

## APPENDIX A PROOF OF CLAIM 1

*Proof:* The proof follows Ren et al. [8, Claim 1]. Specifically,

$$y \in \tilde{C}(\bar{x}_{\text{test}}) \iff \bar{\rho}_y(\bar{x}_{\text{test}}) \geq 1 - \hat{q} \quad (\text{A1})$$

$$\iff \min_{t \in [T]} \rho_y^t(x_{\text{test}}^t) \geq 1 - \hat{q} \quad (\text{A2})$$

$$\iff \rho_y^t(x_{\text{test}}^t) \geq 1 - \hat{q}, \forall t \in [T] \quad (\text{A3})$$

$$\iff y \in C^t(x_{\text{test}}^t), \forall t \in [T] \quad (\text{A4})$$

$$\iff y \in \bigcap_{t=0}^T C^t(x_{\text{test}}^t). \quad (\text{A5})$$

■

## APPENDIX B BACKGROUND: CONFORMAL PREDICTION

We provide a brief overview of conformal prediction (CP) in this section; we refer the reader to [15] for a thorough exposition. Here we describe the single-step setting where a VLM must answer a question pertaining to a *single* image.

Let  $\mathcal{X}$  and  $\mathcal{Y}$  denote the space of inputs (images and corresponding questions) and labels (answers) respectively, and let  $\mathcal{D}$  denote an *unknown* distribution over  $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ . Suppose we have collected a *calibration dataset*  $Z = \{z_i = (x_i, y_i)\}_{i=1}^N$  of such pairs drawn i.i.d. from  $\mathcal{D}$ . Now, given a new i.i.d. sample  $z_{\text{test}} = (x_{\text{test}}, y_{\text{test}})$  with unknown true label  $y_{\text{test}}$ , CP generates a *prediction set*  $C(x_{\text{test}}) \subseteq \mathcal{Y}$  that contains  $y_{\text{test}}$  with high probability [7]:

$$\mathbb{P}(y_{\text{test}} \in C(x_{\text{test}})) \geq 1 - \epsilon. \quad (\text{A6})$$

Here,  $1 - \epsilon$  is a user-defined threshold that impacts the size of  $C(\cdot)$ .

CP provides this statistical guarantee on coverage by utilizing the dataset  $Z$  to perform a calibration procedure with raw (heuristic) confidence scores. In our setting, we define the relevance-weighted confidence score for an input  $x$  as:

$$\rho_y(x) := \text{Rel}(x)(\hat{f}_y(x) - 1). \quad (\text{A7})$$

This quantity is large when it is *both* the case that the VLM is confident in the answer  $y$  and the image is deemed highly relevant. CP utilizes these scores to evaluate the set of *nonconformity scores*  $\{\kappa_i = 1 - \rho_{y_i}(x_i)\}_{i=1}^N$  over the calibration set. Intuitively, the higher the nonconformity score is, the less confident the VLM is in the correct answer or the less relevant the image is deemed to be. We then perform calibration by defining  $\hat{q}$  to be the  $\frac{\lceil (N+1)(1-\epsilon) \rceil}{N}$  empirical quantile of  $\kappa_1, \dots, \kappa_N$ . For a new input  $x_{\text{test}}$ , CP generates  $C(x_{\text{test}}) = \{y \in \mathcal{Y} \mid \rho(x_{\text{test}})_y \geq 1 - \hat{q}\}$ , i.e., the prediction set that includes all labels in which the predictor has at least  $1 - \hat{q}$  relevance-weighted confidence. The generated prediction set ensures that the coverage guarantee in Eq. (A6) holds.

## APPENDIX C ADDITIONAL IMPLEMENTATION DETAILS

### A. Semantic map

**Updating the explored regions.** As introduced in Section III-A, while all voxels seen in the depth image  $I_d^t$  are used to update occupancy at each step, only those within a smaller field of view are used to update whether they have been explored, enabling more fine-grained exploration. In practice, we use a 4 : 3 aspect ratio for the images, and 120 degrees for the horizontal field of view (HFOV) and 105 degrees for the vertical field of view (VFOV) in simulation. Then we mark voxels explored if they correspond to pixels from the middle 50% of the full HFOV and the lower 50% of the full VFOV, as these pixels correspond to voxels closer to the robot.

