Explore until Confident: Efficient Exploration for Embodied Question Answering

Author Names Omitted for Anonymous Review.

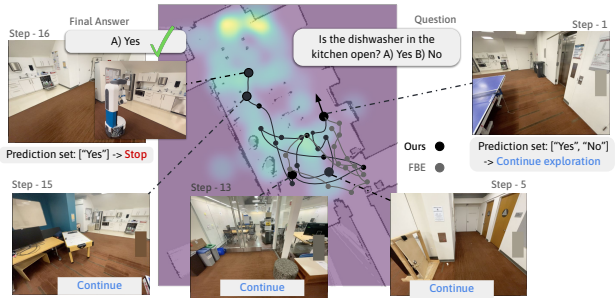

Fig. 1: Our framework leverages a large vision-language model (VLM) to obtain semantic information from the views (visualized by overlaying it on top of the occupancy map), which guides a Fetch robot to explore relevant locations. The robot maintains a set of possible answers and stops when the set reduces to a single answer based on the current view.

***Abstract*—We consider the problem of *Embodied Question Answering (EQA)*, where a robot needs to actively explore an environment to gather information until it is confident about the answer to a question. We leverage the strong semantic reasoning capabilities of large vision-language models (VLMs) to efficiently explore and answer such questions. We first build a semantic map of the scene based on depth information and via visual prompting of a VLM — leveraging its vast knowledge of relevant regions of the scene for exploration. Next, we use conformal prediction to calibrate the VLM's question answering confidence, allowing the robot to know when to stop exploration — leading to a more calibrated and efficient exploration strategy. To test our framework in simulation, we also contribute a new EQA dataset with diverse scenes built upon the Habitat-Matterport 3D Research Dataset (HM3D). Both simulated and real robot experiments show our proposed approach improves the performance and efficiency over baselines.**

I. Introduction

Imagine that a service robot is sent to a home to perform various tasks, and the household owner asks it to check whether the stove is turned off. This setting is referred to as Embodied Question Answering (EQA) [1, 2], where the robot starts at a random location in a 3D scene, explores the space, and stops when it is confident about answering the question. This can be a challenging problem due to highly diverse scenes and lack of an a-priori map of the environment.

Recently, large vision-language models (VLMs) have achieved impressive performance in answering complex questions about static 2D images [3, 4]. They can also help the robot *actively* perceive the 3D scene given partial 2D views and *reason* about future actions for the robot to take [5]. Such capabilities are critical to performing EQA, as the robot can now better reason about relevant regions of the environment, actively explore them, and answer questions that require semantic reasoning. However, there are two main challenges:

1) **Limited Internal Memory of VLMs.** Efficient exploration benefits from the robot tracking previously explored regions and also ones yet to be explored but *relevant* for answering the question. However, VLMs do not have an internal memory for mapping the scene and storing such semantic information;
2) **Miscalibrated VLMs**. VLMs are fine-tuned on pre-trained large language models (LLMs) as the language decoder, and LLMs have been shown to often be miscalibrated [6] – that is they can be over-confident or under-confident about the output. This makes it difficult to determine when the robot is confident enough about question answering in EQA and then stop exploration, affecting overall efficiency.

How can we endow VLMs – with limited memory and potential for miscalibration – with the capability of efficient exploration for EQA? We propose a framework (Fig. 1) that (1) fuses the commonsense/semantic reasoning abilities of a VLM into a global geometric map to enable efficient exploration, and (2) uses the theory of multi-step conformal prediction [7, 8] to formally quantify VLM uncertainty about the question. Through exploration, the robot builds a semantic map of the scene that stores information on occupancy and locations the VLM deems worth exploring. Such semantic information is obtained by annotating the free space in the current image view, prompting the VLM to choose among the unoccupied regions, and querying its prediction (Fig. 2). Throughout an episode, the robot maintains a set of possible answers, updates the set at each step based on new visual information provided to the VLM, and stops exploration when the set of possible answers reduces to a single option. Conformal prediction formally ensures the set covers the true answer with high probability. The set size is also minimized and thus the robot can stop as soon as possible.

