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

Abstract

While current visual captioning models have achieved impressive performance, they often assume that the image is well-captured and provides a complete view of the scene. In real-world scenarios, however, a single image may not offer a good viewpoint, hindering fine-grained scene understanding. To overcome this limitation, we propose a novel task called Embodied Captioning, which equips visual captioning models with navigation capabilities, enabling them to actively explore the scene and reduce visual ambiguity from suboptimal viewpoints. Specifically, starting at a random viewpoint, an agent must navigate the environment to gather information from different viewpoints and generate a comprehensive paragraph describing all objects in the scene. To support this task, we build the ET-Cap dataset with Kubric simulator, consisting of 10K 3D scenes with cluttered objects and three annotated paragraphs per scene. We propose a Cascade Embodied Captioning model (CaBOT), which comprises of a navigator and a captioner, to tackle this task. The navigator predicts which actions to take in the environment, while the captioner generates a paragraph description based on the whole navigation trajectory. Extensive experiments demonstrate that our model outperforms other carefully designed baselines. Our dataset, codes and models are available at https://aim3-ruc.github.io/ExploreAndTell.

1. Introduction

Visual captioning [14, 22, 32, 18, 9] is an essential vision-and-language task which aims to generate natural language descriptions of visual contents. In recent years, many captioning models have been developed and achieved significant improvements in describing major objects and relationships in an image [41, 4, 28, 25, 46, 23, 39]. However, the current models typically rely on well-captured images that provide a good viewpoint of the scene. Unfortunately, in real-world scenarios, capturing such images may not always be feasible. As illustrated in Fig. 1, the initial position to capture an image may not provide a complete view of the scene, and a single image may not be sufficient to capture all objects within it, potentially leading to incomplete or inaccurate visual captions. To address this limitation, it is essential for visual caption models to navigate the environment actively and gather information from multiple viewpoints in order to generate more comprehensive and accurate visual captions.

In this paper, we propose a novel task called Embodied Captioning, which integrates such navigation ability into visual captioning. The task requires an agent, which starts at a random viewpoint in a 3D environment, to navigate the environment to reduce visual ambiguity of the scene, and finally generate a comprehensive paragraph that describes all objects in the scene. Different from existing navigation tasks [5, 15, 44, 30, 24], our Embodied Captioning task does not explicitly define a target location to navigate. Instead, we have an implicit target which is to accurately recognize all objects along with their attributes and relationships in

*Corresponding Author.
the scene as soon as possible.

To support the Embodied Captioning task, we build a high-quality 3D dataset ET-Cap with manually annotated paragraph descriptions. Leveraging high-quality 3D object assets from ShapeNet [11] and GSO [16], we use Kubric [17] to construct 10,000 scenes. For each scene, three annotators are provided with 20 images from different viewpoints of the scene and asked to write a detailed paragraph to describe all visible instances. We also require the annotators to select images with good viewpoints among all the image candidates. Although our dataset is based on synthetic scenes with limited obstacles, it still presents significant challenges for Embodied Captioning. The agent only receives an RGB image of a restricted field of view at each step without any localization information, such as the location of the agent and objects. Therefore, a model must be equipped with long-term memories of previous visual observations and actions in order to efficiently explore the environment, accurately recognize objects, and generate fine-grained scene descriptions.

To address these challenges, we propose a Cascade Embodied Captioning model (CaBOT), which consists of a History-aware Navigator and a Trajectory-aware Captioner. The navigator leverages histories of both observed images and performed actions to predict the next action via a transformer model. The captioner is fed with all images in the predicted trajectory and utilizes a bi-level cross attention former model. The captioner is fed with all images in the predicted trajectory and utilizes a bi-level cross attention former model. The captioner is fed with all images in the predicted trajectory and utilizes a bi-level cross attention former model. The captioner is fed with all images in the predicted trajectory and utilizes a bi-level cross attention former model. The captioner is fed with all images in the predicted trajectory and utilizes a bi-level cross attention former model. The captioner is fed with all images in the predicted trajectory and utilizes a bi-level cross attention former model.

In summary, our contributions are three-fold:

- We propose a novel and challenging Embodied Captioning task which requires agents to explore in 3D environments to generate better visual descriptions.
- A high-quality dataset is constructed to benchmark the Embodied Captioning task, with 10K synthetic 3D scenes and 24K manually annotated good viewpoints and 30K paragraph descriptions.
- We present a Cascade Embodied Captioning model which incorporates navigation histories for captioning, providing a strong starting point for future work.