**Determining the weights of frontiers based on semantic values.** From Section III-B, when the robot plans for the next pose to travel to, it samples from possible frontiers with weights, and the weight of each frontier depends on (1)  $\text{SV}_p$ , the semantic value at the frontier  $p$ , and (2)  $\text{SV}_{p,\text{normal}}$ , the average semantic value of the points with  $d_{\text{SV}}$  distance from  $p$  in the normal direction. Fig. A1 illustrates the setup. We set  $d_{\text{SV}} = 3\text{m}$ . Notice that the frontier near the top of the figure has a slightly higher SV, while the middle frontiers have much higher  $\text{SV}_{p,\text{normal}}$  due to the high semantic value region at about 2m away from the frontiers into the un-explored regions. Balancing between SV and  $\text{SV}_{p,\text{normal}}$  as the sampling weights helps the robot explore and move towards relevant regions.

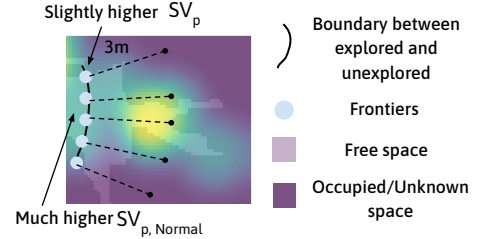


Fig. A1: Sampling weight of the frontier  $p$  depends on both  $\text{SV}_p$  and  $\text{SV}_{p,\text{normal}}$ .

Similar to SV combining LSV and GSV in Section III-B, we again apply temperature scaling ( $\tau_{\text{SV}}$  and  $\tau_{\text{SV, Normal}}$ ) to each of the two values and compute the final weight  $w_p$  of the frontier  $p$ :

$$w_p = \exp(\tau_{\text{SV}} \cdot \text{SV}_p + \tau_{\text{SV, Normal}} \cdot \text{SV}_{p,\text{normal}}). \quad (\text{A8})$$

In practice we use 1 for both scaling values. Online adaptation of these values can potentially further improve the exploration efficiency.

### B. HM-EQA Dataset

In order to generate questions that are realistic in typical household settings, we leverage GPT4-V, the state-of-the-art VLM, to generate such questions based on twelve random views sampled inside an indoor scene from HM3D, and also three sets of example manually-written questions and answers given views of the corresponding scenes (one set per scene).

Afterwards we manually remove some of the questions that are (1) too simple (*e.g.*, “How many sofas are there in the living room for them to sit on?”) or (2) hallucinating objects that cannot be seen from the views by a human (*e.g.*, eyeglasses, watering can, and remote control). We consider option (1) to be too simple as it involves detection of very prominent objects in the scene (large in size). At the end, we generate 500 questions from 312 different scenes. The resulting questions can be roughly divided into five categories (also showing their split within the whole dataset):

- 1) **Identification (16.6%)**: asking about identifying the type of an object, *e.g.*, “Which tablecloth is on the dining table? A) Red B) White C) Black D) Gray”
- 2) **Counting (18.4%)**: asking about the number of objects, *e.g.*, “My friends and I were playing pool last night. Did we leave any cues on the table? A) None B) One C) Two D) Three”.
- 3) **Existence (21.4%)**: asking if an object is present at a location, *e.g.*, “Did I leave my jacket on the bench near the front door? A) Yes B) No”.
- 4) **State (19.8%)**: asking about the state of an object, *e.g.*, “Is the air conditioning in the living room turned on? A) Yes B) No” or “Is the curtain in the master bedroom closed? A) Yes B) No”.
- 5) **Location (23.8%)**: asking about the location of an object, *e.g.*, “Where have I left the black suitcase? A) At the corner of the bedroom B) In the hallway C) In the storage room D) Next to TV in the living room”.

Notice that some of the questions only involve two multiple choices, and our formulation in [Section II](#) assumes four. For consistency, if the question itself does not have four multiple choices, we add additional ones, *e.g.*, “D) (Do not choose this option)” until there are four.