II. Problem Formulation

Distribution of scenarios for EQA. We formalize Embodied Question Answering (EQA) by considering an unknown joint distribution over *scenarios* $\xi \sim \mathcal{D}$ the robot can encounter. A scenario is a tuple $\xi := (e, T, g^0, q, y)$, where e is a simulated or real 3D scene (*e.g.,* , a floor plan with certain dimensions), T is the maximum number of time steps allowed for the robot to navigate in the scene (*e.g.,* a function of scene size), g^0 is the robot's initial pose (2D position and orientation at time 0), q is the question, and y is the ground truth answer. We

will use a subscript to indicate the scenario (*e.g.,* T_ξ for the maximum time horizon in scenario ξ), and a superscript t for time steps (*e.g.,* g^t for the robot's pose at time t). We consider multiple-choice questions q, *e.g.,* "Where did I leave the black suitcase? A) Bedroom B) Living room C) Storage room D) Dining room." Thus the set of labels $\mathcal{Y} := \{\text{'}A\text{'}, \text{'}B\text{'}, \text{'}C\text{'}, \text{'}D\text{'}\}$ contains any answer y.

Robot navigating in a scenario. We do not expect the robot to have any prior knowledge of the scene. We initialize the robot at g^0, and at any time t it can traverse to different poses g^t. The robot's onboard camera provides RGB images $I_c^t \in \mathbb{R}^{H_I \times W_I \times 3}$ and depth images $I_d^t \in \mathbb{R}^{H_I \times W_I}$. We associate a time step with each time the robot stops and takes RGB/depth images. Additionally, we assume access to a collision-free planner π that determines the next pose g^{t+1} to travel to, a maximum of $3\,\mathrm{m}$ away from g^t.

VLM predictions. We pass the RGB image and a text prompt s to the VLM, and query its probability over predicting the next token. For convenience, we denote $x^t = (I_c^t, q)$ consisting of the RGB image I_c^t and the question q. Then, the VLM's prediction given the question q at time t can be denoted as $\hat{f}(x^t) \in [0,1]^{|\mathcal{Y}|}$, which are the softmax scores over the multiple choice set $\mathcal{Y}$. We denote $\hat{f}_y(\cdot)$ as the softmax score for a particular label y.

Goal: efficient exploration. In a new scenario, the robot may stop at any time step $t \leq T_\xi$ and make a final answer. Our goal is to answer the question correctly in *unseen* test scenarios $\xi \in \mathcal{D}$, using a minimal number of time steps.

III. Targeted Exploration using VLM Reasoning

A. Exploration Map and Frontier-Based Exploration

For tracking where the robot has explored, we first adopt a 3D voxel-based representation for the map of size $L \times W \times H$ — M and L expand as the robot explores more areas, and H is fixed as $3.5\,\mathrm{m}$ (typical floor height). Each voxel corresponds to a cube with side length l. At each pose g^t with depth image $I_d^t \in \mathbb{R}^{H_I \times W_I}$ and known camera intrinsics, we apply Volumetric Truncated Signed Distance Function (TSDF) Fusion [9, 10] to update (1) occupancy of the voxels and (2) if they are explored/seen in the current I_d^t. At each time step we then project the 3D voxel map into a 2D point map M: a 2D point is considered free (un-occupied) if all voxels up until $1.5\,\mathrm{m}$ are marked free, which is the height of the camera (in simulation and in reality), and considered explored if all voxels along H have been marked explored.

Based on the 2D map storing occupancy and exploration information, we use a heuristics-based 2D planner that plans new poses around unexplored region. Our strategy is based around Frontier-Based Exploration (FBE), which has been proven a simple yet effective method for navigation tasks [11]. FBE finds the *frontiers* (Fig. 2), the locations at the boundary of the explored and unexplored regions, samples one as the planned location, and uses the normal direction to the unexplored region boundary as the planned orientation.