2. Related Work

**Visual Captioning.** Visual captioning aims to describe visual contents with natural language. A variety of visual captioning tasks have been proposed such as image captioning [14, 22, 32, 18, 9], video captioning [40, 48, 38, 37] and 3D captioning [12]. In image captioning, a model should describe major objects, attributes, relations and even named entities [9] and scene texts [32, 21] in the image. Most of the image captioning datasets [27, 22, 32, 9] use high-quality web images. Such dataset bias results in poor generalization when an image is not well taken as shown in VizWiz-Caption dataset [18] which contains photos taken by the blind. Video captioning tasks [40, 48, 38, 37] take a video as input, and focus more on the actions or events. 3D captioning task [12] is designed to describe a 3D indoor scene with almost complete point clouds as input. Much progress has been made to improve the captioning performance such as attention mechanism [41, 4, 20], transformer architecture [28, 45, 33] and pretraining framework [47, 23, 25, 42, 39]. In this work, we aim to push the standard visual image captioning tasks one step further, which do not passively receive an visual input, but should actively explore in the environment to obtain better visual observations to describe.

**Embodied Vision and Language Tasks.** Growing research attentions have been paid to embodied vision and language tasks [5, 15, 44, 30, 34, 24] in recent years, which require agents to perform actions to achieve various goals specified in natural language. Embodied visual referring expression [30, 24] provides object-oriented high-level instructions and requires both navigation and object grounding. In embodied question answering [15, 44], a question is provided to the agent such as ‘What room is the microwave located in?’, and the agent should firstly find relevant visual information and then generate the answer. Compared with the above tasks, our proposed Embodied Captioning does not provide explicit goals for navigation. In contrast, there is only an implicit goal that the agent should generate more informative captions after its navigation.

The most similar work to ours is embodied scene description (ESD) [8, 7, 34]. Our work is distinguished from ESD in two main aspects. Firstly, the main goal of navigation in ESD is to enlarge the explored area or to find a single viewpoint that captures more objects, while our navigation goal is more entangled with visual captioning to enable to generate accurate and comprehensive captions as soon as possible. Secondly, existing ESD work simply uses pre-trained image captioning models and does not have an in-domain dataset to benchmark the task with systematic evaluation. In this work, we construct a high-quality dataset with human-annotated viewpoints and paragraph descriptions and we also provide multiple strong baselines.

3. ET-Cap Dataset

We build a high-quality synthetic dataset ET-Cap with human annotations to teach agents how to Explore and Tell in 3D environments.

2483
3.1. 3D Scenes Simulation

3D scenes in ET-Cap are constructed with Kubric [17], a recently proposed framework that enables the generation of photo-realistic scenes. We use Kubric to place multiple objects in a scene and render images. 3D object assets come from ShapeNet [11] and Google Scanned Objects [16].

Each scene in ET-Cap is constructed following three steps: Instance Resizing, Instance Selection, and Instance Placing. Specifically, in order to make the scene more realistic, we first resize 3D object assets to their common sizes in real life. Since the indoor environment is often organized by arranging small objects around big furniture, we then randomly choose a big-size furniture instance (called a base instance) and multiple relatively smaller instances (called placing instances). Finally, to ensure the diversity of instance arrangement, we place instances by performing a physic simulation. The base instance is first placed at the center of the scene. The other instances are then dropped above the base instance, leading to diverse poses of instances across scenes because of collision. More details can be found in the supplementary material.

Navigation Space. We discretize the whole environment as a 3D grid world with the total size of $8 \times 8 \times 4 \ m^3$ (length $\times$ width $\times$ height) and each grid size of $0.4 \times 0.4 \times 0.4 \ m^3$. There are on average 4,838 navigable grids in a scene removing the grids occupied by instances. Agents in the 3D grid world can perform five types of actions relative to their current pose: forward-backward move (stop, forward, backward), left-right move (stop, left, right), up-down move (stop, up, down), heading rotate (stop, left, right) and elevation rotate (stop, up, down). The maximum distance traveled in each direction is 1.6m. Each action type involves a combination of a direction classification and a magnitude regression task, e.g., move forward for 0.8m, rotate horizontally to the left for 30 degrees.

