49	15.05
r4	single	✓	×	24.97	22.18	43.66	29.00	16.47	17.49	19.50	37.39	24.80	14.12
r5	multi	✓	×	26.13	23.14	45.09	37.87	18.36	18.98	20.44	38.57	32.29	15.72
r6	multi	✓	✓	**26.45**	**23.56**	**45.48**	**40.83**	**19.43**	**19.22**	**20.82**	**38.87**	**34.88**	**16.62**

B.2. Baseline Details

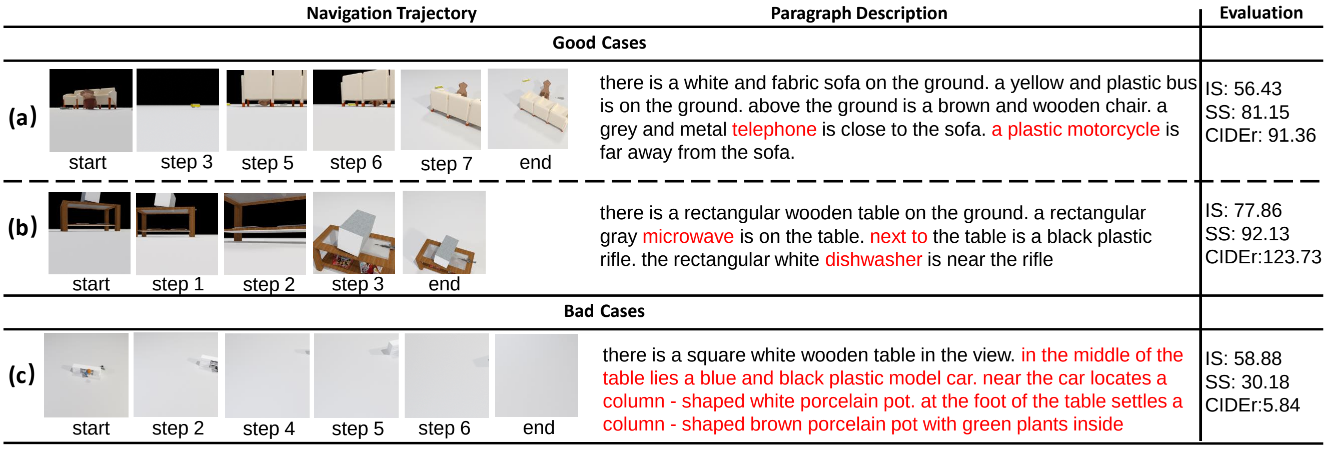
Template-based Captioner first generates descriptions for each viewpoint by fill-in templates and then merges them into a trajectory description. For each viewpoint, it detects salient objects with the in-domain DETR model. Objects with a confidence score higher than 0.9 are selected to construct the description. We attach each object with a color attribute according to its RGB values. After detecting the objects, it first describes the largest object `[obj`$_L$`]` by selecting a template like "`There is a [obj`$_L$`] in the scene.`". Then it describes other objects one by one in descending order of bounding box size. For a smaller object `[obj`$_s$`]`, it describes its spatial relation with a randomly selected larger object `[obj`$_l$`]` with templates such as `A/An [obj`$_s$`] is [Relation] the [obj`$_l$`].`". Note that `[obj`$_l$`]` has occurred in one of the former sentences. The spatial relation is predicted by the relative positions between objects similar to Preposition Functions [2]. After generating captions for each viewpoints, it further ensembles all captions into a trajectory description as follows: 1) choose the best viewpoint with the highest summation of object confidences, and treat its description as the basic description; 2) describe objects that do not occur in the best viewpoint with a template "`There is also a [obj`$_1$`], a [obj`$_2$`], ..., and a [obj`$_n$`] in the view.`". We design multiple synonymous templates according to the style of the ground-truth paragraphs as shown in Tab. 2. We randomly select one template for each object during inference.

Table 2. Templates used in template-based captioner.

Describe objects	Templates
Largest	A/An [obj$_L$] in on the ground. A/An [obj$_L$] is in the view. There is a/an [obj$_L$]. There is a/an [obj$_L$] on the ground. There is a/an [obj$_L$] in the view. On the ground is a/an [obj$_L$]. In the view there is a/an [obj$_L$].
Others	A/An [obj$_s$] is [Relation] the [obj$_l$]. [Relation] the [obj$_l$] is a/an [obj$_s$].
Not occurs	There is also a [obj$_1$], a [obj$_2$], ..., and a [obj$_n$] in the view.