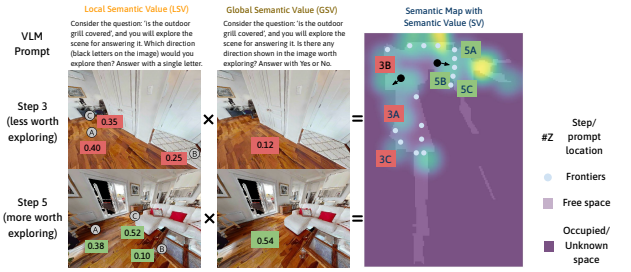

Fig. 2: To query VLM's uncertainty over possible exploration locations, we visually prompt the VLM with possible points in the current view (left column) and also with the entire view (middle column) to obtain the Local Semantic Value (LSV) and Global Semantic Value (GSV) (Section III-B). A weighted combination of them (SV) is saved in a semantic map (right column). The values are used as the weights for sampling the next frontier to navigate to, guiding the robot towards *unknown and relevant* regions (Section III-C)

B. VLM Visual Prompting for Semantic Value (Fig. 2)

As the VLM already has access to rich prior knowledge from large-scale Internet data, we hypothesize that it can potentially provide useful information in determining relevant locations to explore. We achieve this by obtaining the VLM's uncertainty over the possible locations via visual prompting. Given the current RGB image I_c^t, we first identify the free space seen in I_c^t by (a) projecting it onto M, (b) keeping only the free points, and (c) sampling a set of points P using farthest point sampling to ensure coverage. In practice, we use $|P| = 3$, which we find sufficient to cover the possible distinct regions in an image. Then, we de-project the sampled points back onto I_c^t and annotate them with letters $\mathcal{Y}_P = \{\text{'}A\text{'}, \text{'}B\text{'}, \text{'}C\text{'}\}$ on I_c^t to get an annotated image $I_{c,\mathcal{Y}_P}^t$, which can be used for visual prompting (Fig. 2). Now, we have the following prompt:

```
Consider the question: {question}, and you will
explore the scene for answering it. Which
direction (black letters on the image) would you
explore then? Answer with a single letter.
```

We then use the (normalized) probability output of the VLM over each of the three directions to construct the *local semantic value* (LSV) of $p \in P$:

$$\text{LSV}_p(x^t) = \hat{f}_{y_p}(x^t) = \hat{f}_{y_p}(I_c^t, s_{\text{LSV},q}) \in [0,1], \qquad (1)$$

where $x^t = (I_c^t, q)$ is the RGB image and question (cf. Sec. II) and $s_{\text{LSV},q}$ is the prompt above with the question filled in. Note that this is a "local" score because the comparison is from one image, and the locations P are not suited for being compared to those seen in images taken with different poses g^t (*e.g.,* see top and bottom rows in Fig. 2) when planning the next robot pose using M. We additionally need to determine if we should navigate to poses visible from the current pose *at all*: via visual prompting:

```
Consider the question: {question}, and you will
explore the scene for answering it. Is there any
direction shown in the image worth exploring?
Answer with Yes or No.
```

This allows us to obtain the *global semantic value* (GSV) of a given point $p \in P$ by querying the (normalized) probability

of the VLM predicting 'Yes':

$$\text{GSV}_p(x^t) = \hat{f}_{\text{'Yes'}}(x^t) = \hat{f}_{\text{'Yes'}}(I_c^t, s_{\text{GSV},q})) \in [0,1], \quad (2)$$

where again $s_{\text{GSV},q}$ is the prompt above with the question filled in. To determine the overall semantic value (SV), we apply temperature scaling (τ_{LSV} and τ_{GSV}) to each of the two values and compute the following score:

$$\text{SV}_p(x^t) = \exp\left(\tau_{\text{LSV}} \cdot \text{LSV}_p(x^t) + \tau_{\text{GSV}} \cdot \text{GSV}_p(x^t)\right). \quad (3)$$

C. Semantic-value-weighted Frontier Exploration

Now we detail how to incorporate preferences in exploring high semantic-value regions using the semantic map — we apply SV as the weights when sampling the next frontier to navigate to. Each weight are based on two values, SV_p, the semantic value at point p, and also $\text{SV}_{p,\text{Normal}}$, defined as the average semantic value of the points within a certain distance d_{SV} from p in the normal direction. $\text{SV}_{p,\text{Normal}}$ can be particularly useful to better guide the robot *towards* the relevant regions if they are not close to robot's current pose.