3.2. Manual Annotation

After the automatic scene construction, we recruit expert annotators from Appen Platform\(^1\) to select good viewpoints and write detailed descriptions for each scene.

Viewpoint Selection. Explicitly annotating trajectories for Embodied Captioning is time-consuming, subjective, and less flexible due to dependency on initial positions. Therefore, we ask annotators to select good viewpoints for each scene at which the scene can be well-captured. This allows us to automatically generate trajectories from random initial positions to those good viewpoints (Section 3.3). To be specific, as good viewpoints are always a certain distance away from instances in both horizontal and vertical directions, we first sample 20 candidate viewpoints in a highly probable spatial range of the grid world and then set the camera to look at the center of the scene to render images. To help annotators understand the scene, we provide category labels of instances in the scene. The annotators can select multiple good viewpoints as long as the image provides a clear view of (almost) all objects in the scene.

Paragraph Annotation. We ask annotators to write descriptions about objects in the scene, mentioning categories, attributes, and spatial relationships of instances. To ensure that the description is comprehensive, annotators are required to write a paragraph in which the number of sentences roughly matches the number of instances.

For each scene, there are three independent expert annotators recruited to select viewpoints and write paragraphs. Besides, an extra expert inspector is arranged to check the annotation quality. More details about the annotation platform, annotation cost, and annotation samples can be found in the supplementary material.

3.3. Trajectory Generation

We generate ground-truth trajectories based on the annotated good viewpoints and ground-truth 3D environment information in each scene. Specifically, we take a good viewpoint as the target position and randomly select three start positions. In contrast to previous navigation works [5, 15, 34] which use a fixed move step size of 0.25m, we allow our agent to move more flexibly and cover longer distances, with a maximum length of 1.6m, to reduce trajectory length. We employ the Dijkstra algorithm to generate the shortest movement path in the 3D space from the start to the target position. Then, we generate an optimal rotative action of the camera after each move step of the agent. To ensure that most instances are visible, we directly set the camera to look at the center of the scene, as we can use the perfect agent's location and object locations for ground-truth trajectory generation. According to the camera view, position changes are converted to three types of move ac-

\(^1\)https://www.appen.com/
We merge the move action and the camera rotation action at each step, resulting in five ground-truth actions at each step: 3 movements with 3 step lengths, a camera heading rotation with direction and angle, and a camera elevation rotation with direction and angle. Fig. 2 shows an example of a ground-truth trajectory and its paragraph description.

### 3.4. Dataset Statistics

**Scene Statistics.** Our ET-Cap dataset consists of 10,000 scenes and 27,520 unique instances, which include 7,517 base instances and 20,003 placing instances across 10 and 47 object categories respectively. The detailed category distribution is provided in the supplementary material. Notably, about 53% of the instances appear only once in the dataset, as shown in Fig. 3(a), which can prevent models from overfitting the instances. As shown in Fig. 3(b), the number of instances in a scene varies from 3 to 7.

**Trajectory Statistics.** We generate 72,594 trajectories with an average of 6.13 viewpoints and a mean length of 6.3m. The ground-truth trajectories exhibit no significant bias in the action direction. The ‘move down’ direction is relatively less frequent than the ‘move up’ direction because most good viewpoints are in higher positions than the height of the objects, which is consistent with our intuition.

**Caption Statistics.** We collect 30,000 captions in total. A caption contains an average of 5.9 sentences and 50.2 words. The average number of adjectives, nouns and spatial relation words (e.g. ‘left’, ‘right’, ‘center’) in a caption are 9.6, 13.8, and 6.1, respectively. As shown in Tab. 1, our captions are more descriptive than those in existing visual captioning datasets, with a much higher ratio of adjectives. They also contain slightly more spatial words compared to traditional image captions [14, 22] and competitive to 3D object referring expression datasets [1, 12].

**Dataset Splits.** We split our ET-Cap into training, validation, and test sets based on scenes. Tab. 2 shows the statistics of each set. To test the generalization ability of agents, we further divide the validation and test sets into three subsets: a) **Common subset**, in which instances in the scene have all been seen in the training set; b) **Novel Instance subset**, in which the categories are all covered in the training set, but more than 50% of the instances in the scene do not appear in the training set; c) **Novel Category subset**, in which each scene contains at least 1 novel object category that has not been seen in the training set.