Since the different scenes  $e$  from HM3D can have very different sizes (majority of which range from  $100\text{m}^2$  to  $800\text{m}^2$ ), we set the maximum allowed time steps  $T_e$  in each scene to be the square root of the 2D size times a factor of three. The initial pose of the robot  $g^0$  is sampled randomly from the free space in the scene. We have now fully defined the scenarios introduced in [Section II](#),  $\epsilon := (e, T, g^0, q, y)$  ( $q$  for question and  $y$  for answer).

## APPENDIX D ADDITIONAL EXPERIMENT RESULTS

**Hardware experiments.** We focus on comparing the stopping criterion performance in the hardware experiments. First we determine the threshold used for the stopping criterion:  $\epsilon = 0.5$  for the threshold used in CP calibration for our method, 0.4 for Relevance, and 0.1 for Entropy. These thresholds roughly corresponds to 50% success rates based on simulated experiment results from [Fig. 4](#).

[Fig. A7](#) shows all six scenarios (questions/answers, and the initial robot views), the final views after the robot stops exploration using different methods (Ours, Relevance, and Entropy), the final answers chosen, and the number of steps

You will be shown some random views inside a house. You will come up with a simple question related to possible household tasks based on the views, that the household owner may ask the robot. Make sure the question has four options and a definitive answer. Try to be creative in the question and make it sound like interesting scenarios when the household owner needs help.

Note:

- (1) The question is for a robot in the 3D scene, so do not refer to the image in the question.
- (2) The views shown do not cover the full scene and there might be information of the scene missing, and thus do not refer to 'first/second room' or 'the room' or 'the table' or 'the view' or 'the area' in the question.
- (3) Do not ask about, for example, if a door is locked or not, or if the fan is on, since it is hard to tell from static images.

In terms of types of questions to ask:

- (1) Focus on locations of the objects, especially those small, or misplaced, or those that the household owner possibly recently moved.
- (2) Sometimes ask counting questions, such as 'how many chairs in...'.  
(3) You can ask Yes/No type of questions (but just ask sometimes), such as 'is the fire distinguisher near the staircase?' but avoid multiple choices like 'Can't determine' or 'I am not sure.'
- (4) Do not ask questions that the household owner should know, like the color of the sofa, or type of plants."

Fig. A2: Prompt used when GPT4-V generates candidate questions and their answers given multiple views of the scene.

taken at stopping (not normalized). In Scenario 3 and 4, all methods stop at the same time step and answer the question correctly. Looking at other scenarios, Entropy tends to stop early (*e.g.*, Step 1 in Scenario 1 and Step 3 in Scenario 5), but this leads to the failure in Scenario 1 where the other two methods answer correctly based on later views. Relevance achieves the same success rate (4 out of 6) as our method, but it uses more steps in Scenario 2 and Scenario 5, while our approach answers the question based on relevant views from previous steps — in Scenario 2, Ours decides to stop after seeing the lime green stools at Step 6, and in Scenario 5, Ours decides to stop after seeing the monitor under the board at Step 12. Overall, our method achieves the best success rate (same as Relevance) and improves the efficiency.

We again note that the two failed scenarios, 5 and 6, are mostly due to the incorrect prediction of the VLM even when seeing the relevant views. In Scenario 5, the views from Ours and Relevance show the monitor under the board on the ground, and in Scenario 6, the views show the elevators, which have the sign “2” indicating the second floor. However, the VLM answer predictions are both wrong.

**Miscalibrated question-image relevance score.** [Section IV](#) introduces the question-image relevance score  $\text{Rel}(x^t)$  [Eq. \(5\)](#) as a possible way of determining when the robot deems



Fig. A3: Twelve random views sampled within the scene for prompting GPT4-V to generate candidate questions and their answers. In this case, a question about the third image (first row) is generated by GPT4-V: ‘where is the wall clock placed in the kitchen? A) Above the sink B) On the refrigerator C) On the cabinet D) Above the oven Answer: A) Above the sink’.

the current view sufficient for answering the question and stops exploration. However, from experiments we find this score very miscalibrated. Fig. A5 shows the histogram of the highest relevance score within each scenarios (run with maximum time steps), where the answer prediction at the time step with the score is correct. Even though the question is answered correctly with the highest relevance score, the distribution of the scores is very wide — majority of the successful scenarios has the top relevance score lower than 0.5. Ideally the score should be high. This means the raw score alone cannot be used as a reliable indicator as the confidence for answering the question, thus motivating using multi-step conformal prediction (Section IV-A) to rigorously quantify the uncertainty instead.