B.3. Implementation Details

In the navigator, the number of transformer layers of the Region Encoder, Historical Vision Encoder and Historical Action-informed Decoder is 2, 1, 1, respectively. The transformer layer number of the Bi-CrossAttention Decoder in the captioner is set as 2. The maximum trajectory length is set as 12. The maximum caption length is set as 77. The navigation module and the captioning module are trained with 10 epochs and 20 epochs, respectively. Both two modules are trained with the AdamW [3] optimizer. The learning rate for the navigation module and captioning module is $1e^{-4}$ and $3e^{-5}$, respectively. The batch size of these two modules is set as 32.



(a) there is a white and fabric sofa on the ground. a yellow and plastic bus is on the ground. above the ground is a brown and wooden chair. a grey and metal telephone is close to the sofa. a plastic motorcycle is far away from the sofa.

IS: 56.43
SS: 81.15
CIDEr: 91.36

(b) there is a rectangular wooden table on the ground. a rectangular gray microwave is on the table. next to the table is a black plastic rifle. the rectangular white dishwasher is near the rifle

IS: 77.86
SS: 92.13
CIDEr:123.73

(c) there is a square white wooden table in the view. in the middle of the table lies a blue and black plastic model car. near the car locates a column - shaped white porcelain pot. at the foot of the table settles a column - shaped brown porcelain pot with green plants inside

IS: 58.88
SS: 30.18
CIDEr:5.84

Figure 5. More qualitative results of CaBOT on Embodied Captioning test set. Wrong descriptions are marked in red.

Table 3. Ablation study about initialization and training strategy. Oracle-trajectory Captioning are tested withthe captioner of CaBOT. 'DeInit' means whether using transformer encoder of DETR to initialize region encoders of the navigator or the captioner.'RN' and 'EN' refer to ResNet and Transformer Encoder of DETR.

	Backbone		Region En		OracleCap	
	Init	Frozen	DeInit	Frozen	BLEU4↑	CIDEr↑
r1	ResNet50	×	×	×	27.58	42.10
r2	Detr RN	✓	×	×	27.30	40.91
r3	Detr RN	×	×	×	27.62	**45.24**
r4	Detr RN	×	✓	×	**27.94**	44.85
r5	Detr RN	✓	✓	✓	26.33	35.30
r6	Detr RN+En	×	×	×	27.08	43.73
r7	Detr RN+En	✓	×	×	26.78	38.66

C. Captioning Ablation with Predicted Trajectory

In Sec 5.2, we perform captioning ablation study with oracle trajectory to verify the effectiveness of our Bi-CrossAttention Decoder and DETR initialization. To verify that they are also crucial for Embodied Captioning, we further perform an ablation study with trajectory predicted by the CaBOT navigator, as shown in Tab. 1. Firstly, utilizing the mean view of the predicted trajectory outperforms the one utilizing only the end view (r2 vs r1), especially on CIDEr and SPICE. Besides, leveraging the predicted trajectory at the trajectory level is better than at the region level (r3 vs r2). These also support that merging visual information at the trajectory level is better for scene description. Besides, Bi-CrossAttention outperforms single region-level or trajectory-level cross-attention (r5 vs r2, r5 vs r3), initializing the backbone with pre-trained DETR improves the captioning performance (r6 vs r5), both of which are also consistent with the observations in Sec 5.2.

D. Ablation of Instance Recognition Ability

As mentioned in Sec.5.2, knowledge about object recognition from the in-domain DETR model boosts captioning performance. In this section, we compare different strategies of initialization and training to explore how to best leverage this knowledge for Embodied Captioning. As show in Table 3, for the captioner, initializing its backbone with only ResNet of DETR and optimizing it during training is best (r3). It indicates that for captioning, besides category and attributes of each instance, it is also crucial to understand their spatial relationships. Therefore, only leveraging basic knowledge of object detection model and optimizing it during captioning training benefits the captioner more.

E. More Qualitative Results

Fig. 5 shows more qualitative results given by CaBOT. The case (a) shows that the navigator could explore the environment and find a small and relatively far away instance (the yellow toy bus). The captioner could combine vision information provided from different viewpoints to generate the description. In case (c), the agent reaches bad viewpoints during navigation and can not find better viewpoints or return to original viewpoints anymore. This shows that though historical vision and camera information have been leveraged by the navigator, it may still be confused when no instance is visible. Besides, when there are little visual instance information provided in the trajectory, the captioner randomly generates some irrelevant descriptions.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020. 2

[2] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2891–2903, 2013. 3

[3] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net, 2019. 3