### 4. Cascade Embodied Captioning Model

In the Embodied Captioning task, the agent is initialized at a bad viewpoint and can only receive an RGB image at each step. It should navigate in the 3D environment to comprehend the scene and generate a paragraph describing all objects at the end of the navigation. We propose a **Cascade embodied captioning (CaBOT)** model to tackle the task. As shown in Fig. 4, CaBOT is composed of a History-aware Navigator and a Trajectory-aware Captioner. The navigator utilizes an episodic history to explore the environment, while the captioner generates paragraphs using the whole trajectory produced by the navigator.

#### 4.1. History-aware Navigator

Given the observed RGB image at each step, the navigator must generate 5 types of actions: forward-backward movement, left-right movement, up-down movement, elevation rotation, and heading rotation. For each action type, it needs to predict both the direction and magnitude. Once the predictions are made, the agent will move to a new position and adjust its camera orientation accordingly. It stops until all the values of the types of actions are ‘stop’.

Our navigator consists of a backbone to extract image features where we use a ResNet [19], a region encoder to enhance spatial modeling within each image, a historical vision encoder to enhance temporal modeling across images, and a historical camera-informed decoder to predict actions.
Region Encoder. During navigation, to know which direction is possible to capture more visual information of instances, it is necessary to compare the region difference of the current image. Therefore, we apply a Transformer layer [35] with self-attention over the backbone for region-level encoding. For the image $v_t$ rendered from the $t$th viewpoint, its region-level features are denoted as $H_t = [h^1_t, h^2_t, ..., h^R_t]$, where $R$ is the region sequence length.

Historical Vision Encoder. Besides spatial relation within the current image, historical vision information also helps to decide the next actions. For example, observing the change of the central region from blank to one containing partial instances helps to confirm continued movement in the same direction. Thus, we further apply a Historical Vision Encoder to learn the temporal relation of the same region across images at different steps as follows:

$$
\tilde{h}^r_t = \text{Trans}_<(h^r_{t-1}, h^r_t), 1 \leq t \leq R,
$$

where $\text{Trans}_<(\cdot)$ is a Transformer layer with causal-masked self-attention mechanism, which means only previous image information can be attended.

Historical Action-informed Decoder. In addition to historical vision information, historical action information (direction and magnitude) is also helpful for deciding the next actions because it records the viewpoint change during previous navigation. Therefore, we design a Historical Action-informed Decoder leveraging both vision and action information to predict the next actions as follows:

$$
\begin{align*}
    c_t &= W^c a_{t-1} + b^c, \\
    a_t &= \left[ \frac{1}{R} \sum_{r=1}^{R} \tilde{h}^r_t; c_t \right], \\
    \tilde{a}_t &= \text{Trans}_<([a_{t-1}; a_t]),
\end{align*}
$$

where $a_{t-1}$ is a 10D vector consisting of the five independent action classes and magnitudes in the previous step. $W^c$ and $b^c$ are trainable parameters, $[\cdot]$ means concatenation.

With $\tilde{a}_t$ as the input, we apply linear layers to predict the next action $a_t$ including the direction and magnitude for each of the five types of actions. We use a softmax function for the direction classification and a sigmoid function for magnitude regression at the end of the linear layer.

4.2. Trajectory-aware Captioner

The Trajectory-aware Captioner takes images $V = [v_1, v_1, ..., v_T]$ observed in the navigation trajectory as input, and outputs a paragraph to describe the scene. Due to the importance of accurate object recognition for visual captioning [4, 25], we train a DETR [10] model to detect objects (more details can be found in our supplementary material) and use its backbone ResNet [19] to initialize the backbone in our captioner. Similar to the navigator, the captioner also applies a Region Encoder which converts the image sequence into $H = [H_1, ..., H_T]$, where $H_t = [h^1_t, h^2_t, ..., h^R_t], 1 \leq t \leq T, T$ is the trajectory length, and $R$ is the region sequence length.

Bi-CrossAttention Decoder In order to gather visual information from different steps in the navigation trajectory, we propose a Bi-CrossAttention Decoder to leverage all images at both the region level and trajectory level for captioning.