**Semantic exploration results for each question category.** In Fig. 3a we have shown the comparison between our method and two other exploration baselines (CLIP-FBE and FBE) in simulated experiments, using all scenarios from the HMEQA dataset. Here in Fig. A6 we show the results for each of the five question categories (Location, State, Identification, Count, Existence) separately. We find our method shows the most improvement over baselines in Count and Existence. We

Scene 1:  
{twelve views from Scene 1}  
Where did I leave the small silver trash can? A) by the bedroom door B) in the kitchen C) in the bathroom D) by the living room sofa. Answer: A) by the bedroom door

Scene 2:  
{twelve views from Scene 2}  
I forgot where I hung the clock in the basement. A) above the bed B) on the wall C) on the pillar D) next to TV. Answer: C) on the pillar

Scene 3:  
{twelve views from Scene 3}  
Could you go and check how many wooden chairs I left in the garage? A) one B) two C) three D) four. Answer: D) four

Fig. A4: Three example question and answer pairs for GPT4-V. Actual views of the scenes not shown.

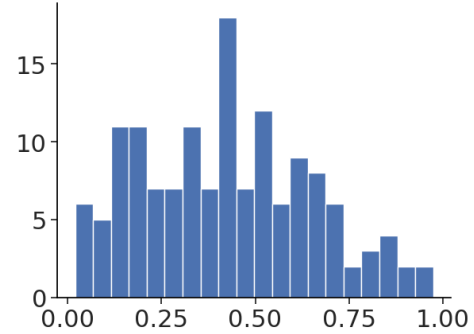


Fig. A5: Histogram of the highest relevance score over steps from each successful scenario. Many scenarios have the top score lower than 0.5, and thus the raw score does not well reflect the confidence for answering the question.

believe the reason is that in these two type of questions, the question itself provides a reasonable amount of information about the location of interest (e.g., ‘how many stools are there at the kitchen counter?’ and ‘are there some towels in the bathroom’), and such information can be used by VLM to indicate possible exploration directions. In contrast, Location questions do not have such information (e.g., ‘where is the piano?’), and the robot needs to explore most areas of the scene for answering them anyway. We also find State questions more difficult than the other categories, as the success rates do not improve with more time steps (e.g., ‘Is the living room air conditioning turned on?’), since it tends to involve small objects that are difficult to see or require very close views. We also note our method does not show improvement for Identification questions — the difference of these questions from Count and Existence is that, the questions only mention the *object* of interest (e.g., the piano) but not the *location* of interest (e.g., the bedroom), and this means identifying relevant exploration directions for them requires a higher degree of semantic reasoning of the VLM. We hope further improvement of the VLM can help better solve these scenarios.

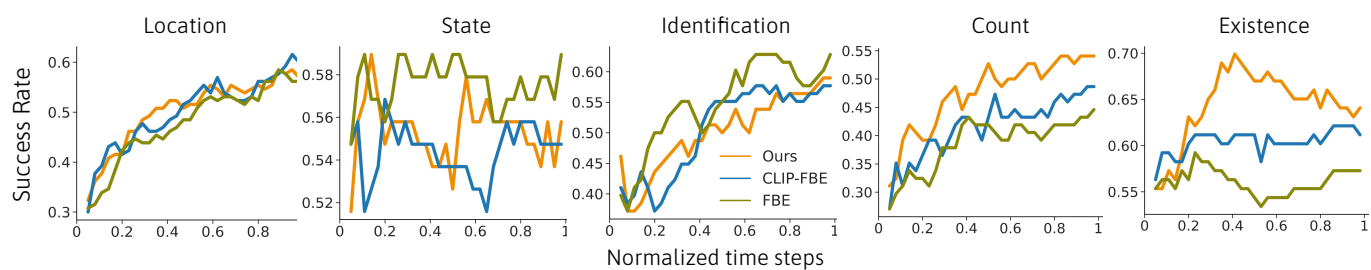


Fig. A6: Normalized time step taken vs. success rate in simulated experiments for comparing different exploration methods, in each of the five question categories: Location, State, Identification, Count, Existence. Our active exploration method shows the most improvement in Count and Existence, where the question provides a reasonable amount of information about the location of interest.