IV. Stopping Criterion for Exploration and Answering the Question

The second piece of efficient exploration is to know when you have enough information to answer the question and realize when you should stop exploring. Techniques for assessing VLM confidence in question answering typically rely on softmax scores. For example, one can compute the entropy of the predicted answer at each time step:

$$H(\hat{f}(x^t)) = -\sum_{y \in \mathcal{Y}} \hat{f}_y(x^t) \log \hat{f}_y(x^t), \quad (4)$$

and stop if this quantity is below a pre-defined threshold. Or we can directly prompt the model:

```
Consider the question {question}. Are you
confident about answering the question given the
current view?
```

We can then look at the probability of the model predicting 'Yes'; we refer to this as the *question-image relevance score*:

$$\text{Rel}(x^t) = \hat{f}_{\text{'Yes'}}(I_c^t, (q, s_{\text{Rel},q})), \quad (5)$$

where $s_{\text{Rel},q}$ is the prompt above with the question filled in. By normalizing this quantity with the sum of confidences of predicting 'Yes' and 'No', one obtains a scalar quantify bounded in $[0,1]$. A scalar threshold $h_{\text{rel}} \in [0,1]$ can then be used as the stopping criterion.

However, the softmax scores from VLMs are often *miscalibrated*, i.e., they are often over- or under-confident. This motivate us to rigorously quantify the VLM's uncertainty and carefully calibrate the raw confidences. Our main insight is to employ multi-step conformal prediction, which allows the robot to maintain a *set* of possible answers (*prediction set*) over time, and stop when the set reduces to a single answer. Conformal prediction uses a moderately sized (e.g., $\sim$300) set of scenarios for carefully selecting a confidence threshold above which answers are included in the prediction set. This procedure allows us to achieve *calibrated confidence*: with a user-specified probability, the prediction set is guaranteed to contain the correct answer for a new scenario. CP also minimizes the prediction set size [7, 8], which helps the robot to stop as quickly as it can while satisfying calibrated confidence.

A. Applying Multi-Step CP for Embodied Question Answering

Here we describe how CP provides a *principled* and more *interpretable* stopping criterion for multi-step exploration by building on the multi-step CP approach presented in [8] (see Section B for background on conformal prediction). Let x^t denote the input at time t consisting of the RGB image I_c^t and the question q. Each episode results in a sequence $\bar{x} = (x^0, x^1, \dots)$ of such inputs. We first define the relevance-weighted confidence score at time t (analogous to the single-step definition Eq. (A7)):

$$\rho_y^t(x^t) := \text{Rel}(x^t)(\hat{f}_y(x^t) - 1). \quad (6)$$

This quantity is large when the input x^t at time t is deemed highly relevant and the VLM is confident in the answer y. We can then define the *episode-level* confidence as:

$$\bar{\rho}_y(\bar{x}) := \min_{t \in [T]} \rho_y^t(x^t), \quad (7)$$

where T is the maximum allowable episode length. Given a calibration dataset $Z = \{z_i = (\bar{x}_i, y_i)\}_{i=1}^N$ of input sequences (collected using the exploration policy) and ground-truth answers, we define the non-conformity score for data point i as $\kappa_i := 1 - \bar{\rho}_{y_i}(\bar{x}_i)$.