We first perform a region-level mean pooling and a trajectory-level mean pooling on $H$ as follows:

$$
\begin{align*}
    p_t &= \frac{1}{R} \sum_{r=1}^{R} h_t^r, 1 \leq t \leq T, \\
    p_t' &= \frac{1}{T} \sum_{t=1}^{T} h_t^r, 1 \leq r \leq R,
\end{align*}
$$

$P = [p_1, ..., p_T]$ and $P' = [p'_1, ..., p'_R]$ are pooled trajectory-level and region-level visual features, respectively.

At the $i$th decoding step of paragraph generation, the decoder first performs causal-masked self-attention on previous words to keep text coherence. Then it applies two cross-attention layers to select relevant visual information at the
we calculate a weight to trajectory length when evaluating descriptions. Specifically, to promote efficient exploration of the scene, we factor in trajectory similarity of objects, object-attribute pairs, and object-on rare words, while SPICE better measures the semantics better for image captions. CIDEr puts more weight on high-frequency words, which typically do not work well for rule-based methods, and hence, we carefully design paragraph templates and object-relations against human-written examples. In the variants of our proposed captioner, we multiply NE by 1/r^l to obtain NE^l, and multiply IS and SS score by r^l to obtain IS^l and SS^l, respectively.

5.2. Captioning Ablation with Oracle Trajectory

In this section, we evaluate the effectiveness of the proposed captioner given ground-truth trajectories. The compared baselines include a template-based method and different variants of our proposed captioner. In the template-based method, we carefully design paragraph templates and insert automatically detected objects from images in the trajectory into the template (more details can be found in the supplementary material). In the variants of our model, we ablate the contributions of region-level cross-attention, trajectory-level cross-attention, and backbone initialization. Tab. 3 presents the comparison results. Firstly, all models outperform the template-based method, indicating the rule-based method is not competitive enough for describing the 3D scene in detail. Secondly, as the end view in an oracle trajectory is the image in good viewpoints annotated by humans, r2 achieves comparable performance with r3.
However, it’s still worse than r4, showing the importance of merging visual information at the trajectory level for scene description. Additionally, by combing both region-level and trajectory-level cross-attention, the captioner achieves better performance (r5 vs r4), indicating region-level features and trajectory-level features are complementary for paragraph generation. Furthermore, by initializing the backbone with ResNet from the object detection model, the captioner performs best (r6), showing knowledge about object detection is also fundamental to describing 3D environments. Finally, there is a large gap (e.g., 51.48 on CIDEr) between model performance and human performance, indicating there is still a large room to improve on our dataset.

### 5.3. Embodied Captioning Results

**Main results.** In this section, we combine the best captioning model (r6) in Tab. 3 with automatically predicted trajectories by a navigator. We compare the proposed navigator with its variants and a carefully designed rule-based method. The rule-based method primarily determines the direction of movement by analyzing the distribution of object areas within the current field of view. Specifically, to decide movement directions, we adopt a cropping strategy that splits the current image into equal-sized left, middle, and right parts. If the upper/lower part contains more instances, we will examine whether this results in fewer instances being visible. If so, the agent will move backward. Otherwise, the agent will move forward depending on where most instance area is located. Moreover, if the agent moves forward, we will examine whether this results in fewer instances being visible. If so, the agent will move backward. Once the agent reaches a new position, we make it turn around by four 90° turns to the left in order to determine the best orientation. The rule-based approach stops when the maximum trajectory length is reached.

Tab. 4 presents both the navigation performance and the captioning performance. Firstly, rule-based navigation is shown to be inferior to variants of CaBOT navigator, which suggests that navigating for better scene descriptions generation in ET-Cap is not a naive task. Secondly, with historical vision information during decoding, the navigator achieves better performance (r3 vs r2), which indicates that visual difference across time at the same image region is helpful for navigation. Besides, introducing historical action encoding further improves navigation performance (r4 vs r3), suggesting that the previous action trajectory is also beneficial to predict the next action. Finally, better navigation leads to better scene description, verifying that navigation ability is crucial for effectively embodied captioning.

**Generalization analysis.** We evaluate the generalization ability of CaBOT on three subsets of the test set. As shown in Tab. 5, CaBOT achieves similar performance on Common Set and Novel Instance Set for all three test settings. It shows that novel instances cause little impact on navigation and captioning ability. We attribute this instance generalization ability to the diverse training data, where most instances appear only one time, as shown in Fig. 3 (a). As for Novel Category Set, there is a small performance drop for navigation but an obvious performance drop for captioning. The navigator can be confused by an instance of novel categories, but may still find good viewpoints by observing other instances. However, for description generation, the captioner struggles to correctly describe instances whose categories have never been seen during training.
Potential benefits from joint navigation and captioning.