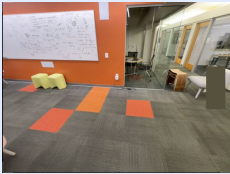


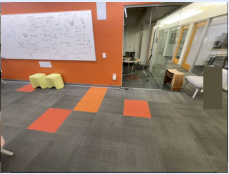




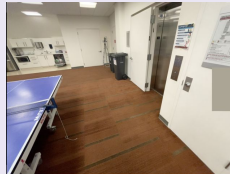
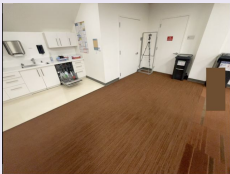
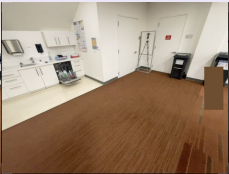
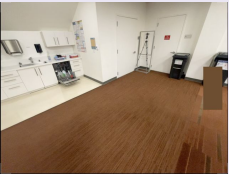
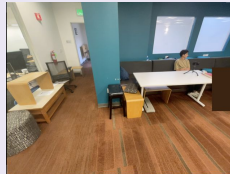


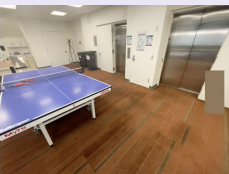
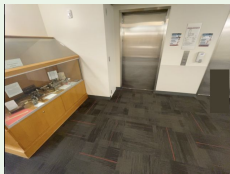

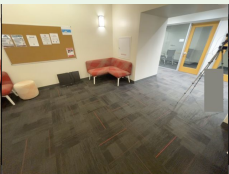

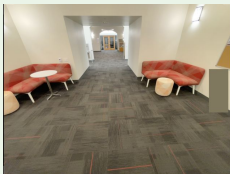
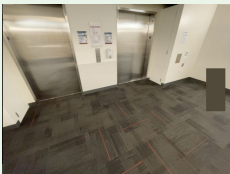
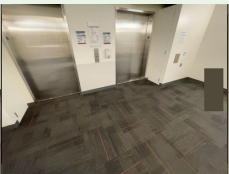

Questions	Initial view	Final View		
		Ours	Relevance	Entropy
<p>1</p> <p>Is there something here that I can bake my cookie dough in?</p> <p>A) Yes B) No</p> <p>C) (Do not choose this option)</p> <p>D) (Do not choose this option)</p> <p>Correct answer: A</p>				
		Answer: A Step: 5 ✓	Answer: B Step: 5 ✓	Answer: B Step: 1 ✗
<p>2</p> <p>What kind of stools are under the white board?</p> <p>A) White ones B) Dark blue ones</p> <p>C) Black ones D) Lime green ones</p> <p>Correct answer: D</p>				
		Answer: D Step: 6 ✓	Answer: D Step: 12 ✓	Answer: D Step: 6 ✓
<p>3</p> <p>Is the dishwasher in the kitchen open or closed? A) Closed B) Open</p> <p>C) (Do not choose this option)</p> <p>D) (Do not choose this option)</p> <p>Correct answer: B</p>				
		Answer: B Step: 16 ✓	Answer: B Step: 16 ✓	Answer: B Step: 16 ✓
<p>4</p> <p>How many ping pong paddles are there on the table? A) None B) One C) Two D) Three</p> <p>Correct answer: C</p>				
		Answer: B Step: 6 ✓	Answer: B Step: 6 ✓	Answer: B Step: 6 ✓
<p>5</p> <p>Where did I leave the monitor? A) On the sofa B) On the table C) Under the board D) On the wall</p> <p>Correct answer: C</p>				
		Answer: B Step: 12 ✗	Answer: B Step: 17 ✗	Answer: B Step: 3 ✗
<p>6</p> <p>Which floor is this?</p> <p>A) First floor (B) Second floor</p> <p>(C) Third floor (D) Basement</p> <p>Correct answer: B</p>				
		Answer: A Step: 15 ✗	Answer: A Step: 15 ✗	Answer: D Step: 5 ✗

Fig. A7: Six scenarios considered in hardware experiments. We show the results for question answering and stopping steps for our method vs. the two baselines. Ours achieves the best success rate while using fewer steps to stop.