We can then perform the standard CP calibration as described in Section B using these non-conformity scores in order to obtain a confidence threshold $\hat{q}$. Then, given a new input sequence $\bar{x}_{\text{test}}$, we can construct a *sequence-level* prediction set $\bar{C}(\bar{x}_{\text{test}}) := \{y \in \mathcal{Y} | \bar{\rho}_y(\bar{x}_{\text{test}}) \geq 1 - \hat{q}\}$. This set is guaranteed to contain the ground-truth answer with probability $1 - \epsilon$. However, at test-time, the robot does not obtain the entire sequence $\bar{x}_{\text{test}}$ at once; instead, the prediction sets must be *causally* constructed over time (i.e., using observations up to the current time). Define the causally constructed prediction set at time t to be:

$$C^t(x_{\text{test}}^t) := \{y \in \mathcal{Y} | \rho_y^t(x_{\text{test}}^t) \geq 1 - \hat{q}\}. \quad (8)$$

Claim 1: For all time $t \in [T]$, the causally constructed prediction set $C^t(x_{\text{test}}^t)$ contains the sequence-level set $\bar{C}(\bar{x}_{\text{test}})$. Moreover, $\cap_{t=0}^T C^t(x_{\text{test}}^t) = \bar{C}(\bar{x}_{\text{test}})$.

Proof: See App. Section A. ■

Proposition 1: With probability $1 - \epsilon$ for test scenarios drawn from $\mathcal{D}$, the ground-truth label y_{test} is contained in the prediction set $\cap_{k=0}^t C^k(x_{\text{test}}^k)$ for all $t \in [T]$.

Proof: This follows directly from the claim above and the fact that the sequence-level prediction set $\bar{C}(\bar{x}_{\text{test}})$ contains the ground-truth label with user-defined probability $1 - \epsilon$ as guaranteed by CP. ■

At test time, we thus construct the set $C^t(x_{\text{test}}^t)$ at each step and maintain the intersection of these sets over time. If the resulting intersection contains only a single element, the robot halts its exploration with $1 - \epsilon$ confidence that the corresponding answer is correct.

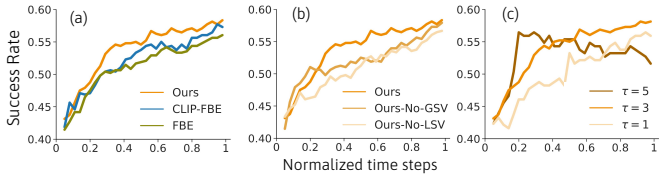

Fig. 3: Normalized time step taken vs. success rate in simulated experiments for (a) comparing different exploration methods, (b) ablating GSV and LSV from our method, and (c) varying semantic value temperature scaling.

V. Experiments and Discussion

Through extensive simulated and hardware experiments, we investigate the following key questions:

- Q1 - **Semantic exploration:** Does our method use fewer steps while achieving the same level of EQA success compared to baselines without leveraging VLM reasoning for exploration?
- Q2 - **Stopping criteria:** Does our method with confidence calibration use fewer steps while achieving the same level of EQA success compared to baselines without calibration?

A. HM-EQA Dataset

While prior work has primarily considered synthetic scenes and simple questions such as "what is the color of the coffee table?" involving basic attributes of relatively large pieces of furniture, we are interested in applying our VLM-based framework in more realistic and diverse scenarios. To this end, we propose HM-EQA, a new EQA dataset based on the Habitat-Matterport 3D Research Dataset (HM3D) [12]. At the end, we generate 500 questions from 312 different scenes. Please see Section C for the details.

B. Q1: Semantic Exploration - Baselines

To evaluate how well our VLM-based semantic exploration performs, we consider the following baselines:

- **FBE**: This baseline applies frontier-based exploration without using any semantic values for weighted sampling.
- **CLIP-FBE**: This baseline considers semantic values when sampling the next frontier like ours. However, the semantic values are from CLIP [13], which provides the relevance of an image given a text input. We apply the multi-scale relevancy extractor from [14] to extract a dense pixel-wise CLIP score to allow fine-grained localization of relevant objects or exploration directions.
- **Ours-No-LSV**: Ours without LSV.
- **Ours-No-GSV**: Ours without GSV.