Although our proposed model separates navigation and caption generation, we believe that simultaneously performing navigation and captioning would lead to better performance in a more efficient manner. The captioner can inform the navigator of objects, attributes, or relations that it is uncertain of, allowing the navigator to explore the environment more efficiently in order to resolve the uncertainty and improve the quality of visual captioning. We conduct an exploratory experiment to demonstrate the potential benefits of joint navigation and captioning. We utilize the CaBOT captioner to generate scene descriptions at each navigation step predicted by the CaBOT navigation, and select the partial trajectory with the best CIDEr performance against the ground-truth captions. As shown in Tab. 6, this method achieves a significant improvement of +25.55 on the CIDEr and +28.02 on the CIDErący. It indicates that imperfect navigation could bring more noise and deteriorate visual captioning, and thus it is necessary to guide the navigation according to the captioning ability.

5.4. Qualitative Analysis

Fig. 5 presents some embodied captioning results of our CaBOT on the test set. Case (a) shows that CaBOT can gradually approach instances and try to wrap around to find better viewpoints. When starting from a relatively lower position, CaBOT can also explore to find a higher viewpoint, as shown in case (b). CaBOT stops at a good viewpoint in case (b) but a bad viewpoint in case (a). However, it generates acceptable descriptions in both cases, which indicates that the overall quality of the trajectory matters more than the end viewpoints. We also show a failure case in case (c).

Table 5. Breakdown analysis on different test subsets.

<table>
<thead>
<tr>
<th>Subset</th>
<th>Navigation</th>
<th>OracleCap</th>
<th>EmboCap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NE↓</td>
<td>IS↑</td>
<td>SS↑</td>
</tr>
<tr>
<td>Common</td>
<td>4.73</td>
<td>58.76</td>
<td>58.17</td>
</tr>
<tr>
<td>Novel Instance</td>
<td>4.88</td>
<td>58.94</td>
<td>58.90</td>
</tr>
<tr>
<td>Novel Category</td>
<td>5.11</td>
<td>57.73</td>
<td>57.08</td>
</tr>
</tbody>
</table>

Table 6. Exploratory experiments of joint navigation and captioning, where we simultaneously generate captions at each navigation step and stop navigation according to the captioning performance.

<table>
<thead>
<tr>
<th>Method</th>
<th>CIDEr</th>
<th>SPICE</th>
<th>CIDErący</th>
<th>SPICEść</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascade</td>
<td>40.83</td>
<td>19.43</td>
<td>34.88</td>
<td>16.62</td>
</tr>
<tr>
<td>Joint</td>
<td>66.38</td>
<td>21.51</td>
<td>62.90</td>
<td>20.41</td>
</tr>
</tbody>
</table>

6. Conclusion

In this work, we present a novel and challenging task called Embodied Captioning, where an agent should navigate in a 3D environment to gather visual information and use natural language to comprehensively describe objects in the scene. To support this task, we build the ET-Cap dataset with manually annotated good viewpoints and paragraph descriptions for 10,000 synthetic scenes. We propose a Cascaded Embodied Captioning (CaBOT) model, which utilizes both visual and action history to perform navigation and then generates captions by leveraging the whole navigation trajectory. Experiments demonstrate the effectiveness of CaBOT and show a promising direction of joint modeling navigation and captioning.

When the agent reaches a bad viewpoint where no instance can be seen, it is less robust to find good viewpoints though historical visual and action information are leveraged. We consider this is mainly due to unseen states in testing, as we only utilize ground-truth trajectories for imitation learning. This problem can be relieved by data augmentation or reinforcement learning, and we leave it to future work.

Acknowledgements. This work was partially supported by the National Key R&D Program of China (No.2020AAA0108600) and the National Natural Science Foundation of China (No. 62072462).
References


[24] Xinghang Li, Di Guo, Huaping Liu, and Fuchun Sun. REVECE: remote embodied visual referring expression in contin-


[34] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, pages 4580–4590. IEEE, 2019. 2