C. Q1: Semantic Exploration - Simulation Results

Fig. 3a shows the average success rate (answer correctly predicted) achieved when the robot is allowed to run for different numbers of time steps. Compared to FBE and CLIP-FBE, our method uses fewer steps to achieve success. The difference becomes significant around $20\% - 30\%$ of the maximum allowable steps, demonstrating the effect of VLM-based active exploration at the early stage of the episodes.

Fig. 3b, we demonstrate that both GSV and LSV are critical to achieving high success rates when using semantic values to guide exploration. Without GSV, the planner can be myopic as it only considers LSV, and thus the robot being stuck in incorrectly chosen locations and unable to explore other locations. Without LSV, the robot explores less efficiently overall due to the less fine-grained semantic values.

In Fig. 3c, we vary the temperature scaling τ_{LSV} and τ_{GSV} applied when determing the semantic values (SV) used for sampling the frontier. The higher the scalings are, the bigger the difference in SV among different regions. Results show that too high τ leads to faster exploration at the beginning, but worse performance in later normalized time steps. This is potentially due to the robot overly prioritizing the semantic regions. Too low τ also leads to inferior efficiency.

D. Q2: Stopping Criterion - Baselines

To evaluate how well our CP-based stopping criterion performs, we consider the following baselines:

- **Entropy**: This baseline uses the entropy of the predicted answer (4) as the metric, and stops exploration once it is lower than some threshold (varied). The final answer is the one with the highest $\hat{f}_y(x^t)$ at the stopping time step.
- **Relevance**: use the question-image relevance score (5) as the metric, and stop exploration once it is higher than some threshold (which can be varied). The final answer is the one with the highest $\hat{f}_y(x^t)$ at the stopping time step.

E. Q2: Stopping Criterion - Simulation Results

For evaluation, we vary the different thresholds used by our method (ϵ), Entropy, and Relevance. We then consider the normalized time step vs. the achieved success rate. Fig. 4 shows that our method significantly outperforms EntropyEq. (4). We find that, as the robot often sees irrelevant views (*e.g.,* facing an empty wall), the VLM still outputs highly confident, biased answers for the question. Such bias leads to low prediction entropy and the robot stops prematurely.

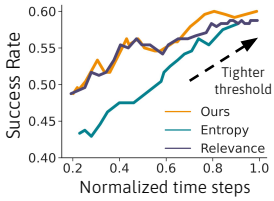

Fig. 4: Normalized time step taken vs. success rate using varying threshold in Ours and the baselines, in simulated experiments.

This observation leads to the necessity of using the question-answer relevance score Eq. (5), which helps the robot ignore some of the irrelevant views and continue exploring. However, we find that, in order to achieve high success rates (upper right side of the plot), Relevance tends to use more time steps. For example, to achieve 58% success rate, our method takes about 71% of the maximum time steps while Relevance takes 85%. Our method, based on the theory of multi-step conformal prediction Section IV-A, calibrates the VLM's confidence and consequently improves exploration efficiency.

F. Q2: Stopping Criterion - Hardware Results.

Please see Section D for results.

References

[1] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10. IEEE, 2018.

[2] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4089–4098. IEEE, 2018.

[3] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[4] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[5] Minae Kwon, Hengyuan Hu, Vivek Myers, Siddharth Karamcheti, Anca Dragan, and Dorsa Sadigh. Toward grounded social reasoning. *arXiv preprint arXiv:2306.08651*, 2023.

[6] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[7] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*, volume 29. Springer, 2005.

[8] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

[9] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011.

[10] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from rgb-d reconstructions. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[11] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.

[12] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-Matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. *arXiv preprint arXiv:2109.08238*, 2021.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

[14] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3D scene understanding from 2D vision-language models. In *Conference on Robot Learning (CoRL)*. PMLR, 2022.

[15] Anastasios N Angelopoulos, Stephen Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.